LINEAR MAPS WITH POINT RULES:

APPLICATIONS TO PATTERN CLASSIFICATION AND ASSOCIATIVE MEMORY

Thesis by

Santosh Subramanyam Venkatesh

In Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy

California Institute of Technology
1987
(Submitted August 29, 1986)

Acknowledgement

My tenure at Caltech has been particularly rewarding, both intellectually and emotionally, and for this I am indebted to my colleagues, and in particular, to several of the faculty.

Prof. Demetri Psaltis has been not only my advisor, but friend, guide, and mentor. He has contributed in no small measure to my intellectual growth and maturity. His objectivity, erudition, and boundless enthusiasm have made these years very productive, and very enjoyable. I am particularly appreciative of the free rein he has given me. And now I am sorry that I beaned him with a tennis ball.

Prof. Edward C. Posner's dry wit has kept me entertained over the years. More to the res, however, I am deeply grateful to him for the time and effort he invested in ploughing through my reports and analyses. His precise comments and intellectually stimulating ideas contributed significantly to the tenor of this thesis.

In Yaser (Prof. Abu-Mostafa to the uninitiated) I have a wonderful friend with whom it has been a pleasure to chew the fat (figuratively speaking), and thresh out an idea. He has been a terrific colleague and friend (and that Burger King is still extant is in large measure due to us).

I would like to thank Prof. Robert J. McEliece for his patience, and his kindness in going through my proofs. His incisive comments contributed in large measure to more precise statements and definitions.

In Prof. Joel Franklin, I have invariably found infectious enthusiasm, and quick suggestions. He has been wonderfully free with his time. I am particularly grateful to him for introducing me to, and piquing my interest in, probability theory and analysis.

While the list of those who have contributed to my endeavour is rapidly becoming legend, I must mention Prof. John Hopfield, whose paper introduced me to the subject of neural networks; Prof. Anatoly Katok, through whom I came into contact with ergodic theory; Prof. R. Wilson who steered me in the right direction when I needed references; Dr. Eugene Rodemich whose proofs contributed greatly to my understanding of probability theory; and Dr. Eung G. Paek who supplied experimental results most generously.

Mrs. Helen Carrier, Mrs. Odessa Myles, and Mrs. Linda Dosza have been very helpful in re administrative matters, and have made my work considerably easier. My thanks go to them.

Last, but by no means the least, my love and gratitude go out to my wife Cecily Anne, who, while serenely maintaining her position as the better three-quarters of the Venkatesh firm, has been wonderfully patient and supportive of a frequently aberrant husband, and has displayed unlooked for talent and virtuosity in quelling the outbursts of a recalcitrant word processor.

Abstract

Generalisations of linear discriminant functions are introduced to tackle problems in pattern classification, and associative memory. The concept of a point rule is defined, and compositions of global linear maps with point rules are incorporated in two distinct structural forms–feedforward and feedback–to increase classification flexibility at low increased complexity. Three performance measures are utilised, and measures of consistency established.

Feedforward pattern classification systems based on multi-channel machines are introduced. The concept of independent channels is defined and used to generate independent features. The statistics of multi-channel classifiers are characterised, and specific applications of these structures are considered. It is demonstrated that image classification invariant to image rotation and shift is possible using multi-channel machines incorporating a square-law point rule. The general form of rotation invariant classifier is obtained. The existence of optimal solutions is demonstrated, and good sub-optimal systems are introduced, and characterised. Threshold point rules are utilised to generate a class of low-cost binary filters which yield excellent classification performance. Performance degradation is characterised as a function of statistical side-lobe fluctuations, finite system space-bandwidth, and noise.

Simplified neural network models are considered as feedback systems utilising a linear map and a threshold point rule. The efficacy of these models is determined for the associative storage and recall of memories. A precise definition of the associative storage capacity of these structures is provided. The capacity of these networks under various algorithms is rigourously derived, and optimal algorithms proposed. The ultimate storage capacity of neural networks is rigourously characterised. Extensions are considered incorporating higher-order networks yielding considerable increases in capacity.

Contents

Methodology

CHAPTER I

Correlators

CHAPTER II

CHAPTER III

Neural Networks

CHAPTER V

CHAPTER VI

CHAPTER VII

SPECTRAL APPROACHES -243-

CHAPTER VIII

MAXIMAL EPSILON CAPACITY -278-

Extensions

CHAPTER IX

CHAPTER X

Beauty is Truth,
Truth Beauty;

John Keats

To my parents, who taught me to behold
beauty in truth, and truth in beauty.

Methodology

CHAPTER I

INTRODUCTION

1. PRELUDE

Linear systems combine the dual advantage of analytical tractability and implementation simplicity. Consequently, they have found ready application in diverse problems in signal detection and pattern classification, as well as in allied situations in associative or content addressable memories. Linear maps together with a non-linear threshold rule for instance can be fruitfully employed in good classification schemes, as in Matched Filtration in signal detection. In such schemes the linear map can be viewed as providing *communication of information* between the various components of a problem, while the threshold decision rule provides the necessary non-linear *computation* adjunct to logical problem solving. The advantages of using linear transformations in conjunction with a simple decision rule in these situations is clear: these systems allow of low cost implementations, and their performance can, in almost all cases, be completely characterised.

The very simplicity of the linear map, however, precludes doing more complex problems. As an instance, it is not possible to achieve rotation invariance in image recognition using purely linear maps with a threshold decision rule.

The approach we will follow is to introduce *low complexity* generalisations of linear transformation to include *point non-linearities.* We demonstrate that the resultant systems expand considerably the set of problems that can be done using just linear machines, at relatively low cost. As the added non-linearities we consider act pointwise on the domain, all that is needed (in addition) is an array of single-input/single-output non-linear devices if the processing is done in parallel, or a single such device if the processing is sequential.

We consider applications of this approach to pattern classification using two particular non-linearities: square-law and threshold rules. The problems considered include: a characterisation of a general class of rotation invariant image recognition systems, and a performance characterisation of a class of low complexity binary filters.

With the introduction of feedback much more complex behaviour than simple classification can be realised in such systems. We focus on associative memory as an application, and obtain precise results on the capacity of simple iterated maps (comprised of linear transformations composed with a threshold rule) for storing complex associations.

2. PATTERN CLASSIFICATION

A. *Canonical Classifiers and Discriminant Functions*

We start with a description of the classical pattern classification problem. Let $\{\Omega^{(1)}, \ldots, \Omega^{(c)}\}$ be a finite set of c *states of nature* which we will also refer to as *classes.* The states of nature are represented by vectors $\mathbf{x}$ in a *pattern space* $\mathbb{H}_p$. We will assume throughout that $\mathbb{H}_p$ is a subset of an inner product space with the inherited inner product. We will denote the inner product space containing $\mathbb{H}_p$ by $\overline{\mathbb{H}}_p$. Instances of pattern spaces that we will use are: the Hilbert space (complex) L^2 of square-integrable functions, the vector space of real n-tuples $\mathbb{R}^n$, and the set of vertices of the binary hypercube $\{-1,1\}^n = \mathbb{B}^n$. (For the first two examples $\mathbb{H}_p = \overline{\mathbb{H}}_p$ while for the last example, $\mathbb{H}_p = \mathbb{B}_n$ while $\overline{\mathbb{H}}_p = \mathbb{R}^n$.) The occurrence of the pattern vectors $\mathbf{x} \in \mathbb{H}_p$ is specified according to the underlying state-conditional probability distributions $F_{\mathbf{x}}(I \mid \Omega^{(s)}) = \mathbf{P}\{\mathbf{x} \in I \mid \Omega^{(s)}\}$ which specify the probability of events $\{\mathbf{x} \in I \subseteq \mathbb{H}_p\}$ conditional upon the occurrence of a state of nature $\Omega^{(s)}$, and the *a priori* probabilities of occurrence, $\mathbf{P}\{\Omega^{(s)}\}$, of the various states of nature. The problem is to classify the patterns $\mathbf{x}$ as arising from the various states of nature.

Definition. A *pattern classifier* is a rule $C : \mathbb{H}_p \rightarrow \{\Omega^{(1)}, \ldots, \Omega^{(c)}\}$.

A couple of remarks are in order:

(1) The classifier partitions the pattern space $\mathbb{H}_p$ into c regions corresponding to the states of nature. Specifically, each feature vector $\mathbf{x} \in \mathbb{H}_p$ is associated with some state of nature.

(2) Note that we do not lose in generality by specifying the domain of the classifier C to be all of $\mathbb{H}_p$. We could, for instance, utilise a dummy state of nature $\Omega^{(0)}$ such that if $\mathbf{x}$ is mapped to $\Omega^{(0)}$ by C, then it indicates positive *non-recognition* of $\mathbf{x}$.

(3) While some simple situations arise when C is constrained to be a fixed mapping on the pattern space, we could, in principle, allow C to be a random mapping predicated upon some random specification of states of nature. Loosely speaking, in the first case the states of nature are fixed, and correspond to some *fixed* partition of the pattern space. In the second case, partitions of the pattern space corresponding to the states of nature are themselves chosen *randomly*. This corresponds to specifying an underlying probability distribution on the state-conditional probability distributions $F_{\mathbf{x}}(I \mid \Omega^{(s)})$ themselves. For this case, a particular realisation of a classifier mapping C is predicated upon a particular realisation of the state-conditional probability distributions. We will utilise both forms of classifiers– fixed and random–fruitfully.

Two issues in re classifiers are their characterisation with regard to some objective measure of performance (we shall return to this issue in section 3), and the complexity of their implementation. Before moving on to these two issues, we first introduce a *canonical form* for pattern classifiers. Our treatment follows that of Duda and Hart [1].

Definition. A set of *discriminant functions* (for classifier C) is a set of c real valued functions on the pattern space, $\delta^{(s)} : \mathbb{H}_p \rightarrow \mathbb{R}$, $s = 1,\ldots,c$, such that for every $\mathbf{x} \in \mathbb{H}_p$,

$$\delta^{(t)}(\mathbf{x}) > \delta^{(s)}(\mathbf{x}) \text{ for all } s \neq t \ , \tag{1.2.1}$$

if $C(\mathbf{x}) = \Omega^{(t)}$.

A canonical form for pattern classifiers is a machine that computes c discriminant functions, and selects the state of nature corresponding to the largest discriminant function. A block-diagrammatic representation of such a classifier is illustrated in fig. 1.1

The choice of discriminant function $\delta^{(s)}$ for a classifier C is not unique, and in fact the following assertion holds, as is easily seen.

Proposition 1.2.1. Let $f : \mathbb{R} \to \mathbb{R}$ be any monotonically increasing function. Then $f \circ \delta^{(s)} : \mathbb{H}_p \to \mathbb{R}$, $s = 1,...,c$, is also a set of discriminant functions for classifier C.

Thus we have an equivalence class of sets of discriminant functions for each classifier, with each set of discriminant functions yielding the same resultant classification. While such sets of discriminant functions are indistinguishable from a theoretical point of view, in practice, however, the appropriate choice of discriminant functions can lead to considerable savings in analysis and implementation.

In the discriminant function methodology, the pattern space is partitioned into c *decision regions* $R^{(1)}, \ldots, R^{(c)} \subseteq \mathbb{H}_p$, with $\mathbf{x} \in R^{(t)}$ if $\delta^{(t)} > \delta^{(s)}$ for all $s \neq t$. The decision regions are separated by *decision surfaces* where two or more discriminant functions take on equal values. Classification of points on the decision surfaces can be made by using any suitable tie-breaking rule.

B. Linear Discriminant Functions

In practice, the cost of realising "optimal" classifiers may well be prohibitive, as they would in general require very complex discriminant functions. A practical alternative to constructing such problem dependent optimal classifiers is to implement

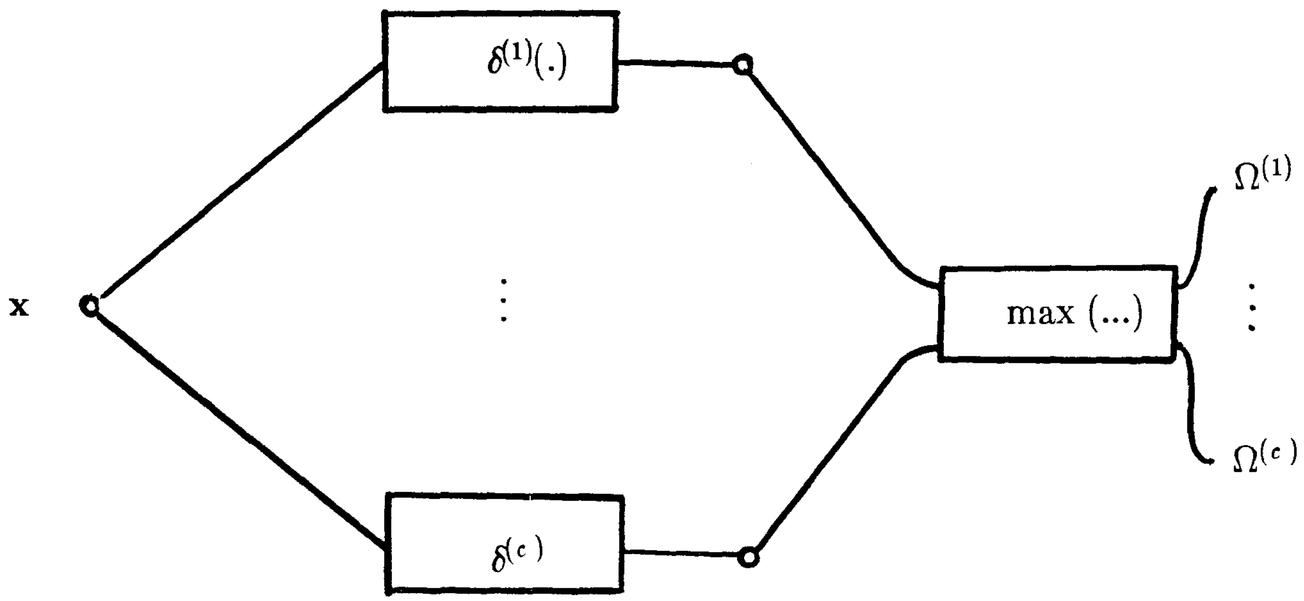

Fig. 1.1. Canonical classifier: Discriminant function realisation.

classifiers of *fixed structure*. While such classifiers of necessity sacrifice some performance, they gain in simplicity of construction, and in being relatively easy to compute.

Linear discriminant functions are classifiers of fixed structure which are affine linear functionals on the pattern space $\mathbb{H}_p$. These are particularly attractive classifiers from the computational point of view as they are among the simplest non-trivial classifiers to implement, and are very tractable analytically. They are even optimal classifiers for an admittedly small set of underlying distributions. As such, linear discriminant functions are attractive candidates for classifiers.

We will denote the set of c linear discriminant functions corresponding to a linear classifier (also called a *linear machine*) by the functionals $\mathbf{L}^{(s)} : \mathbb{H}_p \to \mathbb{R}$, $s = 1,...,c$. With each $\mathbf{L}^{(s)}$ we associate a *weight vector* $\mathbf{l}^{(s)}$ in the parent inner product space $\overline{\mathbb{H}}_p$, and a real scalar threshold $l_0^{(s)}$, so that for every pattern $\mathbf{x} \in \mathbb{H}_p$,

$$\mathbf{L}^{(s)}(\mathbf{x}) = (\mathbf{l}^{(s)} , \mathbf{x}) + l_0^{(s)} \ . \tag{1.2.2}$$

Pattern classification is by means of the threshold comparison rule of equation (1.2.1). *Note that for the two-class case, this is just a threshold rule: decide* $\Omega^{(1)}$ *if* $(\mathbf{l}^{(1)} - \mathbf{l}^{(2)} , \mathbf{x}) + (l_0^{(1)} - l_0^{(2)}) > 0$, *else decide* $\Omega^{(2)}$.

Denoting the c-tuple of linear discriminant functions by $\mathbf{L} = (\mathbf{L}^{(1)}, \ldots, \mathbf{L}^{(c)}) : \mathbb{H}_p \to \mathbb{R}^c$, and the comparator of (1.2.1) by $\mathbf{T} : \mathbb{R}^c \to \{\Omega^{(1)}, \ldots, \Omega^{(c)}\}$, the linear classifier C can be written as the composition $(\mathbf{T} \circ \mathbf{L}) : \mathbb{H}_p \to \{\Omega^{(1)}, \ldots, \Omega^{(c)}\}$. A two-dimensional illustration of possible decision regions produced by such a linear machine is shown in fig. 1.2.

Extending the analogy of fig. 1.2 to higher dimensions, each linear discriminant can be thought of as realising a *separating plane* (hyperplane) in a multi-dimensional space. The decision surfaces of the linear classifier are c separating planes, and these demarcate the decision regions.

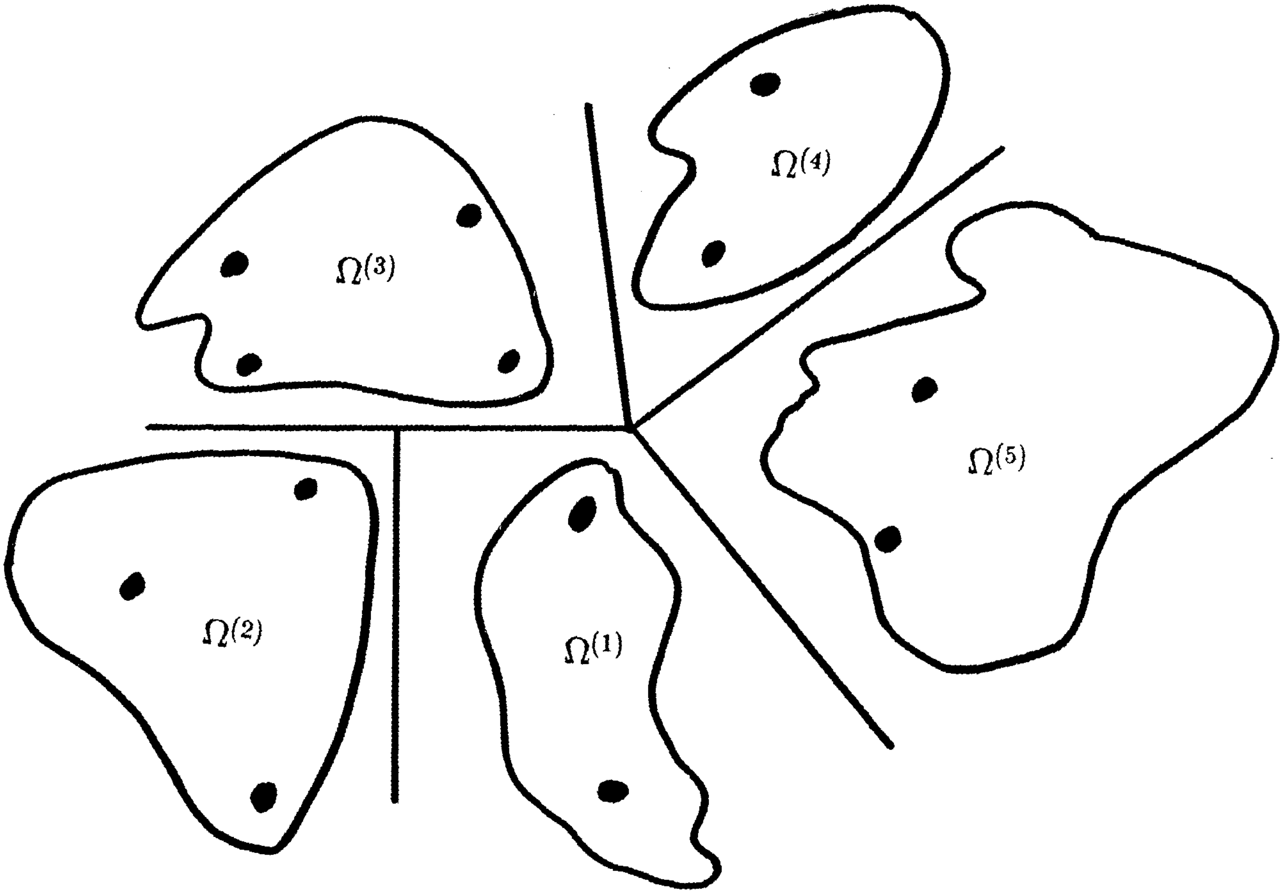

Fig. 1.2. Partitioning of pattern space by a linear machine.

C. Matched Filters

A particular instance of a useful linear classifier is the ***matched filter***. We illustrate this with an example in signal detection. We are presented with a situation where a ***signal*** or ***reference pattern*** $\mathbf{x}_0 \in \mathbb{H}_p$, may or may not have been present in an environment of additive, zero-mean noise, with a positive definite autocorrelation operator $\mathbf{R}_\mathbf{n} : \overline{\mathbb{H}}_p \rightarrow \overline{\mathbb{H}}_p$. There are two states of nature corresponding to the two hypotheses: the presence or absence of the signal. Given a pattern $\mathbf{x} \in \mathbb{H}_p$, (whose state-conditional probability distribution is determined by the random noisy environment, together with the presence, or absence, of the signal) the problem is to find suitable weight vectors to achieve a reliable mapping of $\mathbf{x}$ into one of the two states of nature.

The matched filter is a weight vector (corresponding to a linear discriminant function) defined by

$$\hat{\mathbf{l}} = \mathbb{R}_\mathbf{n}^{-1} \mathbf{x}_0 . \qquad (1.2.3)$$

If the noise is white, as is frequently assumed, then $\hat{\mathbf{l}}$ is just a scaled version of the signal–hence the sobriquet "matched" filter. Note that as there are only two states of nature, it suffices to have a single linear discriminant function corresponding to the hypothesis that the signal was present, with a simple threshold classification rule.

The fact that makes the matched filter important is the following classical result. Let $\hat{\mathbb{H}}_p$ denote the space of positive-definite noise-autocorrelation operators. Define the *signal-to-noise ratio* (SNR) functional $\rho : \mathbb{H}_p \times \overline{\mathbb{H}}_p \times \hat{\mathbb{H}}_p \rightarrow \mathbb{R}^+$ by

$$\rho (\mathbf{x} , \mathbf{l} , \mathbf{R}_\mathbf{n}) = \frac{| (\mathbf{l} , \mathbf{x}) |^2}{(\mathbf{l} , \mathbf{R}_\mathbf{n} \mathbf{l})} . \qquad (1.2.4)$$

Theorem 1.2.1. For a fixed signal $\mathbf{x}_0 \in \mathbb{H}_p$, and a fixed positive definite noise-auto-correlation operator, $\mathbf{R}_\mathbf{n} \in \hat{\mathbb{H}}_p$, the matched filter $\hat{\mathbf{l}} \in \overline{\mathbb{H}}_p$ maximises the signal-to-noise ratio among all weight vectors in the Hilbert Space $\overline{\mathbb{H}}_p$.

We shall return to the issue of signal-to-noise ratios again in Section 3 when we consider performance measures for classifiers. The signal-to-noise ratio is a frequently employed performance measure because of its simplicity, and in some cases it is actually a good measure of classification performance. With the signal-to-noise ratio as a criterion then, the matched filter provides the best performance among all linear discriminant functions. The idea can be simply extended to multiple discriminant functions in the c-class recognition problem.

D. The Capacity of a Separating Plane

In spite of their many appealing features, however, linear discriminant functions are intrinsically limited in scope. The decision regions for a linear machine are constrained to be convex, and this in particular leads to the fact that every decision region must be simply connected. These factors seem to imply that the linear discriminant function approach is best suited for situations where the state-conditional probability densities are unimodal, i.e., the presence of a particular state of nature results in patterns which are clustered together locally in the pattern space, as illustrated in fig. 1.2. An instance where this fails is in image recognition. Associate all rotations and scales of a reference image with one class or state of nature. The patterns in the class are not clustered together in the image space, so that linear discriminant functions do not work well in this instance.

A more serious limitation of linear discriminant functions is that they are seriously limited in the number of states of nature that they can accurately classify. Consider a finite-dimensional Euclidean pattern space, which we take to be $\mathbb{R}^n$ without loss of generality. Let $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(c)}$ be c reference patterns in $\mathbb{R}^n$ chosen to be in general position (i.e., any subset of n or fewer reference patterns is linearly independent). At the very least, we require that the reference patterns themselves be mapped to the appropriate states of nature.

The weight vectors $\mathbf{l}$ corresponding to linear discriminant functions are just n-tuples of real numbers, and the inner product is the natural one. For this case we have

$$L(\mathbf{x}) = (\,\mathbf{l}\,,\mathbf{x}\,) + l_0 = \sum_{j=1}^{n} l_j x_j \, + l_0 \,.$$

For the two-class case, for instance, this leads to a simple threshold decision rule, as illustrated in fig. 1.3 (a).

We query: how large can the number of classes c be made while ensuring that $\exists$ some set of linear discriminant functions which accurately maps the reference patterns to the appropriate class? The answer is furbished in the following result which we prove in chapter IX (also cf. [2]).

Theorem 1.2.2. For every $\lambda \in (0,1)$, as $n \to \infty$, the probability that $\exists$ a linear discriminant function which accurately classifies the c reference patterns approaches one if $c \leq 2(n+1)(1-\lambda)$, and approaches zero if $c \geq 2(n+1)(1+\lambda)$.

Thus, no more than $2\,(\,n+1\,)$ states of nature can be accurately identified by linear discriminant functions if the pattern space is an n-dimensional Euclidean vector space. This result is consistent with an argument based on the available *degrees of freedom*: the n-dimensional weight vector together with the scalar threshold constitute $n+1$ degrees of freedom. If we need more powerful classification capability, and more flexibility in classification, however, we will have to resort to more complex classifiers.

E. Generalisation: Linear Maps with Point Rules

We consider generalisations of the linear discriminant function structure which allow of more flexible classifiers, but which at the same time retain much of the simplicity of linear machines. Our approach is to introduce a "feature extraction" stage prior to actually computing linear discriminant functions, as is indicated schematically in fig. 1.3. The feature extraction procedure is a composite mapping from the pattern space $\mathbb{H}_p$ to a *feature space* $\mathbb{H}_f$. The linear discriminant functions are now computed from vectors in the feature space $\mathbb{H}_f$.

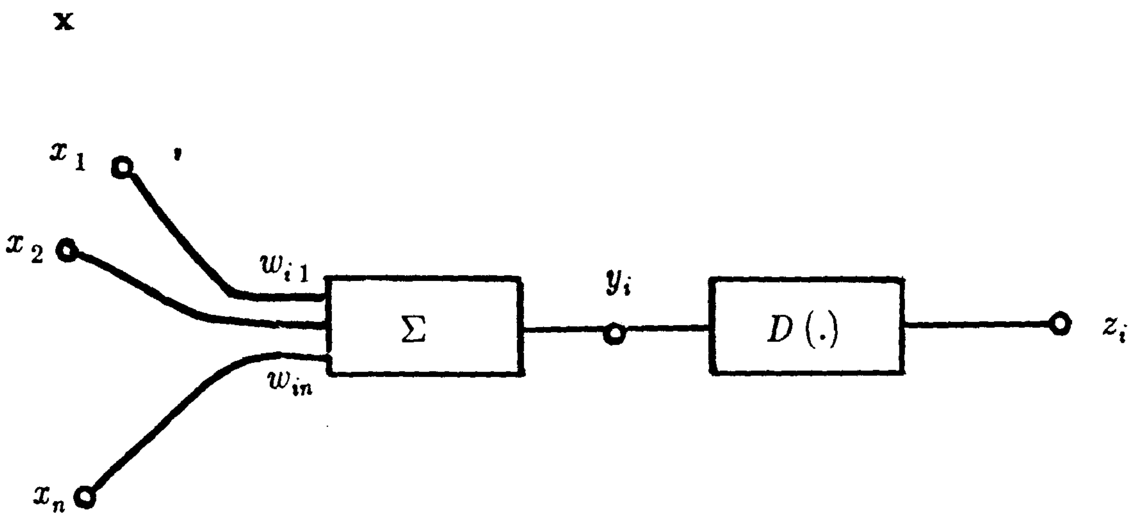

Fig. 1.3 (a). A channel for extracting a single feature component.

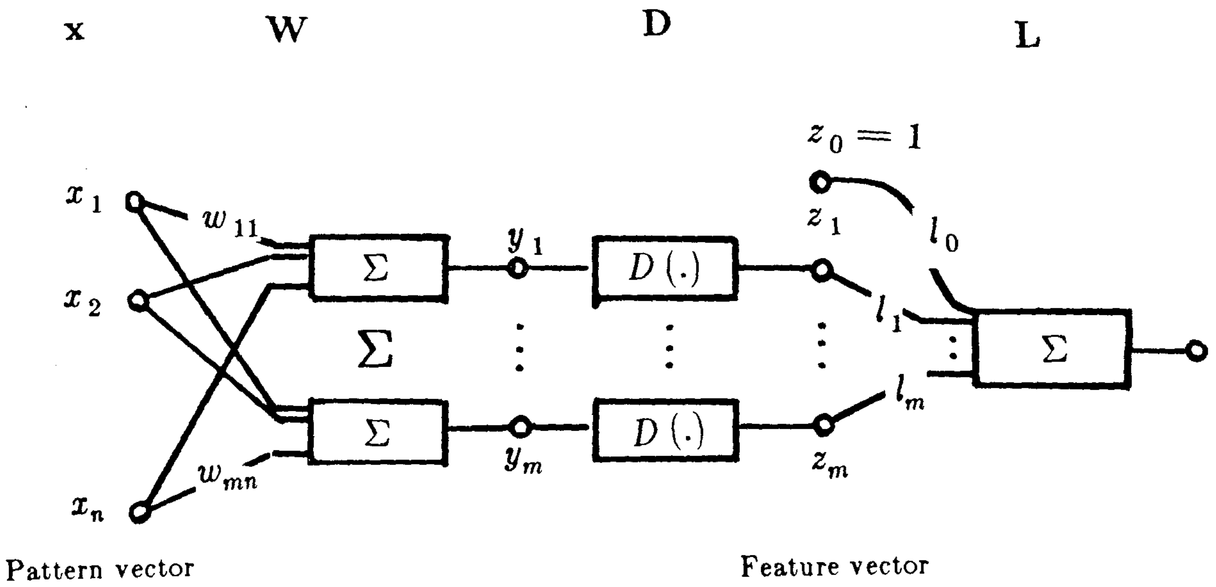

Fig. 1.3 (b). Realisation of a generalised linear discriminant function.

The feature spaces we consider are vector spaces indexed by some set. Specifically, $\mathbb{H}_f$ is a family of real-valued functions $y : \Lambda \rightarrow \mathbb{R}$ satisfying some suitable properties (such as continuity or square-integrability), where Λ is some index set. The following are two examples of feature spaces.

Example 1. Choose $\mathbb{H}_f$ to be the space of all square integrable real-valued functions on the two-dimensional plane. The underlying index set in this instance is $\Lambda = \mathbb{R}^2$. □

Example 2. The vector space of real m-tuples $\mathbb{R}^m$, which is indexed by the finite set $\Lambda = \{1,...,m\}$. □

We will use the representation $(y_\nu : \nu \in \Lambda)$ to explicitly represent feature vectors $\mathbf{y} \in \mathbb{H}_f$ in terms of the underlying index set Λ. We denote the (parent) space of complex-valued functions indexed by Λ by $\overline{\mathbb{H}}_f$, and assume that $\overline{\mathbb{H}}_f$ comes equipped with an inner product.

Definition. A map $\mathbf{D} : \overline{\mathbb{H}}_f \rightarrow \mathbb{H}_f$ is a *point rule* (on the feature space) iff there is a function $D : \mathbb{C} \rightarrow \mathbb{R}$ such that

$$\mathbf{D}(y_\nu : \nu \in \Lambda) = (D\ (y_\nu) : \nu \in \Lambda) \in \mathbb{H}_f \quad \forall\ \mathbf{y} \in \overline{\mathbb{H}}_f\ . \tag{1.2.5}$$

If the indices ν represent time, for instance, then the point rule $\mathbf{D}$ is a memoryless map.

Example 3. (Square-law point rule)
$\mathbf{D}(y_\nu : \nu \in \Lambda) = (\mid y_\nu \rfloor^2 : \nu \in \Lambda)$. □

Example 4. (Threshold point rule)
$\mathbf{D}(y_\nu : \in \Lambda) = (\text{sgn} \{ \text{Re}\ y_\nu \} : \nu \in \Lambda)$. □

Point rules are interesting from a practical point of view, as they are reasonably simple to implement–each vector component generated by a point rule depends solely on the corresponding component of the original vector. Point rules may be realised in parallel by a family of identical single-input, single-output devices (or sequentially by a single such device).

The feature extraction procedure that we will consider throughout is a composite map $(\mathbf{D} \circ \mathbf{W}) : \mathbb{H}_p \rightarrow \mathbb{H}_f$, where $\mathbf{W} : \mathbb{H}_p \rightarrow \overline{\mathbb{H}}_f$ is a linear map, and $\mathbf{D} : \overline{\mathbb{H}}_f \rightarrow \mathbb{H}_f$ is a point rule. (In some cases, we could think of the feature-extraction procedure as being a dimensionality reduction process which effects a sensible reduction of a pattern space of high dimensionality to a lower dimensional feature space.) The final classification stage is by means of discriminant function on the feature space $\mathbf{L} : \mathbb{H}_f \rightarrow \mathbb{R}$.

We will refer to the feature extraction procedure for each component z_ν of the feature vector as a *channel.* If the feature space is m-dimensional, we will have m-channels to compute the components of the feature vector.

Thus, the discriminant functions that we will be considering are of the form $(\mathbf{L} \circ \mathbf{D} \circ \mathbf{W}) : \mathbb{H}_p \rightarrow \mathbb{R}$. These discriminant functions may be considered to be simple generalisations of linear discriminant functions involving the point rule $\mathbf{D}$. Note that the particular choice of $\mathbf{D} = Id$ results in a simple linear discriminant function $(\mathbf{L} \circ \mathbf{W})$ on the pattern space. Thus these new constructs are generalisations of linear machines, which encompass linear classifiers. Additional flexibility in classification is obtained by suitably specifying the (generalised) decisions $\mathbf{D}$, and the linear maps $\mathbf{W}$, and $\mathbf{L}$. If $\mathbf{D}$ is a threshold map, for instance, the procedure is akin to making several partial decisions at an intermediate stage, and using these as features to obtain a final classification using a linear discriminant function.

In fig. 1.3 (b) we have a schematic representation of these generalised linear discriminant functions. The pattern space for this example is Euclidean n-space, while the feature space is Euclidean m-space. Pattern vectors $\mathbf{x} \in \mathbb{R}^n$ are mapped to vectors $\mathbf{y} \in \mathbb{R}^m$ by an $m \times n$ matrix of weights $[\, w_{ij} \,]$, with $y_i = \sum_{j=1}^{n} w_{ij} x_j$. The point rule **D** acts pointwise on each component y_i to produce an m-vector $\mathbf{z}$ with components $z_i = D(y_i)$. Finally, the discriminant function is formed as a linear combination $\sum_{i=1}^{m} l_i z_i + l_0$ of the components of the feature vector $\mathbf{z}$.

As pointed out earlier, these generalised constructs subsume within them the linear discriminant functions as a trivial case. A proper choice of point rules allows of some added flexibility in classification. We illustrate this in the following example.

Example 5. Consider the Boolean mapping XOR : $\{0,1\} \times \{0,1\} \rightarrow \{0,1\}$,

$$\text{XOR}(x_1,x_2) = \begin{cases} 1 \text{ if } (x_1,x_2) = (0,1) \text{ or } (x_1,x_2) = (1,0) \\ 0 \text{ if } (x_1,x_2) = (0,0) \text{ or } (x_1,x_2) = (1,1) \end{cases} .$$

The logical XOR is a two-state classification problem which cannot be realised by any linear discriminant function. As illustrated in fig. 1.4 (a), there is no choice of separating plane which isolates points (0,0) and (1,1) from points (0,1) and (1,0). However, a generalised linear discriminant function using a threshold point rule can be constructed to solve the problem, as illustrated in fig. 1.4 (b). Here, Boolean pairs (x_1,x_2) are first mapped into a two-dimensional Boolean feature space through the linear map with component matrix

$$\mathbf{W} = \begin{bmatrix} 1 & -2 \\ -2 & 1 \end{bmatrix},$$

and a point rule based on the threshold map

$$D(y) = \begin{cases} 0 \text{ if } y < 0.5 \\ 1 \text{ if } y \geq 0.5 \end{cases} .$$

The feature vectors (y_1,y_2) that are realised are hence

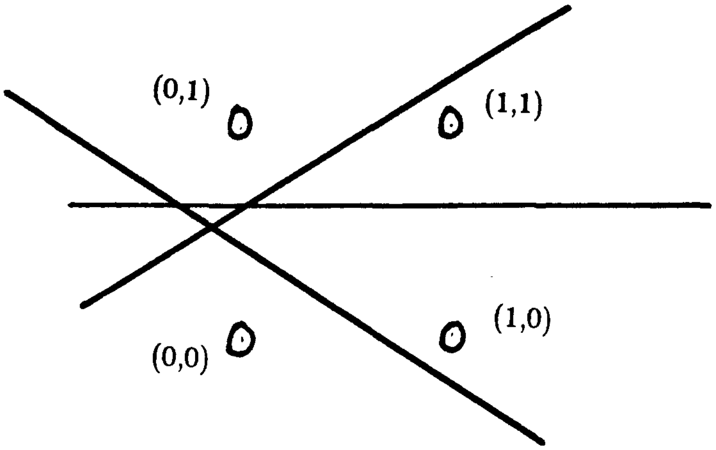

Fig. 1.4 (a). No choice of separating plane can separate (0,0) and (1,1) from (0,1) and (1,0).

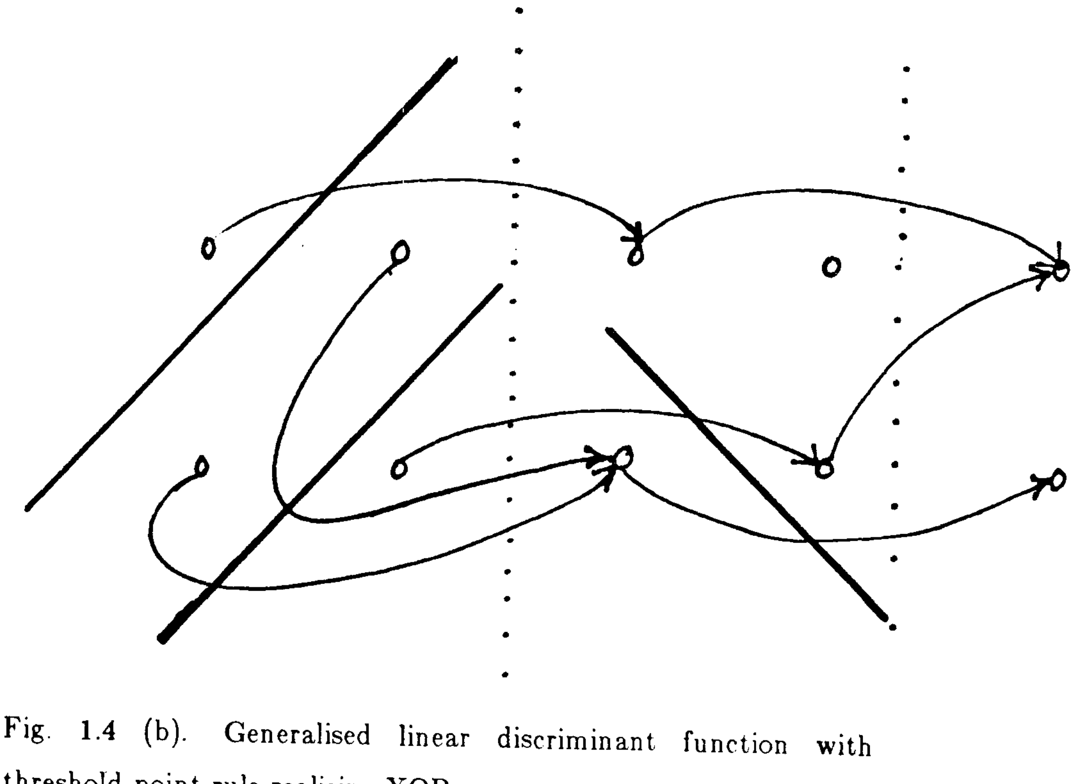

Fig. 1.4 (b). Generalised linear discriminant function with threshold point rule realising XOR.

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} D(\ x_1 - 2x_2) \\ D(-2x_1 + x_2) \end{pmatrix} .$$

It can be easily seen that this procedure maps (0,0), and (1,1) to (0,0), while mapping (0,1) to (0,1), and (1,0) to (1,0). Finally, constructing a linear discriminant function with weight vector $\mathbf{l} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, and threshold $l_0 = -0.5$, on the feature space results in the map

$$(x_1, x_2) \mapsto \begin{cases} 0 & \text{if } D(x_1 - 2x_2) + D(-2x_1 + x_2) < -0.5 \\ 1 & \text{if } D(x_1 - 2x_2) + D(-2x_1 + x_2) \geq -0.5 \end{cases} .$$

This is the required logical XOR. □

Another instance of this acquired flexibility is examined in chapter III, where we analyse the problem of image classification invariant to image shift and rotation in some detail. This is a problem which can be solved via generalised linear discriminant functions, but which linear discriminant functions cannot solve in isolation.

Simple point rules are favoured from the implementation point of view. In the next few chapters, we will examine classifiers of the above structure using simple point rules under a variety of conditions. Note, however, that with classifiers of this structure, the linear discriminant function at the feature space determines the actual *number* of states of nature that can be identified. Thus, if the feature space is an m-dimensional Euclidean space, we can classify at most $2(m + 1)$ patterns by theorem (1.2.2). Thus, with a feedforward system of the sort we have been considering so far, we can achieve more flexibility in classification, but cannot really improve significantly on the capacity of the classifier (over a linear machine) unless we allow of feature spaces of large dimensionality. This of course is in accordance with intuition–to solve complex problems we will need a great number of degrees of freedom, which in this case corresponds to large feature spaces. The best gains, as we shall see, will accrue from dispensing with the linear discriminant function stage, and considering cascades of linear maps and point rules, and feedback.

3. ASSOCIATIVE MEMORY

A problem allied to that of pattern classification is that of associative or content-addressable memory. Formally, ***an associative memory is a classifier where the states of nature are themselves specified patterns from the pattern space.*** Specifically, an associative memory is a map from the pattern space to itself, which maps a subset of the pattern space to a specified set of pattern vectors $\{ \mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(m)} \}$. The specified set of vectors are called the *fundamental memories.*

In a typical problem in associative memory, we specify a "similarity" metric for the pattern space, and require that the associative memory maps all patterns which are near a fundamental memory to the memory itself.

Example 6. Consider a pattern space of binary n-tuples, $\mathbb{B}^n$. Let the specified metric be the Hamming distance between two vectors, and let $0 \leq \rho n < \frac{n}{2}$ be the specified extent of similarity. For a set of m fundamental memories, $\mathbf{u}^{(\alpha)} \in \mathbb{B}^n$, $\alpha = 1,\ldots,m$, which are mutually separated by Hamming distances greater than ρn, we require that all vectors in Hamming balls of radii ρn surrounding the fundamental memories be mapped to the corresponding memories. □

We hence require an associative memory to be a nearest-neighbour pattern classifier, or equivalently, an error correcting code. Note that it may not be absolutely essential that the associative memory map patterns to precisely the fundamental memories. If a certain measure of error tolerance is prescribed, for instance, it may suffice that the mapping results in any of a number of patterns *near* the fundamental memory.

System-theoretic approaches to associative memory have benefited greatly from neurobiological modeling of brain function, and much of the terminology, and approaches in vogue have a strong biological flavour (cf. [3]). Hence, while an associative memory is formally a pattern classifier, we will nevertheless distinguish between selected pattern classification problems (which we treat in the section on

Correlators), and problems in associative memory (which we treat in the section on Neural Networks). For the pattern classification problems we will consider generalised linear machines of the form we introduced earlier. For associative memory, however, we require maps from the pattern space to itself. We hence dispense with the linear discriminant function classification stage, and consider iterated maps of the form $(\mathbf{D} \circ \mathbf{W})^k : \mathbb{H}_p \to \mathbb{H}_p$, where $\mathbf{W} : \mathbb{H}_p \to \overline{\mathbb{H}}_p$ is a linear map, and $\mathbf{D} : \overline{\mathbb{H}}_p \to \mathbb{H}_p$ is a point rule.

Thus, the pattern classifiers we consider are feedforward systems which make ***hard decisions*** upon vectors in the pattern space through the medium of the generalised linear discriminant functions. Clearly, if an error is made in the decision process, then the classification is irretrievably in error. The associative memory structure we consider is a feedback system, and hence has the potential to compensate for occasional errors in decision. In this case, the iterated mapping considered on the pattern space makes ***soft decisions*** which gradually converge to a true classification decision. We will elucidate upon this in greater detail in the section on Neural Networks.

4. MEASURES OF PERFORMANCE

A. Consistency

Thus far, we have alluded only briefly to criteria for judging performance. Objective measures of classifier performance are, however, of prima facie importance in characterising classifiers and rating their relative performance. We will, in main, not distinguish between classifiers and associative memories in this section. The performance measures we develop for classifiers will continue to hold for associative memories as a special family of classifiers.

Let Δ denote the family of discriminant functions. By equation (1.2.1), a c-tuple of discriminant functions $(\delta^{(1)}, \ldots, \delta^{(c)}) \in \Delta^c$ represents a particular discriminant function realisation of a classifier.

Definition. A *performance measure* for classifiers is a mapping $\rho : \Delta^c \rightarrow \mathbb{R}$ which induces a linear order $(\geq)$ on the sets of discriminant functions corresponding to classifiers. We shall say that a classifier realisation $(\delta^{(1)}, \ldots, \delta^{(c)})$ is ***superior to*** a classifier realisation $(\gamma^{(1)}, \ldots, \gamma^{(c)})$ with respect to a performance measure ρ if $\rho(\delta^{(1)}, \ldots, \delta^{(c)}) > \rho(\gamma^{(1)}, \ldots, \gamma^{(c)})$.

Clearly, an arbitrary definition of performance measure is unlikely to subscribe to our intuitive notions of what a good classification scheme entails. If the performance measure is to reflect some "true" measure of goodness of various classifiers, then it must be chosen with some care, and in particular, must reflect the underlying *a priori* probabilities of the states of nature and the state-conditional distributions of the pattern vectors. Before returning to the issue of what constitutes a good performance measure, we first characterise some desirable consistency properties in performance measures which will be useful in classifying different measures.

As we saw earlier, there exist an infinity of sets of discriminant functions satisfying equation (1.2.1), all of which represent a particular classifier. Hence, with every specified classifier $C : \mathbb{H}_p \rightarrow \{\Omega^{(1)}, \ldots, \Omega^{(c)}\}$, we have an associated equivalence class of c-tuples of discriminant functions, $S_C = [\,(\delta^{(1)},\ldots,\delta^{(c)})_C\,]$, with each member of S_C realising the same mapping (specified by the classifier C) from the pattern space to the set of the given states of nature. A desirable consistency property of the performance measure is that it be insensitive to the specific discriminant function realisation of a classifier, so that the classifier order relations are invariant to implementational details.

Definition. A performance measure ρ is *totally consistent* iff for every classifier $C : \mathbb{H}^p \rightarrow \{\Omega^{(1)}, \ldots, \Omega^{(c)}\}$, it yields the same value for every member of the equivalence class S_C.

While total consistency is clearly a very desirable property in a performance measure, we might suspect that this would place too rigourous a constraint on permissible performance measures. In fact, as can be seen from the following assertion, totally consistent performance measures have to be explicitly representable as functions of partitions of the underlying pattern space.

Let $\mathbf{P}$ be the family of all partitions of the pattern space $\mathbb{H}_p$ into c disjoint subsets. For every classifier C, let $R_C = \{ R^{(1)}, \ldots , R^{(c)} \}$ be the induced partition of $\mathbb{H}_p$; specifically, $\mathbf{x} \in R^{(s)} \Leftrightarrow C(\mathbf{x}) = \Omega^{(s)}$. Let $g : \Delta^c \rightarrow \mathbf{P}$ be the natural mapping associating c-tuples of discriminant functions with partitions of the pattern space; $g(\delta^{(1)}, \ldots , \delta^{(c)}) = R_C$ whenever $(\delta^{(1)}, \ldots , \delta^{(c)})$ is a member of the equivalence class S_C of discriminant function c-tuples corresponding to classifier C.

Proposition 1.4.1. A performance measure $\rho : \Delta^c \rightarrow \mathbb{R}$ is totally consistent iff $\exists$ a mapping $\mu : \mathbf{P} \rightarrow \mathbb{R}$ such that $\rho = \mu \circ g$.

Proof. Assume $\exists$ a map $\mu : \mathbf{P} \rightarrow \mathbb{R}$ such that $\rho = \mu \circ g$. Let C be a classifier. Every discriminant function c-tuple in the equivalence class S_C is mapped into the same partition R_C of $\mathbb{H}_p$ by g. Hence, $\rho = \mu \circ g$ is totally consistent.

Now fix a classifier C, and assume ρ is a totally consistent performance measure. Let $(\delta^{(1)}, \ldots , \delta^{(c)})$ be any member of S_C. Now, every partition in $\mathbf{P}$ corresponds to some classifier. Define the map $\mu : \mathbf{P} \rightarrow \mathbb{R}$ by $\mu(R_C) = \rho((\delta^{(1)}, \ldots , \delta^{(c)})_C)$ for every partition R_C in $\mathbf{P}$. Then $\rho = \mu \circ g$. □

The restriction that ρ be specified in terms of partitions of the pattern space can be quite unrealistic, especially for large (infinite!) dimensional pattern spaces. From a practical standpoint, we would like to be able to specify the performance measure directly in terms of the "observables," the discriminant functions. Proposition (1.2.1) inspires the following less demanding requirement of consistency.

Definition. A performance measure ρ is ***monotonically consistent*** iff for every monotonically increasing, piecewise differentiable function $f : \mathbb{R} \to \mathbb{R}$, and every c-tuple of discriminant functions $(\delta^{(1)},...,\delta^{(c)}) \in \Delta^c$,

$$\rho (\delta^{(1)}, \ldots , \delta^{(c)}) = \rho(f \circ \delta^{(1)}, \ldots , f \circ \delta^{(c)}) .$$

We introduce the requirement of piecewise differentiability in the definition to allow of some ease in proving technical results later. The requirement restricts our attention to "useful" functions f .

Monotonic consistency is clearly not as strong as total consistency, and in fact, totally consistent performance measures are also monotonically consistent. The converse is not true, however. Nevertheless, in light of proposition (1.2.1), monotonically consistent performance measures exhibit consistent behaviour over a useful range of discriminant function realisations of classifiers.

We now return to the issue of specifying good performance measures which we hope are consistent in some sense. In the remainder of this section we will specify three measures of performance to which we will frequently have recourse.

B. Probability of Error

In line with intuitive expectation, a good performance measure will reflect the underlying *a priori* probabilities of the states of nature, and the state-conditional probability distributions of the pattern vectors. We illustrate with an example.

Assume the state-conditional distributions are such that the occurrence of any pattern vector depends solely on the presence or absence of one of the states of nature. In effect, the state-conditional distributions are localised in disjoint regions in the pattern space. This is shown schematically in fig. 1.5 (a), where we assume four states of nature. Here, $\mathbf{P} (\mathbf{x} \mid \Omega^{(s)})$ is identically zero outside the indicated support for the probability distribution conditioned on $\Omega^{(s)}$. Now assume we have a classifier C which partitions the pattern space into c regions, as illustrated in fig. 1.5 (b). The partitions here are indicated by bold lines. Clearly, ideal classifier performance would obtain in this instance if the decision boundaries in fig. 1.5 (b) were to coincide with

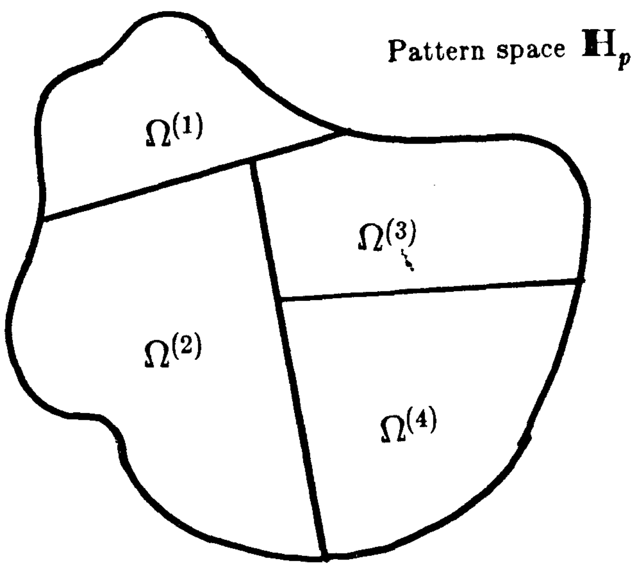

Fig. 1.5 (a). A partition of the pattern space according to prescribed state-conditional probability distributions; a pattern in a particular region will occur only if the corresponding state of nature occurs.

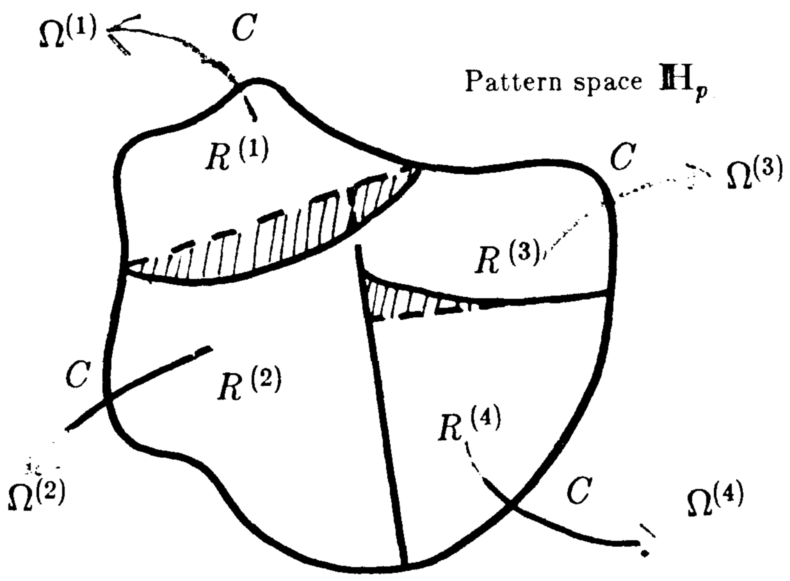

Fig. 1.5 (b). A partition of the pattern space by a particular classifier. The shaded areas denote regions with non-optimal classification.

the boundaries demarcating the support of the state-conditional distributions in fig. 1.5 (a). For the indicated classifier, however, there is a mismatch in the decision boundaries, as indicated in the hatched areas in the figure. Vectors in the hatched areas are erroneously identified by the classifier: in particular, some vectors arising from $\Omega^{(2)}$ are mistakenly identified with $\Omega^{(1)}$, while some vectors arising from $\Omega^{(3)}$ are associated with $\Omega^{(4)}$. A true measure of the performance of the classifier is hence the extent (area) of the hatched area of erroneous classification, suitably weighted by appropriate *a priori* probabilities. Performance, in general, would be deemed to improve if the hatched area decreases in size. More generally, optimum classifier performance is obtained if classification is according to the largest *a posteriori* probability, $\mathbf{P}\left\{\Omega^{(s)} \mid \mathbf{x}\right\}$. This results in the *Bayes classifier*, which unfortunately, takes on a simple form only in exceptional cases. In general, for discriminant functions of the fixed parametric form that we consider, the resultant classification performance will be suboptimal. The probability of error for such classifiers tells us the extent to which we have sacrificed optimality by electing to look at relatively simple parametric structures for discriminant functions.

We generalise this approach to allow of arbitrary state-conditional distributions and *a priori* probabilities for the states of nature.

Let C be a classifier, and let $R_C = \left\{ R^{(1)}, \ldots, R^{(c)} \right\}$ be the partition of $\mathbb{H}_p$ induced by C. Let the state-conditional probability distributions be denoted by $F\,(I \mid \Omega_s) = \mathbf{P}\left\{ \mathbf{x} \in I \mid \Omega^{(s)} \right\}$ for measurable sets $I \subseteq \mathbb{H}_p$, and let $\pi^{(s)} = \mathbf{P}\left\{ \Omega^{(s)} \right\}$ denote the *a priori* probabilities of occurrence of the various states of nature. Then the *probability of (classification) error*, P_e, is defined as

$$P_e \triangleq 1 - \sum_{s=1}^{c} \pi^{(s)} \int_{R^{(s)}} dF\,(\mathbf{x} \mid \Omega^{(s)})\,. \tag{1.4.1}$$

Each term in the sum is just the probability that a particular state of nature occurs, *and* a pattern vector in the decision region $R^{(s)}$ results. Thus the entire sum is the probability, averaged over the states of nature, that classifying patterns according to the decision regions $R^{(s)}$ results in correct classification. The probability that an error occurs in classification is clearly one minus this probability.

The probability of error (or more precisely, the probability of correct classification) is the ultimate performance measure for classifiers. It tells us the expected losses that accrue from the usage of any particular classifier. (In decision-theoretic terminology, the probability of error is the expected risk corresponding to a $0-1$ loss function.)

Proposition 1.4.2. $1-P_e$ is a totally consistent performance measure.

Proof. The proof follows directly from equation (1.4.1), and proposition (1.4.1). □

The probability of error hence imposes an absolute linear order on classifiers, and the goodness of other performance measures depends on how closely they approximate P_e. In practice, however, P_e is frequently too difficult to calculate unless the underlying distributions are cooperative. We hence introduce two more measures of performance.

C. Bhattacharyya Coefficient

We will restrict ourselves to the two-class case for simplicity in the ensuing exposition. Let $\Omega^{(1)}$ and $\Omega^{(2)}$ be the two states of nature with *a priori* probabilities $\pi^{(1)}$ and $\pi^{(2)}$, respectively. Let δ be a discriminant function with the classification rule:

$$\begin{aligned} \mathbf{x} &\mapsto \Omega^{(1)} \quad \text{if } \delta(\mathbf{x}) \geq 0 \\ \mathbf{x} &\mapsto \Omega^{(2)} \quad \text{if } \delta(\mathbf{x}) < 0 \end{aligned}$$

(Note that for the two-class case, it suffices to consider a single discriminant function. Specifically, if $\delta^{(1)}$ and $\delta^{(2)}$ discriminant functions as in (1.2.1), then set $\delta = \delta^{(1)} - \delta^{(2)}$.)

The discriminant function δ is a random variable whose distribution is conditioned upon the states of nature $\Omega^{(1)}$ and $\Omega^{(2)}$. Assume the class-conditional probability density functions $p(v \mid \Omega^{(s)})$, $s=1,2$, exist for δ. Then the *Bhattacharyya coefficient* ρ_B is defined by [4]

$$\rho_B \triangleq (\pi^{(1)}\pi^{(2)})^{1/2} \int_{-\infty}^{\infty} \{p\,(v \mid \Omega^{(1)})\, p\,(v \mid \Omega^{(2)})\}^{1/2}\, dv \ . \tag{1.4.2}$$

In actuality, the Bhattacharyya coefficient ranks classifiers in inverse order. An appropriate performance measure ranking classifiers in their proper order is the *Bhattacharyya distance* $d_B = -\ln \rho_B$. As there is a 1–1 correspondence between d_B and ρ_B, however, we will continue to use ρ_B as a performance measure with the understanding that classifier ranking is reversed.

The Bhattacharyya coefficient is much simpler to compute than the probability of error as it does not require the explicit specification of the various decision regions. Furthermore, it is a good performance measure in the sense that it bounds the probability of error fairly tightly.

Proposition 1.4.3. $\frac{1}{2}\,(\,1 - \sqrt{1 - 4\rho_B^2}) \leq P_e \leq \rho_B$.

Proof. The proof is as in Ref. [5] with a slightly different definition of ρ_B .

Proposition 1.4.4. ρ_B is a monotonically consistent performance measure.

Proof. Let δ be a discriminant function with Bhattacharyya coefficient ρ_B given by equation (1.4.2). Let $f : \mathbb{R} \rightarrow \mathbb{R}$ be a monotonically increasing function. For simplicity assume f^{-1} is differentiable. Let $p^*(\,v \mid \Omega^{(s)}\,)$ denote the probability density function of $f \circ \delta$ conditioned upon the presence of $\Omega^{(s)}$ for $s = 1,2$, and let ρ_B^* denote the associated Bhattacharyya coefficient. We have

$$p^*(\,v \mid \Omega^{(s)}\,) = \frac{df^{-1}(v)}{dv}\, p\,(f^{-1}(v) \mid \Omega^{(s)}\,)\ .$$

The support of $f \circ \delta$ is contained in the open interval $(f\,(-\infty), f\,(\infty))$. Hence,

$$\rho_B^* = (\pi^{(1)}\pi^{(2)})^{1/2} \int_{f\,(-\infty)}^{f\,(\infty)} \{p^*(v \mid \Omega^{(1)})\, p^*(v \mid \Omega^{(2)})\}^{1/2}\, dv$$

$$= (\pi^{(1)}\pi^{(2)})^{1/2} \int_{f(-\infty)}^{f(\infty)} \{p(f^{-1}(v) \mid \Omega^{(1)})\, p(f^{-1}(v) \mid \Omega^{(2)})\}^{1/2} \frac{df^{-1}(v)}{dv}\, dv$$

$$= (\pi^{(1)}\pi^{(2)})^{1/2} \int_{-\infty}^{\infty} \{p(v \mid \Omega^{(1)})\, p(v \mid \Omega^{(2)})\}^{1/2}\, dv$$

$$= \rho_B \ . \qquad \square$$

The usage of Bhattacharyya coefficient as a performance measure, hence results in a consistent ranking of classifiers which is not too far removed from the optimal classifier ordering by means of the probability of error as performance criterion. In fact, it can be shown that P_e approaches ρ_B in the limit of low probabilities of error [5], so that in the region of low errors which we are primarily interested in, ρ_B is actually a very good performance measure.

D. Normalised Mean Separation

We will sometimes encounter situations where both the probability of error, and the Bhattacharyya coefficient are too cumbersome to compute. We hence introduce a variation of the signal-to-noise ratio defined in equation (1.2.4). Again, we consider the two-class case for simplicity, and assume the states of nature are equi-probable.

Let $\delta^{(s)}$, $s = 1,2$, be the discriminant functions for the two-class case. Define

$$\mu^{(s)} = \mathbf{E}\,\{\, |\, \delta^{(s)} \,|\, \}\ ,$$

and

$$\eta^{(s)} = \mathrm{Var}\,(\delta^{(s)})\ \ ,\ s = 1,2\ . \qquad (1.4.3)$$

We now define the *normalised mean separation*, ρ, by

$$\rho = \frac{(\mu^{(1)} - \mu^{(2)})^2}{\eta^{(1)} + \eta^{(2)}} . \tag{1.4.4}$$

The normalised mean separation is related to the signal-to-noise ratio defined in equation (1.2.4). Specifically, the normalised mean separation ρ is one-half the signal-to-noise ratio when the two states of nature are the presence and the absence of a deterministic signal in an additive noise environment.

The performance coefficient, ρ, is not consistent in either of the two senses we have defined. Hence, its efficacy is likely to be limited to ordering classifier performance as parameters are changed within *fixed* functional forms of the discriminant function. However, it has the virtue of simplicity, and can be easily computed in most cases. Besides, under limited circumstances, ρ provides a reasonably accurate measure of performance.

Let δ denote the discriminant function, with pattern vector $\mathbf{x}$ being assigned to $\Omega^{(1)}$ if $\delta(\mathbf{x}) \geq 0$, and to $\Omega^{(2)}$ if $\delta(\mathbf{x}) < 0$. It is intuitively clear that with such a decision rule classification performance improves if the mean difference between the correlation peaks of the two classes, $\mu^{(1)} - \mu^{(2)}$ is made large, and the peak side-lobe variances, $\eta^{(1)}$ and $\eta^{(2)}$ are small; specifically, for such a case we can find a suitable (optimum) threshold t (nominally between $\mu^{(1)}$ and $\mu^{(2)}$) which is several standard deviations from both $\mu^{(1)}$ and $\mu^{(2)}$. The Bayesian risk–or more specifically, the probability of erroneous classification–hence tends to decrease when the average inter-correlation peak, $\mu^{(1)} - \mu^{(2)}$, increases and the side-lobe variances, $\eta^{(1)}$ and $\eta^{(2)}$, decrease. As an instance, the choice of an optimum threshold t for the case of two unimodal probability density functions $f_{\delta^{(1)}}$ and $f_{\delta^{(2)}}$ is illustrated in the schema of fig. 1.6: the probability of error (assuming equal *a priori* class probabilities) is proportional to the area of the shaded region in the figure.

The coefficient ρ defined by equation (1.4.4) increases monotonically with increase in the average peak separation, and decrease in side-lobe variance; in this regard then, the behaviour of ρ is similar to that of the probability of error, P_e, so that ρ is a suitable performance measure. It must be noted however that the

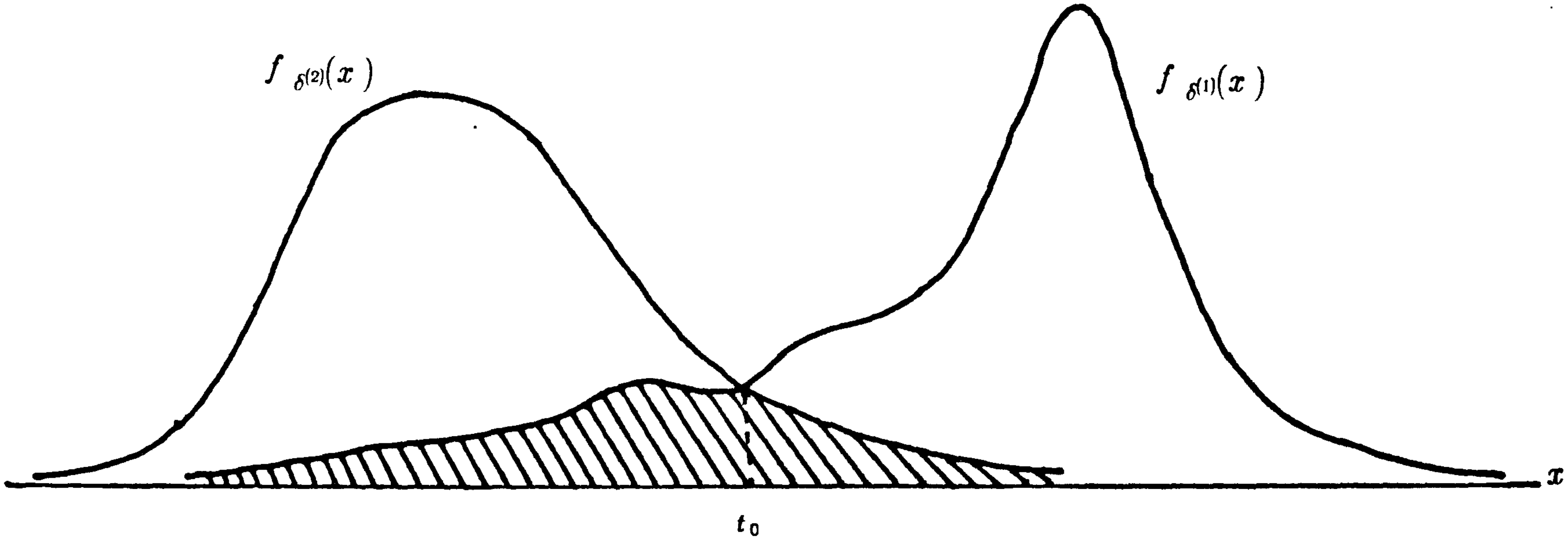

Fig. 1.6. A choice of optimum threshold, t_0, given the two class-conditional densities $f_{\delta^{(1)}}(x)$, and $f_{\delta^{(2)}}(x)$. The shaded area is the minimum attainable probability of misclassification.

performance coefficient ρ is an ***ad hoc*** measure that we adopt because of its simplicity and the sustaining arguments above. In the general case–and especially in the instance of multi-modal densities–it is not necessary that the probability of error, P_e, be expressible as a monotone function of ρ. A simple case where P_e is indeed a monotone function of ρ is when the system outputs conditioned on the two classes, $G_j, j = 1,2$, are Gaussian with equal variances ν. Then, assuming equal *a priori* class probabilities,

$$P_e = \Phi \left(\frac{-\sqrt{\rho}}{2} \right),$$

where $\Phi(x)$ is the cumulative Gaussian distribution function

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{y^2}{2}} dy \ .$$

As we shall see later (cf. [5] also) the Bhattacharyya coefficient is also representable as a monotone (decreasing) function of ρ when the class-conditional distributions of δ are Gaussian, and the *a priori* probabilities of the two classes are the same.

5. Organisation

The dissertation is organised into four sections: Problem Methodology, tackled in the introductory chapter; a section on Correlators (chapters II, III, and IV) dealing with particular applications of the proposed generalised linear discriminant functions to specific problems in image and pattern recognition; a section on Neural Networks (chapters V, VI, VII, and VIII) considering problems in associative memory under a suitable iterated map; and finally, a concluding section (chapters IX and X) detailing some extensions and open problems.

In chapter II we characterise the statistics of multi-channel classifiers realising generalised linear discriminant functions using point rules in an m-dimensional feature space. The states of nature we consider are square-integrable functions in an additive, random noise environment. We introduce the concept of *independent channels*, and obtain necessary and sufficient conditions for realising independent channels. We also characterise discriminant function statistics in some detail when the point rules are chosen to be square-law of threshold.

In chapter III we consider the problem of achieving image recognition invariant to rotations and shifts of images. We demonstrate that square-law point rules in conjunction with suitably chosen linear transformations yield discriminant functions which are invariant to image rotation, and obtain the general form for such systems. Applying the derived statistics from chapter II, we demonstrate the existence of optimal rotation invariant image recognition systems with respect to the performance measure ρ_B. We also demonstrate good sub-optimal rotation invariant classifiers, and characterise the performance sacrificed to gain rotation invariance.

In chapter IV we consider the usage of point threshold rules in conjunction with linear maps to realise certain classes of binary filters which yield considerable savings in system complexity and cost. We demonstrate that these classes of binary filters also yield very satisfactory performance.

Chapter V introduces the form of associative memory structure that we consider, and elucidates neurobiological terminology, and notation. We make precise the notion of capacity of these structures, and identify desired properties and parameters.

In chapter VI we analyse in depth a particular algorithm for memory storage based on the outer products of the desired memories. We demonstrate that the dynamics of the algorithm are such as to emulate a physical content addressable memory, and provide heuristics to estimate its capacity. We then provide fundamental results with rigourous proofs estimating the storage capacity of the algorithm under a variety of preconditions.

In chapter VII we describe alternate algorithms for memory encoding based on spectral approaches which intrinsically store close to the ultimate capacity of the associative network structure itself. We describe various features of the spectral approach, and compare results with the outer product algorithm.

In chapter VIII we present the derivation of the maximal storage capacity of the associative network structure when all algorithms are allowed for consideration. The results bound the performance of any specified algorithm, and take into consideration specified tolerances of error.

Extensions of the basic neural networks structure as embodied in chapter VI through VIII, are considered in chapter IX, and particularly in regard to generalisations of the network to incorporate more communication, and computation, and the gains in capacity thereby, associative memory architectures using distributed non-linearities to compensate for specified distortions, and networks using binarised links. Chapter X concludes with some open problems and questions, and indicates possible lines of research.

References

[1] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis.* New York: Wiley, 1973.

[2] R.O. Winder, "Bounds on threshold gate realizability," *IEEE Trans. Elec. Comp.*, vol. EC-12, pp. 561–564, 1963.

[3] G. E. Hinton and J. A. Anderson (eds.), *Parallel Models of Associative Memory.* Hillsdale, New Jersey: Lawrence-Erlbaum, 1981.

[4] A. Bhattacharyya, "On a measure of divergence between two statistical populations defined by their probability distributions," *Bull. Calcutta Math. Soc.*, vol. 35, pp. 99–109, 1943.

[5] T. Kailath, "The divergence and Bhattacharyya distance measures in signal selection," *IEEE Trans. Comm. Tech.*, vol. COM-15, pp. 52–60, 1967.

Correlators

CHAPTER II

MULTI-CHANNEL MACHINES

1. FINITE-DIMENSIONAL FEATURE SPACES

In this chapter we characterise the statistics of generalised linear discriminant functions with point rules of the form (1.2.5) on a finite number of channels. Two particular instances of point rules that we analyse in some detail are square-law and threshold point rules. We will consider applications of these structures to classification problems in the following chapters.

A. System Structure

The pattern space we consider is a Hilbert space, H, of square-integrable, complex-valued functions, with inner product (,). For definiteness we consider functions $f(x,y)$ of two Cartesian coordinates, with the natural inner product. With each of c states of nature, $\Omega^{(s)}$, we identify functions $f^{(s)}$ from some subset of H. The functions (signals) $f^{(s)}$ are assumed to exist in a random, additive noise environment which determines the class-conditional distributions of pattern vectors in H.

We dub the procedure for realising a generalised linear discriminant function of the form $\mathbf{L} \circ \mathbf{D} \circ \mathbf{W}$ described in the previous chapter, a *multi-channel machine.* A processor realising such a generalised linear discriminant function is illustrated schematically in fig. 2.1.

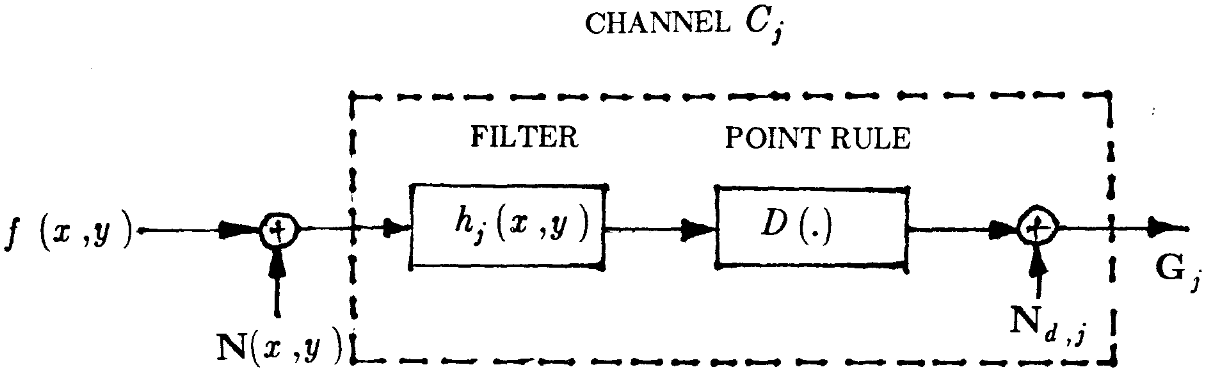

Fig. 2.1 (a). Schema for a single channel.

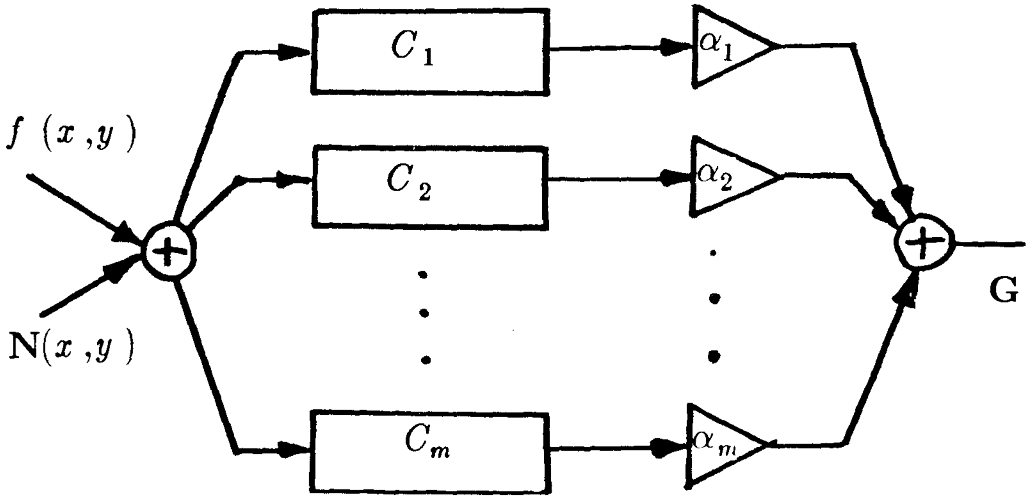

Fig. 2.1 (b). Multi-channel machine.

Pattern vectors f are mapped to points in the m-dimensional feature space $\mathbb{R}^m$ through the agency of a linear map $\mathbf{W} : H \rightarrow \mathbb{C}^m$, followed by a point rule $\mathbf{D} : \mathbb{C}^m \rightarrow \mathbb{R}^m$ as in equation (1.2.5). This, in essence, constitutes a dimensionality reduction procedure. The linear map $\mathbf{W}$ is represented by the m-vector of *channel impulse responses* $[h_1, \ldots, h_m]$, where the h_j's are square-integrable functions in H. Each component of the feature can be thought of as being realised by a single channel, C_j, comprising a linear filter h_j, and a (non-linear) map $\mathbf{D}$. The j-th feature component is hence a real scalar $D\left\{(h_j \ , f \ )\right\}$, when f is the input pattern. Fig. 2.1 (a) illustrates a block-diagrammatic channel realisation.

The discriminant functions of interest are finally realised by projecting feature vectors onto the real line through the linear discriminant functions $\mathbf{L}$ acting on the feature space. We represent $\mathbf{L}$ by the m-vector of weights $(\alpha_1, \ldots, \alpha_m)$. The schema of fig. 2.1 (b) illustrates the realisation of such a discriminant function. Here we have m channels in parallel producing the m components of the feature vector, and the components of the feature vector are then collapsed into a discriminant function through the weights $\alpha_1, \alpha_2, \ldots, \alpha_m$. The weighting factors, α_i, essentially specify the orientation of a hyperplane in the feature space. Note that as a consequence we have m degrees of freedom (corresponding to the m independent dimensions of the feature space) in choosing a generalised linear discriminant function once the feature space is specified.

B. Noise Considerations

If the channels are so specified that feature vectors corresponding to different classes are *linearly separable*, then in principle, in the absence of noise the weight vector can be so chosen as to separate the various classes with no error. (However, for very similar image classes this may call for considerable resolution in the devices used.) The presence of noise, however, causes a spread in the probability distributions of the outputs corresponding to each class. Under such conditions the classes are no longer linearly separable, and the best we can hope to do is minimise the probability of error by choosing channels and weights tailored to the problem.

We consider the following specific noise models at the input and the output of the multi-channel processor:

The input noise process, $\mathbf{N}(x,y)$, (as illustrated in fig. 2.1), is assumed to be wide-sense stationary, additive, white, and Gaussian. We assume $\mathbf{N}(x,y)$ has zero mean, and variance σ_n^2.

We assume some additive "detection" noise at the output of each channel, as indicated in fig. 2.1; the scalar feature at the output of the j-th channel is degraded by an additive Gaussian noise component $\mathbf{N}_{d,j}$. The sequence of random variables $\{\mathbf{N}_{d,j}\}_{j=1}^{m}$ is assumed to be independent, identically distributed, with mean μ_d, and variance σ_d^2.

We now characterise the generalised liner discriminant function obtained at the output of the multi-channel processor. We assume an input image $f \in \Omega$ corrupted by the additive noise process $\mathbf{N}(x,y)$. The multi-channel processor is completely specified by the m-tuple of impulse responses $(h_1,h_2,\ldots,h_m) \in H^m$, and the weights $(\alpha_1,\alpha_2, \ldots, \alpha_m) \in \mathbb{R}^m$. The output of each channel has an additive independent noise component $\mathbf{N}_{d,j}$. With reference to fig. 2.1, we can write the output, $\mathbf{G}_j$, of the j-th channel, for each $j=1,\ldots,m$, as

$$\mathbf{G}_j = D\Big((h_j, f + \mathbf{N})\Big) + \mathbf{N}_{d,j}$$

$$= D\Big(\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\Big(f(x,y) + \mathbf{N}(x,y)\Big)\,\overline{h}_j(x,y)\,dxdy\Big) + \mathbf{N}_{d,j}\ . \qquad (2.1.1)$$

The generalised linear discriminant function, $\mathbf{G}$, is hence

$$\mathbf{G} = \sum_{j=1}^{m} \alpha_j \mathbf{G}_j\ , \qquad (2.1.2)$$

where α_j is the weight for the j-th channel.

C. Statistically Independent Features

The inclusion of more (scalar) features (or equivalently, the usage of more channels), cannot degrade performance; at worst, we can discard non-informative or misleading features (by using a weight of zero) so that performance is unaffected. There is some theoretical evidence, however, supporting the intuitive fact that including more *statistically independent* features in the recognition algorithm improves performance (cf. [1], for example). We are hence motivated to characterise sets of channels which yield statistically independent features.

Definition. Let $\{C_1, C_2, ..., C_m\}$ be a set of m channels. Let $\mathbf{G}_j^{(s)}$ be the output of the j-th channel with input $f^{(s)} \in \Omega^{(s)}$, $s=1,...,c$. We say that $\{C_1, C_2, ..., C_m\}$ constitutes a set of *independent channels* if the random variables $\mathbf{G}_j^{(s)}$, $j=1,...,m$, are independent for each $s=1,...,c$.

We define $\varsigma_j^{(s)} \in \mathbb{C}$, $\eta_j \in \mathbb{R}$, and the complex random variables $\mathbf{N}_j$ for each $s=1,...,c$, and $j=1,...,m$, by

$$\varsigma_j^{(s)} \triangleq (h_j, f^{(s)}) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f^{(s)}(x,y)\,\overline{h}_j(x,y)\,dxdy\ , \tag{2.1.3}$$

$$\eta_j \triangleq \|h_j\|^2 = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |\,h_j(x,y)\,|^2\,dxdy\ . \tag{2.1.4}$$

Define

$$\mathbf{N}_j \triangleq \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathbf{N}(x,y)\,\overline{h}_j(x,y)\,dxdy\ . \tag{2.1.5}$$

With reference to equation (2.1.1), we then have the output of the j-th channel conditioned upon class $\Omega^{(s)}$ being present, given by

$$\mathbf{G}_j^{(s)} = D\Big(\varsigma_j^{(s)} + \mathbf{N}_j\Big) + \mathbf{N}_{d,j}$$

$$= D\Big(\mathrm{Re}\,(\varsigma_j^{(s)} + \mathbf{N}_j) + i\,\mathrm{Im}\,(\varsigma_j^{(s)} + \mathbf{N}_j)\Big) + \mathbf{N}_{d,j}\ .$$

(2.1.6)

We now state a necessary and sufficient condition on the channel impulse responses, h_j, corresponding to each channel C_j, so that the channels,C_j, $j=1,...,m$, are independent.

Theorem 2.1.1. Let $\{C_1,C_2,...,C_m\}$ be a set of m channels, and let impulse response h_j correspond to channel C_j. Then the channels, C_j, $j=1,...,m$, are independent if, and only if, the impulse responses h_j , $j=1,...,m$ satisfy

$$\int \mathrm{Re}\,(h_j)\,\mathrm{Re}\,(h_k) = \int \mathrm{Re}\,(h_j)\,\mathrm{Im}\,(h_k) = \int \mathrm{Im}\,(h_j)\,\mathrm{Re}\,(h_k) = \int \mathrm{Im}\,(h_j)\,\mathrm{Im}\,(h_k) = 0\,, \qquad (2.1.7)$$

for $j,k=1,...,m$ and $j \neq k$, and where all integrals are over the two-dimensional plane.

Proof. We need to show that the random variables $\mathbf{G}_j^{(s)}$ are jointly independent random variables if, and only if, the impulse responses h_j , $j=1,...,m$, satisfy (2.1.7). From a consideration of equation (2.1.6) this is equivalent to showing that for $j \neq k$ the complex random variables $\mathbf{N}_j$ and $\mathbf{N}_k$ are independent if, and only if, the impulse responses are as in the statement of the theorem. (This follows because the random variables $\mathbf{N}_{d,j}$ are independent, and the $\varsigma_j^{(s)}$'s are deterministic quantities). Note that it suffices to show pairwise independence, as the random variables $\mathbf{N}_j$ are jointly Gaussian.

Proof of 'if' part: assume the impulse responses are as in the theorem. From equation (2.1.5) we have for each $j=1,...,m$

$$\mathbf{N}_j = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathrm{N}(x,y)\,\mathrm{Re}\left(h_j(x,y)\right)\,dxdy$$

$$- i\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathrm{N}(x,y)\,\mathrm{Im}\left(h_j(x,y)\right)\,dxdy\ . \qquad (2.1.8)$$

Hence $\mathrm{Re}(\mathbf{N}_j)$ and $\mathrm{Im}(\mathbf{N}_j)$ are the outputs of linear systems with impulse responses $\mathrm{Re}\{h_j(x,y)\}$, and $\mathrm{Im}\{h_j(x,y)\}$. As linear transformations of normal processes are also normal, we have that $\mathrm{Re}(\mathbf{N}_j)$ and $\mathrm{Im}(\mathbf{N}_j)$ are also Gaussian random variables. As $\mathbf{N}(x,y)$ is Gaussian, white, and has zero mean and variance σ_n^2, we have

$$\mathbf{E}\left(\mathrm{Re}(\mathbf{N}_j)\right) = \mathbf{E}\left(\mathrm{Im}(\mathbf{N}_j)\right) = 0\,, \tag{2.1.9}$$

$$\mathrm{Var}\left(\mathrm{Re}\,(\mathbf{N}_j)\right) = \sigma_n^2 \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left(\mathrm{Re}\,(h_j(x,y))\right)^2 dxdy\ . \tag{2.1.10}$$

$$\mathrm{Var}\left(\mathrm{Im}\,(\mathbf{N}_j)\right) = \sigma_n^2 \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left(\mathrm{Im}\,(h_j(x,y))\right)^2 dxdy\ . \tag{2.1.11}$$

We now prove that $\mathrm{Re}(\mathbf{N}_j)$ and $\mathrm{Re}(\mathbf{N}_k)$ are independent. On account of their normality it suffices to show that they are uncorrelated. Now

$$\mathbf{E}\left(\mathrm{Re}\,(\mathbf{N}_j)\,\mathrm{Re}\,(\mathbf{N}_k)\right)$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathbf{E}\left(\mathrm{N}(x_1,y_1)\,\mathrm{N}(x_2,y_2)\right) \mathrm{Re}\left(h_j(x,y)\right) \mathrm{Re}\left(h_k(x,y)\right) dxdy$$

$$= \sigma_n^2 \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathrm{Re}\left(h_j(x,y)\right) \mathrm{Re}\left(h_k(x,y)\right) dxdy = 0\,,$$

as $\mathrm{N}(x,y)$ is white, with $\mathbf{E}\left(\mathrm{N}(x_1,y_1)\mathrm{N}(x_2,y_2)\right) = \sigma_n^2\delta(x_1-x_2,y_1-y_2)$, and $\int \mathrm{Re}(h_j)\,\mathrm{Re}(h_k) = 0$ by assumption.

So $\mathrm{Re}(\mathbf{N}_j)$, and $\mathrm{Re}(\mathbf{N}_k)$ are independent if $j \neq k$. The above argument can be repeated almost *in toto* to prove that $\mathrm{Re}(\mathbf{N}_j)$, and $\mathrm{Im}(\mathbf{N}_k)$ are independent. Hence $\mathrm{Re}(\mathbf{N}_j)$, and $\mathbf{N}_k$ are independent. In very similar manner it can be shown that $\mathrm{Im}(\mathbf{N}_j)$, and $\mathbf{N}_k$ are independent, and hence $\mathbf{N}_j$, and $\mathbf{N}_k$ are independent.

Hence $\mathbf{G}_j$ is independent of $\mathbf{G}_k$ if $j \neq k$.

To prove the 'only if' part: assume $\left\{\mathbf{G}_j\right\}$ is a sequence of independent random variables. Then $\mathrm{Re}(\mathbf{N}_j)$, and $\mathrm{Re}(\mathbf{N}_k)$ are independent if $j \neq k$. So

$$\mathbf{E}\left\{\mathrm{Re}(\mathbf{N}_j)\,\mathrm{Re}(\mathbf{N}_k)\right\} = \mathbf{E}\left\{\mathrm{Re}(\mathbf{N}_j)\right\}\mathbf{E}\left\{\mathrm{Re}(\mathbf{N}_k)\right\} = 0 .$$

But from the proof of the 'if' part we have

$$\mathbf{E}\left\{\mathrm{Re}(\mathbf{N}_j)\,\mathrm{Re}(\mathbf{N}_k)\right\} = \sigma_n^2 \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathrm{Re}\Big(h_j(x,y)\Big)\,\mathrm{Re}\Big(h_k(x,y)\Big)\,dxdy \;,$$

so

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathrm{Re}\Big(h_j(x,y)\Big)\,\mathrm{Re}\Big(h_k(x,y)\Big)\,dxdy = 0 \,, \quad \text{if } j \neq k \;.$$

The proof of the remaining parts of equation (2.1.7) follows similarly. □

We indicate a specialisation of the theorem to two particular cases which we will use in the ensuing chapters.

Corollary 2.1.1. Let $\left\{C_1, C_2, \ldots, C_m\right\}$ be a set of channels. Assume the channel impulse responses h_j are two-dimensional functions, which in polar coordinates are of the form

$$h_j = \hat{h}_{j,k_j}(r)e^{ik_j\theta} \,, \quad k_j \in \mathbb{Z} \,,\; j=1,\ldots,m \tag{2.1.12}$$

with $k_j \neq k_l$ if $j \neq l$. Then $\left\{C_1, C_2, \ldots, C_m\right\}$ is a set of independent channels.

Proof. From equation (2.1.12) we have

$$\mathrm{Re}\left\{h_j\right\} = \mathrm{Re}\left\{\hat{h}_{j,k_j}(r)\right\}\cos k_j\theta - \mathrm{Im}\left\{\hat{h}_{j,k_j}(r)\right\}\sin k_j\theta$$

$$\mathrm{Im}\{h_j\} = \mathrm{Re}\{\hat{h}_{j,k_j}(r)\} \sin k_j \theta - \mathrm{Im}\{\hat{h}_{j,k_j}(r)\} \cos k_j \theta \ .$$

We can hence write

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathrm{Re}\{h_j(x,y)\}\,\mathrm{Re}\{h_l(x,y)\}\; dxdy \;= I_1 + I_2 + I_3 + I_4 \ ,$$

where

$$I_1 = \int_0^{\infty} \mathrm{Re}\{\hat{h}_{j,k_j}(r)\}\,\mathrm{Re}\{\hat{h}_{l,k_l}(r)\}\; rdr \int_0^{2\pi} \cos k_j \theta \cos k_l \theta \; d\,\theta = 0 \ ,$$

$$I_2 = -\int_0^{\infty} \mathrm{Re}\{\hat{h}_{j,k_j}(r)\}\,\mathrm{Im}\{\hat{h}_{l,k_l}(r)\}\; rdr \int_0^{2\pi} \cos k_j \theta \sin k_l \theta \; d\,\theta = 0 \ ,$$

$$I_3 = -\int_0^{\infty} \mathrm{Im}\{\hat{h}_{j,k_j}(r)\}\,\mathrm{Re}\{\hat{h}_{l,k_l}(r)\}\; rdr \int_0^{2\pi} \sin k_j \theta \cos k_l \theta d\,\theta = 0 \ ,$$

$$I_4 = \int_0^{\infty} \mathrm{Im}\{\hat{h}_{j,k_j}(r)\}\,\mathrm{Im}\{\hat{h}_{l,k_l}(r)\}\; rdr \int_0^{2\pi} \sin k_j \theta \sin k_l \theta \; d\,\theta = 0 \ .$$

Hence $\int \mathrm{Re}\{h_j\}\ \mathrm{Re}\{h_l\} = 0$ if $j \neq l$. The remaining parts of equation can be shown to hold true in similar fashion for impulse responses of the form (2.1.12). Hence, by theorem (2.2.1), the channels $\{C_1, C_2, \ldots, C_m\}$ defined by equation (2.1.12) are independent. □

Sets of channels satisfying the hypotheses of corollary (2.1.1) satisfy a further independence property which is useful in computing output statistics. We formally state this property in the following result.

Proposition 2.1.1. For each $j = 1, \ldots, m$, the random variables $\mathrm{Re}\{\mathbf{N}_j\}$, and $\mathrm{Im}\{\mathbf{N}_j\}$ are independent for channels satisfying the property of corollary (2.1.1).

Proof. From the proof of theorem (2.1.1), we have that $\mathrm{Re}\{\mathbf{N}_j\}$ and $\mathrm{Im}\{\mathbf{N}_j\}$ are Gaussian random variables, and hence it suffices to show that they are uncorrelated. From equation (2.1.8)

$$\mathrm{E}\{\mathrm{Re}(\mathbf{N}_j)\,\mathrm{Im}(\mathbf{N}_j)\} = I_1 + I_2 + I_3 + I_4$$

where

$$I_1 = \int_0^{\infty} \left(\mathrm{Re}\{\hat{h}_{j,k_j}(r)\}\right)^2 r dr \int_0^{2\pi} \cos k_j\theta \sin k_j\theta \, d\theta = 0 ,$$

$$I_2 = \int_0^{\infty} \mathrm{Re}\{\hat{h}_{j,k_j}(r)\}\, \mathrm{Im}\{\hat{h}_{j,k_j}(r)\}\, r dr \int_0^{2\pi} \cos^2 k_j\theta \, d\theta$$

$$= \pi \int_0^{\infty} \mathrm{Re}\{\hat{h}_{j,k_j}(r)\}\, \mathrm{Im}\{\hat{h}_{j,k_j}(r)\}\, r dr ,$$

$$I_3 = -\int_0^{\infty} \mathrm{Im}\{\hat{h}_{j,k_j}(r)\}\, \mathrm{Re}\{\hat{h}_{j,k_j}(r)\}\, r dr \int_0^{2\pi} \sin^2 k_j\theta \, d\theta$$

$$= -\pi \int_0^{\infty} \mathrm{Re}\{\hat{h}_{j,k_j}(r)\}\, \mathrm{Im}\{\hat{h}_{j,k_j}(r)\}\, r dr ,$$

$$I_4 = -\int_0^{\infty} \left(\mathrm{Im}\{\hat{h}_{j,k_j}(r)\}\right)^2 r dr \int_0^{2\pi} \sin k_j\theta \cos k_j\theta \, d\theta = 0 .$$

And hence

$$\mathrm{E}\{\mathrm{Re}(\mathbf{N}_j)\mathrm{Im}(\mathbf{N}_j)\} = 0 .$$

From equation (2.1.9), $\mathrm{Re}\{\mathbf{N}_j\}$ and $\mathrm{Im}\{\mathbf{N}_j\}$ have zero mean, and hence they are uncorrelated. □

We will have occasion to utilise channels with impulse responses of the form (2.1.12) in a consideration of image classification systems which are insensitive to image rotation. Another class of useful impulse responses resulting in independent channels obtains on consideration of single lines of raster scanned images.

Corollary 2.1.2. Let H be the set of functions $f : \mathbb{R} \times \mathbb{Z} \rightarrow \mathbb{R}$, with $||f||^2 = \sum_k \int (f(x,k))^2 \, dx < \infty$. Let $\{C_1, \ldots, C_m\}$ be a set of channels with impulse responses h_j given by

$$h_j(x,k) = h(x,k)\,\delta_{jk} \quad , \quad j = 1,\ldots,m \quad , \tag{2.1.13}$$

where $h_j \in H$. Then $\left\{C_1, \ldots, C_m\right\}$ is a set of independent channels.

Proof. As the functions h_j are real, it suffices to show that $(\,h_j \ , h_l\,) = 0$, with the inner product defined as in the corollary. We hence have

$$\sum_k \int h_j(x,k)\, h_l(x,k)\; dx = \sum_k \int \Big(h(x,k)^2\, \delta_{jk}\ \delta_{lk}\ dx = 0 \ .$$

It follows that the channels are independent by theorem (2.1.1). □

Channels with impulse responses as in corollary (2.1.2) occur naturally in certain optical signal processing systems wherein two-dimensional signals are processed through the agency of high quality one-dimensional optical devices. Such systems–sometimes termed *pseudo-correlators* because of the nonlinear correlations they perform–are interesting examples of quadratic machines which are comparable to Matched Filters in classification capability [2]. Their statistical analysis, however, is similar to that for the class of rotation invariant classification systems which we consider in the next chapter. We will not explore the topic further in this monograph.

By virtue of the point rules being local to every channel, the characterisation of independent channels–leading to independent feature components–could be made solely in terms of the channel impulse responses, h_j, irrespective of the actual nature of the point rule **D**. The determination of the class-conditional probability distributions of the discriminant functions, however, is strongly dependent on the nature of the point rule **D**. In the next two sections we consider two simple point rules–square-law and threshold–and obtain expressions for discriminant statistics for these two cases.

2. QUADRATIC MACHINES

A. Square-law Point Rules

For each channel, we consider the case of a square-law point operation with $D(z_j) = |z_j|^2$. The point rule **D** hence utilises a quadratic map in each channel, and in analogy with the terminology, linear machine, we christen the resultant processor a *quadratic machine*. Examples of quadratic machines include a class of rotation invariant image classifiers which we analyse in the next chapter, and the pseudocorrelator–an optical nonlinear correlator.

Substituting for D in equation (2.1.1) we obtain

$$\mathbf{G}_j = \left| \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Big(f(x,y) + \mathrm{N}(x,y) \Big) \, \overline{h}_j(x,y) \, dxdy \right|^2 + \mathbf{N}_{d,j}$$

$$= \mathrm{Re}(\varsigma_j + \mathbf{N}_j) + i \; \mathrm{Im}(\varsigma_j + \mathbf{N}_j) \qquad (2.2.1)$$

where ς_j is the inner product between f and h_j, and $\mathbf{N}_j$ is the inner product between **N**, and h_j. The discriminant function **G** is given by (2.1.1). We will assume that the channel impulse responses h_j satisfy theorem (2.1.1), so that the channels are independent.

B. Single Channel Statistics

In this section we consider the statistics at the output of a single channel. We obtain here an expression for the probability density function (pdf) of the output, $\mathbf{G}_j^{(s)}$, of channel C_j, conditioned upon class $\Omega^{(s)}$ being present at the input of the channel.

Assume the channel impulse responses h_j are as in corollary (2.1.1). Set

$$X_j^{(s)} = \mathrm{Re}\left\{ \varsigma_j^{(s)} + \mathbf{N}_j \right\} ,$$

$$Y_j^{(s)} = \mathrm{Im}\left\{\varsigma_j^{(s)} + \mathbf{N}_j\right\} .$$

The impulse responses, h_j , $j=1,...,m$, of the multi-channel processor are as in corollary (2.1.1), and are of the form (equation (2.1.12))

$$h_j = \hat{h}_{j,k_j}(r)e^{ik_j\theta} , \quad k_j \in \mathbb{Z}$$

and such that $k_j \neq k_l$ if $j \neq l$. Hence

$$||\mathrm{Re}\ h_j||^2 = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left(\mathrm{Re}\left\{h_j(x,y)\right\}\right)^2 dxdy$$

$$= \int_{0}^{\infty}\int_{0}^{2\pi} \left(\mathrm{Re}\left\{\hat{h}_{j,k_j}(r)\right\} \cos k_j\theta - \mathrm{Im}\left\{\hat{h}_{j,k_j}(r)\right\} \sin k_j\theta\right)^2 r d\theta\ dr\ .$$

After some computation we have

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left(\mathrm{Re}\left\{h_j(x,y)\right\}\right)^2 dxdy = \frac{1}{2}\eta_j , \qquad (2.2.2)$$

where η_j is the energy of the function h_j, as defined in equation (2.1.1). Similarly,

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left(\mathrm{Im}\left\{h_j(x,y)\right\}\right)^2 dxdy = \frac{1}{2}\eta_j . \qquad (2.2.3)$$

From equations (2.1.10), and (2.1.11), we then have that $X_j^{(s)} \sim N(\mathrm{Re}\ \varsigma_j^{(s)},\sigma_n^2\eta_j/2)$, and $Y_j^{(s)} \sim N(\mathrm{Im}\ \varsigma_j^{(s)},\sigma_n^2\eta_j/2)$. Furthermore, by proposition (2.1.1), they are also independent. Hence the joint probability density function, $p_{X_j^{(s)}Y_j^{(s)}}(x,y)$, of the jointly Gaussian independent random variables $X_j^{(s)}$, and $Y_j^{(s)}$ is given by

$$p_{X_j^{(s)}Y_j^{(s)}}(x,y) = \frac{1}{\pi\sigma_n^2\eta_j} \exp\left\{-\frac{1}{\sigma_n^2\eta_j}\left[(x - \mathrm{Re}\ \varsigma_j^{(s)})^2 + (y - \mathrm{Im}\ \varsigma_j^{(s)})^2\right]\right\} .$$

If output noise is absent we have

$$\mathbf{G}_j^{(s)} = (X_j^{(s)})^2 + (Y_j^{(s)})^2 \, .$$

Then

$$\mathbf{P}\left\{\mathbf{G}_j^{(s)} \leq v\right\} = \mathbf{P}\left\{(X_j^{(s)})^2 + (Y_j^{(s)})^2 \leq v\right\}$$

$$= \iint_{x^2+y^2 \leq v} p_{X_j^{(s)} Y_j^{(s)}}(x, y)\, dxdy \, .$$

Set

$$x = r \cos\theta \quad \text{and} \quad y = r \sin\theta \, .$$

Then

$$\mathbf{P}\left\{\mathbf{G}_j^{(s)} \leq v\right\} = \frac{1}{\pi\sigma_n^2\eta_j} \int_0^{\sqrt{v}} \int_0^{2\pi} \exp\left\{-\frac{1}{\sigma_n^2\eta_j}\left[(r \cos\theta - \mathrm{Re}\, \varsigma_j^{(s)})^2\right.\right.$$

$$\left.\left. + (r \sin\theta - \mathrm{Im}\, \varsigma_j^{(s)})^2\right]\right\} rdrd\,\theta$$

$$= \frac{1}{\pi\sigma_n^2\eta_j} \int_0^{\sqrt{v}} r \exp\left\{-\frac{1}{\sigma_n^2\eta_j}(r^2 + |\varsigma_j^{(s)}|^2)\right\}$$

$$\times \int_0^{2\pi} \exp\left\{\frac{2r}{\sigma_n^2\eta_j}(Re\ \varsigma_j^{(s)} \cos\theta + \mathrm{Im}\, \varsigma_j^{(s)} \sin\theta)\right\} d\,\theta dr \, .$$

Set

$$\tan\phi = \frac{\mathrm{Im}\, \varsigma_j^{(s)}}{\mathrm{Re}\, \varsigma_j^{(s)}} \, .$$

Then

$$\mathbf{P}\left\{\mathbf{G}_j^{(s)} \leq v\right\} = \frac{1}{\pi\sigma_n^2\eta_j} \int_0^{\sqrt{v}} r \exp\left\{-\frac{1}{\sigma_n^2\eta_j}(r^2 + |\varsigma_j^{(s)}|^2)\right\}$$

$$\times \int_0^{2\pi} \exp\left\{\frac{2r \mid \varsigma_j^{(s)} \mid}{\sigma_n^2 \eta_j} \cos(\theta - \phi)\right\} d\theta dr \ .$$

The modified Bessel function of 0-th order is defined by

$$I_0(x) = \frac{1}{2\pi} \int_0^{2\pi} \exp(x \ \cos\theta) \ d\theta \ .$$

Hence

$$\mathbf{P}\ \{\mathbf{G}_j^{(s)} \leq v\} = \frac{2}{\sigma_n^2 \eta_j} \int_0^{\sqrt{v}} r \ \exp\left\{-\frac{1}{\sigma_n^2 \eta_j} (r^2 + \mid \varsigma_j^{(s)} \mid^2)\right\} I_0\left(\frac{2r \mid \varsigma_j^{(s)} \mid}{\sigma_n^2 \eta_j}\right) dr \ ,$$

for $v \geq 0$. Hence, we obtain

$$p_{\mathbf{G}_j^{(s)}} = \frac{d}{dv} \mathbf{P}\ \{\mathbf{G}_j^{(s)} \leq v\}$$

$$= \frac{1}{\sigma_n^2 \eta_j} \exp\left\{-\frac{1}{\sigma_n^2 \eta_j} (v + \mid \varsigma_j^{(s)} \mid^2)\right\} I_0\left(\frac{2 \mid \varsigma_j^{(s)} \mid}{\sigma_n^2 \eta_j}\right) U(v) \ ,$$

where

$$U(v) = \begin{cases} 1 \ , & \text{if } v \geq 0 \\ 0 \ , & \text{if } v < 0 \end{cases} .$$

The statistics for each channel can thus be completely characterised under the proviso that output noise is absent, and that the channel impulse responses satisfy some orthogonality constraints. In the presence of output noise we are perforce constrained to include a convolution integral because of the addition of an independent noise term at the channel output; we obtain the class-conditional pdf, $p_j^{(s)}(v)$, of $\mathbf{G}_j^{(s)}$ to be

$$p_j^{(s)}(v) = A(v) \int_0^{\infty} \exp\left\{-\left(u - \beta(v)\right)^2\right\} \mathrm{I}_0(\alpha\sqrt{u}) \ du$$

(2.2.4)

where

$$\alpha = \frac{2^{5/4} \mid \varsigma_j^{(s)} \mid \sqrt{\sigma_d}}{\eta_j \sigma_n^2} ,$$

$$\beta(v) = \frac{1}{\sqrt{2}\sigma_d} \left[v - \mu_d - \frac{\sigma_d^2}{\eta_j \sigma_n^2} \right] ,$$

$$A(v) = \frac{\exp \left[\frac{-1}{\eta_j \sigma_n^2} \left\{ v + \mid \varsigma_j^{(s)} \mid^2 - \mu_d - \frac{\sigma_d^2}{2\eta_j \sigma_n^2} \right) \right]}{\sqrt{\pi}\eta_j \sigma_n^2} ,$$

and $I_0(\cdot)$ is the modified Bessel function of 0-th order.

The integral of equation (2.2.4) is analytically intractable, and numerical techniques have to be resorted to for its evaluation.

C. Multi-channel Statistics: The Characteristic Function

In view of the complexity of the expression for the pdf of the output for a single channel, it is not surprising that no closed form expression exists for the pdf of the output of a multi-channel processor. Specifically, a straightforward approach to evaluating the class-conditional pdf of the output yields multiple integrals, which are not analytically tractable. We hence adopt a different tack.

Our approach is to obtain an orthogonal series expansion for the pdf (the Gram-Charlier A-Series), a few terms of which provide adequate approximations to the pdf (cf. [3], for example). The coefficients of the above-mentioned series are specified in terms of certain semi-invariant quantities called the *cumulants*, and these in turn can be computed from the characteristic function of the random variable. We shall hence, first explicitly evaluate the characteristic function, $\Psi^{(s)}(t)$, of the random variable $\mathbf{G}^{(s)}$. (Recall that $\mathbf{G}^{(s)}$ is the generalised linear discriminant function realised at the output of the multi-channel processor, conditioned upon class $\mathbf{\Omega}^{(s)}$ being present at the input of the processor.)

We consider a multi-channel processor specified by the set of independent channels $\left\{C_1, C_2, \ldots, C_m\right\}$, where the impulse responses $h_j \in H$ corresponding to channel C_j are as in theorem (2.1.1). In addition, to the mutual orthogonality provisos of theorem (2.1.1), we also assume that the channel impulse responses further satisfy

$$\int \mathrm{Re}\, h_j \ \mathrm{Im}\, h_j = 0 \ ,$$

so that Re h_j, and Im h_j are orthogonal to each other. This condition is satisfied by h_j specified according to the provisions of corollaries (2.1.1), and (2.1.2), for instance. A net consequence is that proposition (2.1.1) holds for this choice of h_j. Then from equations (2.1.2) and (2.1.6), we have the generalised linear discriminant function conditioned on class $\Omega_{(s)}$ being present, given by

$$\begin{aligned} \mathbf{G}^{(s)} &= \sum_{j=1}^{m} \alpha_j \left[\left(\mathrm{Re}(\varsigma_j^{(s)} + \mathbf{N}_j)\right)^2 + \left(\mathrm{Im}(\varsigma_j^{(s)} + \mathbf{N}_j)\right)^2 + \mathbf{N}_{d,j} \right] \\ &= \sum_{j=1}^{m} \alpha_j \left[\mathbf{X}_j^{(s)} + \mathbf{Y}_j^{(s)} + \mathbf{N}_{d,j} \right], \quad s = 1, \ldots, c \end{aligned} \qquad (2.2.5)$$

where we define the random variables

$$\mathbf{X}_j^{(s)} \triangleq \left(\mathrm{Re}(\varsigma_j^{(s)} + \mathbf{N}_j)\right)^2 ,$$

and

$$\mathbf{Y}_j^{(s)} \triangleq \left(\mathrm{Im}(\varsigma_j^{(s)} + \mathbf{N}_j)\right)^2 .$$

From corollary (2.1.1) and proposition (2.1.1), we have that the random variable, $\mathbf{G}^{(s)}$, is composed of a sum of $3m$ statistically independent random variables. Hence

$$\Psi^{(s)}(t) \triangleq \mathrm{E}\left(e^{i\,\mathbf{G}^{(s)}t} \right)$$

$$= \prod_{j=1}^{m} \mathrm{E}\left(e^{i\,\alpha_j \mathbf{X}_j^{(s)}t} \right) \mathrm{E}\left(e^{i\,\alpha_j \mathbf{Y}_j^{(s)}t} \right) \mathrm{E}\left(e^{i\,\alpha_j \mathbf{N}_{s,j}t} \right) . \tag{2.2.6}$$

The channel impulse responses, h_j, have orthogonal real and imaginary parts by choice. Hence, $\mathrm{Re}\,\mathbf{N}_j$ and $\mathrm{Im}\,\mathbf{N}_j$ are independent random variables. Further, they are jointly normal with $\mathrm{Re}\,\mathbf{N}_j \sim N(0,\sigma_n^2\eta_{j,R})$, and $\mathrm{Im}\,\mathbf{N}_j \sim N(0,\sigma_n^2\eta_{j,I})$, where we define

$$\eta_{j,R} \triangleq \|\mathrm{Re}\, h_j\|^2 \quad \text{and} \quad \eta_{j,I} \triangleq \|\mathrm{Im}\, h_j\|^2$$

in equations (2.1.9), (2.1.10), and (2.1.11). Hence,

$$\mathrm{E}\left(e^{i\,\alpha_j \mathbf{X}_j^{(s)}t} \right) = \int_{-\infty}^{\infty} \exp\left(i\,\alpha_j\, t\,(\,\mathrm{Re}\{\varsigma_j^{(s)}\}+n\,)^2 \right) \frac{\exp\left(\dfrac{-n^2}{2\eta_{j,R}\,\sigma_n^2} \right)}{\sqrt{2\pi\eta_{j,R}\,\sigma_n^2}}\, dn \ .$$

Completing squares and doing an appropriate contour integration gives

$$\mathrm{E}\left(e^{i\,\alpha_j \mathbf{X}_j^{(s)}t} \right) = \frac{\exp\left[\dfrac{(\mathrm{Re}\{\varsigma_j^{(s)}\})^2}{2\eta_{j,R}\,\sigma_n^2} \left(\dfrac{2it\,\alpha_j\,\eta_{j,R}\,\sigma_n^2}{1-2it\,\alpha_j\,\eta_{j,R}\,\sigma_n^2} \right) \right]}{\left(1-2it\,\alpha_j\,\eta_{j,R}\,\sigma_n^2 \right)^{1/2}} \Leftarrow \tag{2.2.7}$$

A similar derivation yields

$$\mathrm{E}\left(e^{i\,\alpha_j \mathbf{Y}_j^{(s)}t} \right) = \frac{\exp\left[\dfrac{(\mathrm{Im}\{\varsigma_j^{(s)}\})^2}{2\eta_{j,I}\,\sigma_n^2} \left(\dfrac{2it\,\alpha_j\,\eta_{j,I}\,\sigma_n^2}{1-2it\,\alpha_j\,\eta_{j,I}\,\sigma_n^2} \right) \right]}{\left(1-2it\,\alpha_j\,\eta_{j,I}\,\sigma_n^2 \right)^{1/2}} \Leftarrow \tag{2.2.8}$$

Finally, the $\mathbf{N}_{d,j}$'s are normally distributed with mean μ_d and variance σ_d^2. Hence (cf. [4] for example)

$$\mathrm{E}\left(e^{i\alpha_j \mathbf{N}_{d,j} t} \right) = \exp\left[i\alpha_j \mu_d t - \alpha_j^2 \sigma_d^2 \frac{t^2}{2} \right] , \quad j=1,\ldots,m \ . \tag{2.2.9}$$

Substituting equations (2.2.7)-(2.2.9) in equation (2.2.6) we get

$$\Psi^{(s)}(t) = \prod_{j=1}^{m} \left[\frac{\exp\left\{ i\alpha_j \mu_d t - \alpha_j^2 \sigma_d^2 \frac{t^2}{2} \right\}}{\left(1 - 2it\,\alpha_j \eta_{j,R}\, \sigma_n^2\right)^{1/2} \left(1 - 2it\,\alpha_j \eta_{j,I}\, \sigma_n^2\right)^{1/2}} \right.$$

$$\times \ \exp\left\{ \frac{(\mathrm{Re}\{\varsigma_j^{(s)}\})^2}{2\eta_{j,R}\,\sigma_n^2} \left(\frac{2it\,\alpha_j \eta_{j,R}\,\sigma_n^2}{1 - 2it\,\alpha_j \eta_{j,R}\,\sigma_n^2} \right) \right\}$$

$$\left. \times \ \exp\left\{ \frac{(\mathrm{Im}\{\varsigma_j^{(s)}\})^2}{2\eta_{j,I}\,\sigma_n^2} \left(\frac{2it\,\alpha_j \eta_{j,I}\,\sigma_n^2}{1 - 2it\,\alpha_j \eta_{j,I}\,\sigma_n^2} \right) \right\} \right] . \tag{2.2.10}$$

Simplifications ensue for the particular choices of impulse responses according to the recipes of corollaries (2.1.1), and (2.1.2). For the former case, we have from equations (2.2.2), and (2.2.3) that $\eta_{j,R} = \eta_{j,I} = \frac{1}{2}\,\eta_j$, while for the latter case, we have $\eta_{j,R} = \eta_j$, and $\eta_{j,I} = 0$.

Note that the pdf, $p^{(s)}(v)$, of the LDF conditioned upon class $\Omega^{(s)}$ being present, $\mathbf{G}^{(s)}$, can in theory be evaluated as the Fourier transform of the characteristic function , $\Psi^{(s)}(t)$. The form of $\Psi^{(s)}(t)$ in equation (2.2.10) is, however, not conducive to analytical Fourier transformation. We hence proceed with an evaluation of the cumulants of the random variable $\mathbf{G}^{(s)}$, and then obtain good approximations to the pdf from a series expansion.

D. The Cumulants of the Output Probability Distribution

The cumulants, originally defined by Thièle, are semi-invariant quantities intimately related to the ordinary moments of a random variable. Besides possessing several intriguing properties (cf. [5] for example), these quantities are of great utility in orthogonal series expansions for pdf's [5]. In what follows, we obtain explicit expressions for the cumulants of the random variables, $\mathbf{G}^{(s)}$, $s=1,...,c$.

The cumulants, $\chi_r^{(s)}$, $r=1,2,...$, of the random variable $\mathbf{G}^{(s)}$ are formally defined by

$$\sum_{r=1}^{\infty} \frac{\chi_r^{(s)}}{r!}(it)^r = \log\left(\Psi^{(s)}(t)\right) . \tag{2.2.11}$$

Substituting equation (2.2.10) for $\Psi^{(s)}(t)$ we have

$$\sum_{r=1}^{\infty} \frac{\chi_r^{(s)}}{r!}(it)^r = \sum_{j=1}^{m} \left[\frac{(\text{Re}\, \varsigma_j^{(s)})^2}{2\eta_{j,R}\,\sigma_n^2} \left(\frac{2it\,\alpha_j\,\eta_{j,R}\,\sigma_n^2}{1-2it\,\alpha_j\,\eta_{j,R}\,\sigma_n^2} \right) \right.$$

$$+ \quad \frac{(\text{Im}\, \varsigma_j^{(s)})^2}{2\eta_{j,I}\,\sigma_n^2} \left(\frac{2it\,\alpha_j\,\eta_{j,I}\,\sigma_n^2}{1-2it\,\alpha_j\,\eta_{j,I}\,\sigma_n^2} \right) + it\,\alpha_j\,\mu_d$$

$$\left. - \quad \frac{\alpha_j^2\sigma_d^2 t^2}{2} - \frac{1}{2}\log\left(1-2it\,\alpha_j\,\eta_{j,R}\,\sigma_n^2\right) - \frac{1}{2}\log\left(1-2it\,\alpha_j\,\eta_{j,I}\,\sigma_n^2\right) \right] . \tag{2.2.12}$$

Now

$$\log(1-itx) = -\sum_{r=1}^{\infty} \frac{1}{r}(itx)^r , \tag{2.2.13}$$

$$\frac{itx}{1-itx} = \sum_{r=1}^{\infty} (itx)^r , \tag{2.2.14}$$

with both series converging for $|\, itx \,| < 1$, i.e., $|\, t \,| < \frac{1}{|\, x \,|}$.

Define $t_j \triangleq | 2\alpha_j \eta_j \sigma_n^2 |^{-1}$, for $j=1,...,m$, and let $t_{\min} = \min(t_1, t_2, \ldots, t_m)$. Then from equations (2.2.12)-(2.2.14) we have

$$\sum_{r=1}^{\infty} \frac{\chi_r^{(s)}}{r!}(it)^r = \sum_{r=1}^{\infty} \left[\sum_{j=1}^{m} (2\alpha_j \eta_{j,R} \sigma_n^2)^r \left(\frac{(\mathrm{Re}\ \varsigma_j^{(s)})^2}{2\eta_{j,R}\sigma_n^2} + \frac{1}{2r} \right) \right] (it)^r$$

$$+ \sum_{r=1}^{\infty} \left[\sum_{j=1}^{m} (2\alpha_j \eta_{j,I} \sigma_n^2)^r \left(\frac{(\mathrm{Im}\ \varsigma_j^{(s)})^2}{2\eta_{j,I}\sigma_n^2} + \frac{1}{2r} \right) \right] (it)^r$$

$$+ \sum_{j=1}^{m} \left(it\, \alpha_j \mu_d - \frac{\alpha_j^2 \sigma_d^2 t^2}{2} \right)$$

when $| t | < t_{\min}$.

Equating corresponding powers of t on both sides of the above equation, we get

$$\chi_r^{(s)} = r! \sum_{j=1}^{m} \left[(2\alpha_j \eta_{j,R} \sigma_n^2)^r \left(\frac{(\mathrm{Re}\ \varsigma_j^{(s)})^2}{2\eta_{j,R}\sigma_n^2} + \frac{1}{2r} \right) + (2\alpha_j \eta_{j,I} \sigma_n^2)^r \left(\frac{(\mathrm{Im}\ \varsigma_j^{(s)})^2}{2\eta_{j,I}\sigma_n^2} + \frac{1}{2r} \right) \right.$$

$$\left. + \quad \alpha_j \mu_d \delta_{r1} + \frac{\alpha_j^2 \sigma_d^2}{2} \delta_{r2} \right], \quad r = 1,2,... \tag{2.2.15}$$

where δ_{rl} is the Kronecker delta.

In particular, we evaluate the following for future reference: the mean, $\mu^{(s)} = \chi_1^{(s)}$, and variance, $\sigma^{(s)^2} = \chi_2^{(s)}$, of $\mathbf{G}^{(s)}$ (cf. [6]).

$$E(\mathbf{G}^{(s)}) \triangleq \mu^{(s)} = \sum_{j=1}^{m} \alpha_j \left[| \varsigma_j^{(s)} |^2 + \eta_j \sigma_n^2 + \mu_d \right], \tag{2.2.16}$$

$$\mathrm{Var}\,(\mathbf{G}^{(s)}) \triangleq \sigma^{(s)^2} = \sum_{j=1}^{m} \alpha_j^2 \left[2\sigma_n^2 \left(\eta_{j,R} (\mathrm{Re}\ \varsigma_j^{(s)})^2 + \eta_{j,I} (\mathrm{Im}\ \varsigma_j^{(s)})^2 \right) \right.$$

$$+ \quad 4\sigma_n^4\left(\eta_{j,R}^2 + \eta_{j,I}^2\right) + \sigma_d^2\Big] . \tag{2.2.17}$$

E. An Orthogonal Series Expansion for the PDF

The Weber-Hermite system of orthogonal polynomials is often used for expansions of the density function, and gives rise to what is known as the Gram-Charlier A-Series [2,4]. This series is particularly well suited for the application at hand because the coefficients of the series expansion find particularly simple expression in terms of the cumulants of the pdf. In terms of this series, the class conditional pdf, $p^{(s)}(v)$, of the generalised linear discriminant function, $\mathbf{G}^{(s)}$, is given by

$$p^{(s)}(v) = \frac{1}{\sqrt{\chi_2^{(s)}}} \sum_{j=0}^{\infty} c_j^{(s)} \phi^{(j)}\left(\frac{v - \chi_1^{(s)}}{\sqrt{\chi_2^{(s)}}}\right) \tag{2.2.18}$$

where

$$\phi^{(j)}(v) \triangleq \frac{d^j}{dv^j}\left(\frac{1}{\sqrt{2\pi}} \exp\left\{\frac{-v^2}{2}\right\}\right)$$

and the coefficients $c_j^{(s)}$ are defined in terms of the cumulants [2,4]. The first few coefficients are

$$c_0^{(s)} = 1$$

$$c_1^{(s)} = c_2^{(s)} = 0$$

$$c_3^{(s)} = \frac{-\chi_3^{(s)}}{3!\chi_2^{(s)\frac{3}{2}}}$$

$$c_4^{(s)} = \frac{\chi_4^{(s)}}{4!\chi_2^{(s)2}} .$$

It is worth noting that rearrangement of the terms of the A-Series leads to better approximations to the pdf when we require to approximate the density using a few terms of the series. Rearrangement of the terms of the A-Series leads to what is known as the Edgeworth Series; this series expansion, however, may have problems of convergence [6]. However, our interest lies not in convergence of the series representation of the pdf, but in obtaining a good approximation to the pdf using a few terms of the expansion. In this respect the Edgeworth series is preferable to the A-Series [3].

A further property of interest of the series expansion (2.2.18) is that this form of the pdf is especially conducive to the evaluation of error probabilities (probabilities of misclassification). This follows because integrating the Hermite polynomials results simply in lower-order Hermite polynomials. As a consequence, we can obtain explicit (if approximate) formulas for the distribution function, thus enabling us to avoid the irksome bother of numerical integration of the pdf's to yield error probabilities.

A consideration of equation (2.2.5) and the results of corollary (2.1.1) and proposition (2.1.1), yields that the random variable $\mathbf{G}^{(s)}$ is the sum of $3m$ independent random variables. It is easy to check that for large m the hypotheses of the *Central Limit Theorem* hold, so that when the number of channels in the multi-channel processor become large the generalised linear discriminant functions $\mathbf{G}^{(s)}$, $s=1,...,c$, approach normality. (A strict upper bound on the deviation of the distribution function from normality can be obtained by use of the Berry-Esseen inequality [7].) The expansion in terms of the Hermite polynomials can hence be expected to fit the pdf very closely, with the use of just a few terms of the series expansion providing good approximations.

F. The PDF for a Pure Noise Input

The probability density function at the output when the input is purely white Gaussian noise is of use in estimating the detectability of the signal term in noise. This corresponds to Neyman-Pearson hypothesis testing where the two hypotheses to be tested are the presence or the absence of a signal. In essence this determines the resolution available in the system.

For the case where the signal is absent at the input, and we have noise alone, the output of the processor is (from equation (2.2.5))

$$\mathbf{G} = \sum_{j=1}^{m} \alpha_j \mid \mathbf{N}_j \mid^2 + \sum_{j=1}^{m} \alpha_j \mathbf{N}_{d,j}$$

$$\triangleq \mathbf{Y}_1 + \mathbf{Y}_2 ,$$

where the random variable $\mathbf{Y}_1$ corresponds to the first summation, and the random variable $\mathbf{Y}_2$ corresponds to the second sum.

Now, the $\mathbf{N}_{d,j}$'s are independent and normal with common mean μ_d and variance σ_d^2. Hence $\mathbf{Y}_2$ is also normal, and the pdf, $p_{Y_2}(v)$, of $\mathbf{Y}_2$ is given by

$$p_{Y_2}(v) = \frac{1}{\left(2\pi\sigma_d^2 \sum_{j=1}^{m} \alpha_j^2\right)^{1/2}} \exp\left\{ - \frac{1}{2\sigma_d^2 \sum_{j=1}^{m} \alpha_j^2} (v - \mu_d \sum_{j=1}^{m} \alpha_j)^2 \right\} . \tag{2.2.19}$$

We obtain the characteristic function, $\Psi_{Y_1}(t)$, of $\mathbf{Y}_1$ by setting the "signal term" $\varsigma_j^{(s)} = 0$ for each $j = 1,...,m$, and by ignoring the terms involving μ_d and σ_d (corresponding to the random variables $\mathbf{N}_{d,j}$) in equation (2.2.10). Hence

$$\Psi_{Y_1}(t) = \prod_{j=1}^{m} (1 - 2it\, \alpha_j \eta_{j,R}\, \sigma_n^2)^{1/2} (1 - 2it\, \alpha_j \eta_{j,I}\, \sigma_n^2)^{1/2} .$$

Consider, for simplicity, that impulse responses are chosen according to the prescription of corollary, (2.1.1), so that $\eta_{j,R} = \eta_{j,I} = \frac{1}{2} \eta_j$. For the case where all the weights, α_j, are strictly positive, the pdf, $p_{Y_1}(v)$, of $\mathbf{Y}_1$ takes on a very simple form. If $\alpha_j > 0$, $j = 1,...,m$ then the poles of $\Psi_{Y_1}(t)$ all occur in the lower half-plane at points $t = -i/(\alpha_j \eta_j \sigma_n^2)$, $j = 1,...,m$. Assuming for simplicity that all the poles are of multiplicity one, i.e., $\alpha_j \eta_j \neq \alpha_l \eta_l$ if $j \neq l$, we use the Calculus of Residues to evaluate $p_{Y_1}(v)$:

$$p_{Y_1}(v) \triangleq \frac{1}{2\pi} \int_{-\infty}^{\infty} \Psi_{Y_1}(t)\, e^{-ivt}\, dt$$

$$= \sum_{j=1}^{m} \left[\frac{\exp\left\{ - \dfrac{v}{\alpha_j \eta_j \sigma_n^2} \right\} \mathrm{U}(v)}{\alpha_j \eta_j \sigma_n^2 \prod\limits_{\substack{l=1 \\ l \neq j}}^{m} \left(1 - \dfrac{\alpha_l \eta_l}{\alpha_j \eta_j} \right)} \right], \tag{2.2.20}$$

where $\mathrm{U}(v)$ is the unit step function $\mathrm{U}(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{if } v < 0 \end{cases}$.

Now $\mathbf{Y}_1$, and $\mathbf{Y}_2$ are independent, and hence the pdf, $p_G(v)$, of $\mathbf{G}$ is the convolution of the individual pdf's. Using equations (2.2.19) and (2.2.20) and performing the integration we obtain

$$p_G(v) = \int_{-\infty}^{\infty} p_{Y_1}(u)\, p_{Y_2}(v-u)\, du$$

$$= \sum_{j=1}^{m} \left[\left[\left\{ \alpha_j \eta_j \sigma_n^2 \prod_{\substack{l=1 \\ l \neq j}}^{m} \left(1 - \frac{\alpha_l \eta_l}{\alpha_j \eta_j} \right) \right\}^{-1} \right.\right.$$

$$\times \ \exp\left\{ - \frac{1}{\alpha_j \eta_j \sigma_n^2} \left(v - \mu_d \sum_{l=1}^{m} \alpha_l - \frac{\sigma_d^2}{2\alpha_j \eta_j \sigma_n^2} \sum_{l=1}^{m} \alpha_l^2 \right) \right\}$$

$$\left.\left. \times \ \Phi \left\{ \frac{v - \mu_d \sum\limits_{l=1}^{m} \alpha_l}{\left(\sigma_d^2 \sum\limits_{l=1}^{m} \alpha_l^2 \right)^{1/2}} - \frac{\left(\sigma_d^2 \sum\limits_{l=1}^{m} \alpha_l^2 \right)^{1/2}}{\alpha_j \eta_j \sigma_n^2} \right\} \right] \right]$$

(2.2.21)

where $\Phi(x)$ is the cumulative Gaussian distribution function,

$$\Phi(x) \triangleq \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{u^2}{2}\right) du$$

and where **G** represents the output of the processor with a pure noise input and with no image class present at the input of the processor.

3. THRESHOLD MACHINES

A. Threshold Point Rules

We now take to a consideration of multi-channel machines whose point rule is specified by independent threshold operations in each channel. In this instance, the point rule (1.2.5) is specified by the hardlimiting map

$$D(z) = \operatorname{sgn}\ (\operatorname{Re} z - t) = \begin{cases} 1 & \text{if } \operatorname{Re} z \geq t \\ -1 & \text{if } \operatorname{Re} z < t \end{cases} \quad \forall z \in \mathbb{C} .$$

Here t is a fixed, real threshold. We will call the resultant processor a *threshold machine*. We will consider an instance of a threshold machine in chapter IV, where we use the threshold point rule to generate a class of low cost binary filters which yield good classification performance.

Without loss of generality, we will take all functions to be real. Substituting for D in equation (2.1.1) we obtain

$$\mathbf{G}_j^{(s)} = \operatorname{sgn}\ \left(\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \{f^{(s)}(x,y) + \mathbf{N}(x,y)\}\, h_j(x,y)\, dxdy - t\right) + \mathbf{N}_{d,j}$$

$$= \operatorname{sgn}\ \left((\varsigma_j^{(s)} - t) + \mathbf{N}_j\right) + \mathbf{N}_{d,j}$$

(2.3.1)

where $\varsigma_j^{(s)}$, and $\mathbf{N}_j$ are as in (2.1.3), and (2.1.5), repectively.

Using equations (2.1.9), (2.1.10), and (2.1.1), we have that $(\varsigma_j^{(s)} - t) + \mathbf{N}_j \sim N(\varsigma_j^{(s)} - t, \sigma_n^2 \eta_j)$. In the following we will assume that the channel impulse responses, h_j, are mutually orthogonal:

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_j(x,y)\, h_{k(x,y)}\, dxdy = 0 \quad \text{if} \quad j \neq k \ .$$

By theorem (2.1.1), it follows that the random variables, $(\varsigma_j^{(s)} - t) + \mathbf{N}_j$, are jointly independent, normal random variables, with mean $(\varsigma_j^{(s)} - t)$, and variance $\sigma_n^2 \eta_j$.

B. Single Channel Statistics

The case where channel output noise is absent is particularly simple. In this case we have the channel output given by

$$\mathbf{X}_j^{(s)} = \operatorname{sgn}\ \left((\varsigma_j^{(s)} - t) + \mathbf{N}_j\right) . \tag{2.3.2}$$

The random variables $\mathbf{X}_j^{(s)}$ clearly take on values -1 and 1 only. As $(\varsigma_j^{(s)} - t) + \mathbf{N}_j \sim N(\varsigma_j^{(s)} - t, \sigma_n^2 \eta_j)$, we have

$$p_j^{(s)} \triangleq \mathbf{P}\left\{\mathbf{X}_j^{(s)} = 1\right\} = \Phi\left(\frac{\varsigma_j^{(s)} - t}{\sigma_n \sqrt{\eta_j}}\right) , \tag{2.3.3}$$

and

$$q_j^{(s)} \triangleq \mathbf{P}\left\{\mathbf{X}_j^{(s)} = -1\right\} = \Phi\left(-\frac{\varsigma_j^{(s)} - t}{\sigma_n \sqrt{\eta_j}}\right) . \tag{2.3.4}$$

Now, from (2.3.1) and (2.3.2), we have

$$\mathbf{G}_j^{(s)} = \mathbf{X}_j^{(s)} + \mathbf{N}_{d,j} \ .$$

The random variable $\mathbf{N}_{d,j} \sim N(\mu_d, \sigma_d^2)$ is independent of $\mathbf{X}_j^{(s)}$. Hence, the characteristic function $\Psi_j^{(s)}(t)$ is given by

$$\Psi_j^{(s)}(t) \triangleq \mathbf{E}\left\{e^{i\mathbf{G}_j^{(s)}t}\right\} = \mathbf{E}\left\{e^{i\mathbf{X}_j^{(s)}t}\right\} \mathbf{E}\left\{e^{iNdjt}\right\} .$$

Using equations (2.3.3), (2.3.4), and (2.2.9), we get

$$\Psi_j^{(s)}(t) = (p_j^{(s)} e^{it} + q_j^{(s)} e^{-it})\, e^{(i\mu_d t - \frac{\sigma_d^2 t^2}{2})} . \tag{2.3.5}$$

The probability density function, $p_j^{(s)}(v)$, of $\mathbf{G}_j^{(s)}$ is obtained as the inverse Fourier transform of $\Psi_j^{(s)}(t)$. Hence

$$\begin{aligned} p_j^{(s)}(v) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \Psi_j^{(s)}(t)\, e^{-ivt}\, dt \\ &= \frac{p_j^{(s)}}{\sigma_d}\, \phi\left(\frac{v - \mu_d - 1}{\sigma_d}\right) + \frac{q_j^{(s)}}{\sigma_d}\, \phi\left(\frac{v - \mu_d + 1}{\sigma_d}\right), \end{aligned} \tag{2.3.6}$$

where $\phi(x)$ is the standard Gaussian probability density function

$$\phi(x) = \frac{1}{\sqrt{2\pi}}\, e^{\frac{-x^2}{2}} .$$

C. Multi-channel Statistics

Rewriting equation (2.1.2) for the processor output, we have the discriminant function $\mathbf{G}^{(s)}$ given by

$$\mathbf{G}^{(s)} = \sum_{j+1}^{m} \alpha_j \mathbf{G}_j^{(s)} .$$

By choice of orthogonal impulse responses, the channels are independent. Hence the characteristic function, $\Psi^{(s)}(t)$, of $\mathbf{G}^{(s)}$ is given by

$$\Psi^{(s)}(t) = \mathbf{E}\left(e^{i\mathbf{G}^{(s)}t}\right)$$

$$= \prod_{j=1}^{m} \mathbf{E}\left(e^{i\alpha_j \mathbf{G}_j^{(s)}t}\right).$$

Using equation (2.3.5), and taking cognisance of the weighting factors α_j, we have

$$\Psi^{(s)}(t) = \prod_{j=1}^{m} \Psi_j^{(s)}(\alpha_j t)$$

$$= \prod_{j=1}^{m} \left(p_j^{(s)} e^{i\alpha_j t} + q_j^{(s)} e^{-i\alpha_j t}\right) e^{(i\alpha_j \mu_d t - \alpha_j^2 \sigma_d^2 \frac{t^2}{2})}.$$

Let P denote the family of ordered pairs of subsets (J,K), with $J \bigcup K = \{1,\dots,m\}$, and $J \bigcap K = \emptyset$. Using the convention $\prod_{j \in \emptyset} x_j = 1$, we have

$$\Psi^{(s)}(t) = \sum_{(J,K) \in P} \left[\left(\prod_{j \in J} p_j^{(s)}\right)\left(\prod_{k \in K} q_k^{(s)}\right)\right.$$

$$\left. \times \ \exp\left\{it\left((1+\mu_d)\sum_{j \in J}\alpha_j - (1-\mu_d)\sum_{k \in K}\alpha_k\right) - \left(\sum_{l=1}^{m}\alpha_l^2\right)\sigma_d^2\frac{t^2}{2}\right\}\right]. \tag{2.3.7}$$

The probability density function, $p^{(s)}(v)$, of $\mathbf{G}^{(s)}$ is hence given by

$$p^{(s)}(v) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \Psi^{(s)}(t)\, e^{-ivt}\, dt$$

$$= \frac{1}{\sigma_d} \sum_{(J,K)\in P} \left(\prod_{j\in J} p_j^{(s)} \right) \left(\prod_{k\in K} q_k^{(s)} \right) \phi \left\{ \frac{v \; - (1+\mu_d) \sum\limits_{j\in J} + (1-\mu_d) \sum\limits_{k\in K} \alpha_k}{\sigma_d} \right\} . \qquad (2.3.8)$$

Except for the simplest cases, where we have very few channels, equation (2.3.8) would in the general case be very cumbersome to evaluate. An alternate approach for this case would be to compute the cumulants of $\mathbf{G}^{(s)}$, and then use the first few terms of the Edgeworth series expansion for the probability density function to obtain reasonable approximations to (2.3.8).

Assume without loss of generality that $p_j^{(s)} > q_j^{(s)}$ for $j = 1,...m$; i.e., from (2.3.3), and (2.3.4), we require that $\varsigma_j^{(s)} > t$. (If $\varsigma_j^{(s)} < t$, the following results continue to hold by interchanging $p_j^{(s)}$ and $q_j^{(s)}$.) Rewriting equation (2.3.7), we have

$$\Psi^{(s)}(t) = \prod_{j=1}^{m} p_j^{(s)} e^{i \alpha_j t} \left(1 + \frac{q_j^{(s)}}{p_j^{(s)}} e^{-2i \alpha_j t} \right) e^{(i \alpha_j \mu_d t \; - \alpha_j^2 \sigma_d^2 \frac{t^2}{2})} .$$

From the defining equation (2.2.11) for the cumulants, $\chi_r^{(s)}$, of the random variable $\mathbf{G}^{(s)}$, we have

$$\sum_{r=1}^{\infty} \frac{\chi_r^{(s)}}{r\,!} (it)^r = \log \Psi^{(s)}(t)$$

$$= \sum_{j=1}^{m} \left[\log p_j^{(s)} + i \alpha_j (1 + \mu_d) t \; - \alpha_j^2 \sigma_d^2 \frac{t^2}{2} + \log \left(1 + \frac{q_j^{(s)}}{p_j^{(s)}} e^{-2i \alpha_j t} \right) \right] .$$

We have $\left| \dfrac{q_j^{(s)}}{p_j^{(s)}} e^{-2i \alpha_j t} \right| < 1$, so that the Taylor series expansion

$$\log \left(1 + \frac{q_j^{(s)}}{p_j^{(s)}} e^{-2i \alpha_j t} \right) = \sum_{l=1}^{\infty} (-1)^{l-1} \left(\frac{q_j^{(s)}}{p_j^{(s)}} \right)^l \frac{e^{-2i \alpha_j lt}}{l}$$

converges for all t. Hence,

$$\sum_{r=1}^{\infty} \frac{\chi_r^{(s)}}{r!} (it)^r = \sum_{r=1}^{\infty} \sum_{j=1}^{m} \Bigg[\alpha_j (1+\mu_d)\delta_{r1} + \alpha_j^2 \sigma_d^2 \delta_{r2}$$

$$+ (-2\alpha_j)^r \sum_{l=1}^{\infty} (-1)^{l-1} \left(\frac{q_j^{(s)}}{p_j^{(s)}} \right)^l l^{r-1} \Bigg] \frac{(it)^r}{r!} .$$

Equating corresponding powers of t on both sides, we obtain

$$\chi_r^{(s)} = \sum_{j=1}^{m} \left[\alpha_j (1+\mu_d)\delta_{r1} + \alpha_j^2 \sigma_d^2 \delta_{r2} - (-2\alpha_j)^r \sum_{l=1}^{\infty} \left(- \frac{q_j^{(s)}}{p_j^{(s)}} \right)^l l^{r-1} \right], \quad r = 1,2,\ldots \tag{2.3.9}$$

The mean, $\mu^{(s)}$, of $\mathbf{G}^{(s)}$, can be directly obtained from (2.3.9) as

$$\mu^{(s)} = \mathbf{E}\,(\mathbf{G}^{(s)}) = \chi_1^{(s)} = \sum_{j=1}^{m} \alpha_j \left\{ \mu_d + \frac{1}{p_j^{(s)} - q_j^{(s)}} \right\} .$$

The form (2.3.9) is not particularly illuminating for the general form of cumulant. However, in the limit as $p_j^{(s)} \rightarrow 1$, we obtain the following asymptotic estimate for the cumulants.

$$\chi_r^{(s)} \sim \sum_{j=1}^{m} \left[\alpha_j (1+\mu_d)\delta_{r1} + \alpha_j^2 \sigma_d^2 \delta_{r2} + (2\alpha_j)^r \frac{q_j^{(s)}}{p_j^{(s)}} \right] .$$

Using these estimates for the cumulants of $\mathbf{G}^{(s)}$, the first few terms of the series (2.2.18) can be used to approximate the probability density of $\mathbf{G}^{(s)}$.

REFERENCES

[1] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[2] D. Psaltis, E. G. Paek, and S. S. Venkatesh, "Acousto-optic image processor,"

SPIE 10-th Intl. Optical Computing Conf., MIT, Cambridge, Massachusetts, April 1983.

[3] H. Cramér, *Mathematical Methods of Statistics.* Princeton: Princeton University Press, 1946.

[4] A. Papoulis, *Probability, Random Variables, and Stochastic Processes.* New York: McGraw-Hill Kogakusha, 1965.

[5] M. G. Kendall, *The Advanced Theory of Statistics*, vol. 1. London: Charles Griffin and Co., 1943.

[6] H. Cramér, *Random Variables and Probability Distributions.* Cambridge: Cambridge Tracts in Mathematics, No. 36, 1937.

[7] W. Feller, *An Introduction to Probability Theory and its Applications,* vol. 2. New York: Wiley, 1966.

CHAPTER III

ROTATION INSENSITIVE FILTERS

1. INTRODUCTION

A. Background

The first application we consider is the use of quadratic machines to achieve invariance to image rotation in classification. While matched spatial filters have been demonstrated to provide an effective and simple tool for shift-invariant image recognition, the technique of matched filtration fails when rotations of images are encountered. In recent papers, Hsu, et al., [1,2] demonstrate that linear filtration using a circular harmonic filter, followed by an amplitude extraction operation yields rotation and shift invariance in image classification. The use of a single circular harmonic filter, however, corresponds to using only a small fraction of the information content in an image (except in rare cases where an image is dominated by a single circular harmonic component), and can lead to poor noise performance [3]. The problem is further exacerbated when we are faced with multiple classes in the recognition problem, where it may prove difficult, if not impossible, to find a single circular harmonic filter which provides adequate discrimination against all classes.

The use of a single circular harmonic filter is equivalent to using a single feature for discrimination purposes. The addition of more independent features can be intuitively expected to yield better results, and can in fact be theoretically justified (cf.

[4], for example). It is reasonable hence, to consider algorithms utilising several independent rotation invariant features. Two approaches proposed recently incorporate more robustness in the classification procedure by using several circular harmonic filters to generate a sequence of independent rotation invariant features which are used in the decision algorithm: Hsu and Arsenault [5] propose a multi-dimensional decision-making approach, while Wu and Stark [6] propose a vector signature-based algorithm.

B. Quadratic Machines

Ideally, we would like to incorporate robustness in the decision making procedure, while at the same time avoiding the computational complexities of multi-dimensional decision making. With this in mind we investigate generalised linear discriminant functions of a rotation invariant character (of which [6] is a special case) thereby utilising several features, while at the same time retaining a simple decision rule. The particular machines we will investigate in this regard are the quadratic machines analysed in the previous chapter. With the nature of the point rule fixed to be square-law, much of our effort will devolve around appropriately choosing channel impulse responses (corresponding to filters) which achieve the necessary rotation invariant discriminant functions.

We generate a finite-dimensional subspace of a (countably) infinite-dimensional feature space which has the desired (rotation) invariance properties, as indicated in the schema of fig. 2.1 (b). Here each channel generates one independent rotation (and shift) invariant feature, and the resultant feature vector is projected onto the real line by an appropriate choice of weights, to obtain the linear discriminant function. Note that any choice of weights yields a generalised linear discriminant function of a rotation invariant character. (Consequently, we have several degrees of freedom, with each rotation invariant feature used in the linear discriminant function contributing one degree of freedom.) Each channel consists of a filter which produces correlation peaks insensitive to rotation, followed by a square-law device to eliminate any unwanted phase terms, as shown in fig. 2.1 (a). We call these filters "*R*otation *I*nsensitive *F*ilters," or RIFs for brevity.

Multi-class image classification is achieved by constructing a series of processors as in fig. 2.1 (b), each of which is geared to distinguish one particular image class from the other classes, as in equation . As all these processors have similar constructions, we consider the problem of generating just one representative multi-channel rotation invariant processor which distinguishes one image class from all the other classes.

In section 2 we characterise the most general type of channel impulse response which yields outputs insensitive to input rotations (up to a phase factor).

The remaining sections 3 and 4 tackle the issue of linear discriminant function processor performance. Clearly any approach toward optimising classification efficiency (minimising the probability of error) will require consideration of the following:

(1) The choice of rotation insensitive filters–characterised in section 2–for each channel,

(2) The optimum choice of weights for the linear discriminant function, and

(3) The number of channels to be chosen in the multi-channel processor.

As an alternative to strictly optimal solutions, we obtain near-optimal solutions which maximally separate image classes in mean in section 3. We also explore the possibility of *ad hoc* selection of weights in conjunction with maximally separating filters. As a final note we tackle the issue of the discrimination information content in multiple stages of RIFs. We demonstrate that multiple stages of RIFs can potentially yield good performance, using the performance of matched filters as a convenient yardstick. In section 4 we demonstrate that the problem of optimisation over an abstract function space can be reduced to a more concrete problem of optimisation over sets of real numbers. We prove that optimum solutions exist when output noise is absent, and demonstrate that an optimal solution can be obtained by consideration of a compact set of k-tuples of real numbers.

2. THE GENERAL FORM OF ROTATION INSENSITIVE FILTERS

Each channel of the multi-channel processor realising the generalised linear discriminant function (figure 2.1) has a rotation insensitive filter as a key element. In this section we obtain expressions for the most general form of a rotation insensitive filter.

Let $f\ (x\ ,y\ )$ represent a generic image belonging to one of c classes of images (which are symbolically represented by $\Omega^{(1)},\Omega^{(2)},\ .\ .\ .\ ,\Omega^{(c)}$). We denote by $f^{\ (i\ )}(x\ ,y\ )$ a particular image in class $\Omega^{(i)}$. Each class of images, $\Omega^{(i)}$, is assumed to consist of all shifted and rotated versions of the specified image, $f^{\ (i\ )}(x\ ,y\ )$. We use the nonce notation $f_{\ \phi}(x\ ,y\ )$ to denote an image, $f\ (x\ ,y\ )$, which has been rotated through an angle of ϕ radians. All functions, f , considered, are assumed to belong to the Hilbert space, H (complex L_2), of square-integrable (finite-energy) functions.

We define $\Omega\ \subseteq H$ by $\Omega\ \triangleq \bigcup_{i=1}^{c} \Omega^{(i)}$. Clearly, $\{\Omega^{(1)},\Omega^{(2)},\ .\ .\ .\ ,\Omega^{(c)}\}$ forms a partition of Ω .

Definition. Let $\Lambda\ \subseteq H$ be a subset of H. We define a function $h\ \in H$ to be *rotation insensitive to* Λ if for each $f\ \in \Lambda$, and any angle of rotation ϕ, the magnitudes of the correlation between $f\ (x\ ,y\ )$ and $h\,(x\ ,y\ )$, and of that between $f_{\ \phi}(x\ ,y\ )$ and $h\ (x\ ,y\ )$, are equal at some point in the correlation plane.

We say that h is *rotation insensitive to H* (or simply *rotation insensitive*) if we set $\Lambda\ = H$ in the above definition.

In what follows we investigate the behaviour of two general sub-classes of finite energy functions, $H_0^{\Omega}\ \subseteq H$ and $H_0\ \subseteq H$, which conform to the above definition, viz.,

$$H_0^{\Omega}\ \triangleq \left\{h\ \in H : h \text{ is rotation insensitive to } \Omega\right\},$$

$H_0 \triangleq \{h \in H : \text{h is rotation insensitive to } H\}$.

Clearly, if h is rotation insensitive to H, then it is also rotation insensitive to Ω; hence $H_0^{\Omega} \supseteq H_0$.

Note that we do not require invariance to rotation at all points in the correlation plane; rather, we require invariance at only a single point. By specifying the filters in such a fashion that we obtain correlation peaks at the specified "invariance point", we could locate the "invariance point" in the correlation plane even when there are shifts of the input image. If input image position is controllable, then we could specify a fixed point in the correlation plane as our "invariance point", and restrict our attention to this point alone. Furthermore, note that in the definition of H_0^{Ω}, we required invariance to rotation not just for one specified image class of interest, but for all the image classes. This was motivated by the fact that it is desirable to have rotation invariance extend to all the image classes so that the decision rule does not have to take into account the vagaries in the output due to rotations in the other classes.

Let $h(x,y) \in H$, $f(x,y) \in \Lambda \subseteq H$, and let $g(\xi,\eta)$ denote the correlation between $f(x,y)$ and $h(x,y)$ at the point (ξ,η) in the correlation plane. Then we have

$$g(\xi,\eta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x+\xi,y+\eta)\bar{h}(x,y)dxdy$$

where $\bar{h}$ is the complex conjugate of $h(x,y)$.

As correlation is a shift-invariant operation we restrict ourselves to considering solely rotated versions of the specified images $f^{(i)}(x,y)$, $i=1,...,c$. Now for simplicity we require our "invariance point" to be at the origin, (0,0), of the correlation plane. Setting $g(0,0) = g$, and rewriting the correlation integral in polar coordinates, we have

$$g = \int_0^{\infty}\int_0^{2\pi} f(r\cos\theta, r\sin\theta)\,\bar{h}(r\cos\theta, r\sin\theta)\,r\,d\theta dr$$

$$= \int_0^{\infty} \int_0^{2\pi} f'(r,\theta)\, \overline{h}'(r,\theta)\, r \; d\theta \, dr \; . \tag{3.2.1}$$

Clearly, for any angle of rotation, ϕ, we have

$$f_{\phi}(x,y) = f_{\phi}(r \cos\theta, r \sin\theta) = f'(r,\theta-\phi) \; .$$

Letting g_{ϕ} denote the correlation between $f_{\phi}(x,y)$ and $h(x,y)$ at (0,0) in the correlation plane, we have

$$g_{\phi} = \int_0^{\infty} \int_0^{2\pi} f'(r,\theta-\phi)\overline{h}'(r,\theta) r \; d\theta \, dr \; . \tag{3.2.2}$$

For each $r \in [0,\infty)$, the functions $f'(r,\theta)$ and $h'(r,\theta)$ are periodic functions of θ with period 2π. We can hence formally expand $f'(r,\theta)$ and $h'(r,\theta)$ in a Fourier series (the circular harmonic expansion, cf. [7]):

$$f'(r,\theta) = \sum_{n=-\infty}^{\infty} \hat{f}_n(r) e^{in\theta} \tag{3.2.3}$$

$$h'(r,\theta) = \sum_{n=-\infty}^{\infty} \hat{h}_n(r) e^{in\theta} \tag{3.2.4}$$

where $i = \sqrt{-1}$,and the circular harmonic-coefficients, $\hat{f}_n(r)$ and $\hat{h}_n(r)$, are given by

$$\hat{f}_n(r) = \frac{1}{2\pi} \int_0^{2\pi} f'(r,\theta)\, e^{-in\theta} \, d\theta \quad , \quad n \in \mathbb{Z} \, ,$$

$$\hat{h}_n(r) = \frac{1}{2\pi} \int_0^{2\pi} h'(r,\theta)\, e^{-in\theta} \, d\theta \quad , \quad n \in \mathbb{Z} \, .$$

Substituting (3.2.3) and (3.2.4) in (3.2.2), we have

$$g_\phi = \int_0^\infty \int_0^{2\pi} \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \hat{f}_n(r)\, \overline{\hat{h}_m}(r)\, e^{i(n-m)\theta - n\phi}\, r\, d\theta\, dr$$

$$= \sum_{n=-\infty}^{\infty} \left\{ 2\pi \int_0^\infty \hat{f}_n(r) \overline{\hat{h}_n}(r)\, r\, dr \right\} e^{-in\phi} . \tag{3.2.5}$$

We now make some definitions to facilitate further discussion. Let V be a unitary space defined as follows: the constituent vectors, $\mathbf{p}$, of V are complex-valued functions, $p : [0,\infty) \to \mathbb{C}$, and such that $\int_0^\infty | p(r) |^2 r\, dr < \infty$, with natural notions of addition and multiplication by a scalar; the inner product, $\langle\, ,\, \rangle$, in V is defined by

$$\langle \mathbf{p}, \mathbf{q} \rangle \triangleq 2\pi \int_0^\infty p(r)\, \overline{q}(r)\, r\, dr$$

for each pair of vectors $\mathbf{p}$, $\mathbf{q}$ in V.

As a consequence of the finite energy requirements imposed on all the functions considered, we have from equations (3.2.3) and (3.2.5), and Parseval's identity, that $\hat{\mathbf{f}}_n^{(i)} \triangleq \hat{f}_n^{(i)}(r) \in V$, and $\hat{\mathbf{h}}_n \triangleq \hat{h}_n(r) \in V$, for each $n \in \mathbb{Z}$, and each $i = 1,...,c$ (i.e., the circular harmonic components of the finite-energy functions considered are all elements of V).

We can now simply rewrite equation (3.2.5) in vector space notation as

$$g_\phi = \sum_{n=-\infty}^{\infty} \langle \hat{\mathbf{f}}_n , \hat{\mathbf{h}}_n \rangle e^{-in\phi} \tag{3.2.6}$$

Note that from equations (3.2.1) and (3.2.2), $g_0 \equiv g$. We now characterise the general form of RIF in the following result:

Theorem 3.2.1. Let $\Lambda \subseteq H$. Let $h \in H$, and let $\{\hat{\mathbf{h}}_n\}$ be the circular harmonic coefficients of h. Then h is rotation insensitive to Λ if, and only if, for each $f \in \Lambda$, (with corresponding circular harmonic coefficients $\{\hat{\mathbf{f}}_n\}$)

$$\left\langle \hat{\mathbf{f}}_n \, , \hat{\mathbf{h}}_n \right\rangle = \begin{cases} \varsigma(f\,) & \text{if } n = k(f\,) \\ 0 & \text{if } n \neq k(f\,) \end{cases}, \tag{3.2.7}$$

where $k : \Lambda \rightarrow \mathbb{Z}$, and $\varsigma : \Lambda \rightarrow \mathbb{C}$ are maps satisfying:

(1) $k(f_\phi) = k(f\,) \; \forall \, f \, , f_\phi \in \Lambda$, and

(2) $\varsigma(f_\phi) = e^{-ik(f\,)\phi} \varsigma(f\,) \quad , \forall \, f \; \in \Lambda \;\; , \phi \in \mathbb{R}$.

(Note that in obtaining the circular harmonic coefficients we specify as coordinate origin some appropriately chosen point–say the geometric image centre–and exclude from our considerations RIFs using circular harmonic coefficients corresponding to a different origin. The justification is two-fold. On the one hand, we expect the averaging effect of using a linear discriminant function on the feature space to wash out, to some extent, the effects of not choosing the best centre of expansion for the circular harmonics [1]. The second argument is based on statistical grounds: RIFs using different centres of expansion result in features which are no more statistically independent.)

Proof. To prove the "only if" part:

Assume h is rotation insensitive to Λ , and fix $f \; \in \Lambda$. We now define the complex numbers, $\varsigma_n(f\,)$, $n \in \mathbb{Z}$, by

$$\varsigma_n(f\,) = \; | \, \varsigma_n(f\,) \, | \, e^{\,i\,\omega_n(f\,)} \triangleq \left\langle \hat{\mathbf{f}}_n \, , \hat{\mathbf{h}}_n \right\rangle , \tag{3.2.8}$$

where we make explicit the dependence of the sequence $\left\{ \varsigma_n(f\,) \right\}_{n=-\infty}^{\infty}$ on $f \; \in \Lambda$. We then have from (3.2.6), and the definition of $\varsigma_n(f\,)$,

$$g_\phi = \sum_{n=-\infty}^{\infty} , | \, \varsigma_n(f\,) \, | \, e^{\,i(\omega_n(f\,) - n\phi)} . \tag{3.2.9}$$

After some algebra we can then show that

$$|g|^2 - |g_\phi|^2 = 4 \sum_{n=-\infty}^{\infty} \sum_{k=1}^{\infty} |\varsigma_n(f)|\,|\varsigma_{n+k}(f)|$$

$$\sin\left(\frac{k\phi}{2}\right) \sin\left(\{\omega_{n+k}(f) - \omega_n(f)\} + \frac{k\phi}{2}\right).$$

Now by definition, h satisfies $|g|^2 - |g_\phi|^2 = 0$, $\forall\, \phi \in \mathbb{R}$. Clearly this can hold true for all angles of rotation ϕ if, and only if, $|\varsigma_n(f)|\,|\varsigma_{n+k}(f)| = 0$, for all $k \in \mathbb{Z}^+$, and for each $n \in \mathbb{Z}$. Hence the sequence $\{\varsigma_n(f)\}_{n=-\infty}^{\infty}$ must be of the form

$$\varsigma_n(f) = \varsigma(f)\,\delta_{nk(f)}, \quad \text{some } k(f) \in \mathbb{Z}, \tag{3.2.10}$$

where $\varsigma : \Lambda \to \mathbb{C}$ is some complex function defined on Λ, and $\delta_{nk(f)}$ is the Kronecker delta,

$$\delta_{nk(f)} \triangleq \begin{cases} 1 & \text{if } n = k(f) \\ 0 & \text{if } n \neq k(f) \end{cases} .$$

Using the above result, equation (3.2.3) and the definition of $\varsigma_n(f)$ we have

$$\varsigma_n(f_\phi) = \varsigma(f_\phi)\delta_{nk(f_\phi)},$$

and

$$\varsigma_n(f_\phi) = \varsigma_n(f)e^{-in\phi}.$$

Hence

$$\varsigma(f_\phi)\delta_{nk(f_\phi)} = \varsigma(f)e^{-in\phi}\delta_{nk(f)},$$

so that for each $f \in \Omega$,

$$k(f_\phi) = k(f),$$

$$\varsigma(f_{\phi}) = \varsigma(f)e^{-ik(f)\phi} .$$

From (3.2.8) and (3.2.10), we see that equation (3.2.7) holds. Choosing $\varsigma \colon \Lambda \rightarrow \mathbb{C}$, and $k : \Lambda \rightarrow \mathbb{Z}$ as above, the proof of the 'only if' part is completed.

To prove the converse: assume there exist mappings k and ς as in the theorem. Fix $f \in \Lambda$, and let ϕ be some (arbitrary) angle of rotation. Then from (3.2.6) and (3.2.7) we have

$$g_{\phi} = \sum_{n=-\infty}^{\infty} \varsigma(f)\, \delta_{nk(f)}\, e^{-in\phi} = \varsigma(f)\, e^{-ik(f)\phi} ,$$

and

$$|g|^2 - |g_{\phi}|^2 = |\varsigma(f)|^2 - |\varsigma(f)e^{-ik(f)\phi}|^2 = 0 .$$

This is true for all $f \in \Lambda$, and as ϕ was arbitrary, h is rotation insensitive to Λ. □

We can now characterise the elements of H_0^{Ω}. As Ω has c distinct classes, the map k in the theorem can be specialised as follows; $k : \Omega \rightarrow \{k_1, k_2, \ldots, k_c\} \subseteq \mathbb{Z}$, with $k(f) = k^{(2)}$ if $f \in \Omega^{(2)}$. (Note that the integers $k^{(2)}$ need not be distinct.) Concisely put then, the theorem requires that for any $h \in H_0^{\Omega}$, the circular harmonic coefficients, $\hat{\mathbf{h}}_n$, satisfy two orthogonality conditions:

(1) $\hat{\mathbf{h}}_n$ is orthogonal to each of the c vectors, $\hat{\mathbf{f}}_n^{(1)}, \hat{\mathbf{f}}_n^{(2)}, \ldots, \hat{\mathbf{f}}_n^{(c)}$, if $n \notin \{k_1, k_2, \ldots, k_c\}$, for some set of integers k_i, $i=1,\ldots,c$,

(2) $\hat{\mathbf{h}}_{k_i}$ is orthogonal to the vectors, $\hat{\mathbf{f}}_{k_i}^{(j)}$, $j \neq i$, for $i=1,\ldots,c$,

where $\hat{\mathbf{f}}_n^{(i)}$, $n \in \mathbb{Z}$, are the circular harmonic coefficients of the characteristic images, $f^{(i)}(x,y)$, of each class, Ω_i, $i=1,\ldots,c$. This then is a complete characterisation of a general function $h \in H_0^{\Omega}$ which is rotation insensitive to Ω.

It is easily seen that non-trivial solutions rotation insensitive to Ω exist. In fact, the following result holds:

Corollary 3.2.1. Let $h \in H$, and let $\hat{\mathbf{h}}_n$, $n \in \mathbb{Z}$, be the circular harmonic coefficients of h. Then h is rotation insensitive to H if, and only if,

$$\hat{\mathbf{h}}_n \equiv \mathbf{0} \quad \text{, if } n \neq k \text{, some } k \in \mathbb{Z} \text{ .}$$

Proof. We substitute $\mathbf{\Lambda} = H$ in the theorem. As $h \in H$, we must have $\left\langle \hat{\mathbf{h}}_n , \hat{\mathbf{h}}_n \right\rangle = ||\hat{\mathbf{h}}_n||^2 = \varsigma(h)\delta_{n,k(h)}$ from the theorem. ($\delta_{n,k(h)}$ is the Kronecker delta). Setting $k(h) \triangleq k$, we have by properties of the norm that $\hat{\mathbf{h}}_n = \mathbf{0}$ if $n \neq k$.
□

Corollary (3.2.1) essentially enunciates a necessary and sufficient condition for functions in H to be rotation insensitive: all functions of the form $h'(r,\theta) = \hat{h}_k(r)e^{ik\theta}$, where $k \in \mathbb{Z}$, and $\hat{h}_k(r) = \hat{\mathbf{h}}_k \in V$, are rotation insensitive, and hence also rotation insensitive to Ω. A ready example of functions in this class is the circular harmonic filters of Hsu, et al., [1,2], which fall in this category, and are hence rotation insensitive to H.

Corollary provides an easy method of obtaining functions rotation insensitive to Ω. The ease of formulation of the rotation insensitive functions in H_0 lends itself to some simplicity in notation; we shall hence concentrate on these functions for purposes of analysis, while indicating briefly how extensions may be made to the more general case of functions rotation insensitive to $\mathbf{\Omega}$ in H_0^{Ω}.

3. OPTIMAL CLASSIFICATION

A. Asymptotic Optimality

For simplicity, we consider the two-class problem. We construct a family of multi-channel rotation invariant processors (which we call RIPs for brevity), by choosing the channel impulse responses, h_j, of an m-channel quadratic machine according to the prescription of corollary (2.1.1). The output of each channel is rotation invariant, as the magnitude square operation cancels the rotation dependent phase term of theorem (3.2.1). Hence, the generalised linear discriminant function that accrues from the machine is also rotation invariant as it is the linear combination of rotation invariant features from the channels.

For the two-class case we require that the discriminant functions satisfy $\mathbf{G}^{(1)} > t$, and $\mathbf{G}^{(2)} < t$, for some specified threshold t. The classification rule is, as before: decide $\boldsymbol{\Omega}^{(1)}$ if $\mathbf{G} > t$, and decide $\boldsymbol{\Omega}^{(2)}$ if $\mathbf{G} < t$.

As we saw in our discussion of performance criteria, a general approach toward optimising system parameters to obtain best classification performance requires the minimisation of the probability of error, P_e. With a choice of channel impulse responses, h_j, as in corollary (2.1.1), we see that corollary (2.1.1) and proposition (2.1.1) hold, so that the results of Section II (3) hold *in toto*. In principle, then, we could utilise the expressions obtained for the statistics of the quadratic machine to compute the required probabilities, and optimise this by suitable choice of parameters. As we saw, however, simple analytical expressions do not obtain for P_e for the quadratic machine, so that the procedure outlined above could be quite labourious.

Some gratuitous simplicity obtains, however, when we consider the limit of a large number of channels. By choice of channel impulse responses according to corollary (2.1.1), we obtain a quadratic machine with independent channels. Consequently, for a large enough number of channels, we expect the *Central Limit Theorem* to come into force, so that the discriminant function, $\mathbf{G}$, approaches normality. (This, of course, corresponds to approximating the pdf, $p^{(2)}(v)$, by the first term in the A-series expansion (2.2.18).) Note that the class pdf's become unimodal asymptotically by virtue of the approaching Gaussian behaviour. The discriminant functions may hence be expected to perform well in classification.

For independent channels, the optimum (classification performance) can be expected to be monotonic with the number of channels used. Hence, in the ensuing discussion, we will assume a simplicity of notation; we assume the channels are rotation insensitive to H, and satisfy the hypotheses of corollaries (2.1.1) and (2.1.2). The general optimisation problem, where the channels are rotation insensitive to Ω, can be treated in similar fashion; in addition to this, generalisations to arbitrary noise models are discussed briefly at the end of this section.

Let us first consider the probability of error. Using the first term of the A-series expansion (2.2.18), we obtain central tendency asymptotically with m with

$$p^{(2)}(v) \sim \frac{1}{\sigma^{(2)}} \phi \left(\frac{v - \mu^{(2)}}{\sigma^{(2)}} \right) \tag{3.3.1}$$

where $\mu^{(2)}$ and $\sigma^{(2)}$ are given in equations (1.2.17) and (1.2.18), respectively. The probability of error for a fixed threshold t is then given asymptotically with m by

$$P_e(t) \sim \pi^{(1)} \Phi \left(- \frac{\mu^{(1)} - t}{\sigma^{(1)}} \right) + \pi^{(2)} \Phi \left(- \frac{t - \mu^{(2)}}{\sigma^{(2)}} \right) . \tag{3.3.2}$$

The shaded area in fig. 3.1 corresponds to the probability of error. Here, of course, we assume that the RIFs and threshold are so chosen that $\mu^{(2)} < t < \mu^{(1)}$. (Clearly, other choices give much larger probabilities of error.) For given system parameters, there, of course, exists an optimum threshold, t_0, corresponding to the point of crossover of the two densities in fig. 3.1. Differentiating (3.3.2) with respect to t, we obtain the optimum threshold (for fixed parameters) given by the following equation:

$$\frac{d\,P_e}{dt}(t_{0)} \sim \frac{\pi^{(1)}}{\sigma^{(1)}} \Phi \left(- \frac{\mu^{(1)} - t_0}{\sigma^{(1)}} \right) - \frac{\pi^{(2)}}{\sigma^{(2)}} \Phi \left(- \frac{t_0 - \mu^{(2)}}{\sigma^{(2)}} \right) = 0 \tag{3.3.3}$$

Now, $\mu^{(s)}$ and $\sigma^{(s)}$ depend on the particular parametric realisation of the RIP. Rewriting equations (1.2.17) and (1.2.18) for ease of reference,

$$\mu^{(s)} = \sum_{j=1}^{m} \alpha_j \; [\; | \varsigma_j^{(s)} |^2 + \eta_j \sigma_n^2 + \mu_d] \, ,$$

and

$$\sigma^{(s)2} = \sum_{j=1}^{m} \alpha_j^2 \, [\, 2\eta_j \; \sigma_n^2 \; | \varsigma_j^{(s)} |^2 + \eta_j^2 \sigma_n^4 + \sigma_d^2] .$$

(Recall that $\eta_{j,R} = \eta_{j,I} = \frac{\eta}{2}$, in this instance.)

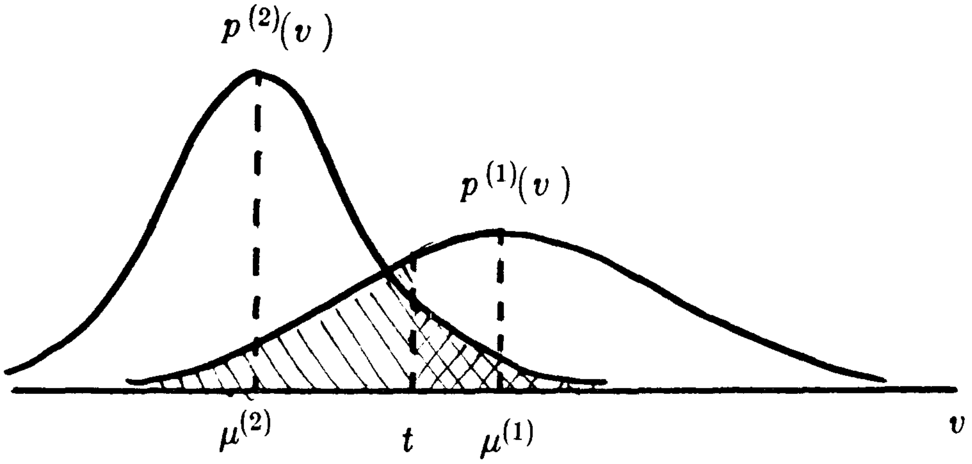

Fig. 3.1. Probability of classification error for a choice of threshold t. (Equi-probable states of nature assumed.)

The optimisation problem is then to choose system parameters–filters and weights–to minimise $P_e(t_0)$. This, however, involves the solution of the transcendental equation (3.3.3) for every choice of system parameters (which result in changes in the mean, and variance), before P_e can be computed from (3.3.2).

We introduce some analytical and computational simplicity by considering the Bhattacharyya distance as our performance measure instead. The Bhattacharyya coefficient (1.4.2) takes on a simple form when the class-conditional densities are normal. Specifically, by substituting (3.3.1), and carrying out the indicated integration, we obtain that asymptotically with m ,

$$\rho_B \sim \left\{ \frac{2\pi^{(1)}\pi^{(2)}\sigma^{(1)}\sigma^{(2)}}{\sigma^{(1)^2} + \sigma^{(2)^2}} \right\}^{1/2} \exp\left\{ -\frac{1}{4} \frac{(\mu^{(1)} - \mu^{(2)})^2}{\sigma^{(1)^2} + \sigma^{(2)^2}} \right\} . \qquad (3.3.4)$$

As we saw earlier, ρ_B is a good fixed performance measure, which can frequently be used as a satisfactory alternative to P_e . In this case, we clearly save much in computation when we consider optimal systems with respect to ρ_B.

Specifically, given distinct integers, $k_1, \ldots, k_m$ (corresponding to various circular=harmonic orders), we seek to choose impulse responses of the form $h_j = \hat{h}_{k_j}(r) e^{ik_j\theta}$, and weights α_j , so as to minimise ρ_B. (We assume that the circular harmonic orders $k_j, j = 1,\ldots,m$, are chosen that $\exists$ at least one non-trivial generalised linear discriminant function with $\mu^{(1)} > \mu^{(2)}$.)

B. Existence of Optimal RIPs

We concentrate for the nonce on RIPs which are rotation insensitive to H, and satisfy the hypotheses of corollary (3.3.2). We assume the circular-harmonic orders $k_1, k_2, \ldots, k_m$, are fixed. Let $\mathbf{h} = (h_1, h_2, \ldots, h_m) \in (H_0)^m$ be an m-tuple of impulse responses as in corollary (3.3.1), and let $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_m) \in \mathbb{R}^m$ be the corresponding m-tuple of weights. The RIP is hence completely specified by $(\boldsymbol{\alpha}, \mathbf{h}) \in \mathbb{R}^m \times (H_0)^m$.

Definition. We say that $(\boldsymbol{\alpha}^0,\mathbf{h}^0) \in \mathbb{R}^m \times (H_0)^m$ is an *optimum RIP* if $\rho_B(\boldsymbol{\alpha}^0,\mathbf{h}^0) = \min_{\substack{\boldsymbol{\alpha}\in \mathbb{R}^m \\ \mathbf{h}\in (H_0)^m}} \left\{ \rho_B(\boldsymbol{\alpha},\mathbf{h}) : \mu^{(1)}(\boldsymbol{\alpha},\mathbf{h}) \geq \mu^{(2)}(\boldsymbol{\alpha},\mathbf{h}) \text{ , } s=1,...,c \right\}$.

Note that in our definition, we specialise to generalised linear discriminant functions for which $\mu^{(1)} \geq \mu^{(2)}$, $s=1,...,c$ so that the processing yields adequate discrimination between class $\Omega^{(1)}$ and $\Omega^{(2)}$.

The optimisation problem can be reduced from a "difficult" problem of optimisation over an abstract functional space to a comparatively "simple" problem of optimising over sets of real numbers, as we now show:

Recall that we had defined the inner product space V, with product $\langle \, , \rangle$ in section 2. Now, for each $k \in \mathbb{Z}$, we define the subspace V_k of V to consist of all linear combinations of the k-th circular-harmonics of each of the specified images $f^{(2)}$, $s=1,2$:

$$V_k \triangleq \mathrm{sp}\{\hat{\mathbf{f}}_k^{(1)},\hat{\mathbf{f}}_k^{(2)}\}$$

$$= \left\{ \mathbf{f} : \mathbf{f} = a_1\hat{\mathbf{f}}_k^{(1)} + a_2\hat{\mathbf{f}}_k^{(2)} \text{ , } a_1 \text{ , } a_2 \in \mathbb{C} \right\} .$$

Without loss of generality, assume $\hat{\mathbf{f}}_k^{(1)}$ and $\hat{\mathbf{f}}_k^{(2)}$ are linearly independent. Then the dimension of V_k is 2. Let $\{\mathbf{e}_{k,1},\mathbf{e}_{k,2}\}$ be any set of orthonormal basis vectors for V_k. Then, for some complex numbers $\gamma_{k,l}^{(2)}$, we can write

$$\hat{\mathbf{f}}_k^{(2)} = \sum_{l=1}^{2} \gamma_{k,l}^{(2)} \mathbf{e}_{k,l} \text{ , } \quad \gamma_{k,l}^{(2)} \in \mathbb{C} \text{ , } s=1,2. \qquad (3.3.5)$$

Let V_k^* be the orthogonal subspace of V_k, and let $\mathbf{e}_{k,0}$ be any vector of unit norm in V_k^*; i.e., $\langle \mathbf{e}_{k,0} \text{ , } \mathbf{e}_{k,l} \rangle = \delta_{0l}$, where δ_{0l} is the Kronecker delta.

Now, for each $j=1,...,m$, the impulse responses are of the form $h_j = \hat{h}_{k_j}(r)e^{ik_j\theta}$, with $\hat{h}_{k_j}(r) = \hat{\mathbf{h}}_{k_j} \in V$. Clearly we can write

$$\hat{\mathbf{h}}_{k_j} = \hat{\mathbf{h}}_{k_j}' + \hat{\mathbf{h}}_{k_j}'' \ , \tag{3.3.6}$$

where $\hat{\mathbf{h}}_{k_j}' \in V_{k_j}$, and $\hat{\mathbf{h}}_{k_j}'' \in V_{k_j}^*$, so that $\left\langle \hat{\mathbf{h}}_{k_j}' , \hat{\mathbf{h}}_{k_j}'' \right\rangle = 0$. Now $\hat{\mathbf{h}}_{k_j}''$ lies in the orthogonal subspace of V_{k_j}, and does not interact with the circular-harmonic coefficients $\hat{\mathbf{f}}_{k_j}^{(i)}$, $i=1,2$. Its only contribution to ρ_B is hence in the energy term η_j of equation (2.1.1). Now

$$\eta_j = ||\hat{\mathbf{h}}_{k_j}'||^2 + ||\hat{\mathbf{h}}_{k_j}''||^2 \ .$$

Hence all vectors of equal norm in $V_{k_j}^*$ are equivalent as far as performance is concerned as their contributions to ρ_B are the same. Without loss of generality we can hence consider only vectors $\hat{\mathbf{h}}_{k_j}''$ of the form $\hat{\mathbf{h}}_{k_j}'' = \beta_{k_j,0}\mathbf{e}_{k_j,0}$, for some $\beta_{k_j,0} \in \mathbb{C}$. Also, $\hat{\mathbf{h}}_{k_j}' = \sum_{l=1}^{2} \beta_{k_j,l}\mathbf{e}_{k_j,l}$ for some $\beta_{k_j,l} \in \mathbb{C}$. Hence, in the optimisation problem, we can without loss of generality restrict ourselves to considering circular-harmonic coefficients of the form

$$\hat{\mathbf{h}}_{k_j} = \sum_{l=0}^{2} \beta_{k_j,l}\mathbf{e}_{k_j,l} \ , \quad \text{for some } \beta_{k_j,l} \in \mathbb{C} \ . \tag{3.3.7}$$

Substituting (3.3.5) and(3.3.7) in (2.1.3) and (2.1.1), we have for each $j=1,\ldots,m$,

$$\varsigma_j^{(2)} = \sum_{l_j=1}^{2} \gamma_{k_j,l_j}^{(2)} \overline{\beta_{k_j,l_j}} \ , \quad s=1,2 \ , \tag{3.3.8}$$

$$\eta_j = \sum_{l_j=0}^{2} | \beta_{k_j,l_j} |^2 \ . \tag{3.3.9}$$

Now, let $\boldsymbol{\beta} = (\mathrm{Re}(\beta_{k_j,l_j}) , \mathrm{Im}(\beta_{k_j,l_j}))$, $l_j = 0,1,2$, $j=1,\ldots,m$, be a $6m$-tuple of real numbers (corresponding to the coefficients in the expansion (3.3.7)). Using equations (3.3.8), (3.3.9), (2.2.16), and (2.2.17) in equation (3.3.4), we see that the optimisation problem over the RIPs $(\boldsymbol{\alpha},\mathbf{h})$ is equivalent to optimising over the $m + 6m$ real variables $(\boldsymbol{\alpha},\boldsymbol{\beta})$, with the proviso that $\mu^{(1)}(\boldsymbol{\alpha},\boldsymbol{\beta}) \geq \mu^{(2)}(\boldsymbol{\alpha},\boldsymbol{\beta})$.

In this context, we define our performance criterion in terms of the Bhattacharyya coefficient. Let $Q \subseteq \mathbb{R}^m \times \mathbb{R}^{6m}$ be defined by

$$Q \triangleq \{(\boldsymbol{\alpha},\boldsymbol{\beta}) : \mu^{(1)}(\boldsymbol{\alpha},\boldsymbol{\beta}) \geq \mu^{(2)}(\boldsymbol{\alpha},\boldsymbol{\beta}) \} . \tag{3.3.10}$$

Clearly, $Q \neq \emptyset$ by choice of the circular-harmonic orders k_j , $j=1,\ldots,m$. We now define the performance criterion, $\rho : Q \rightarrow [0,1]$, by

$$\rho(\boldsymbol{\alpha},\boldsymbol{\beta}) = \begin{cases} \rho_B(\boldsymbol{\alpha},\boldsymbol{\beta}) & \text{if } \boldsymbol{\alpha}\neq\mathbf{0} \text{ , } \boldsymbol{\beta}\neq\mathbf{0} \\ 1 & \text{otherwise} \end{cases} .$$

(This is clearly consistent with using the Bhattacharyya distance as our distance measure, with the added constraint that we require $\mu^{(1)}(\boldsymbol{\alpha},\boldsymbol{\beta}) \geq \mu^{(2)}(\boldsymbol{\alpha},\boldsymbol{\beta})$ for adequate discrimination).

We now demonstrate the existence of an optimum RIP when output noise is absent in all the channels:

Lemma 3.3.1. Assume that $\mu_d = \sigma_d = 0$. Then, for all real, non-zero constants ω_1 and ω_2, the RIPs $(\boldsymbol{\alpha},\mathbf{h})$ and $(\omega_1\boldsymbol{\alpha},\omega_2\mathbf{h})$ yield the same performance.

Proof. Follows as a special case of the monotone consistency of the Bhattachryya coefficient, proposition (1.4.4). □

The proof follows by straightforward substitution in equation (3.3.4). (Note that scaling the impulse responses, $\mathbf{h}$, by ω_2 is equivalent to scaling $\boldsymbol{\beta}$ by ω_2.) Thus in the absence of output noise, the performance is invariant to scaling the weight vector

$\boldsymbol{\alpha}$ or the m-tuple of impulse responses $\mathbf{h}$.

Now, note that $\boldsymbol{\alpha} = \mathbf{0}$, or $\boldsymbol{\beta} = \mathbf{0}$ cannot be a solution to the optimisation problem (either case corresponds to no processing at all). The probability of error for either case is $P_e = 1 - \pi_1$, which is the worst case performance for all generalised linear discriminant functions with $\mu^{(1)}(\boldsymbol{\alpha},\boldsymbol{\beta}) \geq \mu^{(2)}(\boldsymbol{\alpha},\boldsymbol{\beta})$. Any other assignment of weights and filters, $\boldsymbol{\alpha} \neq \mathbf{0}$, $\boldsymbol{\beta} \neq \mathbf{0}$, will hence give at least as good a performance.

We now show that we can restrict attention to *compact* sets. We define the compact sets $S_\alpha \subseteq \mathbb{R}^m$, and $S_\beta \subseteq \mathbb{R}^{6m}$, by

$$S_\alpha \triangleq \{\boldsymbol{\alpha} : \sum_{j=1}^{m} |\alpha_j| = 1\} ,$$

$$S_\beta \triangleq \{\boldsymbol{\beta} : \sum_{j=1}^{m} \sum_{l_j=0}^{2} [\ |\mathrm{Re}(\beta_{k_j,l_j})| + |\mathrm{Im}(\beta_{k_j,l_j})|\ ] = 1\} .$$

Now let $A \subseteq S_\alpha \times S_\beta$ be defined by

$$A \triangleq \{(\boldsymbol{\alpha},\boldsymbol{\beta}) \in S_\alpha \times S_\beta : \mu^{(1)}(\boldsymbol{\alpha},\boldsymbol{\beta}) \geq \mu^{(1)}(\boldsymbol{\alpha},\boldsymbol{\beta}) .$$

Lemma 3.3.2. A is compact.

Proof. A is bounded as S_α and S_β are bounded. So it suffices to prove that A is closed.

Let $(\boldsymbol{\alpha}^0,\boldsymbol{\beta}^0) = \mathbf{s}^0$ be any accumulation point of A. Clearly, $\mathbf{s}^0 \in S_\alpha \times S_\beta$ as $S_\alpha \times S_\beta$ is compact.

Claim: $\mu^{(1)}(\mathbf{s}^0) \geq \mu^{(2)}(\mathbf{s}^0)$.

We prove the claim by contradiction. Suppose $\mu^{(1)}(\mathbf{s}^0) - \mu^{(2)'}(\mathbf{s}^0) = -\epsilon < 0$ for some real, positive number ϵ. Now, without loss of generality, we can assume that each of the image classes has unit energy $\int f^{(2)^2} = 1$. Hence, for $s = 1,2$, $||\hat{\mathbf{f}}_k^{(2)}||^2 \leq 1\ \forall$ $k \in \mathbb{Z}$. Hence, from equation (3.3.5), $|\gamma_{k,l}^{(2)}| \leq 1\ \forall\ l = 1,2$, $k \in \mathbb{Z}$.

Now, there is a sequence $\{\mathbf{s}^n\}$ such that $\mathbf{s}^n \rightarrow \mathbf{s}^0$ as $n \rightarrow \infty$. So, $\exists N < \infty : n \geq N \Longrightarrow | \beta_{k_j,l_j}^{(0)} - \beta_{k_j,l_j}^{(n)} | < \frac{\epsilon}{16c}$ for each $l_j = 1,\ldots,n_{k_j}$, $j = 1,\ldots,m$. It then follows that

$$| \varsigma_{k_j}^{(1)}(\boldsymbol{\beta}^0) |^2 = | \sum_{l_j=1}^{c} \gamma_{k_j,l_j}^{(1)} \overline{\beta_{k_j,l_j}^{(0)}} |^2 > | \varsigma_{k_j}^{(1)}(\boldsymbol{\beta}^n) |^2 - \frac{\epsilon}{2} \quad \text{if } n \geq N \ ,$$

and

$$| \varsigma_{k_j}^{(2)'}(\boldsymbol{\beta}^0) |^2 < | \varsigma_{k_j}^{(2)'}(\boldsymbol{\beta}^n) |^2 + \frac{\epsilon}{2} \quad \text{if } n \geq N \ .$$

So

$$-\epsilon = \mu^{(1)}(\mathbf{s}^0) - \mu^{(2)'}(\mathbf{s}^0)$$

$$= \sum_{j=1}^{m} \alpha_j \left[| \varsigma_{k_j}^{(1)}(\boldsymbol{\beta}^0) |^2 - | \varsigma_{k_j}^{(2)'}(\boldsymbol{\beta}^0) |^2 \right]$$

$$> [\mu^{(1)}(\mathbf{s}^n) - \mu^{(2)'}(\mathbf{s}^n)] - \epsilon \quad \text{if } n \geq N \ .$$

Now $\mathbf{s}^n \in \boldsymbol{A} \Longrightarrow \mu^{(1)}(\mathbf{s}^n) \geq \mu^{(2)'}(\mathbf{s}^n) \ \forall \ n \in \mathbb{Z}$ by definition of $\boldsymbol{A}$. So $\epsilon > \epsilon$.

So we have a contradiction, and this proves the claim. Consequently, by definition of $\boldsymbol{A}$, $\mathbf{s}^0 \in \boldsymbol{A}$. As $\mathbf{s}^0$ was an arbitrary accumulation point of $\boldsymbol{A}$, we have that $\boldsymbol{A}$ is closed. So $\boldsymbol{A}$ is compact. □

Theorem 3.3.1. Assume $\mu_d = \sigma_d = 0$. Then there exists an optimum RIP $(\boldsymbol{\alpha}^0,\boldsymbol{\beta}^0) \in \boldsymbol{A}$.

Proof. From equation (2.3.4) it suffices to show that $\exists \ (\boldsymbol{\alpha}^0,\boldsymbol{\beta}^0) \in \boldsymbol{A} : \rho(\boldsymbol{\alpha}^0,\boldsymbol{\beta}^0) = \min\{\rho(\boldsymbol{\alpha},\boldsymbol{\beta}) : (\boldsymbol{\alpha},\boldsymbol{\beta}) \in \boldsymbol{Q}\}$. We first demonstrate that in the optimisation problem we can, without loss of generality, restrict our attention to $\boldsymbol{A} \subseteq \boldsymbol{Q}$. Let $\nu = \inf\{\rho(\boldsymbol{\alpha},\boldsymbol{\beta}) : (\boldsymbol{\alpha},\boldsymbol{\beta}) \in \boldsymbol{Q}\}$. Then $\exists$ a sequence $\{(\boldsymbol{\alpha}_k,\boldsymbol{\beta}_k)\} \subset \boldsymbol{Q}$ such that $\rho(\boldsymbol{\alpha}_k,\boldsymbol{\beta}_k) \rightarrow \nu$ as $k \rightarrow \infty$. From the discussion following lemma (3.3.1), we can without loss of generality choose the sequence so that $\boldsymbol{\alpha}_k$ and $\boldsymbol{\beta}_k$ do not approach 0 as

$k \to \infty$. So for k large enough, $\boldsymbol{\alpha}_k \neq 0$ and $\boldsymbol{\beta}_k \neq 0$. So $\exists$ $(\boldsymbol{\alpha}_k^0,\boldsymbol{\beta}_k^0) \in A$, and nonzero, real constants $\omega_{1,k}$ and $\omega_{2,k}$ such that $(\omega_{1,k}\,\boldsymbol{\alpha}_k^0,\omega_{2,k}\,\boldsymbol{\beta}_k^0) = (\boldsymbol{\alpha}_k\,,\boldsymbol{\beta}_k)$. And now by lemma (3.3.1), we have $\rho(\boldsymbol{\alpha}_k^0,\boldsymbol{\beta}_k^0) = \rho(\boldsymbol{\alpha}_k\,,\boldsymbol{\beta}_k) \to \nu$ as $k \to \infty$. For the optimisation problem, we can hence restrict ourselves to the set A without any loss of generality.

Now by lemma (3.3.2), A is compact, and as $\mathbf{0} \notin A$, we have from equation (3.3.4) that ρ is continuous on A. It then follows that $\exists$ $(\boldsymbol{\alpha}^0,\boldsymbol{\beta}^0) \in A\ :\ \rho(\boldsymbol{\alpha}^0,\boldsymbol{\beta}^0) = \min\{\rho(\boldsymbol{\alpha},\boldsymbol{\beta}) : (\boldsymbol{\alpha},\boldsymbol{\beta}) \in A\ \}$.

Claim: $\rho(\boldsymbol{\alpha}^0,\boldsymbol{\beta}^0) = \nu$.

We prove the claim by contradiction. Assume $\rho(\boldsymbol{\alpha}^0,\boldsymbol{\beta}^0) \neq \nu$. Then $\rho(\boldsymbol{\alpha}^0,\boldsymbol{\beta}^0) > \nu$ by definition of ν. Set $\epsilon = \rho(\boldsymbol{\alpha}^0,\boldsymbol{\beta}^0) - \nu > 0$. Now, $\rho(\boldsymbol{\alpha}_k^0,\boldsymbol{\beta}_k^0) \to \nu$ as $k \to \infty$. So for k large enough, $\rho(\boldsymbol{\alpha}_k^0,\boldsymbol{\beta}_k^0) < \nu + \frac{\epsilon}{2} = \rho(\boldsymbol{\alpha}^0,\boldsymbol{\beta}^0) - \frac{\epsilon}{2} < \rho(\boldsymbol{\alpha}^0,\boldsymbol{\beta}^0)$. This is a contradiction as ρ achieves its minimum on A at $(\boldsymbol{\alpha}^0,\boldsymbol{\beta}^0)$, and $\{(\boldsymbol{\alpha}_k^0,\boldsymbol{\beta}_k^0)\} \subset A$. So $\rho(\boldsymbol{\alpha}^0,\boldsymbol{\beta}^0) = \nu$.

□

(In the theorem, we have used the coefficients $\boldsymbol{\beta}^0$ to represent the corresponding m-tuple of impulse responses).

Some comments are in order:

(1) The definitions of the compact sets S_α and S_β are not sacrosanct. We could have chosen, for instance, the boundaries of balls of radius $r > 0$.

(2) We could restrict our attention to the compact set S_α even if output noise is present; i.e., the discriminant function is invariant to scaling. This, of course, follows from proposition (1.4.4)-the Bhattacharyya coefficient is monotonically constant.

(3) The generalisation of the above results to filters rotation insensitive to Ω is straightforward. We expand each circular harmonic of the filters in expansions of the form (3.3.7), and optimise over the set of real variables specified by the coefficients. The results, specifically theorem (3.3.1), remain unchanged. Note, however, that we still require independent channels as characterised in theorem (2.1.1). Also, proposition (2.1.1) does not hold anymore; the variance at the output of each channel, C_j , $j=1,...,m$, is now given by:

$$\mathrm{Var}\left\{\mathbf{G}_j^{(2)}\right\} = \mathrm{Var}\left\{\,|\,\mathrm{Re}(\varsigma_j^{(2)} + \mathbf{N}_j)\,|\,^2\right\} + \mathrm{Var}\left\{\,|\,\mathrm{Im}(\varsigma_j^{(2)} + \mathbf{N}_j)\,|\,^2\right\}$$

$$+\, 4r^2\, \mathrm{Var}^{\,2}\left\{\mathrm{Re}(\mathbf{N}_j)\right\}\, \mathrm{Var}^{\,2}\left\{\mathrm{Im}(\mathbf{N}_j)\right\}\ ,$$

where r is the correlation coefficient between $\mathrm{Re}(\mathbf{N}_j)$ and $\mathrm{Im}(\mathbf{N}_j)$.

(4) The assumption of additive, white, Gaussian noise at the input was made simply for ease of notation. The results are valid for any noise model as long as independent channels are used, and the number of channels is large enough that the processor output is approximately normal. However, theorem (2.1.1) has to be modified for this case. If R_n is the noise autocorrelation operator, for instance, the first integral in equation (2.1.7) must be replaced by an expression of the form ($\mathrm{Re}\, h_j$, R_n [$\mathrm{Re}\, h_k$]), where (,) is the natural inner product in the Hilbert space, and $R_n : H \rightarrow H$; similar inner products replace the other integrals. Equation (3.3.4) is still valid, but the means and variances must be replaced by more complicated expressions. In all cases, the means and variances can still be expressed in terms of real coefficients, β_{k_j, l_j}, from expansions as in equation (3.3.7).

Thus, we have demonstrated that the optimisation problem can be reduced from the conceptually more difficult one of optimising over a function space, to the simpler problem of optimising over sets of real numbers. Further simplicity is effected because we can restrict ourselves to bounded (in fact, compact) sets, so that the optimisation procedure does not have to deal with unbounded terms. The proof of the existence of optimal solutions for the case of no output noise is a useful result in this regard; we can have recourse to numerical algorithms to minimise ρ over a compact set of real numbers in order to extract an optimum solution. (The optimality, or otherwise, of any trial point can be checked by the Kuhn-Tucker conditions (cf. [8] for example).)

Thus, in principle, we can find optimal solutions, though this may be computationally very expensive from the practical point of view because of the large number of variables to be handled in the optimisation procedure. Note, however, that given the image classes and the noise model, this expense is a one time "set-up" cost;

once the optimum RIP is found, no further computational costs are incurred as long as the problem specifications do not change.

4. SUB-OPTIMAL CLASSIFICATION

A. Maximally Separating Rotation Insensitive Filters

Optimum RIPs, even when they do exist, may be computationally difficult to find as we saw in the last section. We hence consider here a sub-optimal solution to the optimisation problem which is easy to obtain, and can be expected to yield good (near-optimal) performance.

To ensure reasonable performance, any RIP chosen must clearly belong to the set of RIPs, Q, defined in equation (3.3.10) of the last section. Now, for any set of specified circular harmonic orders, $k_1,k_2, \ldots, k_m$, the circular harmonic coefficients $\hat{\mathbf{f}} = (\hat{\mathbf{f}}_{k_1}^{(1)},\hat{\mathbf{f}}_{k_2}^{(1)}, \ldots, \hat{\mathbf{f}}_{k_m}^{(1)})$, of the image $f^{(1)}$ can serve as rotation insensitive filters (RIF's) for the RIP. Clearly, for appropriately chosen circular harmonic orders, $k_1,k_2,\ldots,k_m$, and weights, $\boldsymbol{\alpha} = (\alpha_1,\alpha_2, \ldots, \alpha_m)$, the RIP $(\boldsymbol{\alpha}, \hat{\mathbf{f}}) \in Q$, and it is hence a legitimate RIP with reasonable performance. However, this is clearly not an optimum choice in general, and performance may be barely tolerable if the image classes are very similar.

Again we consider functions rotation insensitive to H for notational simplicity. From equation (3.3.4) for the Bhattacharyya coefficient, we note that, if all other factors are being held constant, performance improves monotonically with the mean class separation $\mu^{(1)} - \mu^{(2)}$. It is hence reasonable to seek filters for which $\mu^{(1)} - \mu^{(2)}$ is large. Now, from equation (2.2.16), we have

$$\mu^{(1)} - \mu^{(2)} = \sum_{j=1}^{m} \alpha_j (| \varsigma_j^{(1)} |^2 - | \varsigma_j^{(2)} |^2) .$$

(Recall that $\varsigma_j^{(2)}$ was defined as the output of the j- th filter when class $\Omega^{(2)}$ is present at the input.) The minimum value that $| \varsigma_j^{(2)} |^2$ can take is zero. So we would like to find filters for which $|\varsigma_j^{(2)}|$ is as close to zero as possible, and for which $|\varsigma_j^{(1)}|$ is comparatively large.

In order to formalise the above discussion, we introduce the mean-square error term, $\epsilon_j : V \to \mathbb{R}$, for each channel, $j=1,...,m$. Let $\tau_j > 0$ be fixed. Then we define for each $\hat{\mathbf{h}}_{k_j} \in V$

$$\epsilon_j(\hat{\mathbf{h}}_{k_j}) \triangleq |\varsigma_j^{(1)}(\hat{\mathbf{h}}_{k_j}) - \tau_j|^2 + |\varsigma_j^{(2)}(\hat{\mathbf{h}}_{k_j})|^2 \quad , j=1,...,m \ ,$$

where $\varsigma_j^{(2)}$ is as in equation (2.1.3). (We make explicit here the dependence of $\varsigma_j^{(2)}$ on the filter corresponding to channel C_j.)

Definition. Let $\tau_j > 0$ be a given positive real number, and let the circular harmonic orders k_j be fixed for each $j=1,...,m$. Then we say that $\hat{\mathbf{h}}_{k_j}^0$ is a *maximally separating RIF* for channel C_j if $\epsilon_j(\hat{\mathbf{h}}_{k_j}^0) = \min\{\epsilon_j(\hat{\mathbf{h}}_{k_j}) : \hat{\mathbf{h}}_{k_j} \in V\}$.

Thus, maximally separating filters achieve the best possible mean separation between the image class $\Omega^{(1)}$ and $\Omega^{(2)}$ (in a mean-square sense). We now demonstrate that maximally separating filters can be easily constructed:

From equation (3.3.6), we have $\hat{\mathbf{h}}_{k_j} = \hat{\mathbf{h}}_{k_j}' + \hat{\mathbf{h}}_{k_j}''$, where $\hat{\mathbf{h}}_{k_j}' \in V_{k_j}$, and $\hat{\mathbf{h}}_{k_j}'' \in V_{k_j}^*$. (Recall that the subspace V_{k_j} of V was defined to be dense in $\mathrm{sp}\{\hat{\mathbf{f}}_{k_j}^{(1)}, \hat{\mathbf{f}}_{k_j}^{(2)}\}$, and $V_{k_j}^*$ was defined to be the orthogonal subspace of V_{k_j}.) Choosing a set of orthonormal basis vectors, $\{\mathbf{e}_{k_j,1}, \mathbf{e}_{k_j,2}\}$, for V_{k_j}, and a vector of unit norm, $\mathbf{e}_{k_j,0} \in V_{k_j}^*$, we obtain the expansions (3.3.5) and (3.3.7).

Now, let $\mathbb{C}^2$ denote the 2-dimensional inner product space composed of pairs of complex numbers, and let $\mathbf{r}^{(1)}, \mathbf{r}^{(2)}$, be the standard basis for $\mathbb{C}^2$. We define the linear transformation $\hat{\mathrm{F}}_{k_j} : V \to \mathbb{C}^2$ as follows: (1) $V_{k_j}^*$ lies in the null space of $\hat{\mathrm{F}}_{k_j}$, i.e., $\hat{\mathbf{h}}_{k_j}'' \in V_{k_j}^* \implies \hat{\mathrm{F}}_{k_j} \hat{\mathbf{h}}_{k_j}'' = \mathbf{0}$, and (2) the restriction of $\hat{\mathrm{F}}_{k_j}$ to V_{k_j}, $\mathrm{F}_{k_j} : V_{k_j} \to \mathbb{C}^c$, has matrix elements $(\mathrm{F}_{k_j})_{s,l} = \gamma_{k_j,l}^{(2)}$, $s=1,2$, $l=1,2$, in the basis $\{\mathbf{e}_{k_j,l}\}_{l=1}^2$ (where $\gamma_{k_j,l}^{(2)}$ is as defined in equation (3.3.5)). From equations (3.3.6) and (3.3.8), we have

$$\hat{F}_{k_j}\hat{h}_{k_j} = F_{k_j}\hat{h}_{k_j}' = \varsigma_{k_j}^{(1)}\mathbf{r}^{(1)} + \varsigma_{k_j}^{(2)}\mathbf{r}^{(2)} \ .$$

We hence require to find $\hat{\mathbf{h}}_{k_j}^{0'} \in V_{k_j}$ such that the norm of the (error) vector $F_{k_j}\hat{\mathbf{h}}_{k_j}' - \tau_j \mathbf{r}^{(1)}$ is minimised; i.e., $\hat{\mathbf{h}}_{k_j}^{0'}$ is the RIF such that $\epsilon_j = ||F_{k_j}\hat{\mathbf{h}}_{k_j}' - \tau_j \mathbf{r}^{(1)}||^2$ is a minimum. The solution can be easily found to be

$$\hat{\mathbf{h}}_{k_j}^{0'} = \tau_j F_{k_j}^{-1}\mathbf{r}^{(1)} \ ,$$

where $F_{k_j}^{\dagger}$ is the adjoint of the linear operator F_{k_j}.

Clearly $\hat{\mathbf{h}}_{k_j}'' \in V_{k_j}^{*}$ does not interact with the image classes, and so can be arbitrarily chosen. In accordance with equation (3.3.7) we can write the general maximally separating RIF for the j-th channel to be of the form

$$\hat{\mathbf{h}}_{k_j}^{0} = \tau_j F_{k_j}^{-1}\mathbf{r}^{(1)} + \beta_{k_j,0}\mathbf{e}_{k_j,0} \quad , \ j=1,...,m \ , \tag{3.4.1}$$

for some $\beta_{k_j,0} \in \mathbb{R}$.

The form of the maximally separating RIF above has the additional advantage of being independent of basis chosen for V_{k_j}, so that any convenient basis can be chosen.

In the general c-class case, we can proceed similarly. Here, however, some dependencies may exist across classes, so that the $\dim(V_{k_j}) = n_{k_j} < e$. In that case, the solution (mean-square optimal) is the pseudo-inverse $\hat{h}_{k_j}^{(0)'} = \tau_j (F_{k_j}^{\dagger} F_{k_j})^{-1} F_{k_j}\mathbf{r}^{(1)} + \beta_{k_j,0}\mathbf{e}_{k_j,0}$, where $F_{k_j}^{\dagger}$ is the adjoint of F_{k_j}. (Note that if V_{k_j} is c-dimensional, i.e., the vectors $\hat{\mathbf{f}}_{k_j}^{(1)},\hat{\mathbf{f}}_{k_j}^{(2)}, \ldots, \hat{\mathbf{f}}_{k_j}^{(c)}$, are all linearly independent, then we can find $\hat{\mathbf{h}}_{k_j}^{0'} \in V_{k_j}$ such that the minimum mean-square error $\epsilon_j = 0$. For this case we would have $\hat{\mathbf{h}}_{k_j}^{0'} = \tau_j F_{k_j}^{-1}\mathbf{r}_t$, so that class $\mathbf{\Omega}^{(t)}$ is mapped to $\tau_j > 0$, and all the other classes are mapped to 0 by the maximally separating filter.

In general, however, $\dim\{V_{k_j}\} = n_{k_j} < c$ so that $\epsilon_{j,min} > 0$.)

With such an *ad hoc* choice of RIFs, the dimensionality of the optimisation problem can be reduced considerably. (The original problem required optimisation over the roughly $6m$ real variables $(\boldsymbol{\alpha},\boldsymbol{\beta})$; the use of maximally separating filters reduces this to an optimisation problem over the $3m$ variables $(\tau_j, \beta_{k_j,0}, \alpha_j)$, $j=1,\ldots,m$).

An orthonormal basis for V_{k_j} can easily be constructed using the Gram-Schmidt orthogonalisation procedure. Set

$$\mathbf{e}'_{k_j,1} = \hat{\mathbf{f}}_{k_j}^{(L_1)} ,$$

$$\mathbf{e}'_{k_j,l} = \hat{\mathbf{f}}_{k_j}^{(2)} - \left[\frac{\left\langle \hat{\mathbf{f}}_{k_j}^{(2)} , \mathbf{e}'_{k_j} \right\rangle}{\left\langle \mathbf{e}'_{k_j} , \mathbf{e}'_{k_j} \right\rangle} \right] \mathbf{e}'_{k_j} .$$

Then the vectors $\mathbf{e}_{k_j,l} = \dfrac{\mathbf{e}'_{k_j,l}}{||\mathbf{e}'_{k_j,l}||}$, $l=1,2$, are the required set of orthonormal basis vectors for V_{k_j}.

Note also, that in all but the most pathological cases, a unit vector in $V^*_{k_j}$ may also be constructed using a similar procedure by considering the complex conjugates of the circular harmonic coefficients, $\hat{\mathbf{f}}_{k_j}^{(2)*} = \overline{f}_{k_j}^{(2)}(r)$. Successively orthogonalising $\hat{\mathbf{f}}_{k_j}^{(2)*}$ with respect to each of the vectors $\mathbf{e}_{k_j,l}$ yields a vector in $V^*_{k_j}$.

Again, the procedure is simply generalised to multiple classes. Also note that the Bhattacharyya coefficient assumes an even simpler form when maximally separating RIFs are used. Assuming that the circular harmonic coefficients of the images are linearly independent for the channels considered (so that image classes $\Omega^{(2)}$, are mapped to 0 by each RIF), we have that $\mu^{(1)} - \mu^{(2)} = \sum_{j=1}^{m} \alpha_j \tau_j$ if $s \neq t$. We can hence write equation (3.3.4) as

$$\rho_B = \left(\frac{2\sigma^{(1)}\sigma^{(2)}}{\sigma^{(1)^2}+\sigma^{(2)^2}} \right)^{\frac{1}{2}} \exp\left\{ -\frac{1}{4} \frac{(\sum_{j=1}^{m} \alpha_j \tau_j)^2}{\sigma^{(1)^2}+\sigma^{(2)^2}} \right\} . \qquad (3.4.2)$$

A particularly simple *ad hoc* choice of RIP which yields good performance can be realised as follows:

Each channel of the RIP utilises a maximally separating filter of the form (3.4.1) with $\tau_j = \tau =$ constant, and $\beta_{k_j,0} = 0$ for each $j=1,...,m$; (i.e., each channel yields the same mean class separation of τ, and we restrict our attention to the unique maximally separating RIF in V_{k_j} for each $j=1,...,m$). Further, we choose the simple weighting rule wherein all channels have the same weight, $\alpha_j = \frac{1}{m}$, $j=1,...,m$. (From the discussion at the end of section 3(C), it is clear that choosing the constant weight $\frac{1}{m}$ is permissible as scaling the weights leaves the performance unchanged). Then the generalised linear discriminant function is given by

$$\mathbf{G} = \frac{1}{m} \sum_{j=1}^{m} \mathbf{G}_j$$

where $E\{\mathbf{G}_j^{(1)}\} - E\{\mathbf{G}_j^{(2)}\} = \tau$.

In this scheme, each independent feature is given equal weight. Here we give an intuitive justification of the scheme. Assuming the features $\mathbf{G}_1, \mathbf{G}_2, \ldots, \mathbf{G}_m$, have similar distributions (i.e., assuming the maximally separating filter energies, η_j, $j=1,...,m$, are approximately equal), then for large m, $\mathbf{G}$ approaches normality, and the mean-to-standard deviation ratio determines performance. Now, the mean-to-standard deviation ratio of $\mathbf{G}$ is of the order of $\sqrt{m}$; hence $\mathbf{G}^{(1)} - \mathbf{G}^{(2)} \rightarrow \mu^{(1)} - \mu^{(2)} = \tau$ for large m, as a consequence of the *Law of Large Numbers*. Thus, as long as the image energy is distributed uniformly over a large number of circular harmonics, such an *ad hoc* weighting scheme may be expected to give good performance.

Now, for independent channels, performance is monotonic with the number of channels used when the optimal weights are used. A point of interest in *ad hoc* weighting schemes such as the one above, is that an optimum number of channels may exist. We illustrate this for the above scheme in what follows.

Let $\{C_j\}_{j=1}^{\infty}$ be a sequence of independent channels using maximally separating filters, and arranged so that the filter energies are monotonically increasing; $\eta_1 \geq \eta_2 \geq \cdots$. For convenience let us assume that the circular harmonic coefficients of the image classes are linearly independent. Now, without loss of generality, let us assume that all images have unit energy: $\int f^{(1)^2} = \sum_{j=1}^{\infty} ||\hat{\mathbf{f}}^{(1)}_{k_j}||^2 = 1.$ Hence, the sequence $\{||\hat{\mathbf{f}}^{(1)}_{k_j}||^2\}$ decreases faster than $\frac{1}{j}$ for sufficiently large j. Now,

$$\langle \hat{\mathbf{f}}^{(1)}_{k_j} , \hat{\mathbf{h}}^{0}_{k_j} \rangle = \tau > 0 .$$

Clearly, by the Cauchy-Schwarz inequality,

$$\eta_j = ||\hat{\mathbf{h}}^{0}_{k_j}||^2 \geq \frac{|\langle \hat{\mathbf{f}}^{(1)}_{k_j} , \hat{\mathbf{h}}^{0}_{k_j} \rangle|^2}{||\hat{\mathbf{f}}^{(1)}_{k_j}||^2} = \frac{\tau^2}{||\hat{\mathbf{f}}^{(1)}_{k_j}||^2} .$$

Hence the sequence $\{\eta_j\}$ increases faster than j for sufficiently large j. In order to keep the variances of equation (2.2.17) finite (in fact, uniformly bounded above) so that the Central Limit Theorem holds, we require that the weights α_j decrease faster than $\frac{1}{j}$. Now, while an optimal choice of weights can be found satisfying this constraint, it is clear that the *ad hoc* choice of equal weights will violate this constraint when the number of channels becomes sufficiently large.

Thus, with the choice of equal weights for each channel, performance will actually *deteriorate* beyond a certain number of channels (essentially because the variances of the individual features start blowing up while the mean class separation remains constant). Consequently, an optimum number of channels can be found for which best performance is obtained.

B. Comparison of System Performance with Standard Matched Filters

So far we've espoused the use of generalised linear discriminant functions because of the simplicity attendant upon dimensionality reduction, and the fact that the unimodal class distributions are well suited for analysis by these generalised linear discriminant functions. However, even with the use of optimal or sub-optimal solutions, it is not clear whether the generalised linear discriminant function can yield good class-discrimination performance (i.e., low probabilities of error). We then attempt to characterise the discrimination information content of multiple stages of RIFs in comparison with that of standard matched filters [9].

A signal-to-noise ratio comparison between the matched filter, and a single channel RIP where the filter is matched to a particular circular harmonic of the input image, shows that there is considerable degradation in the signal-to-noise ratio for the single channel RIP [3]. Though the comparison done was to estimate signal detectability in noise, the same result holds true for the classification problem with the distance measure of equation (3.3.4). The drop in performance of the single channel RIP compared to the matched filter is clearly a consequence of the fact that each circular harmonic contains only a small portion of the total information content in the image. The price to be paid for increasing within-class tolerance (rotation invariance) is therefore a loss in between-class discrimination. However, generalised linear discriminant functions constructed using multiple RIF stages may be expected to improve performance significantly, especially if the number of channels is large, as each channel incorporates more of the essential information content of the image.

For purposes of comparison, we consider a single channel composite "matched" filter which maximally separates image $f^{(1)}$ from the image $f^{(2)}$ [10,11] on the one hand, and a multi-channel RIP using maximally separating RIFs on the other. We assume for simplicity that there is no degeneracy in the images or in the circular harmonics; i.e., the images $f^{(2)}$ are linearly independent, as are the circular harmonics $\hat{\mathbf{f}}_{k_j}^{(2)}$ for each $j=1,\ldots,m$. The noise models at the input and the output are as before.

The single channel maximally separating filter:

We introduce some notation for simplicity. Let (,) denote the standard inner product in the Hilbert space H, i.e., if $f, h \in H$,

$$(f , h) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y)\bar{h}(x,y)\, dxdy \ .$$

The output, $\mathbf{G}_1^{(2)}$, of the single channel processor conditioned upon $f^{(2)}$ being present at the input is given by

$$\mathbf{G}_1^{(2)} = (f^{(2)}, h) + \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbf{N}(x,y)\bar{h}(x,y)\, dxdy + \mathbf{N}_d \ .$$

We define the maximally separating filter $h \in H$, such that

$$\mu_1^{(2)} \triangleq \mathrm{E}\{\mathbf{G}_1^{(2)}\} = \delta_{s1} ,$$

where δ_{st} is the Kronecker delta. Also, we define

$$\eta \triangleq ||h||^2 = (h , h) .$$

$\mathbf{G}_1^{(2)}$ is a Gaussian random variable, with mean $\mu_1^{(2)}$, and variance $\sigma_1^{(2)^2} = \sigma_n^2\eta + \sigma_d^2$. The variance at the output is the same for both classes, and $\mu_1^{(1)} - \mu_1^{(2)} = 1$. Hence, using equation (3.3.4) we get the Bhattacharyya distance, $d_{B,1}$, to be

$$d_{B,1} = -\ln \rho_{B,1} = \frac{1}{8(\eta\sigma_n^2 + \sigma_d^2)} \ . \tag{3.4.3}$$

Multiple maximally separating RIF stages:

We choose the maximally separating RIFs, $\hat{\mathbf{h}}_{k_j}$ and the weights α_j , $j=1,...,m$,such that

$$\left\langle \hat{\mathbf{f}}_{k_j}^{(2)} , \hat{\mathbf{h}}_{k_j} \right\rangle = \delta_{s1} ,$$

and $\sum_{j=1}^{m} \alpha_j = 1$. (This maintains a class separation of unity in mean, as for the single channel filter). From equation (3.4.2), the Bhattacharyya distance, $d_{B,2}$, is given by

$$d_{B,2} = -\ln\rho_{B,2} = \frac{1}{4(\sigma^{(1)^2} + \sigma^{(2)^2})} - \frac{1}{2}\ln\left(\frac{2\sigma^{(1)}\sigma^{(2)}}{\sigma^{(1)^2} + \sigma^{(2)^2}} \right) . \tag{3.4.4}$$

Substituting for $\sigma^{(2)^2}$ from equations (3.4.3) and (3.4.4), we get the performance ratio, Δ, given by

$$\Delta = \frac{d_{B,2}}{d_{B,1}} = \frac{\eta\sigma_n^2 + \sigma_d^2}{\sum_{j=1}^{m} \alpha_j^2 \left(\eta_j^2\sigma_n^4 + \eta_j\sigma_n^2 + \sigma_d^2 \right)} + 2(\eta\sigma_n^2 + \sigma_d^2)$$

$$\times \left[2\ln\left\{ \sum_{j=1}^{m} \alpha_j^2(\eta_j^2\sigma_n^4 + \eta_j\sigma_n^2 + \sigma_d^2) \right\} - \ln\left\{ \sum_{j=1}^{m} \alpha_j^2(\eta_j^2\sigma_n^4 + \sigma_d^2) \right\} \right.$$

$$\left. - \ln\left\{ \sum_{j=1}^{m} \alpha_j^2(\eta_j^2\sigma_n^4 + 2\eta_j\sigma_n^2 + \sigma_d^2) \right\} \right]$$

$$= \Delta_1 + \Delta_2 , \tag{3.4.5}$$

where Δ_1 corresponds to the first term, and Δ_2 corresponds to the second.

Limiting expressions when the noise level is high:

We consider first the case of very noisy inputs, i.e., large σ_n^2. It is not difficult to show that both Δ_1 and Δ_2 approach zero as σ_n^2 grows large. Hence, the performance ratio, $\Delta \rightarrow 0$ as $\sigma_n^2 \rightarrow \infty$. For large input noise we have the usual effect of square law processing causing considerable deterioration in performance compared to the linear filter. Note, however, that Δ_2 approaches zero very slowly as σ_n^2 grows large, so that the deterioration in performance is not very rapid.

We now consider the case of large output noise terms, $\sigma_d^2 \rightarrow \infty$. Again, it is simple to show that Δ_2 approaches zero, while Δ_1 approaches $\frac{1}{\sum \alpha_j^2}$ as the output noise grows large. Hence, the performance ratio $\Delta \rightarrow \Delta^0 = \frac{1}{\sum_{j=1}^{m} \alpha_j^2}$ as $\sigma_d^2 \rightarrow \infty$. By choosing $\{\alpha_j\}$ to be a discrete probability distribution so that $\sum_{j=1}^{m} \alpha_j = 1$, $\alpha_j \geq 0$, we have that

$$\Delta^0 = \frac{1}{\sum_{j=1}^{m} \alpha_j^2} \geq 1 .$$

Consequently, with a proper choice of weights, the performance of the multichannel RIP can be better than that of the composite matched filter for large output noise. (This is simply a consequence of the Law of Large Numbers; summing a large number of independent random variables essentially washes out the noise terms, thus giving improvement in classification performance).

Limiting expressions for the case of low input noise when output noise is absent:

For small σ_n^2, we have $\sigma_n^4 << \sigma_n^2$; neglecting the σ_n^4 term in equation (3.4.5), and setting $\sigma_d^2 = 0$, we get that $\Delta_2 \rightarrow 0$ as $\sigma_n^2 \rightarrow 0$. Hence,

$$\lim_{\sigma_n^2 \to 0} \Delta = \frac{\eta}{\sum\limits_{j=1}^{m} \alpha_j^2 \eta_j} \quad ; \sigma_d^2 = 0 \ . \tag{3.4.6}$$

The relative performance is determined entirely by the energies of the respective maximally separating filters, the number of channels, and the choice of weights. While the expression is simple, it is not clear whether either scheme enjoys any advantage over the other. We then make some simplifications to obtain bounds for the performance ratio.

Maximising Δ with regard to the weights α_j, and subject to the constraint $\sum \alpha_j = 1$, $\alpha_j \geq 0$, yields that the optimum choice of weights maximising (3.4.6) is

$$\alpha_k^{(0)} = \frac{\left(\sum\limits_{j \neq k} \eta_j^{-1} \right)^{-1}}{\eta_k} \quad , k = 1, \ldots, m \ .$$

Substituting in (3.4.6) we get

$$\Delta = \frac{\eta}{\left(\sum\limits_{j=1}^{m} \eta_j^{-1} \right)^{-1}} \ . \tag{3.4.7}$$

The bank of m maximally separating RIFs in parallel is equivalent to a single maximally separating filter of energy $\left(\sum\limits_{j=1}^{m} \eta_j^{-1} \right)^{-1}$ when the input noise level is low. Note that $\left(\sum\limits_{j=1}^{m} \eta_j^{-1} \right)^{-1} < \min(\eta_j)$, so that this "equivalent" filter has less energy than any of the individual RIFs. From (3.4.4) we see that this implies that performance improves monotonically with the number of independent stages added, and this augurs good results in comparison with matched filters.

Now, bounds on the performance ratio can be obtained by means of spectral analysis. From equation (3.4.1), $\hat{\mathbf{h}}_{k_j} = \mathrm{F}_{k_j}^{-1}\mathbf{r}^{(1)}$, $j=1,...,m$. Hence

$$\eta_j = \left\langle \hat{\mathbf{h}}_{k_j} , \hat{\mathbf{h}}_{k_j} \right\rangle = \left\langle \mathbf{r}_t , (\mathrm{F}_{k_j}\mathrm{F}_{k_j}^{\dagger})^{-1}\mathbf{r}_t \right\rangle > 0 .$$

$\mathrm{F}_{k_j}\mathrm{F}_{k_j}^{\dagger}$ is a positive definite Hermitian transformation, and thus has 2 positive eigenvalues. Let $\{\lambda_{k_j,l}\}_{l=1}^{2}$ be the eigenvalues of $\mathrm{F}_{k_j}\mathrm{F}_{k_j}^{\dagger}$, arranged so that

$$0 < \lambda_{k_j,1} \leq \lambda_{k_j,2} .$$

The eigenvalues of $(\mathrm{F}_{k_j}\mathrm{F}_{k_j}^{\dagger})^{-1}$ are then $\{\lambda_{k_j,l}^{-1}\}_{l=1}^{2}$. Let $r_l^{(1)}$, $l=1,2,$ be the projections of the unit vector $\mathbf{r}^{(1)}$ on the basis of eigenvectors of $(\mathrm{F}_{k_j}\mathrm{F}_{k_j}^{\dagger})^{-1}$. Then

$$\eta = \sum_{l=1}^{2} \lambda_{k_j,l}^{-1} \mid r_l^{(1)} \mid^2 .$$

Now, $\mathbf{r}^{(1)}$ is a vector of unit norm; $||\mathbf{r}^{(1)}||^2 = \sum_{l=1}^{c} \mid r_l^{(1)} \mid^2 = 1$. Hence $\mid r_l^{(1)} \mid^2 \leq 1$. By arrangement of $\{\lambda_{k_j,l}\}$, we then have

$$0 < \lambda_{k_j,2}^{-1} \leq \eta_j \leq \lambda_{k_j,1}^{-1} \quad , j=1,...,m .$$

Similarly, defining the (deterministic) correlation matrix F with elements $\mathrm{F}_{r,s} = (f^{(r)} , f^{(2)})$, we obtain

$$0 < \lambda_2^{-1} \leq \not{\eta} \leq \lambda_1^{-1} ,$$

where $0 < \lambda_1 \leq \lambda_2$ are the 2 positive eigenvalues of the positive definite matrix $FF^{\dagger}$. Substituting in (3.4.7) we get

$$0 < \lambda_2^{-1} \sum_{j=1}^{m} \lambda_{k_j,1} \leq \frac{d_{B,\ell}}{d_{B,1}} \leq \lambda_1^{-1} \sum_{j=1}^{m} \lambda_{k_j,2} \ . \tag{3.4.8}$$

If the weights α_j are not optimally chosen, the bound is

$$0 < \frac{\lambda_2^{-1}}{\sum_{j=1}^{m} \alpha_j^2 \lambda_{k_j,1}^{-1}} \leq \frac{d_{B,\ell}}{d_{B,1}} \leq \frac{\lambda_1^{-1}}{\sum_{j=1}^{m} \alpha_j^2 \lambda_{k_j,2}^{-1}} \ .$$

Thus, for the low input noise case, good performance relative to the matched filter can be achieved. Also, even for very similar image classes, the difference in the images is likely to be exhibited strongly in a few circular harmonics. By choosing just those circular harmonics for which there is good discrimination between classes, good performance can be achieved.

Note that from equation (3.4.7), the performance of the multi-channel RIP improves monotonically as the energy of the "equivalent" filter, $\sum_{j=1}^{m} \hat{h}_{k_j}(r) e^{ik_j\theta}$, decreases. Further, $\left(\sum_{j=1}^{m} \eta_j^{-1} \right)^{-1} < \min(\eta_j)$. This dictates a good *ad hoc* approach to feature selection. Let $\{C_j\}_{j=1}^{\infty}$ be a sequence of channels arranged so that the energies of the corresponding maximally separating filters, $\hat{\mathbf{h}}_{k_j}$, are monotonic; i.e., $\eta_1 \leq \eta_2 \leq \cdots$. Choosing channels $C_1, C_2, \ldots, C_m$, for the RIP then yields the "equivalent" filter with the least energy, and hence the best performance for the low noise case.

REFERENCES

[1] Y. N. Hsu, H. H. Arsenault, and G. April, "Rotation invariant digital pattern recognition using circular harmonic expansion," *Appl. Opt.*, vol. 21, pp. 4012–4015, 1982.

[2] Y. Hsu and H. H. Arsenault, "Optical pattern recognition using circular harmonic expansion," *Appl. Opt.*, vol. 21, pp. 4016–4019, 1982.

[3] H. H. Arsenault, Y. N. Hsu, K. Chalasinka-Macukow, and Y. Yang, "Rotation invariant pattern recognition," *Proc. Soc. Photo-Opt. Instrumen. Eng.*, vol. 359, pp. 266–272, 1982.

[4] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[5] Y. N. Hsu and H. H. Arsenault, "Pattern discrimination by multiple circular harmonic components," *Appl. Opt.*, vol. 23, pp. 841–844, 1984.

[6] R. Wu and H. Stark, "Rotation-invariant pattern recognition using a vector reference," *Appl. Opt.*, vol. 23, pp. 838–840, 1984.

[7] A. M. Cormack, "Representation of a function by its line integral, with some radiological applications," *Jnl. Appl. Phys.*, vol. 34, pp. 2722–2727, 1963.

[8] J. Franklin, *Methods of Mathematical Economics*. New York: Springer-Verlag, 1980.

[9] G. L. Turin, "An introduction to matched filters," *IRE Trans. Inform. Theory*, vol. IT-6, pp. 311–329, 1960.

[10] H. J. Caulfield and W. J. Maloney, "Improved discrimination in optical character recognition," *Appl. Opt.*, vol. 8, pp. 2354–2356, 1969.

[11] H. J. Caulfield, "Linear combinations of filters for character recognition: a unified treatment," *Appl. Opt.*, vol. 19, pp. 3877–3878, 1980.

CHAPTER IV

BINARY FILTERS

1. INTRODUCTION

In this chapter, we examine an application of threshold point rules to generating low cost filtration systems which emulate Matched Filter performance. As noted earlier, Matched Filters are commonly used in divers applications in communication systems, signal processing, and pattern recognition, where the objective is frequently the recognition of a particular signal or pattern immersed in noise. Recapitulating theorem (1.2.1), the principal theoretical result supporting the use of Matched Filters can be encapsulated as follows: *among the class of all linear, shift-invariant systems, Matched Filters maximise a (suitably defined) signal-to-noise ratio* [1]. Specifically, if the (deterministic) signal of interest corrupted by additive, white noise is presented as the input to a linear, shift-invariant system, the ratio of the output signal power to the output noise power is maximised by the Matched Filter. The importance of this statistic, of course, depends on how well the signal-to-noise ratio reflects the (Bayesian) risk, or error probability. If the output correlation peak is a normal random variable, (which would be the case if the noise itself was a normal process, or if it was an independent process), then the signal-to-noise ratio statistic coincides with the risk function, so that the matched filter is *ipso facto* the optimum filter for this case.

Practical implementations of Matched Filters–and linear, shift-invariant systems in general–are much facilitated by the fundamental Fourier convolution theorem wherein convolutions (or correlations) in one domain are transformed into products in the Fourier domain. As a consequence, relatively simple analog implementations such as optical Fourier-plane correlators [2], and digital implementations using algorithms such as the Fast Fourier Transform abound.

We focus on three issues in regard to traditional Matched Filters:

First, the storage of the system transfer function (matched to the Fourier Transform of the signal of interest) presupposes an adequate available dynamic range (for instance, in a hologram in an optical correlator, or in computer memory in digital correlators). Of interest then is the amount of redundant information encoded in the transfer function. More precisely, *at issue is the determination of the minimum number of bits of information required to characterise the pattern or signal, and for which good classification performance still obtains.* The actual determination of an optimal, minimal cost filter, however, depends on the precise statement of the classification problem, as well as on the actual distribution of the classes. Our concern in this paper is with a class of heuristically derived filters; we demonstrate that pattern classes can be accurately classified using filters containing just n bits of information, where n is the space- (time-) bandwidth product of the signal classes.

We investigate a form of extreme quantisation in the Fourier domain–a class of binarisation (or hardlimiting) operations wherein each point in the domain of the transfer function is mapped to a single bit: -1 or +1. The resulting binary filter is then a function whose range is just $\{-1,1\}$. (If the space- (or time-) bandwidth product of the signal is n , then the binarisation operation characterises the transfer function by just n bits of information. Clearly the binarisation process need not be confined to the filter alone, but can also be applied to the Fourier Transform of the input signal. This leads us naturally to the consideration of three systems: the *Matched Filter* with the reference signal (corresponding to the filter transfer function) matched to the desired signal, the *Binary Filter* with the filter being a binarised version of the matched filter, and the *Matched Binary Filter* where both the input signal, and the reference undergo a hardlimiting operation in the Fourier domain. We will see that all three systems can be viewed within the framework of a natural generalisation of a traditional correlational system. The schema of fig. 4.1 illustrates the model system under consideration in block diagrammatic form.

Our concern in this paper will be mainly with the derivation of results characterising the performance of these systems for certain classes of signals; experimental results detailing the use of such processors in recognition systems have been tabulated in [4], [5]. fig. 4.2 demonstrates experimental results for a system using a Binary Filter which was in practice implemented as an optical correlator. The system effectively recognises the letter O from the words MOD and OSF, as can be seen from the prominent auto-correlation peaks. In fig. 4.3 we demonstrate two correlations of a random one-dimensional input sequence; in fig. 4.3 (a) the correlation was accomplished using a Matched Filter, while in fig. 4.3 (b) the correlation was performed using a Binary Filter. Note that a correlation peak is clearly visible for the correlation using the Binary Filter, and that the side-lobe fluctuation levels for the Binary Filter are comparable to that for the Matched Filter.

Binary systems such as the above can be of considerable practical importance. The requirement of a large dynamic range for the filter (corresponding to the many bits required to represent each sample point) is obviated, and just a single representation bit is utilised per sample point. The resultant decrease in required memory storage paves the way for low cost, low complexity systems. Of interest in an optical implementation is the recent availability of a two-dimensional binary spatial light modulator–the magneto-optic device–which has been successfully used in such filtration systems [4], [5].

The second issue we address is how classification is affected by side-lobe energy in the correlation output. High side-lobe peak levels can lead to confusion with the main correlation peak, and consequent erroneous classification. Now, for deterministic signals not much can be said *a priori* about side-lobe structure. We introduce signal statistics and a performance measure akin to a signal-to-noise ratio which incorporates the expected height of the correlation peak, and the maximum spread in side-lobe energy averaged over the ensemble. In this regard it is helpful, though not essential, to think of the signal as being a sample representation of an ergodic process. The

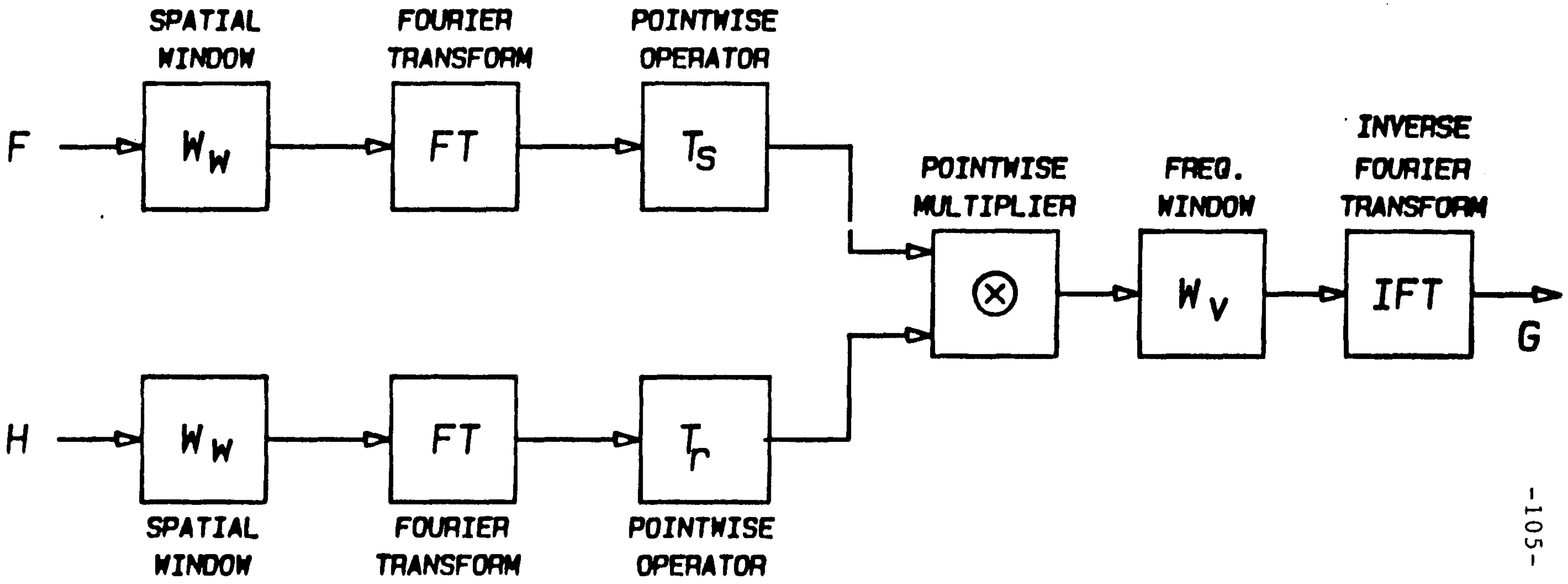

Fig. 4.1. System block diagram.

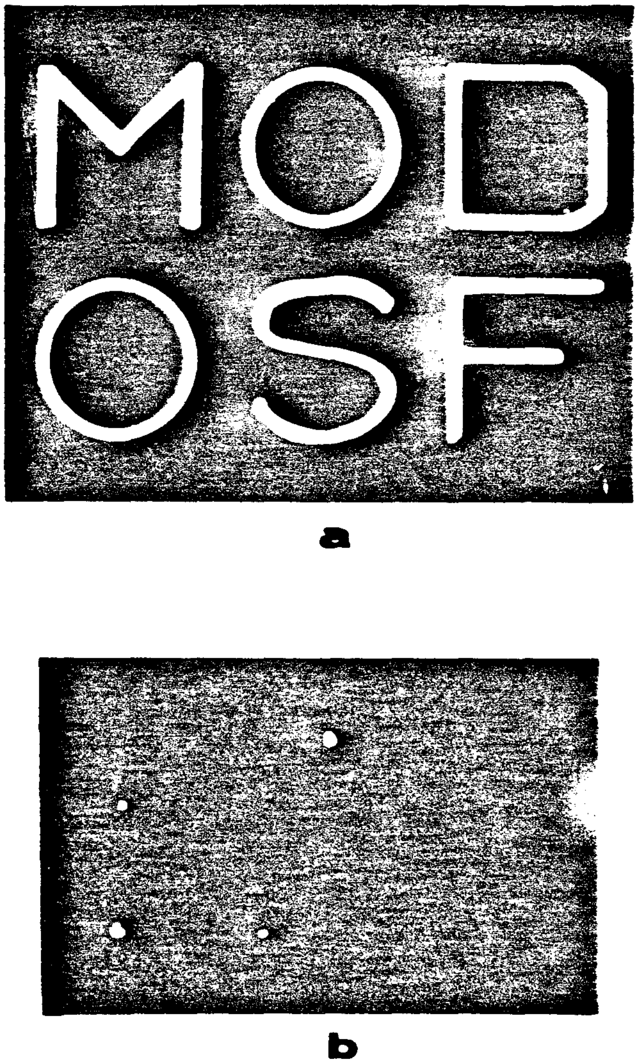

Fig. 4.2. Recognition of the letter O from the words MOD and OSF by an optically implemented Binary Filter.

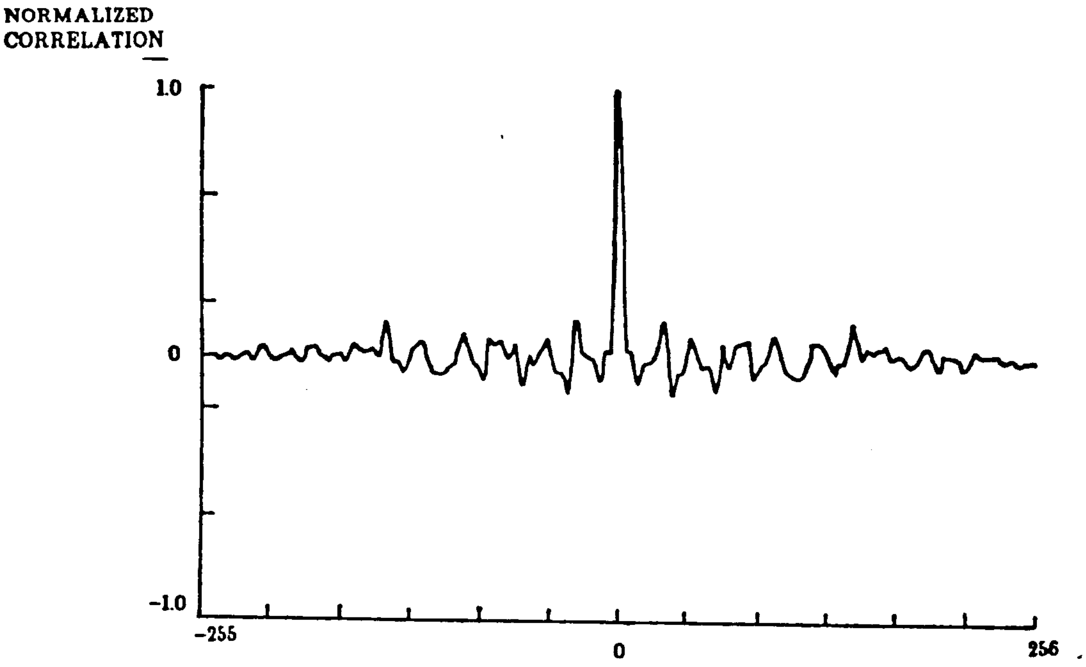

Fig. 4.3 (a). Correlation of a random sequence using a Matched Filter (after [4]).

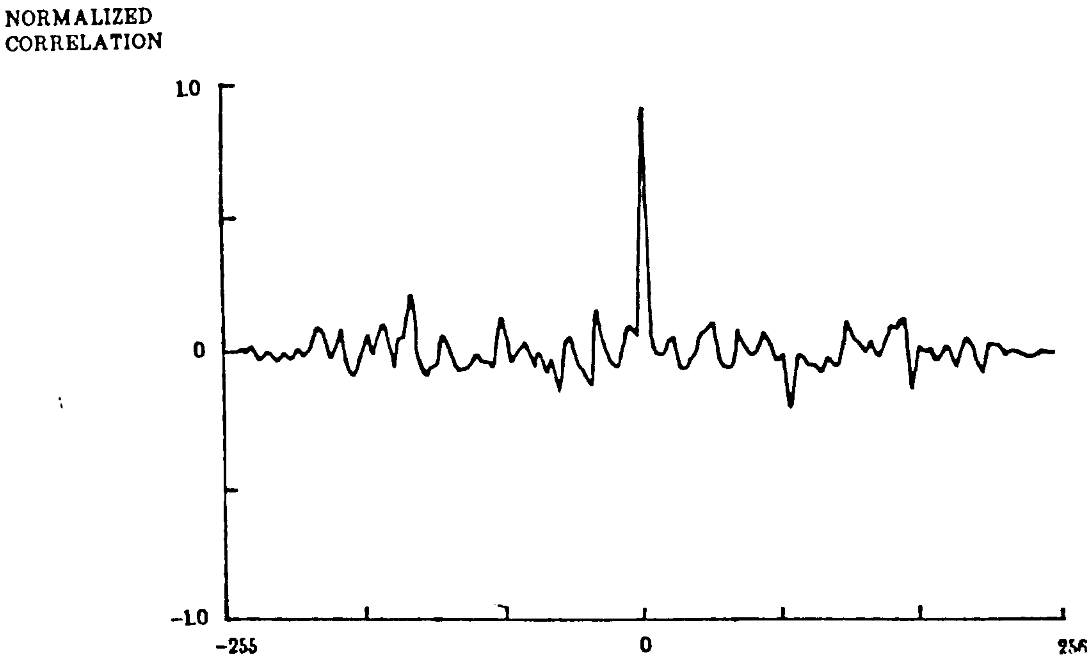

Fig. 4.3 (b). Correlation of a random sequence using a Binary Filter (after [4]).

ensemble average of side-lobe energy is then equivalent to averaging side-lobe energy along the correlation output itself.

The classical idealised Matched Filter requires an infinite system space-bandwidth product, and this, of course, can only be approximated by physical systems. The third issue we address is the effect of restricting the system space-bandwidth product to be finite. (To this extent then, our systems become shift-variant. However, if the system space-bandwidth product is considerably larger than the space-bandwidth product of the signal, shift-invariance is achieved over a reasonable range). Our analysis is statistical, and we do not have recourse to prolate spheroidal wave functions–the eigenmodes of this class of shift-variant systems [6]. In this regard, new results are obtained characterising the performance of the Matched Filter, the Binary Filter, and the Matched Binary Filter as a function of the system space-bandwidth product.

In section 2 we present a detailed picture of the three correlation systems we consider, and introduce a performance measure as a yardstick of their relative performance. We describe the signal statistics in section 3, and derive certain results needed for the analysis. In section 4 we analyse the performance of the three systems in a two-class pattern recognition problem where the patterns belong to well-defined statistical classes, and are noise-free. The attrition in classification performance of the systems when the patterns are corrupted by additive noise is traced in section 5. Discussions of the relative performance of the systems under consideration are covered in sections 6 and 7.

2. SYSTEM MODELS AND PERFORMANCE MEASURE

A. A Generalisation of the Fourier-Plane Correlator

We consider a class of systems which can be described as generalisations of conventional Fourier-plane correlators. These systems are characterised by a space-bandwidth (time-bandwidth) product, p, and may incorporate point-wise nonlinearities in the form of hardlimiting. Fig. 4.1 depicts the generic three-port system under consideration in block-diagram form.

The system has two inputs: a signal, F, which is a sample realisation from some (statistically characterised) signal or pattern class, and a reference signal, H, which is matched to the sample realisation of a specific pattern class. (In practical realisations of such correlator systems, the reference signal would be assimilated within the system as it is known *a priori*. We represent the reference as input to a second port in order to better illustrate the effect of non-linear operations on the signal and the reference.) For definiteness we assume that the input signals are real-valued functions of a real variable (space or time for instance), $F : \mathbb{R} \rightarrow \mathbb{R}$. Henceforth we will refer to the domain of definition of the input signal as "space." We will consider the two-class case to aid in clarity of exposition, and denote by F_1 and F_2, respectively, sample realisations of pattern classes C_1, and C_2. For definiteness, we henceforth denote the (unknown) signal F by F_j, $j=1,2$, corresponding to either F_1 or F_2 being present at the input. We will assume that the reference signal H is matched to the sample realisation F_1 of class C_1.

Both inputs are passed through a spatial window W_ω which effectively limits the support of the input signal and the reference to $[-\omega\, ,\, \omega]$. The windowed functions are then Fourier transformed to yield the complex-valued functions $\hat{\mathrm{F}}_j$ and $\hat{H}$. These are simply the finite-domain (or short term) Fourier transforms of F_j and H:

$$\hat{\mathrm{F}}_j(u) = \int_{-\omega}^{\omega} \mathrm{F}_j(x)\, e^{-i\,2\pi u x}\, dx \ ,$$

$$\hat{H}(u) = \int_{-\omega}^{\omega} H(x)\, e^{-i\,2\pi u x}\, dx \ , \tag{4.2.1}$$

We will refer to the (real) transform variable u as the "frequency." The signal term $\hat{\mathrm{F}}_j$, and the reference term $\hat{H}$ are then subjected to the pointwise operations $T_s : \mathbb{C} \rightarrow \mathbb{C}$, and $T_r : \mathbb{C} \rightarrow \mathbb{C}$, respectively. We restrict our attention to two maps: the identity map $T(\alpha) = \alpha$, and a hardlimiting operation $T(\alpha) = \mathrm{sgn}\,\{\mathrm{Re}\,(\alpha)\} = \begin{cases} 1 & \text{if Re}\,(\alpha) \geq 0 \\ -1 & \text{if Re}\,(\alpha) < 0 \end{cases}$. The signal term and the complex-conjugate of the reference are then multiplied pointwise, and the product is then passed through a frequency window W_ν which limits the support of the product to $[-\nu, \nu]$. An inverse-Fourier transform operation finally yields the complex-valued output function G_j of the system. To summarise then, with inputs $\mathrm{F}_j(x)$ and $H(x)$, the output $G_j(x)$ of the system is given by

$$G_j(x) = \int_{-\nu}^{\nu} T_s\{\hat{\mathrm{F}}_j(u)\} \, \overline{T_r\{\hat{H}(u)\}} \, e^{i2\pi ux} \, du \tag{4.2.2}$$

with $\hat{\mathrm{F}}_j$ and $\hat{H}$ given by equation (4.2.1), and where we use the notation that a bar above a complex variable denotes the complex-conjugate of that particular variable.

Note that with $T_s = T_r = Id$, and $\omega = \nu = \infty$, we just have a conventional Fourier-plane correlator (Matched Filter). The system under consideration represents a more general version of a Fourier-plane correlator. The *system space-bandwidth product* which we denote by p is given by the product of the width of the spatial and frequency windows, $p = 4\omega\nu$. (If the domain of the input-signal is time, then causality requirements enforce that the reference signal H be a suitably delayed version of F_1. To obtain reasonable correlation peaks for discrimination purposes, we then require that the system space-bandwidth product be at least comparable to that of the signal.)

Within the structure of the system, we have some flexibility in the choice of the operations T_s and T_r. We consider three cases.

(1) Linear operation: $T_s = Id$, $T_r = Id$. This is essentially a *Matched Filter* with a (finite) system space-bandwidth product p.

(2) Non-linearity imposed on the reference signal: $T_s = Id$, $T_r = \mathrm{sgn} \ o \, \mathrm{Re}$. This corresponds to a Fourier-plane correlator with a *Binary Filter.*

(3) Non-linearity imposed on both input signal and reference: $T_s = \mathrm{sgn} \ o \, \mathrm{Re}$, $T_r = \mathrm{sgn} \ o \, \mathrm{Re}$. This case corresponds to binarisation of the real part of the Fourier Trapsform of the signal, and a *Matched Binary Filter.*

In terms of the canonical generalised linear discriminant function we have been considering, we can consider system inputs to be a pair of functions $(F^{(j)}(x), H(x))$. The linear transformation $\mathbf{W}$ in this instance, realises a pair of Fourier transforms over finite windows, $\mathbf{W}(F^{(j)}(x),H(x)) = (\hat{\mathrm{F}}^{(j)}(u),\hat{H}(u))$. Here the feature space is the space of the Fourier transforms, indexed by $\nu \in \mathbb{R}$. The point rule **D** operates at each point ν in the Fourier plane: $D(\hat{\mathrm{F}}^{(j)}(u),\hat{H}(u)) = T_s\left\{\hat{\mathrm{F}}^{(j)}(u)\right\} \times \overline{T_r\left\{\hat{H}(u)\right\}}$. The final linear discriminant function, **L**, is realised by an inverse Fourier transform over a finite window. Note that the system is somewhat different from the threshold machines we introduced in chapter II. Specifically, as we will see, we no more have independent channels.

B. Performance Measure

In characterising the performance of the three correlation schemes, we concentrate on two key measures: the strength of the correlation peak, and the side-lobe structure. For specific sample realisations, not much can be said about the size of the side-lobes; however, if signal statistics are known we can extract peak and side-lobe information from a consideration of the ensemble. In the next section we describe a specific statistical structure for the two signal classes from which we can obtain quantitative estimates of the performance of the three proposed schemes.

We define the discrimination efficiency of the correlators considered in terms of the normalised mean separation, ρ, which incorporates information about correlation-peak size, as well as the energy in the side-lobes. For $j = 1,2$ let

$$\mu_j = \sup_x\left\{ \mid \mathrm{E}\left\{G_j(x)\right\} \mid \right\} ,$$

$$\eta_j = \sup_x\left\{\mathrm{Var}\left\{G_j(x)\right\}\right\} .$$

We then define the performance coefficient to be

$$\rho = \frac{(\mu_1 - \mu_2)^2}{\eta_1 + \eta_2} . \qquad (4.2.3)$$

We denote by ρ_m, ρ_b, and ρ_{mb}, respectively, the performance coefficient for the Matched Filter, the Binary Filter, and the Matched Binary Filter. We shall take system peformance to be a monotonically increasing function of the coefficient ρ, with the system with the largest ρ realising the best performance.

Note that the form of the coefficient ρ is similar to a ***signal-to-noise ratio***, the "signal" corresponding to class C_1, and the "noise" to class C_2. (In fact, when the output variable G is Gaussian, and the ***a priori*** probabilities of the two classes are the same, it turns out that the form of the Bhattacharyya coefficient [7] is identical to equation (4.2.3) for ρ.) From classical communication theory we have that for correlational-systems which are linear functionals of the input signal, the peak signal-to-noise ratio for a signal immersed in white noise is obtained for the Matched Filter. Hence we expect that the classification performance of the Binary Filter is bounded by that of the Matched Filter–at least when the system space-bandwidth product is large. The same cannot be concluded ***a priori*** for the Matched Binary Filter, however, because of the non-linearity introduced in the signal path.

3. SIGNAL STATISTICS

We assume that the signals $\mathrm{F}_1(x)$ and $\mathrm{F}_2(x)$ corresponding to the two classes C_1 and C_2 are sample realisations of mutually independent, white random processes with

$$\mathrm{E}\left\{\mathrm{F}_j(x)\right\} = 0 \, ,$$

$$\mathrm{E}\left\{\mathrm{F}_j(x)\mathrm{F}_j(y)\right\} = \sigma_j^2\delta(x-y) \, . \tag{4.3.1}$$

The signal classes have been restricted to be stationary and white in order to effect some simplicity in the ensuing analysis. The stationarity constraint can be relaxed to allow of correlation functions of the form $r_j(x)\delta(x-y)$; the analysis for this case is essentially the same as for the case we consider. With the added constraint that the process be Gaussian, one or both constraints can be relaxed to encompass general correlation functions of the form $r_j(x,y)$.

In what follows we derive expressions for various signal statistics that we need to characterise the performance of the three schemes.

With $\hat{F}^{(j)}(u)$ given by equation (4.2.1) we have

$$\mathrm{Re}\ \hat{F}^{(j)}(u) = \int_{-\omega}^{\omega} F_j(x) \cos 2\pi ux \ dx \ ,$$

$$\mathrm{Im}\ \hat{F}^{(j)}(u) = \int_{-\omega}^{\omega} F_j(x) \sin 2\pi ux \ dx \ . \qquad (4.3.2)$$

Define

$$\mathrm{sgn}\left\{\mathrm{Re}\ \hat{F}^{(j)}(u)\right\} \triangleq \begin{cases} 1 & \text{if Re } \hat{F}^{(j)}(u) \geq 0 \\ -1 & \text{if Re } \hat{F}^{(j)}(u) < 0 \end{cases} .$$

The random processes $F^{(j)}(x)$ are independent, and we then have by virtue of the *Central Limit Theorem* that Re $\hat{F}^{(j)}(u)$ and Im $\hat{F}^{(j)}(u)$ are Gaussian random processes. Further,

$$\mathbf{E}\left\{\mathrm{Re}\ \hat{F}^{(j)}(u)\right\} = \int_{-\omega}^{\omega} \mathbf{E}\left\{F^{(j)}(x)\right\} \cos 2\pi ux \ dx \ = 0 \ ,$$

and

$$\mathbf{E}\left\{\mathrm{Im}\ \hat{F}^{(j)}(u)\right\} = \int_{-\omega}^{\omega} \mathbf{E}\left\{F^{(j)}(x)\right\} \sin 2\pi ux \ dx \ = 0 \ , \qquad (4.3.3)$$

from equation (4.3.1). The Gaussian processes Re $\hat{F}^{(j)}(u)$ and Im $\hat{F}^{(j)}(u)$ are hence zero-mean. Now, using (4.3.1) again we have

$$\mathbf{E}\left\{\mathrm{Re}\ \hat{F}^{(j)}(u)\,\mathrm{Im}\ \hat{F}^{(j)}(1)\right\} = \int_{-\omega}^{\omega}\int_{-\omega}^{\omega} \mathbf{E}\left\{F^{(j)}(x)\,F^{(j)}(y)\right\} \cos 2\pi ux \ \sin 2\pi ty \ dx \ dy$$

$$= \sigma_j^2 \int_{-\omega}^{\omega} \cos 2\pi ux \ \sin 2\pi tx \ dx$$

$$= 0 \ , \qquad (4.3.4)$$

as $\cos 2\pi ux$ is an even function of x , and $\sin 2\pi tx$ is an odd function of x .

Thus Re $\hat{F}^{(j)}(u)$ and Im $\hat{F}^{(j)}(u)$ are uncorrelated random processes, and as a consequence of their being normal, they are also independent. As $F^{(1)}(x)$ and $F^{(2)}(x)$ are mutually independent, we have that Re $\hat{F}^{(1)}(u)$, Im $\hat{F}^{(1)}(u)$, Re $\hat{F}^{(2)}(u)$, and Im $\hat{F}^{(2)}(u)$ are mutually independent Gaussian random processes with zero-mean. To completely characterise these processes it is sufficient now to obtain their second-order statistics. Having recourse again to equations (4.3.1), and (4.3.2), we have for $j=1,2$

$$E\left\{\mathrm{Re}\ \hat{F}^{(j)}(u)\,\mathrm{Re}\ \hat{F}^{(j)}(1)\right\} = \int_{-\omega}^{\omega}\int_{-\omega}^{\omega} E\left\{F^{(j)}(x)\,F_j(y)\right\}\cos 2\pi ux\ \cos 2\pi ty\ dxdy$$

$$= \frac{\sigma_j^2}{2}\int_{-\omega}^{\omega}\left\{\cos 2\pi(u-t)x + \cos 2\pi(u+t)x\right\} dx$$

$$= \sigma_j^2\omega\left\{\mathrm{sinc}\ 2\omega(u-t) + \mathrm{sinc}\ 2\omega(u+t)\right\} \qquad (4.3.5)$$

where

$$\mathrm{sinc}\ x \triangleq \begin{cases} 1 & \text{if } x=0 \\ \dfrac{\sin \pi x}{\pi x} & \text{if } x\neq 0 \end{cases}.$$

In similar fashion we find

$$E\left\{\mathrm{Im}\ \hat{F}^{(j)}(u)\mathrm{Im}\ \hat{F}^{(j)}(1)\right\} = \sigma_j^2\omega\left\{\mathrm{sinc}\ 2\omega(u-t) - \mathrm{sinc}\ 2\omega(u+t)\right\}. \qquad (4.3.6)$$

We also require expressions for the fourth-order moments. Using the fact that Re $\hat{F}^{(j)}(u)$ is a Gaussian process, we have

$$E\left\{\mathrm{Re}\ \hat{F}^{(j)}(u)\,\mathrm{Re}\ \hat{F}^{(j)}(1)\,\mathrm{Re}\ \hat{F}^{(j)}(2)\,\mathrm{Re}\ \hat{F}^{(j)}(r)\right\}$$

$$= E\left\{\mathrm{Re}\ \hat{F}^{(j)}(u)\,\mathrm{Re}\ \hat{F}^{(j)}(1)\right\} E\left\{\mathrm{Re}\ \hat{F}^{(j)}(2)\,\mathrm{Re}\ \hat{F}^{(j)}(r)\right\}$$

$$+ E\left\{\mathrm{Re}\ \hat{F}^{(j)}(u)\,\mathrm{Re}\ \hat{F}^{(j)}(2)\right\} E\left\{\mathrm{Re}\ \hat{F}^{(j)}(1)\,\mathrm{Re}\ \hat{F}^{(j)}(r)\right\}$$

$$+ \mathbf{E}\left\{\mathrm{Re}\ \hat{F}^{(j)}(u)\,\mathrm{Re}\ \hat{F}^{(j)}(r)\right\}\mathbf{E}\left\{\mathrm{Re}\ \hat{F}^{(j)}(1)\,\mathrm{Re}\ \hat{F}^{(j)}(2)\right\}, \qquad (4.3.7)$$

which can be evaluated with the aid of equation (4.3.5). Similarly, using equation (4.3.6) we can estimate

$$\mathbf{E}\left\{\mathrm{Im}\ \hat{F}^{(j)}(u)\,\mathrm{Im}\ \hat{F}^{(j)}(1)\,\mathrm{Im}\ \hat{F}^{(j)}(2)\,\mathrm{Im}\ \hat{F}^{(j)}(r)\right\}$$

$$= \mathbf{E}\left\{\mathrm{Im}\ \hat{F}^{(j)}(u)\,\mathrm{Im}\ \hat{F}^{(j)}(1)\right\}\mathbf{E}\left\{\mathrm{Im}\ \hat{F}^{(j)}(2)\,\mathrm{Im}\ \hat{F}^{(j)}(r)\right\}$$

$$+ \mathbf{E}\left\{\mathrm{Im}\ \hat{F}^{(j)}(u)\,\mathrm{Im}\ \hat{F}^{(j)}(2)\right\}\mathbf{E}\left\{\mathrm{Im}\ \hat{F}^{(j)}(1)\,\mathrm{Im}\ \hat{F}^{(j)}(r)\right\}$$

$$+ \mathbf{E}\left\{\mathrm{Im}\ \hat{F}^{(j)}(u)\,\mathrm{Im}\ \hat{F}^{(j)}(r)\right\}\mathbf{E}\left\{\mathrm{Im}\ \hat{F}^{(j)}(1)\,\mathrm{Im}\ \hat{F}^{(j)}(2)\right\}. \qquad (4.3.8)$$

We finally need to estimate the first and second moments of the random processes sgn $\left\{\mathrm{Re}\ \hat{F}^{(j)}(u)\right\}$ and $|\,\mathrm{Re}\ \hat{F}^{(j)}(u)\,|$. Define $r : \mathbb{R}^3 \rightarrow [-1,1]$ by

$$r(u,t;\omega) \triangleq \frac{\mathrm{sinc}\ 2\omega(u-t) + \mathrm{sinc}\ 2\omega(u+t)}{(1+\mathrm{sinc}\ 4\omega u)^{1/2}(1+\mathrm{sinc}\ 4\omega t)^{1/2}}. \qquad (4.3.9)$$

Note that from equation (4.3.5) it follows that for each u and t, $r(u,t;\omega)$ is just the correlation coefficient of the random variables Re $\hat{F}^{(j)}(u)$ and Re $\hat{F}^{(j)}(1)$.

The derivation of the following results is postponed to Appendices A and B. Identifying the random process Re $\hat{F}^{(j)}(u)$ with the process $X(u)$ in the appendices, we replace the correlation $r_{u,t}$ of equation (A.1) by the expression on the RHS of equation (4.3.5);

$$r_{u,t} = \sigma_j^2\omega\left\{\mathrm{sinc}\ 2\omega(u-t) + \mathrm{sinc}\ 2\omega(u+t)\right\}.$$

Substituting the above in equations (A.2), (B.1), (A.8), and (B.4), we obtain

$$\mathbf{E}\left(\mathrm{sgn}\left\{\mathrm{Re}\ \hat{F}^{(j)}(u)\right\}\right) = 0,$$

(4.3.10)

$$\mathbf{E}\left\{ \,|\, \mathrm{Re}\ \hat{\mathrm{F}}^{(j)}(u)\,|\, \right\} = \left(\frac{2\omega\sigma_j^2}{\pi}\right)^{1/2} (1 + \mathrm{sinc}\ \ 4\omega u\,)^{1/2}\,, \tag{4.3.11}$$

$$\mathbf{E}\,(\mathrm{sgn}\ \left\{\mathrm{Re}\ \ \hat{\mathrm{F}}^{(j)}(u\,)\right\}\ \mathrm{sgn}\ \left\{\mathrm{Re}\ \ \hat{\mathrm{F}}^{(j)}(1)\right\}) = \frac{2}{\pi}\ \sin^{-1} r\,(u\,,t\,;\omega)\ , \tag{4.3.12}$$

and

$$\mathbf{E}\left\{ \,|\, \mathrm{Re}\ \ \hat{\mathrm{F}}^{(j)}(u\,)\,|\quad |\,\mathrm{Re}\ \ \hat{\mathrm{F}}^{(j)}(1)\,|\, \right\}$$

$$= \frac{2\omega\sigma_j^2}{\pi}\ \Big[(1 + \mathrm{sinc}\ \ 4\omega u\,)^{1/2}(1 + \mathrm{sinc}\ \ 4\omega t\,)^{1/2}(1 - [r\,(u\,,t\,;\omega)]^2)^{1/2}$$

$$+\ (\mathrm{sinc}\ \ 2\omega(u - t\,) + \mathrm{sinc}\ \ 2\omega(u + t\,))\ \sin^{-1} r\,(u\,,t\,;\omega)\Big]\,. \tag{4.3.13}$$

We now utilise these results to quantitatively estimate the performance of the three proposed correlator schemes.

4. TWO-CLASS DISCRIMINATION: NO ADDITIVE NOISE

We consider the case where we have two classes of patterns C_1 and C_2. We assume that the signals $\mathrm{F}^{(1)}(x\,)$ and $\mathrm{F}^{(2)}(x\,)$ corresponding to classes C_1 and C_2 are uncontaminated by noise, and have the statistics described in the last section (equation (4.3.1)). The signals $\mathrm{F}^{(1)}(x\,)$ and $\mathrm{F}^{(2)}(x\,)$ are sample realisations of the corresponding statistical classes, and the reference signal is chosen to be $\mathrm{F}^{(1)}(x\,)$.

A. The Matched Filter

Referring back to fig. 4.1, we have $T_s = Id$ and $T_r = Id$ for the Matched Filter. The system is linear in both inputs. Substituting in equation (4.2.2) we have the output $G^{(j)}(x\,)$ corresponding to an input signal $\mathrm{F}^{(j)}(x\,)$ and reference signal $\mathrm{F}^{(1)}(x\,)$ given by

$$G^{(j)}(x) = \int_{-\nu}^{\nu} \hat{\mathrm{F}}^{(j)}(u)\, \overline{\hat{\mathrm{F}}^{(1)}(u)}\, e^{i2\pi ux}\, dx \qquad (4.4.1)$$

with $\hat{\mathrm{F}}^{(j)}(u)$ given by equation (4.2.1).

Our consideration of the Matched Filter as a correlational system described by equation (4.4.1) differs somewhat from the classical deterministic Matched Filter [1]. An immediate point of divergence from the classical system is the incorporation of finite system windows in space and frequency–namely, the inclusion of a finite system space-bandwidth product, $p = 4\omega\nu$, in the governing system equations. A second point of departure from the traditional deterministic filter is the representation of both signal and reference as members of a statistical class rather than as deterministic entities. As a consequence, the performance coefficient, ρ, obtained by averaging the relative signal strength and the "noisy" side-lobe fluctuation across the ensemble, yields a different performance characterisation of the system under consideration than the classical signal-to-noise ratio characterisation of Matched Filters.

We now estimate the performance coefficient ρ_m given by equation (4.2.3) for the case of the Matched Filter.

<u>CLASS 1</u> : For class C_1, we have the system output given by

$$G^{(1)}(x) = \int_{-\nu}^{\nu} \hat{\mathrm{F}}^{(1)}(u)\, \overline{\hat{\mathrm{F}}^{(1)}(u)}\, e^{i2\pi ux}\, dx$$

$$= \int_{-\nu}^{\nu} \left\{[\mathrm{Re}\ \hat{\mathrm{F}}^{(1)}(u)]^2 + [\mathrm{Im}\ \hat{\mathrm{F}}^{(1)}(u)]^2\right\} e^{i2\pi ux}\, du\ .$$

The output mean is hence

$$\mathrm{E}\left\{G^{(1)}(x)\right\} \doteq \int_{-\nu}^{\nu} (\mathrm{E}\left\{[\mathrm{Re}\ \hat{\mathrm{F}}^{(1)}(u)]^2\right\} + \mathrm{E}\left\{[\mathrm{Im}\ \hat{\mathrm{F}}^{(1)}(u)]^2\right\})\, e^{i2\pi ux}\, du\ .$$

From equations (4.3.5) and (4.3.6) we have

$$\mathbf{E}\left\{G^{(1)}(x)\right\} = 2\omega\sigma_1^2 \int_{-\nu}^{\nu} e^{i2\pi ux}\, du = 4\omega\nu\sigma_1^2 \operatorname{sinc} 2\nu x \ . \tag{4.4.2}$$

$$\mu_1 = \sup_x \left\{ \mid \mathbf{E}\left\{G^{(1)}(x)\right\} \mid \right\} = 4\omega\nu\sigma_1^2 \, . \tag{4.4.3}$$

We now estimate the variance:

$$\mathbf{E}\left\{ \mid G^{(1)}(x) \mid^2 \right\} = \int_{-\nu}^{\nu}\int_{-\nu}^{\nu} \mathbf{E}\left\{([\mathrm{Re}\ \hat{\mathrm{F}}^{(1)}(u)]^2 + [\mathrm{Im}\ \hat{\mathrm{F}}^{(1)}(u)]^2)([\mathrm{Re}\ \hat{\mathrm{F}}^{(1)}(1)]^2 + [\mathrm{Im}\ \hat{\mathrm{F}}^{(1)}(1)]^2)\right\}$$

$$\times\ e^{i2\pi(u-t)x}\, dudt \ .$$

Using equations (4.3.5), (4.3.6), (4.3.7), and (4.3.8), and the independence of Re $\hat{\mathrm{F}}^{(1)}(u)$, and Im $\hat{\mathrm{F}}^{(1)}(u)$, we have, after some manipulation, that

$$\mathbf{E}\left\{ \mid G^{(1)}(x) \mid^2 \right\} = 4\omega^2\sigma_1^4 \int_{-\nu}^{\nu}\int_{-\nu}^{\nu} [1 + (\operatorname{sinc}\ 2\omega(u-t))^2 + (\operatorname{sinc}\ 2\omega(u+t))^2]\ e^{i2\pi(u-t)x}\, dudt \ .$$

Using (4.4.2) we have

$$\mathrm{Var}\ G^{(1)}(x)$$

$$= \mathbf{E}\left\{ \mid G^{(1)}(x) \mid^2 \right\} - \mid \mathbf{E}\left\{G^{(1)}(x)\right\} \mid^2$$

$$= 4\omega^2\sigma_1^4 \int_{-\nu}^{\nu}\int_{-\nu}^{\nu} [(\operatorname{sinc}\ 2\omega(u-t))^2 + (\operatorname{sinc}\ 2\omega(u+t))^2] \cos 2\pi(u-t)x\ dudt \ . \tag{4.4.4}$$

Hence,

$$\eta_1 = \sup_x \left\{\mathrm{Var}\ G^{(1)}(x)\right\}$$

$$= \mathrm{Var}\ G'_1(0)$$

$$= 4\omega^2\sigma_1^4 \int_{-\nu}^{\nu}\int_{-\nu}^{\nu} [(\operatorname{sinc}\ 2\omega(u-t))^2 + (\operatorname{sinc}\ 2\omega(u+t))^2]\ du\ \ dt$$

$$= 8\omega^2\sigma_1^4 \int_{-\nu}^{\nu}\int_{-\nu}^{\nu} [\text{sinc } 2\omega(u-t)]^2 \, du \; dt \; .$$

Setting $u - t = s$, and exchanging the order of integration we find that this evaluates to

$$\eta_1 = 8\omega^2\sigma_1^4 \int_{-2\nu}^{2\nu} (2\nu - \mid s \mid)[\text{sinc } 2\omega s]^2 ds$$

$$= 64\omega^2\nu^2\sigma_1^4 \int_{0}^{1} (1 - t)(\text{sinc } 4\omega\nu t)^2 dt \; , \qquad (4.4.5)$$

where we've changed the variable of integration by setting $s = 2\nu t$, and used the fact that $(1 - \mid t \mid)$, and $(\text{sinc } 4\omega\nu t)^2$ are both even functions of t.

<u>CLASS 2</u> : For class C_2, we have the system output given by

$$G^{(2)}(x) = \int_{-\nu}^{\nu} \hat{\mathrm{F}}^{(2)}(u) \, \overline{\hat{\mathrm{F}}^{(1)}(u)} \, e^{i 2\pi u x} \, du \; . \qquad (4.4.6)$$

Hence,

$$\mathbf{E}\left\{G^{(2)}(x)\right\} = \int_{-\nu}^{\nu} \mathbf{E}\left\{\hat{\mathrm{F}}^{(2)}(u) \, \overline{\hat{\mathrm{F}}^{(1)}(u)}\right\} e^{i 2\pi u x} \, du \; .$$

By independence of the processes $\mathrm{F}^{(1)}(x)$ and $\mathrm{F}^{(2)}(x)$, and from equation (4.3.3) we then have

$$\mathbf{E}\left\{G^{(2)}(x)\right\} = 0 \; . \qquad (4.4.7)$$

Hence

$$\mu_2 = \sup_x \left\{ \mid \mathbf{E}\left\{G^{(2)}(x)\right\} \mid \right\} = 0 \; . \qquad (4.4.8)$$

To estimate the output variance, we have

$$\mathbf{E}\left\{ \mid G^{(2)}(x) \mid ^2\right\} = \int_{-\nu}^{\nu}\int_{-\nu}^{\nu} \mathbf{E}\left\{\hat{\mathrm{F}}^{(2)}(u) \, \overline{\hat{\mathrm{F}}^{(2)}(1)}\right\} \mathbf{E}\left\{\overline{\hat{\mathrm{F}}^{(1)}(u)} \, \hat{\mathrm{F}}^{(1)}(1)\right\} e^{i 2\pi(u-t)x} \, dudt$$

where we have again used the statistical independence of the two classes. Now from equations (4.3.4), (4.3.5), and (4.3.6) we have

$$\mathbf{E}\left\{\hat{\mathrm{F}}^{(j)}(u)\,\overline{\hat{\mathrm{F}}^{(j)}(1)}\right\} = 2\omega\sigma_j^2 \operatorname{sinc}\ 2\omega(u-t)\ . \tag{4.4.9}$$

Hence

$$\mathbf{E}\left\{\,|\,G^{(2)}(x)\,|\,^2\right\} = 4\omega^2\sigma_1^2\sigma_2^2\int_{-\nu}^{\nu}\int_{-\nu}^{\nu}[\operatorname{sinc}\ 2\omega(u-t)]^2\cos 2\pi(u-t)x\ dudt\ , \tag{4.4.10}$$

and by the same procedure as before, we obtain

$$\eta_2 = 32\omega^2\nu^2\sigma_1^2\sigma_2^2\int_0^1 (1-t)(\operatorname{sinc}\ 4\omega\nu t)^2 dt\ . \tag{4.4.11}$$

Define α as a function of the space-bandwidth product p by

$$\alpha(p) = \int_0^1 (1-t)(\operatorname{sinc}\ pt)^2 dt\ . \tag{4.4.12}$$

Combining equations (4.2.3), (4.4.3), (4.4.5), (4.4.8), (4.4.11), and (4.4.12) we have the performance coefficient given by

$$\rho_m = \frac{\dfrac{\sigma_1^2}{\sigma_2^2}}{2\,\alpha(p)\left(1+2\dfrac{\sigma_1^2}{\sigma_2^2}\right)}\ . \tag{4.4.13}$$

Asymptotic results: The expression (4.4.13) can be readily evaluated for extreme values of the system space-bandwidth product. For very low space-bandwidth products, $p \rightarrow 0$, the integral in (4.4.13) converges to 1/2, so that

$$\rho_m \rightarrow \frac{\dfrac{\sigma_1^2}{\sigma_2^2}}{\left(1+2\dfrac{\sigma_1^2}{\sigma_2^2}\right)} \quad \text{as } p \rightarrow 0 \,.$$

For very high space-bandwidth products, $p \rightarrow \infty$, on the other hand, the integral asymptotically approaches the value $p/2$, so that

$$\rho_m \rightarrow \frac{\dfrac{p\,\sigma_1^2}{\sigma_2^2}}{\left(1+2\dfrac{\sigma_1^2}{\sigma_2^2}\right)} \quad \text{as } p \rightarrow \infty \,.$$

The asymptotic results correspond well with intuition. For very low space-bandwidth products we expect a low processing gain for the system since very little correlation matching can be obtained. For very high space-bandwidth products on the other hand, the use of uncorrelated signals at the input yields optimal (very high) processing gains. (The fact that ρ_m grows unboundedly for large space-bandwidth products should not be disturbing. The use of wide-sense stationary signals at the input results in a very rapid growth of the correlation peak with increasing space-bandwidth product; for wide-sense stationary processes almost all members of the ensemble have infinite energy. Furthermore, the uncorrelated nature of the processes (they are statistically independent) results in very small sidelobes; in fact, the sidelobes decrease very rapidly as the space bandwidth product increases. In fine, the performance measure ρ_m grows proportionally with increase in the space-bandwidth product. In a sense, then, the statistical structure of the input signals is particularly well suited for such correlation matching.) In practice, the results will be tempered by the practical constraints of finite signal energies, and the fact that signals tend to have some measure of correlation (a correlation length); the one lowers the correlation peak value, while the other increases the energy in the sidelobes. Our results for the case under consideration should, however, serve to illustrate the performance trends in these processors.

It is instructive to compare the performance measure given by equation (4.4.13) with the classical Matched Filter result for the signal-to-noise ratio of a deterministic signal immersed in white noise. The *processing gain* of a classical system (defined to be the ratio of the output signal-to-noise ratio to the input signal-to-noise ratio) is given essentially by the signal space-bandwidth product [1]. If we define σ_1^2/σ_2^2 to be a measure of the input signal-to-noise ratio for the statistical case under consideration, then the processing gain of our system is given by $\frac{1}{2\alpha(p)(1+2\sigma_1^2/\sigma_2^2)}$. The term $2\sigma_1^2/\sigma_2^2$ appears because the statistical side-lobe fluctuations of the "signal" term F_1 itself also introduces some "noisy" variance. Neglecting this for the nonce, we see that the processing gain of the system is given by $1/2\alpha(p)$. If the system space-bandwidth product p is large, $1/2\alpha(p) \approx p$, which is consistent with the the classical result (assuming the signal space-bandwidth product and the system space-bandwidth product are reasonably well matched). For small p, however, the processing gain is approximately unity so that there is effectively no gain in the system. Thus the presence of a finite system space-bandwidth product manifests itself in a loss of processing gain; the larger the space-bandwidth product, the more the processing gain realised by the system.

B. The Binary Filter

The Binary Filter is obtained by introducing the pointwise hardlimiting nonlinearity into the path of the reference signal input in fig. 4.1; specifically, $T_s = Id$, and $T_r = \text{sgn} \circ \text{Re}$. Substituting in equation (4.2.2) we have the correlation output of the system given by

$$G^{(j)}(x) = \int_{-\nu}^{\nu} \hat{F}^{(j)}(u)\,\text{sgn}\left\{\text{Re}\ \hat{F}^{(1)}(u)\right\} e^{i2\pi ux}\,du\ . \tag{4.4.14}$$

<u>CLASS 1</u> : The output of the system with $F^{(1)}(x)$ at the input is given by substitution in equation (4.4.14):

$$G^{(1)}(x) = \int_{-\nu}^{\nu} \hat{F}^{(1)}(u) \operatorname{sgn} \left\{ \operatorname{Re} \ \hat{F}^{(1)}(u) \right\} e^{i2\pi ux} \ du$$

$$= \int_{-\nu}^{\nu} | \operatorname{Re} \ \hat{F}^{(1)}(u) | \ e^{i2\pi ux} \ du \ + i \int_{-\nu}^{\nu} \operatorname{Im} \ \hat{F}^{(1)}(u) \operatorname{sgn} \left\{ \operatorname{Re} \ \hat{F}^{(1)}(u) \right\} e^{i2\pi ux} \, du \qquad (4.4.15)$$

The output mean is hence given by

$$\mathbf{E} \left\{ G^{(1)}(x) \right\} = \int_{-\nu}^{\nu} \mathbf{E} \left\{ \, | \operatorname{Re} \ \hat{F}^{(1)}(u) \, | \, \right\} e^{i2\pi ux} \ du$$

$$+ \, i \int_{-\nu}^{\nu} \mathbf{E} \left\{ \operatorname{Im} \ \hat{F}^{(1)}(u) \right\} \mathbf{E} \left\{ \operatorname{sgn} \left\{ \operatorname{Re} \ \hat{F}^{(1)}(u) \right\} \right\} e^{i2\pi ux} \ du$$

where we have used the fact that Re $\hat{F}^{(1)}(u)$ and Im $\hat{F}^{(1)}(u)$ are independent. Then using equations (4.3.3), (4.3.10), and (4.3.11),

$$\mathbf{E} \left\{ G^{(1)}(x) \right\} = \left(\frac{2\omega\sigma_1^2}{\pi} \right)^{1/2} \int_{-\nu}^{\nu} (1 + \operatorname{sinc} \ 4\omega u)^{1/2} \, e^{i2\pi ux} \ du \ , \qquad (4.4.16)$$

and

$$\mu_1 = \sup_x \left\{ \, | \, \mathbf{E} \left\{ G^{(1)}(x) \right\} | \, \right\}$$

$$= \mathbf{E} \left\{ G_1(0) \right\}$$

$$= \left(\frac{8\omega\nu^2\sigma_1^2}{\pi} \right)^{1/2} \int_0^1 (1 + \operatorname{sinc} \ pt)^{1/2} \, dt \ , \qquad (4.4.17)$$

where, as before, p is the space-bandwidth product $4\omega\nu$.

We now obtain the output variance for class C_1. In equation (4.4.15) set

$$H_1(x) = \int_{-\nu}^{\nu} | \operatorname{Re} \ \hat{F}^{(1)}(u) | \ e^{i2\pi ux} \ du \ ,$$

$$H_2(x) = \int_{-\nu}^{\nu} \mathrm{Im}\ \hat{F}^{(1)}(u)\, \mathrm{sgn}\left\{\mathrm{Re}\ \hat{F}^{(1)}(u)\right\} e^{i2\pi ux}\ du\ .$$

Then

$$G^{(1)}(x) = H_1(x) + iH_2(x)\ .$$

Hence

$$\mathbf{E}\left\{\, | G^{(1)}(x) |^2\right\} = \mathbf{E}\left\{\, | H_1(x) |^2\right\} + \mathbf{E}\left\{\, | H_2(x) |^2\right\}$$

$$- i\mathbf{E}\left\{H_1(x)\,\overline{H_2(x)}\right\} + i\mathbf{E}\left\{\overline{H_1(x)}\, H_2(x)\right\}$$

$$= \mathbf{E}\left\{\, | H_1(x) |^2\right\} + \mathbf{E}\left\{\, | H_2(x) |^2\right\} - 2\mathrm{Im}\left(\mathbf{E}\left\{H_1(x)\,\overline{H_2(x)}\right\}\right)\ .$$

Now

$$\mathbf{E}\left\{H_1(x)\,\overline{H_2(x)}\right\} = \int_{-\nu}^{\nu}\int_{-\nu}^{\nu} \mathbf{E}\left\{\, |\mathrm{Re}\ \hat{F}^{(1)}(u) |\ \ \mathrm{sgn}\left\{\mathrm{Re}\ \hat{F}^{(1)}(t)\right\}\right\}$$

$$\times\ \ \mathbf{E}\left\{\mathrm{Im}\ \hat{F}^{(1)}(t)\right\} e^{i2\pi(u-t)x}\ dudt$$

$$= 0$$

as $|\mathrm{Re}\ \hat{F}^{(1)}(u)|\ \mathrm{sgn}\left\{\mathrm{Re}\ \hat{F}^{(1)}(t)\right\}$ is independent of $\mathrm{Im}\ \hat{F}^{(1)}(t)$, and $\mathrm{Im}\ \hat{F}^{(1)}(t)$ has zero-mean (equation (4.3.3)). Also, from equation (4.4.13) we have

$$\mathbf{E}\left\{\, | H_1(x) |^2\right\} = \int_{-\nu}^{\nu}\int_{-\nu}^{\nu} \mathbf{E}\left\{\, |\mathrm{Re}\ \hat{F}^{(1)}(u) |\ \ |\mathrm{Re}\ \hat{F}^{(1)}(t) |\,\right\} e^{i2\pi(u-t)x}\ dudt$$

$$= \frac{2\omega\sigma_1^2}{\pi}\int_{-\nu}^{\nu}\int_{-\nu}^{\nu} \Big[(1 + \mathrm{sinc}\ \ 4\omega u)^{1/2}(1 + \mathrm{sinc}\ \ 4\omega t)^{1/2}(1 - [r(u,t;\omega)]^2)^{1/2}$$

$$+ \left\{\mathrm{sinc}\ \ 2\omega(u-t) + \mathrm{sinc}\ 2\omega(u+t)\right\} \sin^{-1} r(u,t;\omega)\Big] \cos 2\pi(u-t)x\ \ dudt\ \ ,$$

and from equations (4.3.6), and (4.3.12) we have

$$\mathbf{E}\left\{\, | H_2(x) |^2\right\}$$

$$= \int_{-\nu}^{\nu}\int_{-\nu}^{\nu} \mathrm{E}\left\{\mathrm{Im}\ \hat{F}^{(1)}(u)\ \mathrm{Im}\ \hat{F}^{(1)}(t)\right\}$$

$$\times\ \mathrm{E}\left\{\mathrm{sgn}\left\{\mathrm{Re}\ \hat{F}^{(1)}(u)\right\}\ \mathrm{sgn}\left\{\mathrm{Re}\ \hat{F}^{(1)}(t)\right\}\right\}\ e^{i2\pi(u-t)x}\ dudt$$

$$= \frac{2\omega\sigma_1^2}{\pi}\int_{-\nu}^{\nu}\int_{-\nu}^{\nu}\left\{\mathrm{sinc}\ 2\omega(u-t) - \mathrm{sinc}\ 2\omega(u+t)\right\}\ \sin^{-1}r(u,t;\omega)\ \cos 2\pi(u-t)x\ \ dudt\ ,$$

where $r(u,t;\omega)$ as defined in equation (4.3.9) is given by

$$r(u,t;\omega) = \frac{\mathrm{sinc}\ 2\omega(u-t) + \mathrm{sinc}\ 2\omega(u+t)}{(1+\mathrm{sinc}\ 4\omega u)^{1/2}(1+\mathrm{sinc}\ 4\omega t)^{1/2}}.$$

Using expression (4.4.16) for $\mathrm{E}\left\{G^{(1)}(x)\right\}$, we finally get

$$\mathrm{Var}\ G^{(1)}(x) = \mathrm{E}\left\{\ |\ G^{(1)}(x)\ |^2\right\} - \ |\ \mathrm{E}\left\{G^{(1)}(x)\right\}\ |^2$$

$$= \mathrm{E}\left\{\ |\ H_1(x)\ |^2\right\} + \mathrm{E}\left\{\ |\ H_2(x)\ |^2\right\} - \ |\ \mathrm{E}\left\{G^{(1)}(x)\right\}\ |^2$$

$$= \frac{2\omega\sigma_1^2}{\pi}\int_{-\nu}^{\nu}\int_{-\nu}^{\nu}\Big[2\mathrm{sinc}\ \ 2\omega(u-t)\ \sin^{-1}r(u,t;\omega) - (1+\mathrm{sinc}\ \ 4\omega u)^{1/2}$$

$$\times\ \ (1+\mathrm{sinc}\ \ 4\omega t)^{1/2}\left\{1-(1-[r(u,t;\omega)]^2)^{1/2}\right\}\Big]\cos 2\pi(u-t)x\ \ dudt$$

$$= \frac{4omeg\,\nu^2\sigma_1^2}{\pi}\int_{-1}^{1}\int_{-1}^{1}\Big[\mathrm{sinc}\ \ \frac{p}{2}(u-t)\ \sin^{-1}r(u,t;0.25p) - \frac{1}{2}\ (1+\mathrm{sinc}\ \ pu)^{1/2}$$

$$\times\ \ (1+\mathrm{sinc}\ \ pt)^{1/2}(1-\sqrt{1-[r(u,t;0.25p)]^2})\Big]\cos 2\pi(u-t)\nu x\ \ dudt\ , \qquad (4.4.18)$$

where we've used the identity $r(\nu u,\nu t;\omega) = r(u,t;0.25p)$, which can be verified by direct substitution in the defining equation (4.3.9) for r, with $p = 4\omega\nu$.

No analytic expression is available in general for $\eta_1 = \sup_x \left\{ \text{Var } G^{(1)}(x) \right\}$, and we have to resort to numerical evaluation for specified parameters p, σ_1^2, and σ_2^2. (Note that, in general, the supremum does not occur at $x = 0$).

<u>CLASS 2</u> : From equation (4.4.14), the output for class C_2 is given by

$$G^{(2)}(x) = \int_{-\nu}^{\nu} \hat{F}^{(2)}(u) \text{ sgn} \left\{ \text{Re } \hat{F}^{(1)}(u) \right\} e^{i 2\pi u x} \, du \ .$$

The output mean is hence

$$\mathbf{E} \left\{ G^{(2)}(x) \right\} = \int_{-\nu}^{\nu} \mathbf{E} \left\{ \hat{F}^{(2)}(u) \right\} \mathbf{E} \left\{ \text{sgn} \left\{ \text{Re } \hat{F}^{(1)}(u) \right\} \right\} e^{i 2\pi u x} \, du = 0$$

from equations (4.3.3), and (4.3.10), and using the independence of $F^{(1)}(x)$, and $F^{(2)}(x)$. Hence

$$\mu_2 = \sup_x \left\{ \mid \mathbf{E} \left\{ G^{(2)}(x) \right\} \mid \right\} = 0 \ . \tag{4.4.19}$$

Since $G^{(2)}(x)$ has zero-mean, we can estimate the class variance as

$$\text{Var } G^{(2)}(x) = \mathbf{E} \left\{ \mid G^{(2)}(x) \mid^2 \right\}$$

$$= \int_{-\nu}^{\nu} \int_{-\nu}^{\nu} \mathbf{E} \left\{ \hat{F}^{(2)}(u) \overline{\hat{F}^{(2)}(t)} \right\} \mathbf{E} \left(\text{sgn} \left\{ \text{Re } \hat{F}^{(1)}(u) \right\} \text{sgn} \left\{ \text{Re } \hat{F}^{(1)}(t) \right\} \right)$$

$$\times \, e^{i 2\pi (u-t) x} \, dudt \ .$$

Using the results of equations (4.4.9), and (4.3.12), we have

$$\text{Var } G^{(2)}(x) = \frac{4\omega\sigma_2^2}{\pi} \int_{-\nu}^{\nu} \int_{-\nu}^{\nu} \text{sinc } 2\omega(u-t) \sin^{-1} r \, (u, t; \omega) \cos 2\pi(u-t) x \ dudt$$

$$= \frac{4\omega\sigma_2^2}{\pi} \int_{-1}^{1}\int_{-1}^{1} \text{sinc}\ \frac{p}{2}(u-t)\sin^{-1} r(u,t;0.25p)\cos 2\pi(u-t)\nu x\ dudt\ . \tag{4.4.20}$$

Again, no analytic expression can be found for $\eta_2 = \sup_x \left\{ \text{Var}\ G^{(2)}(x) \right\}$, in general, and numerical evaluation must be used.

Define β_0, β_1, and β_2 as functions of the space-bandwidth product p by

$$\beta_0(p) = \left[\int_0^1 (1 + \text{sinc}\ pt)^{1/2}\, dt \right]^2 , \tag{4.4.21}$$

$$\beta_1(p) = \sup_x \left\{ \int_{-1}^{1}\int_{-1}^{1} \text{sinc}\ \frac{p}{2}(u-t)\sin^{-1} r(u,t;0.25p)\cos 2\pi(u-t)\nu x\ du\ dt \right\} \tag{4.4.22}$$

and

$$\beta_2(p) = \sup_x \left\{ \int_{-1}^{1}\int_{-1}^{1} \left\{ \text{sinc}\ \frac{p}{2}(u-t)\sin^{-1} r(u,t;0.25p) - \frac{1}{2}(1 + \text{sinc}\ pu)^{1/2} \right.\right.$$

$$\left.\left. \times\ (1 + \text{sinc}\ pt)^{1/2}\left\{1 - (1 - [r(u,t;0.25p)]^2)^{1/2}\right\} \right\} \cos 2\pi(u-t)\nu x\ du\ dt \right\} . \tag{4.4.23}$$

Combining the results of equations (4.2.3), (4.4.17), (4.4.18), (4.4.19), and (4.4.20), and using the defining equations (4.4.21), (4.4.22), and (4.4.23) we obtain the performance coefficient, ρ_b, of the Binary Filter to be

$$\rho_b = \frac{2\beta_0(p)\dfrac{\sigma_1^2}{\sigma_2^2}}{\beta_1(p) + \beta_2(p)\dfrac{\sigma_1^2}{\sigma_2^2}} . \tag{4.4.24}$$

Asymptotic expansions, and approximations of any form are rather hard to come by for the above expression in view of its rather complicated structure. We will hence have recourse to numerical solutions in section 6.

C. The Matched Binary Filter

For this, the third of the three proposed correlator schemes, the pointwise thresholding operation is introduced into both signal pathways in fig. 4.1. Both the input signal and the reference are hence subjected to a pointwise binarisation operation, $T_s = \text{sgn} \circ \text{Re}$, and $T_r = \text{sgn} \circ \text{Re}$. On substituting in equation (4.2.2), we find the correlation output when the input is $\mathrm{F}^{(j)}(x)$ to be given by

$$G^{(j)}(x) = \int_{-\nu}^{\nu} \text{sgn}\left\{\text{Re } \hat{\mathrm{F}}^{(j)}(u)\right\} \text{sgn}\left\{\text{Re } \hat{\mathrm{F}}^{(1)}(u)\right\} e^{i2\pi ux}\, du \ . \qquad (4.4.25)$$

We now obtain the output mean, and variance for the two classes.

<u>CLASS 1</u> : The system output when the sample realisation $\mathrm{F}^{(1)}(x)$ of class C_1 is present at one input port with the reference signal $\mathrm{F}^{(1)}(x)$ being the second input is given from (4.4.25) by

$$\begin{aligned} G^{(1)}(x) &= \int_{-\nu}^{\nu} \text{sgn}\left\{\text{Re } \hat{\mathrm{F}}^{(1)}(u)\right\} \text{sgn}\left\{\text{Re } \hat{\mathrm{F}}^{(1)}(u)\right\} e^{i2\pi ux}\, du \\ &= \int_{-\nu}^{\nu} e^{i2\pi ux}\, du \\ &= 2\nu \, \text{sinc } 2\nu x \ . \end{aligned}$$

The output $G^{(1)}(x)$ is purely deterministic, and has no random component. Hence $\mathrm{E}\left\{G^{(1)}(x)\right\} = G^{(1)}(x)$, and $\text{Var}\left\{G^{(1)}(x)\right\} = 0$. Consequently,

$$\mu_1 = \sup_x \left(\mathrm{E}\left\{G^{(1)}(x)\right\}\right) = 2\nu \ , \qquad (4.4.26)$$

and

$$\eta_1 = \sup_x \left\{\text{Var } G^{(1)}(x)\right\} = 0 \ . \qquad (4.4.27)$$

<u>CLASS 2</u> : With $\mathrm{F}^{(2)}(x)$ as the input signal, equation (4.4.25) yields the system output

$G^{(2)}(x)$ to be

$$G^{(2)}(x) = \int_{-\nu}^{\nu} \text{sgn}\left\{\text{Re}\ \hat{\text{F}}^{(2)}(u)\right\} \text{sgn}\left\{\text{Re}\ \hat{\text{F}}^{(1)}(u)\right\} e^{i2\pi ux}\ du\ .$$

Hence from (4.3.10), and independence of $\text{F}^{(1)}(x)$, and $\text{F}^{(2)}(x)$ we have

$$\mathbf{E}\left\{G^{(2)}(x)\right\} = \int_{-\nu}^{\nu} \mathbf{E}\left\{\text{sgn}\left\{\text{Re}\ \hat{\text{F}}^{(2)}(u)\right\}\right\} \mathbf{E}\left\{\text{sgn}\left\{\text{Re}\ \hat{\text{F}}^{(1)}(u)\right\}\right\} e^{i2\pi ux}\ du\ = 0\ .$$

$$\mu_2 = \sup_x \left(\mathbf{E}\left\{G^{(2)}(x)\right\}\right) = 0\ . \tag{4.4.28}$$

To estimate the variance of $G^{(2)}(x)$, we see that as a consequence of $G^{(2)}(x)$ being zero mean

$$\begin{aligned}\text{Var}\left\{G^{(2)}(x)\right\} &= \mathbf{E}\left\{\ |\ G^{(2)}(x)\ |\ ^2\right\} \\ &= \int_{-\nu}^{\nu}\int_{-\nu}^{\nu} \mathbf{E}\left\{\text{sgn}\left\{\text{Re}\ \hat{\text{F}}^{(2)}(u)\right\} \text{sgn}\left\{\text{Re}\ \hat{\text{F}}^{(2)}(1)\right\}\right\} \\ &\quad \times \mathbf{E}\left\{\text{sgn}\left\{\text{Re}\ \hat{\text{F}}^{(1)}(u)\right\} \text{sgn}\left\{\text{Re}\ \hat{\text{F}}^{(1)}(1)\right\}\right\} e^{i2\pi(u-t)x}\ dudt\ .\end{aligned}$$

Referring to equation (4.3.12) we have

$$\text{Var}\left\{G^{(2)}(x)\right\} = \frac{4}{\pi^2}\int_{-\nu}^{\nu}\int_{-\nu}^{\nu} [\sin^{-1} r\,(u\,,t\,;\omega)]^2 \cos 2\pi(u-t)x\ \ dudt\ . \tag{4.4.29}$$

Hence

$$\begin{aligned}\eta_2 &= \sup_x \left\{\text{Var}\ G^{(2)}(x)\right\} = \text{Var}\left\{G_2(0)\right\} \\ &= \frac{4}{\pi^2}\int_{-\nu}^{\nu}\int_{-\nu}^{\nu} [\sin^{-1} r\,(u\,,t\,;\omega)]^2\ du\ \ dt\end{aligned}$$

$$= \frac{16\nu^2}{\pi^2} \int_0^1 \int_0^1 [\sin^{-1} r\,(u\,,t\,;0.25p\,)]^2\, du\;\, dt\;, \tag{4.4.30}$$

as $r\,(\nu u\,,\nu t\,;\omega) = r\,(u\,,t\,;0.25p\,)$, and $r\,(u\,,t\,;0.25p\,) = r\,(u\,,-t\,;0.25p\,) = r\,(-u\,,t\,;0.25p\,) = r\,(-u\,,-t\,;0.25p\,)$ from the defining equation (4.3.9) of the correlation coefficient r.

Define γ as a function of the space-bandwidth product p by

$$\gamma(p\,) = \int_0^1 \int_0^1 [\sin^{-1} r\,(u\,,t\,;0.25p\,)]^2\, dudt\;. \tag{4.4.31}$$

From equations (4.4.26), (4.4.27), (4.4.28), and (4.4.30), and the defining equation (4.4.31) we get the performance coefficient ρ_{mb} for the Binary Matched Filter to be

$$\rho_{mb} = \frac{\pi^2}{4\gamma(p\,)}\;. \tag{4.4.32}$$

The most remarkable feature of the performance coefficient ρ_{mb} is that it depends only on the system space-bandwidth product p, and is independent of the class variances σ_1^2 and σ_2^2.

5. CLASSIFICATION IN ADDITIVE NOISE

In practice, the issue of *system robustness* in the face of signal degradations, and noise becomes important. We illustrate how noisy signals result in performance attrition in the three correlator systems considered.

We consider the case where the input signal $F(x)$ is contaminated by an additive noise term $N(x)$. (We assume that the reference signal $H(x)$ being known *a priori* can hence be represented in a reasonably accurate, and noise-free manner). We take $N(x)$ to be an independent noise process which is additive and white with

$$\mathrm{E}\left\{N(x\,)\right\} = 0\;,$$

and

$$\mathbf{E}\left\{N(x)N(y)\right\} = \sigma_n^2\delta(x-y)\ .$$

The input signal term is then $\mathrm{F}^{(j)}(x) + N(x)$, and the reference signal term (matched to class C_1) is $\mathrm{F}^{(1)}(x)$.

A. The Matched Filter

We first demonstrate the rate of decay of system performace as the input noise level increases for the case of the Matched Filter. We denote by $G_n^{(j)}(x)$ the (noisy) correlation output of the system when the input signal is a noisy realisation of class C_j, viz., $\mathrm{F}^{(j)}(x) + N(x)$. The second system input–the reference signal $H(x)$–is as before matched to the sample realisation $\mathrm{F}^{(1)}(x)$ of class C_1. For simplicity of notation we denote by Ξ_m the system operator for the Matched Filter; Ξ_m maps the two inputs–the signal and the reference–to the output correlation function. For the noise-free input case, we have from equation (4.4.1) that

$$G^{(j)}(x) = \Xi_m\left\{\mathrm{F}^{(j)}(x), \mathrm{F}^{(1)}(x)\right\} \qquad (4.5.1)$$

$$= \int_{-\nu}^{\nu} \hat{\mathrm{F}}^{(j)}(u)\, \overline{\hat{\mathrm{F}}^{(1)}(u)}\ e^{\,i\,2\pi ux}\ du\ .$$

Let $G^{(n)}(x)$ be the system output when the input signal is a pure noise term $N(x)$. Then

$$G^{(n)}(x) = \Xi_m\ \left\{N(x), \mathrm{F}^{(1)}(x)\right\}\ . \qquad (4.5.2)$$

Note that $G^{(n)}(x)$ has the same form as the noise-free system output when class C_2 is present at the input; simply replacing F_2 by N in equation (4.4.6) yields $G^{(n)}(x)$. The analysis for class C_2 in section 4(A) hence holds in entirety for $G^{(n)}(x)$, as $N(x)$ and $\mathrm{F}^{(2)}(x)$ have similar statistics, and are both independent of $\mathrm{F}^{(1)}(x)$. In particular, from equations (4.4.7), and (4.4.10), we have

$$\mathbf{E}\left\{G^{(n)}(x)\right\} = 0\,, \tag{4.5.3}$$

and

$$\mathrm{Var}\left\{G^{(n)}(x)\right\} = 4\omega^2\sigma_1^2\sigma_n^2 \int_{-2\nu}^{2\nu} (2\nu - \mid s \mid)(\mathrm{sinc}\ 2\omega s)^2 \cos 2\pi s x \ ds\ . \tag{4.5.4}$$

Furthermore, $G^{(n)}(x)$ and $G^{(j)}(x)$ are uncorrelated processes, $\mathrm{E}\left\{G^{(n)}(x)G^{(j)}(x)\right\} = 0$. This follows from the mutual independence of $\mathrm{F}^{(j)}(x)$ and $N(x)$, as can be verified by straightforward substitution in equations (4.5.1), and (4.5.2).

Now, the noisy output of interest when the input is corrupted by an additive noise component is given by

$$G_n^{(j)}(x) = \Xi_m \ \left\{\mathrm{F}^{(j)}(x) + N(x), \mathrm{F}^{(1)}(x)\right\} \tag{4.5.5}$$

$$= \Xi_m \ \left\{\mathrm{F}^{(j)}(x), \mathrm{F}^{(1)}(x)\right\} + \Xi_m \ \left\{N(x), \mathrm{F}^{(1)}(x)\right\}$$

as the system operator Ξ_m is linear in its first argument–the input signal term. Hence

$$G_n^{(j)}(x) = G^{(j)}(x) + G^{(n)}(x)\ . \tag{4.5.6}$$

It follows then that

$$\mathbf{E}\left\{G_n^{(j)}(x)\right\} = \mathbf{E}\left\{G^{(j)}(x)\right\} + \mathbf{E}\left\{G^{(n)}(x)\right\} = \mathbf{E}\left\{G^{(j)}(x)\right\}\,,$$

as $G^{(n)}(x)$ has zero-mean from equation (4.5.3). Hence the noisy correlation peak has mean value

$$\mu_{j,n} = \sup_x \left\{ \mid \mathrm{E}\left\{G_n^{(j)}(x)\right\} \mid \right\} = \sup_x \left\{ \mid \mathrm{E}\left\{G^{(j)}(x)\right\} \mid \right\} = \mu_j\ ,$$

with μ_j given by equations (4.4.3) and (4.4.8). Furthermore, as $G^{(j)}(x)$ and $G^{(n)}(x)$ are uncorrelated, we have

$$\text{Var } G_n^{(j)}(x) = \text{Var } G^{(j)}(x) + \text{Var } G^{(n)}(x) .$$

Combining equations (4.4.4), (4.4.10), and (4.5.4), we have the peak variances $\eta_{j,n}$ for each of the two classes given by

$$\eta_{1,n} = \sup_x \left\{ \text{Var } G_n^{(1)}(x) \right\}$$

$$= \text{Var } G_1(0) + \text{Var } G_n(0)$$

$$= 2p^2\sigma_1^2(2\sigma_1^2 + \sigma_n^2) \int_0^1 (1 - t)(\text{sinc } pt)^2 dt ,$$

and

$$\eta_{2,n} = 2p^2\sigma_1^2(\sigma_2^2 + \sigma_n^2) \int_0^1 (1 - t)(\text{sinc } pt)^2 dt .$$

The performance coefficient $\rho_{m,n}$ for the Matched Filter when input noise is present is hence given by

$$\rho_{m,n} = \frac{(\mu_{1,n} - \mu_{2,n})^2}{\eta_{1,n} + \eta_{2,n}}$$

$$= \frac{\left(\dfrac{\sigma_1^2}{\sigma_2^2 + 2\sigma_n^2}\right)}{2\alpha(p)\left(1 + 2\dfrac{\sigma_1^2}{\sigma_2^2 + 2\sigma_n^2}\right)} , \tag{4.5.7}$$

where, as in equation (4.4.12),

$$\alpha(p) = \int_0^1 (1 - t)(\text{sinc } pt)^2 \, dt ,$$

is solely a function of the system space-bandwidth product $p = 4\omega\nu$. A comparison of equations (4.4.13) and (4.5.7) shows that the presence of additive input noise is equivalent to an additive increase in the variance (or spread) of class C_2 by exactly twice the spread of the noise.

B. The Binary Filter

The case of the Binary Filter can be tackled in an exactly analogous manner. Defining Ξ_b to be the system operator for the Binary Filter, we have the noisy correlation output given as in equation (4.5.5) by

$$G_n^{(j)}(x) = \Xi_b \ \{\mathrm{F}^{(j)}(x) + N(x), \mathrm{F}^{(1)}(x)\} \ .$$

Again, Ξ_b is linear in its first argument (the input signal term) so that $G_n^{(j)}(x)$ is given as in (4.5.6) to be the sum of the noise-free system output $G^{(j)}(x)$, and the system output $G^{(n)}(x)$ when the input signal is a pure noise term. Tracing through the same analysis yields the performance coefficient $\rho_{b,n}$ for the Binary Filter when the input is degraded by an additive noise.

In general, however, it turns out that the form of $\rho_{b,n}$ is not conducive to a convenient representation as in equation (4.4.24) for the noise-free case; specifically, in equation (4.4.23), the functional $\beta_2(p)$ has to be replaced by a more complicated supremum taken over the sum of two integrals, the coefficient of one being σ_1^2, and of the other being σ_n^2. (The supremum is now a function of not only the space-bandwidth product p, but also of the signal and noise variances.) Using the fact that $\sup\{A + B\} \leq \sup\{A\} + \sup\{B\}$, we can arrive at the following convenient lower bound estimate for $\rho_{b,n}$ for the sake of comparison:

$$\rho_{b,n} \geq \frac{2\beta_0(p)\dfrac{\sigma_1^2}{\sigma_2^2 + 2\sigma_n^2}}{\beta_1(p) + \beta_2(p)\dfrac{\sigma_1^2}{\sigma_2^2 + 2\sigma_n^2}} \tag{4.5.8}$$

with the functionals $\beta_0(p)$, $\beta_1(p)$, and $\beta_2(p)$ given by equations (4.4.21), (4.4.22), and (4.4.23).

On comparing (4.4.24), and (4.5.8) we see that the effect of additive noise is to create a larger effective spread for class C_2 just as in the case of the Matched Filter. In both cases, the noise effectively reduces the ability of the system to pick out class C_1 by increasing side-lobe energy, and at the same time increasing the correlation spread of class C_2 so that the probability of erroneous identification of a spurious peak (side-lobes from either class) with the correlation peak of class C_1 is increased.

C. The Matched Binary Filter

The situation becomes more involved for the case of the Matched Binary Filter. Here the system operator Ξ_{mb} is linear in neither input. As a consequence, the evaluation of the performance coefficient requires the estimation of rather complicated expressions for the fourth order moments of hard-limited random processes which are mutually correlated. We can give a simple heuristic argument, however, to demonstrate that additive noise is considerably more inimical to this correlation scheme than to the previous two schemes we've discussed.

For the purpose of analysis, it is expedient to think of the Matched Binary Filter as a two-stage decison making system, with partial decisions being made initially at each pont in the Fourier domain, with a final classification decision being made based upon all the previous partial decisions. Assume $\mathrm{F}_j + N$ is the noisy input signal to the Matched Binary Filter. The thresholding operations on the signal and the reference, followed by the pointwise multiplication in essence realise a hard decision at each point in the Fourier domain:

$$T_s\,(\hat{\mathrm{F}}^{(j)}(u) + \nu)\; \overline{T_r\,(\hat{\mathrm{F}}^{(1)}(u))} = \begin{cases} 1 & \text{if } \mathrm{sgn}\,\left\{\mathrm{Re}\ (\hat{\mathrm{F}}^{(j)}(u) + \hat{N}(u))\right\} = \mathrm{sgn}\,\left\{\mathrm{Re}\ \hat{\mathrm{F}}^{(1)}(u)\right\} \\ -1 & \text{if } \mathrm{sgn}\,\left\{\mathrm{Re}\ (\hat{\mathrm{F}}^{(j)}(u) + \hat{N}(u))\right\} \neq \mathrm{sgn}\,\left\{\mathrm{Re}\ \hat{\mathrm{F}}^{(1)}(u)\right\} \end{cases}.$$

The final classification is simply a threshold decision based upon the weighted integral (Fourier transform) of the individual partial decisions. The reliability of the final classification decision is clearly a function of the reliability of the earlier partial decisions. If $\mathbf{P}\ [\mathrm{sgn}\ \left\{\mathrm{Re}\ (\hat{\mathrm{F}}^{(1)}(u) + \hat{N}(u))\right\} = \mathrm{sgn}\ \left\{\mathrm{Re}\ \hat{\mathrm{F}}^{(1)}(u)\right\}] \approx 1$ for each point u in the Fourier domain, (or if $\mathbf{P}\ [\mathrm{sgn}\ \left\{\mathrm{Re}\,(\hat{F}(u)+\nu)\right\} = -\mathrm{sgn}\ \left\{\mathrm{Re}\ \hat{\mathrm{F}}^{(1)}(u)\right\}] \approx 1$ for each u, in which case we just have the negative of the reference), then we have maximally reliable partial decisions for class C_1 inputs, which result in reliable classification for class C_1 as correlation peaks will be well above threshold; if, on the other hand, $\mathbf{P}\,[\text{sgn}\,\{\text{Re}\ (\hat{\mathrm{F}}^{(1)}(u)+\hat{N}(u))\}{=}\text{sgn}\,\{\text{Re}\ \hat{\mathrm{F}}^{(1)}(u)\}] \approx 1/2$ for each u, then we have maximally unreliable partial decisions for class C_1 inputs, which result in a high probability of overall misclassification as correlation peaks will be low. The reverse situation is true for class C_2–we would like $\mathbf{P}\,[\text{sgn}\,\{\text{Re}\ (\hat{\mathrm{F}}^{(2)}(u)+\hat{N}(u))\}{=}\text{sgn}\,\{\text{Re}\ \hat{\mathrm{F}}^{(1)}(u)\}] \approx 1/2$ for each u so that the resultant correlation peak is low.

By virtue of the two signals and the noise being mutually independent, we have $\mathbf{P}\,[\text{sgn}\,\{\text{Re}\ (\hat{\mathrm{F}}^{(2)}(u)+\hat{N}(u))\}{=}\text{sgn}\,\{\text{Re}\ \hat{\mathrm{F}}^{(1)}(u)\}] =$ $\mathbf{P}\,[\text{sgn}\,\{\text{Re}\ \hat{\mathrm{F}}^{(2)}(u)\}{=}\text{sgn}\,\{\text{Re}\ \hat{\mathrm{F}}^{(1)}(u)\}] = 1/2$ for each u. Hence the overall reliability of the partial decisions is determined entirely by what happens to a noisy class C_1 input. For the sake of simplicity we make the following definitions. Let $p = \mathbf{P}\,[\text{sgn}\,\{\text{Re}\ (\hat{\mathrm{F}}^{(1)}(u)+\hat{N}(u))\}{=}\text{sgn}\,\{\text{Re}\ \hat{\mathrm{F}}^{(1)}(u)\}]$, $\sigma_{\mathrm{F}}^2 = \text{Var}\,\{\text{Re}\ \hat{\mathrm{F}}^{(1)}(u)\}$, and $\sigma_N^2 = \text{Var}\,\{\text{Re}\,\hat{N}(u)\}$, and define the Gaussian density and distribution functions

$$\phi(x) = \frac{1}{\sqrt{2\pi}}\, e^{-x^2/2} ,$$

and

$$\Phi(x) = \int_{-\infty}^{x} \phi(y)\, dy \ .$$

Then

$$p = \int_0^{\infty} \mathbf{P}\,\{\text{Re}\ (\hat{\mathrm{F}}^{(1)}(u)+\hat{N}(u)) \geq 0 \ |\ \text{Re}\ \hat{\mathrm{F}}^{(1)}(u) = f\ \}\ \frac{1}{\sigma_{\mathrm{F}}}\ \phi(\frac{f}{\sigma_{\mathrm{F}}})\ df$$

$$+ \int_{-\infty}^{0} \mathbf{P}\,\{\text{Re}\ (\hat{\mathrm{F}}^{(1)}(u)+\hat{N}(u)) < 0 \ |\ \text{Re}\ \hat{\mathrm{F}}^{(1)}(u) = f\ \}\ \frac{1}{\sigma_{\mathrm{F}}}\ \phi(\frac{f}{\sigma_{\mathrm{F}}})\ df$$

$$= \frac{1}{\sigma_{\mathrm{F}}} \int_0^{\infty} \Phi(\frac{f}{\sigma_N})\ \phi(\frac{f}{\sigma_{\mathrm{F}}})\ df \ + \frac{1}{\sigma_{\mathrm{F}}} \int_{-\infty}^{0} \Phi(-\ \frac{f}{\sigma_N})\ \phi(\frac{f}{\sigma_{\mathrm{F}}})\ df$$

$$= 2 \int_0^\infty \Phi(\frac{f\ \sigma_F}{\sigma_N})\ \phi(f\ )\ df\ . \qquad (4.5.9)$$

For $\sigma_N \ll \sigma_F$, we clearly have $p \approx 1$, in line with expectations as for the low noise case we expect highly reliable decisions. If $\sigma_N \gg \sigma_F$, on the other hand, equation (4.5.9) yields $p \approx 1/2$; i.e., when the signal is swamped by noise we obtain unreliable partial decisions, as expected. Now consider the intermediate case with equal signal and noise power, $\sigma_N = \sigma_F = \sigma$. From equation (4.5.9) we have

$$p = 2 \int_0^\infty \Phi(f\ )\ \phi(f\ )\ df$$

$$= 2 \int_{1/2}^{1} \Phi\ d\,\Phi$$

$$= 2\ \frac{\Phi^2}{2} \Big|_{1/2}^{1} = \frac{1}{2}\ .$$

Thus, even when the noise level is not very high, the partial decisions are very unreliable, leading to poor performance. Note that both the Matched Filter and the Binary Filter still perform very well for noise levels of the order of the signal; in both cases, the noise tends to average out so that the processing gain of the system is sufficient to pull out a peak. Note from equations (4.5.7) and (4.5.8) that for $\sigma_1 = \sigma_2 = \sigma_n$, both $\rho_{m,n}$ and $\rho_{b,n}$ decrease by at most a factor of 1/3 from the noise-free case.

(The behaviour of p as a function of the noise level adds an interesting footnote: for fixed σ_F, p is not a monotonically decreasing function of σ_N, as might perhaps be expected intuitively. In fact, worst case performance for the system occurs when $\sigma_N = \sigma_F$ (at which point $p = 1/2$). As σ_N increases beyond σ_F, system performance actually improves slightly before it drops again when σ_N becomes large (at which point p approaches 1/2 again). However, such improvement is marginal at best, and system performance is poor whenever the noise level exceeds that of the signal. Note that this sort of fine shade of performance distinction is not mirrored in our performance measure ρ, which yields overall trends, but is insensitive to small perturbations in performance.

The Matched Binary Filter hence does not perform very well for high noise levels at the input. With this in mind, we develop an upper bound for the performance coefficient which is reasonably sharp for low noise levels at the input. The output of the system when the input signal is corrupted by additive noise is given by analogy with equation (4.4.25) by

$$G_n^{(j)}(x) = \int_{-\nu}^{\nu} \text{sgn}\left\{\text{Re}\ (\hat{\text{F}}^{(j)}(u) + \hat{N}(u))\right\} \text{sgn}\left\{\text{Re}\ \hat{\text{F}}^{(1)}(u)\right\} e^{i2\pi ux}\, du \qquad (4.5.10)$$

where $\hat{N}(u)$ is given by

$$\hat{N}(u) = \int_{-\omega}^{\omega} N(x)\, e^{i2\pi ux}\, dx \ .$$

<u>CLASS 1</u> : When the input signal is a noisy version of a sample realisation of class C_1, the output is given by

$$G_n^{(1)}(x) = \int_{-\nu}^{\nu} \text{sgn}\left\{\text{Re}\ (\hat{\text{F}}^{(1)}(u) + \hat{N}(u))\right\} \text{sgn}\left\{\text{Re}\ \hat{\text{F}}^{(1)}(u)\right\} e^{i2\pi ux}\, du \ .$$

$N(x)$ and $\text{F}^{(1)}(x)$ are mutually independent zero-mean random processes, and hence Re $\{\hat{N}(u)\}$ and Re $\hat{\text{F}}^{(1)}(u)$ are mutually independent normal processes with zero-mean *vide* the *Central Limit Theorem*, and the linearity of the expectation operator. It then follows trivially that Re $\hat{\text{F}}^{(1)}(u)$ and Re $(\hat{\text{F}}^{(1)}(u) + \hat{N}(u))$ are jointly normal as they are obtained by the linear transformation

$$\begin{bmatrix} \text{Re}\ \hat{\text{F}}^{(1)}(u) \\ \text{Re}\ (\hat{\text{F}}^{(1)}(u) + \hat{N}(u)) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \text{Re}\ \hat{\text{F}}^{(1)}(u) \\ \text{Re}\ \hat{N}(u) \end{bmatrix}$$

from the jointly normal processes Re $\hat{N}(u)$ and Re $\hat{\text{F}}^{(1)}(u)$. Proceeding in a manner similar to the derivation of equations (4.3.5) and (4.3.6), we obtain

$$\mathbf{E}\left\{\mathrm{Re}\ (\hat{\mathrm{F}}^{(j)}(u)+\hat{N}(u))\ \mathrm{Re}\ (\hat{\mathrm{F}}^{(j)}(1)+\hat{N}(t))\right\}$$

$$=\omega(\sigma_j^2+\sigma_n^2)(\mathrm{sinc}\ 2\omega(u-t)+\mathrm{sinc}\ 2\omega(u+t)) \qquad (4.5.11)$$

and

$$\mathbf{E}\left\{\mathrm{Re}\ (\hat{\mathrm{F}}^{(j)}(u)+\hat{N}(u))\ \mathrm{Re}\ \hat{\mathrm{F}}^{(1)}(1)\right\}=\omega\sigma_1^2(\mathrm{sinc}\ 2\omega(u-t)+\mathrm{sinc}\ 2\omega(u+t))\ .$$

We then have from equation (4.3.12) that,

$$\mathbf{E}\left\{G_n^{(1)}(x)\right\}$$

$$=\int_{-\nu}^{\nu}\mathbf{E}\left[\mathrm{sgn}\left\{\mathrm{Re}\ (\hat{\mathrm{F}}^{(1)}(u)+\hat{N}(u))\right\}\mathrm{sgn}\left\{\mathrm{Re}\ \hat{\mathrm{F}}^{(1)}(u)\right\}\right]e^{i2\pi ux}\,du$$

$$=\frac{2}{\pi}\int_{-\nu}^{\nu}\sin^{-1}\left\{\frac{\omega\sigma_1^2(1+\mathrm{sinc}\ 4\omega u)}{[\omega\sigma_1^2(1+\mathrm{sinc}\ 4\omega u)]^{1/2}[\omega(\sigma_1^2+\sigma_n^2)(1+\mathrm{sinc}\ 4\omega u)]^{1/2}}\right\}e^{i2\pi ux}\,du$$

$$=\frac{4\nu}{\pi}\sin^{-1}\left\{\left(\frac{\sigma_1^2}{\sigma_1^2+\sigma_n^2}\right)^{1/2}\right\}\mathrm{sinc}\ 2\nu x\ .$$

Hence

$$\mu_{1,n}=\sup_x\left\{\ |\ \mathbf{E}\left\{G_n^{(1)}(x)\right\}\ |\ \right\}=\frac{4\nu}{\pi}\sin^{-1}\left\{\left(\frac{\sigma_1^2}{\sigma_1^2+\sigma_n^2}\right)^{1/2}\right\}. \qquad (4.5.12)$$

Rather complicated fourth-order moments are required for the estimation of $\eta_{1,n}$-the maximum variance of the noisy output correlation. We avoid these complications with the intention of obtaining a manageable upper bound on performance. (An *ad hoc* reason for neglecting the contribution of the variance of the class C_1 output is that $\eta_1\equiv 0$ for the noise-free case–cf. section 4(c). Hence, at least for the low noise case the error that accrues in neglecting $\eta_{1,n}$ will not be large. When noise levels are high, however, as seen earlier, the performance of the Matched Binary Filter drops rapidly, and the upper bound we derive may not be sharp).

<u>CLASS 2</u> : From equation (4.5.10), the noisy output for class C_2 is given by

$$G_n^{(2)}(x) = \int_{-\nu}^{\nu} \text{sgn}\left\{\text{Re}\ (\hat{\text{F}}^{(2)}(u) + \hat{N}(u))\right\}\ \text{sgn}\left\{\text{Re}\ \hat{\text{F}}^{(1)}(u)\right\}\ e^{i2\pi ux}\ du\ .$$

The zero-mean normal processes Re $(\hat{\text{F}}^{(2)}(u) + \hat{N}(u))$ and Re $\hat{\text{F}}^{(1)}(u)$ are mutually independent so that

$$\mathbf{E}\left\{G_n^{(2)}(x)\right\} = \mu_{2,n} = 0\ . \tag{4.5.13}$$

To estimate the variance we see from equations (4.3.12), and (4.5.11) that

$$\text{Var}\ G_n^{(2)}(x) = \int_{-\nu}^{\nu}\int_{-\nu}^{\nu} \text{E}\left\{\text{sgn}\left\{\text{Re}\ (\hat{\text{F}}^{(2)}(u) + \hat{N}(u))\right\}\ \text{sgn}\left\{\text{Re}\ (\hat{\text{F}}^{(2)}(1) + \hat{N}(t))\right\}\right\}$$

$$\times\ \mathbf{E}\left\{\text{sgn}\left\{\text{Re}\ \hat{\text{F}}^{(1)}(u)\right\}\ \text{sgn}\left\{\text{Re}\ \hat{\text{F}}^{(1)}(1)\right\}\right\}\ e^{i2\pi(u-t)x}\ dudt$$

$$= \frac{4}{\pi^2}\int_{-\nu}^{\nu}\int_{-\nu}^{\nu} [\sin^{-1} r\,(u,t;\omega)]^2 \cos 2\pi(u-t)x\ dudt\ ,$$

where $r\,(u,t;\omega)$ is the correlation coefficient defined in equation (4.3.9). Hence

$$\eta_{2,n} = \sup_x \left\{\text{Var}\ G_n^{(2)}(x)\right\}$$

$$= \frac{16\nu^2}{\pi^2}\int_0^1\int_0^1 [\sin^{-1} r\,(u,t;0.25p)]^2\ dudt\ . \tag{4.5.14}$$

Using equations (4.5.12), (4.5.13), and (4.5.14) we have the noisy performance coefficient $\rho_{mb,n}$ bounded by

$$\rho_{mb,n} \le \frac{(\mu_{1,n} - \mu_{2,n})^2}{\eta_{2,n}}$$

$$= \frac{1}{\gamma(p)} \left[\sin^{-1} \frac{\sigma_1}{\sqrt{\sigma_1^2 + \sigma_n^2}} \right]^2 , \qquad (4.5.15)$$

where

$$\gamma(p) = \int_0^1 \int_0^1 [\sin^{-1} r(u,t;0.25p)]^2 \, dudt$$

(as in equation (4.4.31)) is a function solely of the system space-bandwidth product $p = 4\omega\nu$.

Equation (4.5.15) is readily reduced to a simpler form when either signal or noise dominates. When the signal term dwarfs the noise term, $\sigma_1^2 >> \sigma_n^2$, we obtain $\rho_{mb,n} \approx \frac{\pi^2}{4\gamma(p)}$ in accordance with equation (4.4.32) for the noise-free case. When the noise dominates the signal term however, $\sigma_n^2 >> \sigma_1^2$, we have, using the small angle approximation, $\sin^{-1}x \approx x$ for small x, that $\rho_{mb,n} \approx \frac{\sigma_1^2}{\gamma(p)\sigma_n^2}$.

Note that for $\sigma_1 = \sigma_2$, we have $\rho_{mb,n} \leq \frac{\pi^2}{16\gamma(p)}$, so that the drop in performance is at least a factor of 1/4 from the noise-free system performance (equation (4.4.32)). This is to be compared with the performance drop of by at most a factor of 1/3 for the Matched Filter, and the Binary Filter.

6. NUMERICAL SOLUTIONS AND DISCUSSION

We return now to a consideration of the relative performance of the three systems under advisement. Let σ^2 represent either $\frac{\sigma_1^2}{\sigma_2^2}$ for the noise-free case, or $\frac{\sigma_1^2}{\sigma_2^2 + \sigma_n^2}$ for the noisy case. We will refer to σ^2 as the *class spread ratio*; in essence σ^2 is a statistical measure of the relative strengths of "signal" (class C_1) and "noise" (class C_2, and additive noise) at the input of the correlational system. Recapitulating the expressions for the performance coefficients derived in section 4 for easy reference, we have (equations (4.4.13), (4.4.24), (4.5.7), (4.5.8)),

$$\rho_m = \frac{\sigma^2}{2\alpha(p) + 4\alpha(p)\sigma^2} ,$$

$$\rho_b = \frac{2\beta_0(p)\sigma^2}{\beta_1(p) + \beta_2(p)\sigma^2} ,$$

where the functionals $\alpha(p)$, $\beta_0(p)$, $\beta_1(p)$, and $\beta_2(p)$ are defined in equations (4.4.12), (4.4.21), (4.4.22), and (4.4.23), respectively; and for the noise-free case, the performance coefficient of the Matched Binary Filter is given by equation (4.4.32)

$$\rho_{mb} = \frac{\pi^2}{4\gamma(p)} ,$$

with $\gamma(p)$ defined by equation (4.4.31). (Recall that the performance of the Matched Binary Filter deteriorates very rapidly in the presence of additive noise. Hence, for comparisons of system performance in noise, we consider just the Matched Filter and the Binary Filter.)

Numerical solutions of the performance of the three systems are depicted in figs. 4.4, 4.5, and 4.6. Figure 4.4 depicts the performance coefficient, ρ_{mb} , of the Matched Binary Filter plotted as a function of the system space-bandwidth product, p , for the noise-free case. Figs. 4.5 and 4.6, respectively, depict a family of performance curves for the Matched Filter and the Binary Filter. In each figure the performance coefficient ρ is plotted as a function of the class spread ratio σ^2, and the family of curves is generated by varying the space-bandwidth parameter p between 8 and 256. In order to facilitate comparison between the Matched Filter and the Binary Filter, for values of p =8, and p =256, the corresponding performance curves of the two systems are extracted from figs. 4.5 and 4.6, and plotted on the same graph in figs. 4.7 and 4.8

Note that *if the input patterns are noise-free, the Matched Binary Filter yields better performance than both the Matched Filter, and the Binary Filter* for all values of class spread ratio and system space-bandwidth product. As anticipated in the discussion earlier, for noise-free (or low noise) systems, the Matched Binary Filter makes maximally reliable partial decisions at the intermediate decision stage. Thus,

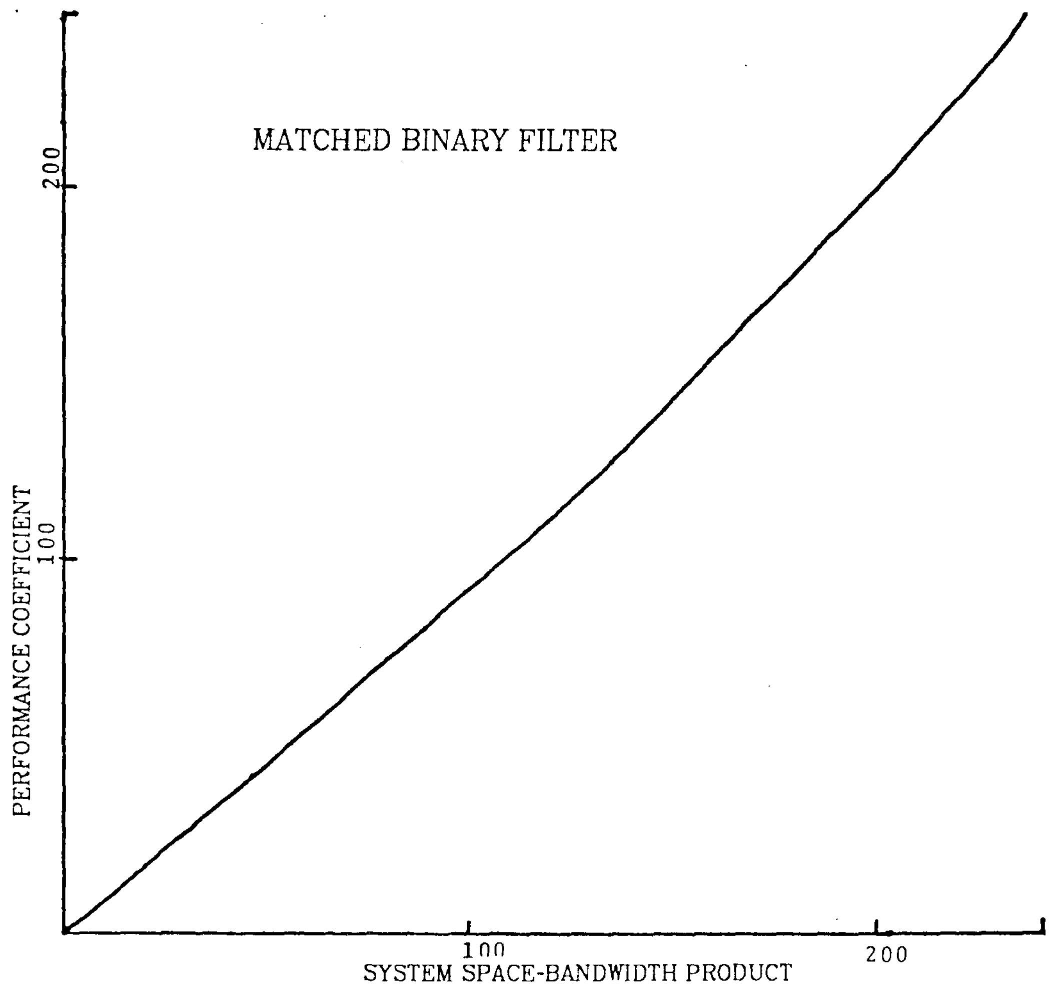

Fig. 4.4. Plot of the performance coefficient, ρ_{mb} , of the Matched Binary Filter vs. the system space-bandwidth product, p , when the input is noise free.

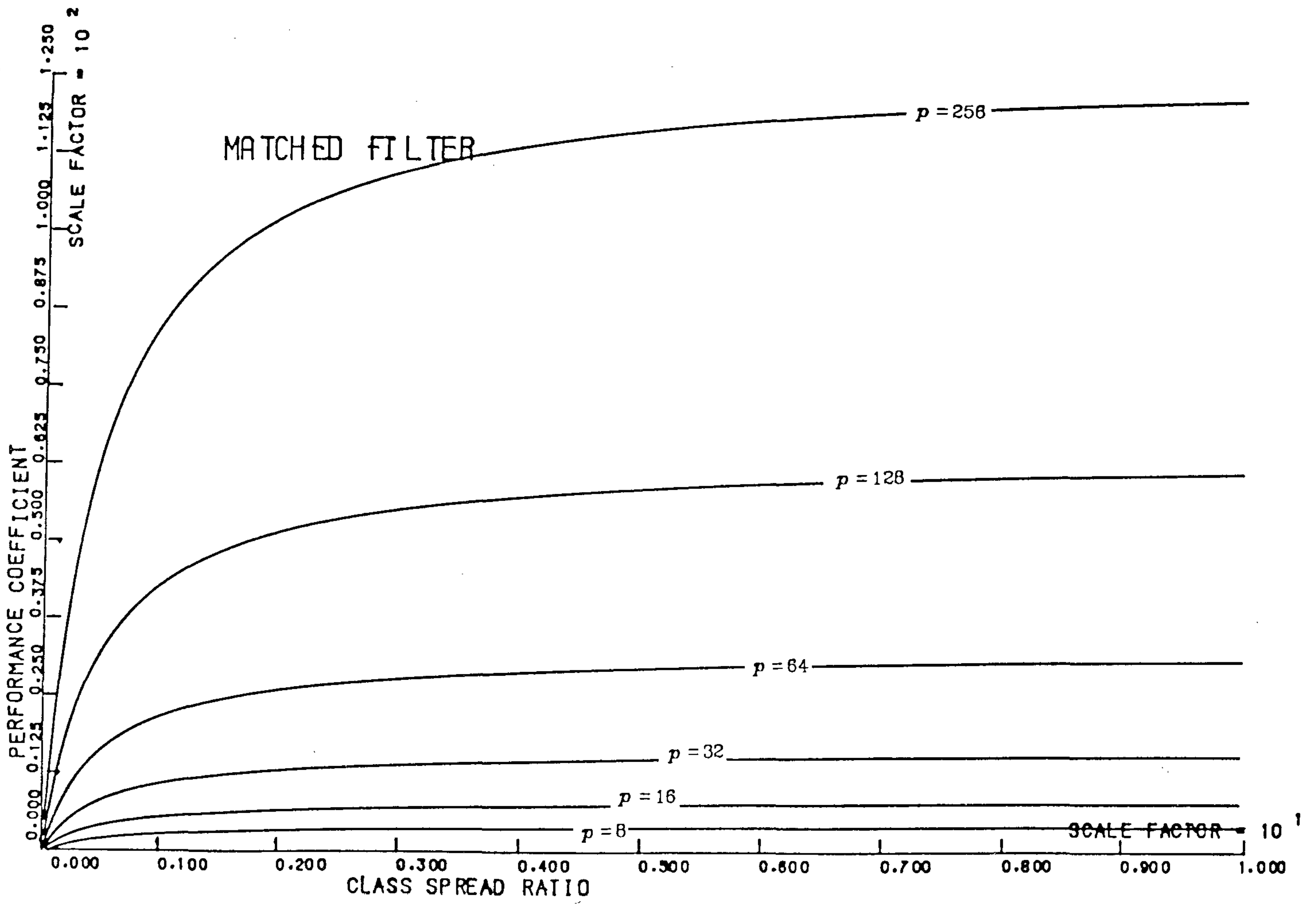

Fig. 4.5. Plot of the performance coefficient, ρ_m , of the Matched Filter vs. the class spread ratio, σ^2, with the system space-bandwidth product, p , as a parameter.

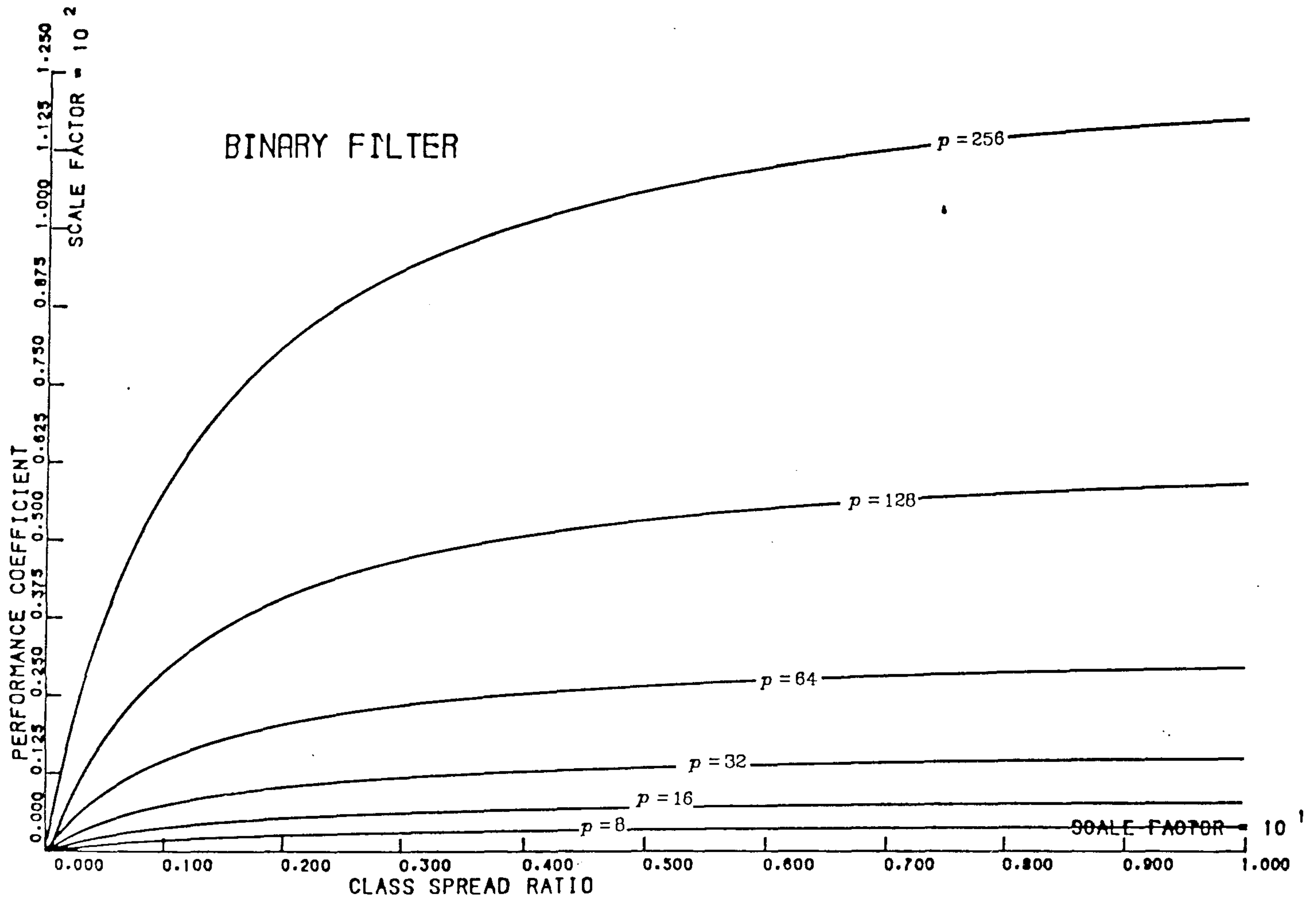

Fig. 4.6. Plot of the performance coefficient, ρ_b , of the Binary Filter vs. the class spread ratio, σ^2, with the system space-bandwidth product, p , as a parameter.

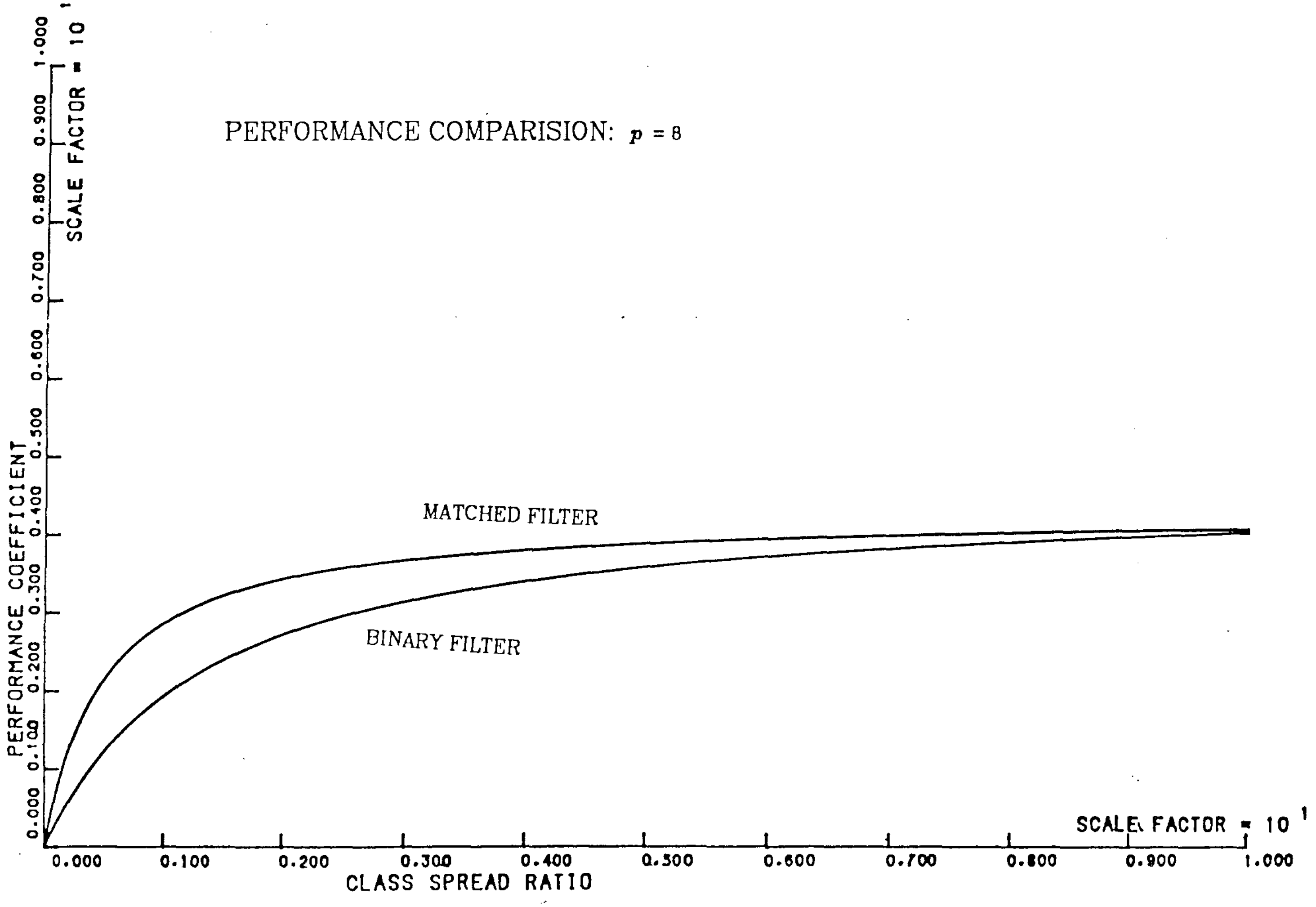

Fig. 4.7. Plots of the relative peformance of the Matched Filter and the Binary Filter for a given space-bandwidth product $p = 8$.

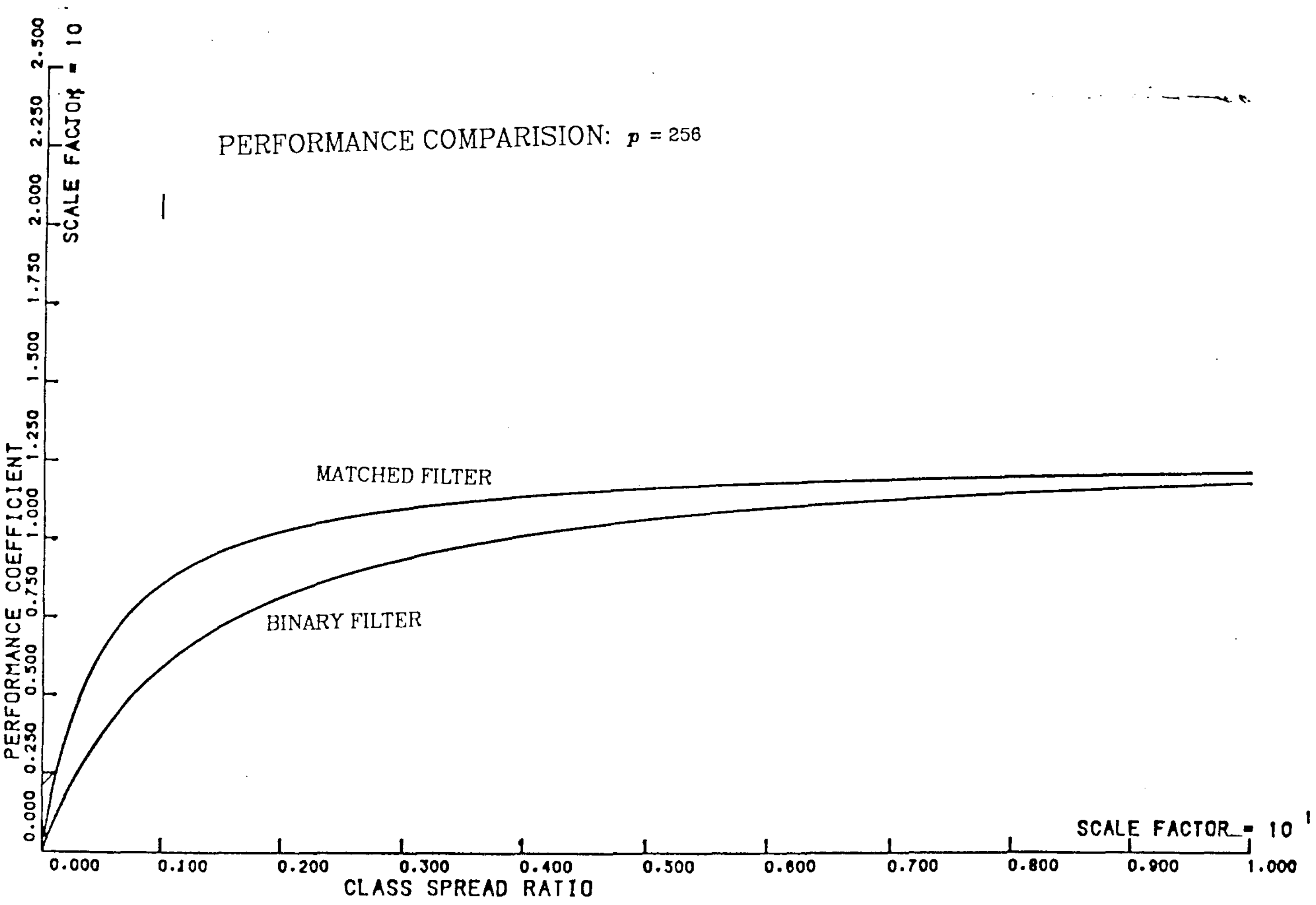

Fig. 4.8. Plots of the relative peformance of the Matched Filter and the Binary Filter for a given space-bandwidth product $p = 256$.

even points in the Fourier domain with small signal amplitudes are assigned the appropriate sign; this procedure essentially assigns the same weight to every point in the signal domain, somewhat akin to an inverse filtering operation. (Again, as in inverse filtering, the procedure is very sensitive to noise.) The overall threshold classification based upon partial decisions over the entire Fourier domain is very reliable, as a consequence of the huge processing gain obtained by making accurate decisions on large segments of low amplitude points. Thus, for systems with very low noise levels, Matched Binary Filters are viable, low-cost alternatives which can yield comparable, or superior performance to the other two schemes.

It can be immediately seen from the figures that, all other things being held constant, *the performance coefficient* ρ *is a monotonically increasing function of the system space-bandwidth product* for all three systems. This is clearly in accordance with our expectations; increasing the system space-bandwidth product is equivalent to increasing the size of the windows W_ω and W_ν in the space and frequency domains (cf., fig. 4.1), and consequently, a greater degree of correlation matching can be obtained.

Now, when the class spread ratio, σ^2, is large, we have a situation where the noise power, σ_n^2, and the class C_2 spread, σ_2^2, are both much smaller than the class C_1 spread, σ_1^2. This can be viewed as essentially saying that patterns of class C_1 can take on values from a much wider set than can patterns of class C_2 and the noise patterns. The probability of significant cross-correlation in any particular case is then quite small, so that we expect good classification performance for large values of σ. This intuitive expectation is echoed in figs. 4.5–4.8, where we see that for the Matched Filter and the Binary Filter, *the performance coefficient* ρ *is a monotonically increasing function of the class spread ratio* σ^2, for each performance curve (corresponding to fixed p).

For the Matched Filter, a close examination of the asymptotes and the slope near the origin of each performance curve reveals that *"large* p*" behaviour holds for relatively small values of the system space-bandwidth product* (as small as $p = 8$). For most cases of interest then, the second of the asymptotic results following equation (4.4.13) holds true; hence, the asymptote of the performance curve for the Matched

Filter is approximately $p/2$, and the graph near the origin is a straight line with positive slope p.

As anticipated earlier, the classification performance of the Binary Filter is always inferior to that of the Matched Filter. However, *the performance of the Binary Filter is surprisingly close to that of the Matched Filter* though ρ_b is always bounded from above by ρ_m. Note that for large values of class spread ratio, the performance curve of the Binary Filter approaches the same asymptote $p/2$ as the Matched Filter, so that their performance is virtually identical. An examination of their relative performance for each p (in the range considered) indicates that when the class spread ratio is unity (i.e., the two classes have the same variance or spread), we have $\rho_b \approx \frac{2}{3}\rho_m$.

These numerical simulations, coupled with the prior success of experimental systems utilising binary filters, tend to bolster the intuitively accepted fact that the phase of the Fourier transform of the signal contains most of the information content in the signal. The significance of the results lies in the demonstration that for classification purposes, most of the information content in the signal can be extracted with filters of low complexity.

Admittedly, the statistical structure of the signals we have considered is particularly well suited for this sort of correlational matching, as discussed in section 4. Our results, however, indicate performance trends. Practical constraints of finite signal energies, and the fact that patterns tend to be correlated over some length will, of course, temper our theoretical results–smaller ρ's will result as a consequence of a decrease in the correlation peak value and increase in the energy of the side-lobes–but these may be expected to follow the general trend of the theoretical results we have obtained, as evidenced in the experimental results in [4] and [5]; specifically, we expect the *relative* performance of the schemes to qualitatively mirror the results of our statistical analysis.

Extensions of the analysis to two-dimensional signals (images) are straightforward. The statistical structure of the signals can also be generalised somewhat in the analysis. For instance, we could in an analogous fashion treat the case where the input signals $F_j(x)$ are correlated Gaussian processes with $E\left\{F_j(x)\,F_j(y)\right\} = r_j(x,y)$, for some correlation function $r_j(x,y)$. The analysis will be similar (though somewhat more complicated) with each sinc function being effectively replaced by a function of the form

$$\int_{-\omega}^{\omega}\int_{-\omega}^{\omega} r_j(x,y)\cos 2\pi u\,(x \pm y)\;dx\;\;dy\ \ .$$

While the success of these schemes is very encouraging, several questions remain: we have demonstrated binary correlator structures based on heuristic algorithms; however, it is not immediately obvious whether we can specify *optimum* binary correlator structures for a given problem. As a specific instance, we can obtain real/complex filter structures which maximally separate pattern classes in that the filter is orthogonal to all unwanted patterns, thus yielding a significant correlation only if the desired pattern is present. It is not clear, however, whether an algorithm can be specified which yields the binary filter which is the best approximation to any such maximally separating filter. Another related area is the determination of optimum binary representations of patterns, so that they can be reconstructed with high fidelity [8].

The surprisingly good performance of the binary systems we have discussed leads us to conjecture that considerable redundancy in information storage in traditional filtration systems can be eliminated for a class of recognition problems. This can be of considerable practical import: low dynamic range requirements for the filter can lead to a decrease in required memory storage over conventional correlators, leading to low implementation costs and low system complexity. Clearly, much is also saved in computation. The availability of binary spatial light modulators motivates the utilisation of such binary techniques in optical correlators; the lack of availability of suitable spatial light-modulators has long been one of the constraints on such analog correlators.

Appendix A

Let $X(u)$ be a normal random process with

$$\mathbf{E}\left\{X(u)\right\} = 0 \,,$$

$$\mathbf{E}\left\{X(u)X(t)\right\} = r_{u,t} \,, \tag{4.A.1}$$

and define the process $Y(u)$ by

$$Y(u) = \text{sgn}\, X(u).$$

$Y(u)$ is easily seen to be zero mean as

$$\mathbf{E}\left\{Y(u)\right\} = P\left\{Y(u)=1\right\} - P\left\{Y(u)=-1\right\}$$

$$= P\left\{X(u)\geq 0\right\} - P\left\{X(u)<0\right\} = 0 \,. \tag{4.A.2}$$

The second moment of $Y(u)$ is given by

$$\mathbf{E}\left\{Y(u)Y(t)\right\} = P\left\{Y(u)Y(t)=1\right\} - P\left\{Y(u)Y(t)=-1\right\}$$

$$= 2P\left\{Y(u)Y(t)=1\right\} - 1 \,.$$

Now by symmetry we have that

$$P\left\{Y(u)=1,Y(t)=1\right\} = P\left\{Y(u)=-1,Y(t)=-1\right\} = 0.5\, P\left\{Y(u)Y(t)=1\right\} \,,$$

so that

$$\mathbf{E}\left\{Y(u)Y(t)\right\} = 4P\left\{Y(u)=1,Y(t)=1\right\} - 1$$

$$= 4P\left\{X(u)\geq 0, X(t)\geq 0\right\} - 1 . \qquad (4.A.3)$$

For simplicity of notation let $p_{u,t} = P\left\{X(u)\geq 0, X(t)\geq 0\right\}$. It hence suffices to determine $p_{u,t}$ to specify the second moment $\mathbf{E}\left\{Y(u)Y(t)\right\}$ of the hardlimited process $Y(u)$. Now, $p_{u,t}$ is simply the probability mass in the first quadrant of the joint normal density function

$$\frac{1}{2\pi \mid \Sigma_{u,t} \mid^{1/2}} e^{-\frac{1}{2}\left\langle \mathbf{x}, \Sigma_{u,t}^{-1}\mathbf{x}\right\rangle} ,$$

where $\Sigma_{u,t}$ is the correlation matrix

$$\Sigma_{u,t} = \begin{bmatrix} r_{u,u} & r_{u,t} \\ r_{u,t} & r_{t,t} \end{bmatrix} \qquad (4.A.4)$$

and $\mathbf{x} = \begin{pmatrix} x_u \\ x_t \end{pmatrix} \in \mathbb{R}^2$. We hence get

$$p_{u,t} = \int_0^{\infty}\int_0^{\infty} \frac{1}{2\pi \mid \Sigma_{u,t} \mid^{1/2}} e^{-\frac{1}{2}\left\langle \mathbf{x}, \Sigma_{u,t}^{-1}\mathbf{x}\right\rangle} d\mathbf{x} .$$

Define $\mathbf{y} = \begin{pmatrix} y_u \\ y_t \end{pmatrix} \in \mathbb{R}^2$ by the linear transformation $\mathbf{y} = \Sigma_{u,t}^{-1/2}\mathbf{x}$. The Jacobian of the transformation is $\mid \Sigma_{u,t} \mid^{1/2}$ so that by transforming the variable of integration we then have

$$p_{u,t} = \iint_{\bar{D}(y_u, y_t)} \frac{1}{2\pi} e^{-\frac{1}{2}\left\langle \mathbf{y}, \mathbf{y}\right\rangle} d\mathbf{y} ,$$

where $\overline{D}(y_u, y_t)$ is the region in the y_u, y_t plane corresponding to the first quadrant in the x_u, x_t plane. Expressing the result in polar coordinates, we have

$$p_{u,t} = \iint\limits_{D(\rho_{u,t},\theta_{u,t})} \frac{1}{2\pi} e^{-\frac{1}{2}\rho_{u,t}^2} \rho_{u,t} \; d\rho_{u,t} \; d\theta_{u,t} \; ,$$

where

$$\rho_{u,t} = \sqrt{y_u^2 + y_t^2} \, ,$$

$$\theta_{u,t} = \tan^{-1} \frac{y_t}{y_u} \, ,$$

and

$$D(\rho_{u,t} \, , \theta_{u,t}) = \overline{D}(y_u \, , y_t) \, .$$

Let $\mathbf{e}_u = \begin{pmatrix}1\\0\end{pmatrix}$, and $\mathbf{e}_t = \begin{pmatrix}0\\1\end{pmatrix}$ be the standard basis vectors for the linear vector space $\mathbb{R}^2$. Set $\mathbf{f}_u = \Sigma_{u,t}^{-1/2}\mathbf{e}_u = \begin{pmatrix} f_{u,u} \\ f_{u,t} \end{pmatrix}$, and $\mathbf{f}_t = \Sigma_{u,t}^{-1/2}\mathbf{e}_t = \begin{pmatrix} f_{t,u} \\ f_{t,t} \end{pmatrix}$. By continuity of the linear transformation $\Sigma_{u,t}^{-1/2}$ then, $D(\rho_{u,t} \, , \theta_{u,t})$ is simply the region between the vectors $\mathbf{f}_u$, and $\mathbf{f}_t$, indicated schematically by the shaded region in fig. 4.9. Hence we obtain

$$D(\rho_{u,t} \, , \theta_{u,t}) = \{(\rho,\theta) : 0 \leq \rho < \infty \, , \theta_u \leq \theta \leq \theta_t \} \, ,$$

with

$$\theta_\alpha = \tan^{-1} \left(\frac{f_{\alpha,t}}{f_{\alpha,u}} \right) \, , \quad \alpha = u, t \; .$$

We hence get

$$p_{u,t} = \int_{\theta_u}^{\theta_t} \int_0^\infty \frac{1}{2\pi} e^{-\frac{1}{2}\rho_{u,t}^2} \rho_{u,t} \; d\rho_{u,t} \; d\theta_{u,t}$$

$$= \frac{\theta_t - \theta_u}{2\pi} \, . \qquad (4.A.5)$$

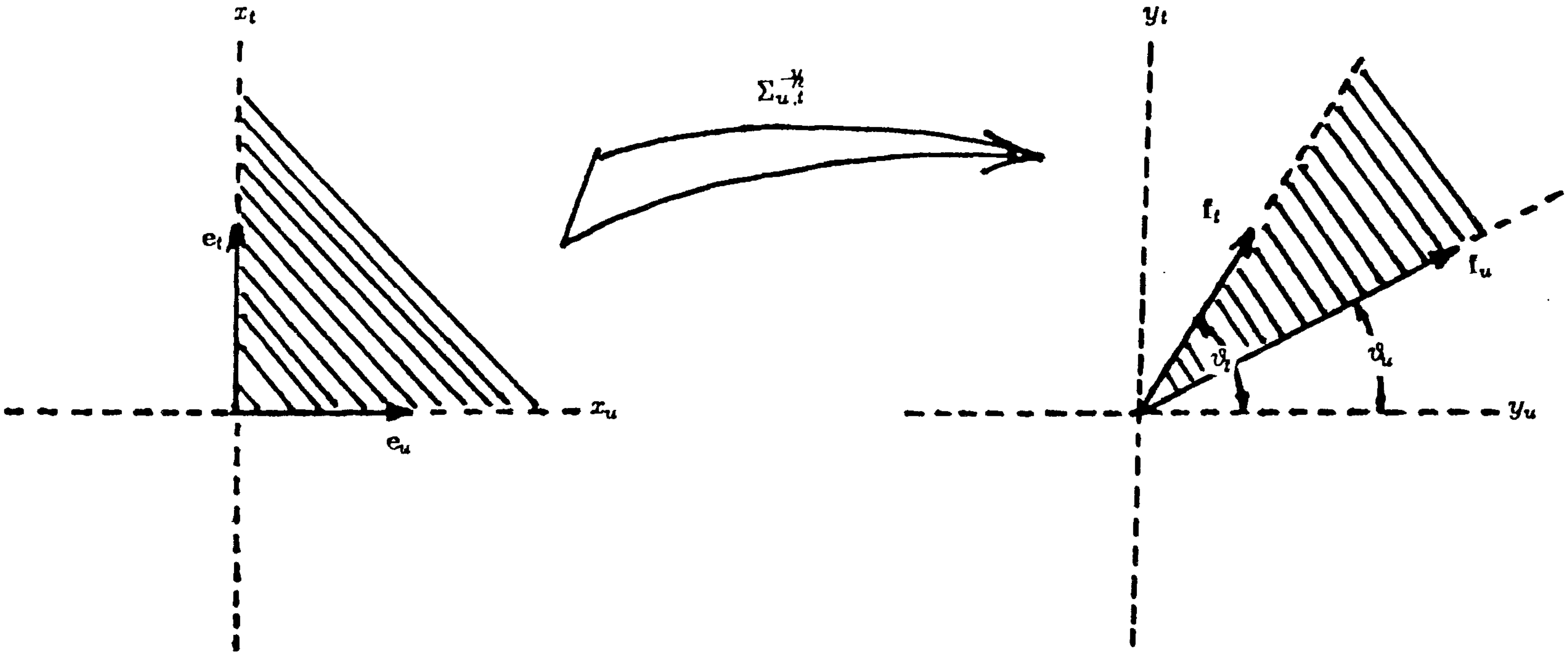

Fig. 4.9. A representation of the transformation of coordinates by the linear map $\Sigma_{u,t}^{-1/2}$. The shaded area between the standard basis vectors $\mathbf{e}_u$, and $\mathbf{e}_t$, corresponds to the first quadrant of the x_u ,x_t plane, and this is mapped into the shaded area between the vectors $\mathbf{f}_u$, and $\mathbf{f}_t$ in the y_u ,y_t plane.

To estimate $p_{u,t}$ it hence suffices to estimate the angle between the basis vectors $\mathbf{f}_u$, and $\mathbf{f}_t$. We have

$$\cos(\theta_t - \theta_u) = \frac{\left\langle \mathbf{f}_u , \mathbf{f}_t \right\rangle}{||\mathbf{f}_u|| \, ||\mathbf{f}_t||}$$

$$= \frac{\left\langle \Sigma_{u,t}^{-1/2}\mathbf{e}_u , \Sigma_{u,t}^{-1/2}\mathbf{e}_t \right\rangle}{||\Sigma_{u,t}^{-1/2}\mathbf{e}_u|| \, ||\Sigma_{u,t}^{-1/2}\mathbf{e}_t||} . \qquad (4.A.6)$$

From equation (4.A.4) we have

$$\Sigma_{u,t}^{-1} = \frac{1}{(r_{u,u} r_{t,t} - r_{u,t}^2)} \begin{bmatrix} r_{u,u} & -r_{u,t} \\ -r_{u,t} & r_{t,t} \end{bmatrix} .$$

We now estimate each of the inner products in equation (4.A.6). The linear transformation $\Sigma_{u,t}^{-1/2}$ is symmetric, and hence it follows that

$$\left\langle \Sigma_{u,t}^{-1/2}\mathbf{e}_u , \Sigma_{u,t}^{-1/2}\mathbf{e}_t \right\rangle = \left\langle \mathbf{e}_u , \Sigma_{u,t}^{-1}\mathbf{e}_t \right\rangle$$

$$= \frac{1}{(r_{u,u} r_{t,t} - r_{u,t}^2)} \begin{pmatrix} 1 \, 0 \end{pmatrix} \begin{bmatrix} r_{u,u} & -r_{u,t} \\ -r_{u,t} & r_{t,t} \end{bmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$= \frac{-r_{u,t}}{r_{u,u} r_{t,t} - r_{u,t}^2} ,$$

$$||\Sigma_{u,t}^{-1/2}\mathbf{e}_u|| = \left(\left\langle \Sigma_{u,t}^{-1/2}\mathbf{e}_u , \Sigma_{u,t}^{-1/2}\mathbf{e}_t \right\rangle\right)^{1/2}$$

$$= \left(\left\langle \mathbf{e}_u , \Sigma_{u,t}^{-1}\mathbf{e}_u \right\rangle\right)^{1/2}$$

$$= \left(\frac{1}{(r_{u,u}\, r_{t,t} - r_{u,t}^2)} \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{bmatrix} r_{u,u} & -r_{u,t} \\ -r_{u,t} & r_{t,t} \end{bmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right)^{1/2}$$

$$= \left(\frac{r_{u,u}}{(r_{u,u}\, r_{t,t} - r_{u,t}^2)} \right)^{1/2} ,$$

and similarly,

$$||\Sigma_{u,t}^{-1/2} \mathbf{e}_t\, || = \left(\frac{r_{t,t}}{(r_{u,u}\, r_{t,t} - r_{u,t}^2)} \right)^{1/2} .$$

It hence follows that

$$\theta_t - \theta_u = \cos^{-1} \left(\frac{-r_{u,t}}{\sqrt{r_{u,u}\, r_{t,t}}} \right) . \tag{4.A.7}$$

Substituting equations (4.A.5), and (4.A.7) in equation (4.A.3), we get

$$\mathbf{E}\left\{ Y(u)Y(t) \right\} = \frac{4}{2\pi} \cos^{-1} \left(\frac{-r_{u,t}}{\sqrt{r_{u,u}\, r_{t,t}}} \right) - 1$$

$$= \frac{2}{\pi} \left[\frac{\pi}{2} - \sin^{-1} \left(\frac{-r_{u,t}}{\sqrt{r_{u,u}\, r_{t,t}}} \right) - 1 \right]$$

$$= \sin^{-1} \left(\frac{r_{u,t}}{\sqrt{r_{u,u}\, r_{t,t}}} \right) . \tag{4.A.8}$$

Appendix B

Let $X(u)$ be a normal random process as in Appendix A. Define the process $Z(u)$ by

$$Z(u) = |X(u)| \ .$$

For the mean of the process $Z(u)$ we have

$$\mathrm{E}\left\{Z(u)\right\} = \frac{1}{\sqrt{2\pi r_{u,u}}} \int_{-\infty}^{\infty} |x| \; e^{-\frac{x^2}{2r_{u,u}}} \, dx$$

$$= \left(\frac{2}{\pi r_{u,u}}\right)^{1/2} \int_{0}^{\infty} x \; e^{-\frac{x^2}{2r_{u,u}}} \, dx$$

$$= -\left(\frac{2r_{u,u}}{\pi}\right)^{1/2} e^{-\frac{x^2}{2r_{u,u}}} \Bigg|_{0}^{\infty}$$

$$= \left(\frac{2r_{u,u}}{\pi}\right)^{1/2} . \qquad (4.B.1)$$

We now estimate the second moment of $Z(u)$. As a consequence of Price's Theorem [9], we have

$$\frac{\partial \mathrm{E}\left\{Z(u)Z(t)\right\}}{\partial \mathrm{E}\left\{X(u)X(t)\right\}} = \mathrm{E}\left\{\frac{\partial^2\{Z(u)Z(t)\}}{\partial X(u)\,\partial X(t)}\right\}$$

$$= \mathrm{E}\left\{\frac{dZ(u)}{dX(u)}\;\frac{dZ(t)}{dX(t)}\right\} .$$

Now we have the relation

$$\frac{dZ(u)}{dX(u)} = \frac{d\,|X(u)|}{dX(u)} = \operatorname{sgn} X(u)\,,$$

which holds at all points except $X(u) = 0$, which is of probability measure zero. Setting $\mathbf{E}\left\{X(u)X(t)\right\} = r$ as the variable of integration, we hence have

$$\frac{\partial \mathbf{E}\left\{Z(u)Z(t)\right\}}{\partial r} = \mathbf{E}\left\{\operatorname{sgn} X(u)\operatorname{sgn} X(t)\right\}$$

$$= \frac{2}{\pi}\sin^{-1}\left(\frac{r}{\sqrt{r_{u,u}\,r_{t,t}}}\right), \tag{4.B.2}$$

which follows from equation (4.A.8). Now, if the jointly normal random variables $X(u)$ and $X(t)$ were uncorrelated (and hence also independent) we would have $r = 0$, and

$$\mathbf{E}\left\{Z(u)Z(t)\right\}\Big|_{r=0} = \mathbf{E}\left\{Z(u)\right\}\Big|_{r=0}\,\mathbf{E}\left\{Z(t)\right\}\Big|_{r=0}$$

$$= \frac{2}{\pi}\sqrt{r_{u,u}\,r_{t,t}} \tag{4.B.3}$$

from equation (4.B.1). Integrating both sides of equation (4.B.2) between the limits $r = 0$ and $r = r_{u,t}$, we get

$$\mathbf{E}\left\{Z(u)Z(t)\right\}\Big|_{r=r_{u,t}} - \mathbf{E}\left\{Z(u)Z(t)\right\}\Big|_{r=0} = \int_0^{r_{u,t}} \frac{2}{\pi}\sin^{-1}\left(\frac{r}{\sqrt{r_{u,u}\,r_{t,t}}}\right) dr\;.$$

Set $s = \dfrac{r}{\sqrt{r_{u,u}\,r_{t,t}}}$, and $s_{u,t} = \dfrac{r_{u,t}}{\sqrt{r_{u,u}\,r_{t,t}}}$. Integrating by parts, we have

$$\mathbf{E}\left\{Z(u)Z(t)\right\}\Big|_{r=r_{u,t}} - \mathbf{E}\left\{Z(u)Z(t)\right\}\Big|_{r=0}$$

$$= \frac{2\sqrt{r_{u,u}\,r_{t,t}}}{\pi}\int_0^{s_{u,t}} \sin^{-1} s \;\, ds$$

$$= \frac{2\sqrt{r_{u,u}\,r_{t,t}}}{\pi}\left[s\,\sin^{-1} s\,\Big|_0^{s_{u,t}} - \int_0^{s_{u,t}} \frac{s}{\sqrt{1-s^2}}\,ds \right]$$

$$= \frac{2\sqrt{r_{u,u}\,r_{t,t}}}{\pi}\left[s_{u,t}\,\sin^{-1} s_{u,t} + \sqrt{1-s^2}\,\Big|_0^{s_{u,t}} \right]$$

$$= \frac{2\sqrt{r_{u,u}\,r_{t,t}}}{\pi}\left[s_{u,t}\,\sin^{-1} s_{u,t} + \left(1 - s_{u,t}^2\right)^{½} - 1 \right].$$

Replacing $\mathbf{E}\left\{Z(u)Z(t)\right\}\Big|_{r=r_{u,t}}$ by $\mathbf{E}\left\{Z(u)Z(t)\right\}$, and $s_{u,t}$ by $\dfrac{r_{u,t}}{\sqrt{r_{u,u}\,r_{t,t}}}$ and using equation (4.B.3) we finally obtain

$$\mathbf{E}\left\{Z(u)Z(t)\right\} = \frac{2}{\pi}\left[r_{u,t}\sin^{-1}\left(\frac{r_{u,t}}{\sqrt{r_{u,u}\,r_{t,t}}}\right) + \sqrt{r_{u,u}\,r_{t,t}}\left(1 - \frac{r_{u,t}^2}{r_{u,u}\,r_{t,t}}\right)^{½} \right]. \tag{4.B.4}$$

REFERENCES

[1] G. L. Turin, "An introduction to matched filters," *IRE Trans. Inform. Theory,* vol. IT-6, pp. 311–329, 1960.

[2] A. Vander Lugt, "Signal detection by complex spatial filtering," *IEEE Trans. Inform. Theory,* vol. IT-10, pp. 139–145, 1964.

[3] A. V. Oppenheim and R. V. Schafer, *Digital Signal Processing.* Englewood Cliffs, New Jersey: Prentice-Hall, 1975.

[4] D. Psaltis, E. G. Paek, and S. S. Venkatesh, "Optical image correlation using a binary spatial light modulator," *Opt. Eng.*, vol. 23, No. 6, pp. 698–704, 1984.

[5] D. Psaltis, F. Mok, and E. G. Paek, in *Spatial Light Modulators and Applications*, Uzi Efron, ed., Proc. SPIE, vol. 29, p. 465, 1984.

[6] D. Slepian and H. O. Pollak, "Prolate spheroidal wave functions, Fourier analysis and uncertainty–I," *Bell Sys. Tech. Jnl.*, vol. 40, pp. 43–64, 1961.

[7] T. Kailath, "The divergence and Bhattacharyya distance measures in signal selection," *IEEE Trans. Comm. Tech.*, vol. COM-15, No. 1, pp. 52–60, 1967.

[8] S. R. Curtis, A. V. Oppenheim, and J. S. Lim, "Signal reconstruction from Fourier Transform sign information," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-33, no. 3, pp. 643–657, 1985.

[9] R. Price, "A useful theorem for nonlinear devices having Gaussian inputs," *IRE Trans. Inform. Theory*, vol. IT-4, pp. 69–72, 1958.

Neural Networks

CHAPTER V

ASSOCIATIVE NEURAL NETS

1. NEURAL NETWORK MODELS

A. Iterated Maps

We preface our discussion of associative neural networks with a mathematical modeling of these structures, and indications as to how these models can result in useful computation.

Thus far in our discussion of correlator strategies we have addressed the classification problem–the making of *hard decisions* assigning unknown patterns from some pattern space $\mathbb{H}_p$ to one (or possibly none) of a set of prescribed pattern classes of interest. To recapitulate, the classification decison involves two stages: a feature extraction stage $\mathbf{D} \circ \mathbf{W} : \mathbb{H}_p \rightarrow \mathbb{H}_f$ composed with a hard decision (classification) stage $\mathbf{T} \circ \mathbf{L} : \mathbb{H}_f \rightarrow \mathbb{B}$. The first stage maps patterns in the pattern space $\mathbb{H}_p$ to a feature space $\mathbb{H}_f$ (of possibly reduced dimensionality); the linear map $\mathbf{W} : \mathbb{H}_p \rightarrow \mathbb{H}_f$ serves to extract critical features for classification, while the non-linear pointwise decision rule $\mathbf{D} : \mathbb{H}_f \rightarrow \mathbb{H}_f$ provides non-linear logical computational capability. The second stage is a linear discriminant function (or threshold gate, or perceptron) classification rule which determines the choice of separating planes in the feature space $\mathbb{H}_f$ which optimally partition the space of feature vectors according to the prescribed pattern classes.

The non-linear correlator structure we have considered so far is clearly a feed forward system with a final classification stage $\mathbf{T} \circ \mathbf{L}$ where a hard decision is made in a single pass. In this section we consider systems which make *soft decisions* in the pattern space itself through the introduction of feedback. In particular, we consider iterated maps of the form $(\mathbf{D} \circ \mathbf{W})^k : \mathbb{H}_p \to \mathbb{H}_f$ which map the pattern space $\mathbb{H}_p$ to itself. In the terminology of the last section, this is equivalent to an infinite cascade of "feature extraction stages" $\mathbf{D} \circ \mathbf{W}$; here, however, the "feature space" is identical to the pattern space itself, and we eschew the hard decision stage $\mathbf{T} \circ \mathbf{L}$.

There is clearly potential for storing associations of patterns within such a feedback system. Specifically, *a priori* unknown patterns could be mapped to appropriate pattern classes by successive iterations in the pattern space itself. Within such a feedback system there is a potential for a gradual or soft correction of errors in distorted or noisy patterns (as opposed to the single step, hard decisions of the previous section) by clever choices of global linear maps $\mathbf{W} : \mathbb{H}_p \to \mathbb{H}_p^*$ and pointwise nonlinear rules $\mathbf{D} : \mathbb{H}_p^* \to \mathbb{H}_p$. Latent possibilities include the capability to recover from occasional incorrect decisions in the course of iteration. (This, of course, is not possible when a single hard decision is made; an error remains–quite incontrovertibly–an error.)

B. Neurobiological Modeling

We will typically consider patterns drawn from subsets of Euclidean n-space; specifically, in chapters VI and VII, we consider patterns chosen from the vertices of a binary n-cube, $\mathbb{H}_p = \mathbb{B}^n \triangleq \{-1,1\}^n$, while in chapters VII and IX we consider cases where patterns are chosen from real n-space, $\mathbb{H}_p = \mathbb{R}^n$. The map that we will consider in this section will typically be iterations of global linear maps in conjunction with pointwise threshold rules. Such maps find extensive application in neurobiology as mathematical models of brain function, and in turn, these neuro-anatomical models have proved fertile ground in the development of efficient systems of associative or content addressable memory. We will hence have recourse to (grossly simplified) models and ideas borrowed from neurobiology, and adopt neurobiological terminology. So first, some neurobiological motivation.

Neural network models based upon mathematical idealizations of biological memory typically consist of a densely interconnected, dynamical cellular cluster [1]. The processing nodes in such a structure are the *neurons*, and the neuronal interconnections are through the medium of linear *synaptic conduits*. Describing the instantaneous state of a neural network to be the collective states of each of the individual neurons (firing or non-firing) in the system then leads to a characterisation of the dynamics of the system as a motion in time through the state space of the system. In this form, then, the mathematical abstraction of neural function leads to a consideration of a finite state automaton with specified state transition rules. Other dynamical systems much akin to neural networks in this regard include the Ising spin glass models (cf. [2], for instance) and cellular automata (cf. [3]).

We consider an associative structure based upon such a neural net. The model (McCulloch-Pitts) neurons we consider are simple bistable elements each being capable of assuming two values: -1 (off) and 1 (on). The *state* of each neuron then represents one bit of information, and the state of the *system* as a whole is described by a binary n-tuple if there are n neurons in the system. We assume that the neural net is (possibly) densely interconnected, with neuron i transmitting information to neuron j through a linear synaptic connection w_{ij}. The neural interconnection weights w_{ij} are considered throughout to be *fixed*; i.e., learning of associations has already taken place, and no further synaptic modifications are made in the neurobiological interpretation. We will frequently assume that the connection matrix is symmetric with zero diagonal. This, of course, implies that there is no self-reinforcement for neural sites, and that in practical implementations, the neural pathways need not be distinct.

The schema of fig. 5.1 illustrates a typical example of the structure that we envisage for our associative memory thought of as a neural network. A five-neuron densely interconnected neural network is shown. The circles represent the direction of inter-neural information flow through the corresponding synaptic weight w_{ij}. The instantaneous state of the system depicted is $(u_1,u_2,u_3,u_4,u_5) = (1,-1,1,-1,-1)$. While it is not necessary for the matrix of weights $\mathbf{W}$ to be symmetric, useful computational behaviour obtains if, in fact, it is symmetric.

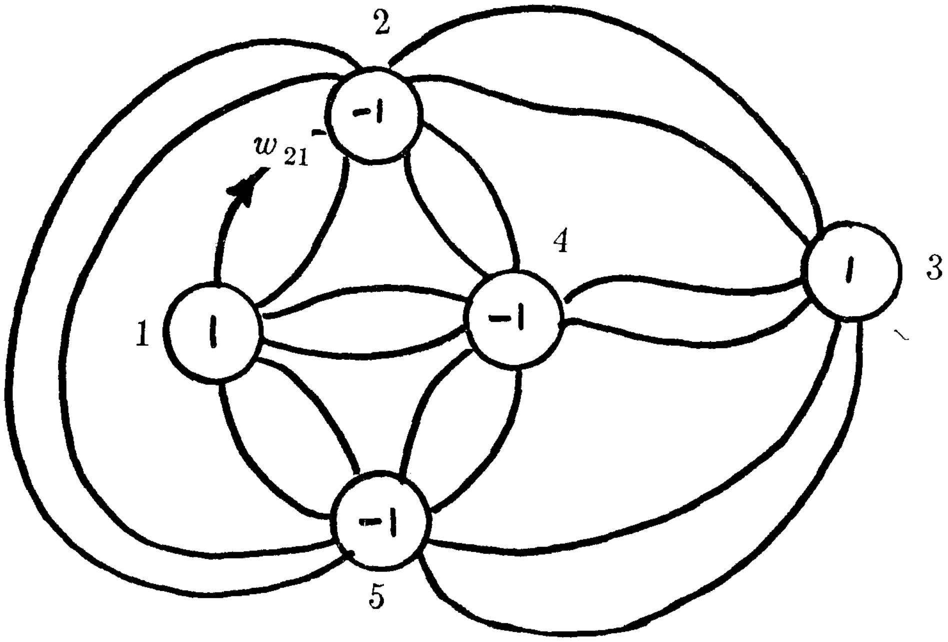

Fig. 5.1. A five-neuron densely interconnected network.

Logical computation in the network takes place at each neural site by means of a simple threshold decision rule. Each neuron evaluates the weighted sum of the binary states of all the neurons in the system; the new state of the neuron is -1 if the sum is negative and +1 if the sum (equals or) exceeds zero. (In this, and in what follows, we almost always assume a threshold of zero.) Specifically, if $\mathbf{u} = (u_1, u_2, \ldots, u_n)$ is the present state of the system (with $u_j = \pm 1$ being the state of the j-th neuron), the new state u_i^* of the i-th neuron is determined by the rule

$$u_i^* = \operatorname{sgn} \left\{ \sum_{j=1}^{n} w_{ij} u_j \right\} = \begin{cases} 1 & \text{if } \sum w_{ij} u_j \geq 0 \\ -1 & \text{if } \sum w_{ij} u_j < 0 \end{cases} . \qquad (5.1.1)$$

We will discuss two modes of changing state $\mathbf{u} \mapsto \mathbf{u}^*$. In *synchronous* operation, each of the n neurons *simultaneously* evaluates and updates its state according to rule (5.1.1). In *asynchronous* operation, the components of the current "probe" vector $\mathbf{u}$ are updated one at a time according to (5.1.1), to produce a new probe vector. At any given "refresh" instant, the component i chosen to be updated is selected from among the n indices i with equal probability $1/n$, independently of which components were updated previously and of what the values of the probe vector were before and after update.

In terms of our previous formalism, the states of the system are binary n-tuples $\mathbf{u} = (u_1, \ldots, u_n)$ belonging to the pattern space $\mathbb{B}^n$. State transitions are dictated by iterates of the composite map $\mathbf{D} \circ \mathbf{W} : \mathbb{B}^n \rightarrow \mathbb{B}^n$, where $\mathbf{W}$ is a linear map corresponding to communication through the synaptic weights, and $\mathbf{D}$ is a threshold rule. Both synchronous and asynchronous modes of operation can be accommodated within this system-theoretic formalism.

In synchronous operation, the linear transformation $\mathbf{W}$ corresponds to the $n \times n$ matrix of interconnection weights $[w_{ij}]$, and $\mathbf{D}$ is a threshold operator which accepts n-vectors as input, and returns n-vectors whose components are the signs of the input vector. (We, as before, use the convention that sgn $(0) = 1$.) If $\mathbf{u}$ is the present state of the system, then the new system state is

$$(\mathbf{D} \circ \mathbf{W})\mathbf{u} = \begin{bmatrix} \operatorname{sgn}\ (\sum w_{1j}\, u_j) \\ \vdots \\ \operatorname{sgn}\ (\sum w_{nj}\, u_j) \end{bmatrix}.$$

Modeling is a bit more complex for the asynchronous case as we have to accommodate the fact that only a single, randomly chosen component of a state vector is evaluated at a time. We can model the situation by treating $\mathbf{W}$ as a random operator with an ensemble of n linear maps $\mathbf{W}_1, \ldots, \mathbf{W}_n$, each occurring with probability $1/n$. Specifically, we assume that map $\mathbf{W}_i$ is in operation if the i-th component of the state vector is chosen to be updated. The matrix of components of $\mathbf{W}_i$ is simply obtained from the identity matrix by replacing the i-th row with $(w_{i1}, \ldots, w_{in})$ – the interconnection weights leading to the i-th neuron. The non-linear map $\mathbf{D}$ is a point threshold map as before. Thus if $\mathbf{u} = (u_1, \ldots, u_n)$ is the present system state, and the i-th neuron is to be updated, then $\mathbf{W}_i$ is the linear operator in use, and the new state is

$$(\mathbf{D} \circ \mathbf{W}_i)\mathbf{u} = \begin{bmatrix} \operatorname{sgn}\ (u_1) \\ \vdots \\ \operatorname{sgn}\ (u_{i-1}) \\ \operatorname{sgn}\ (\sum w_{ij}\, u_j) \\ \operatorname{sgn}\ (u_{i+1}) \\ \vdots \\ \operatorname{sgn}\ (u_n) \end{bmatrix} = \begin{bmatrix} u_1 \\ \vdots \\ u_{i-1} \\ \operatorname{sgn}\ (\sum w_{ij}\, u_j) \\ u_{i+1} \\ \vdots \\ u_n \end{bmatrix}.$$

In this neural network model, the linear synaptic weights provide global communication of information, while the non-linear logical operations essential to computation take place at the neurons. Thus, in spite of the simplicity of the highly stylised neural network structure that we utilise, considerable computational power is inherent in the system. The implementation of models of learning (the *Hebbian hypothesis*, [4]), and associative recall ([5], [6], [7], [8], [9], [10], [11]), and the solution of complex minimisation problems ([12], [13]) using such neural networks is indicative of the computational power latent in the system.

The central features of such associative computational systems are: (1) the powerful highly fanned-out distributed information processing that is evidenced as a natural consequence of collective system dynamics; (2) the extreme simplicity of the individual processing nodes; and (3) the massive parallelism in information processing that accrues from the global flow of information, and the concurrent processing at the individual neural sites of the network. To recapitulate, keynotes of such neural network structures include a high degree of parallelism, distributed storage of information, robustness, and very simple basic elements performing tasks of low computational complexity.

2. ASSOCIATIVE MEMORY

We now consider neural associative nets. Questions that we will attempt to answer include: Can problem specific neural networks be designed to "store" sets of prescribed associations? What is the storage capacity of such networks? How are capacity estimates modified if we require error correction or allow some error tolerance?

In the rest of the chapter, we will define the associative structure that we consider, and make precise the notion of capacity.

A. Association, Attraction, and Tolerance

We will consider in main auto-associative storage wherein prescribed binary n-tuples $\mathbf{u} \in \mathbb{B}^n$ are stored as memories in suitably chosen neural networks. (The case where arbitrary associations $\mathbf{u} \mapsto \mathbf{v}$ are required to be stored is considered in chapter VIII; the results of chapters VI and VII also extend simply to hetero-associative storage.) We define memory in a natural fashion for these systems: we typically require that vectors $\mathbf{u}$ in the state space of the neural network that are labelled as memories be fixed points of the system. Specifically, if $\mathbf{u} \in \mathbb{B}^n$ is specified as a memory, then we require that for each neuron $i = 1,\ldots,n$,

$$u_i = \operatorname{sgn} \left\{ \sum_{j=1}^{n} w_{ij} \, u_j \right\},$$

where, as before, w_{ij} denotes the directed weight linking neuron j to neuron i. Specified states $\mathbf{u} \in \mathbb{B}^n$ that are to be stored will be called *fundamental memories.* It is clear that the fundamental definition of memories is independent of whether we have the asynchronous or synchronous models, as fixed points are the same in either mode. But in the structure of associations, it is a desideratum that the stored memories are also *attractors*; i.e., they exercise a region of influence around them so that states which are sufficiently similar to the memory are mapped to the memory by repeated iterates of the system operator.

In essence, then, we shall require that if the initial state of the neural network is "close" to a memory then the system dynamics will proceed in a direction so that the neural network settles in a stable state centered at the memory, or at least close to it. Here we use the Hamming distance as the natural similarity measure between two states in the binary n-space under consideration. With this interpretation, we can think of an associative memory as a basket of n memories (fig. 5.2) which corrects all (or most of) the errors in an initial *probe vector* within a certain specified distance (the prescribed error correction capability) of a stored memory. It turns out that in many contexts, situations with incorrectly specified or distorted memories are redressable by the neural network as long as the number of components in error is fewer than $n/2$; i.e., no more than half of all the components are incorrect initially.

There is a persuasive analogy between these ideas of associative memory and concepts from information theory and coding. We could view the associative memory as a kind of decoder for a code consisting of the m fundamental memories; the distorted memories or probes could be viewed as noisy patterns, and the mechanism of decoding would be the iterated mapping performed by the network. But the codes will, as we shall see, have very low rates, and hence be of limited usefulness for channel coding. Nonetheless, the techniques that we will use are quite reminiscent of coding theory, especially random coding and sphere hardening.

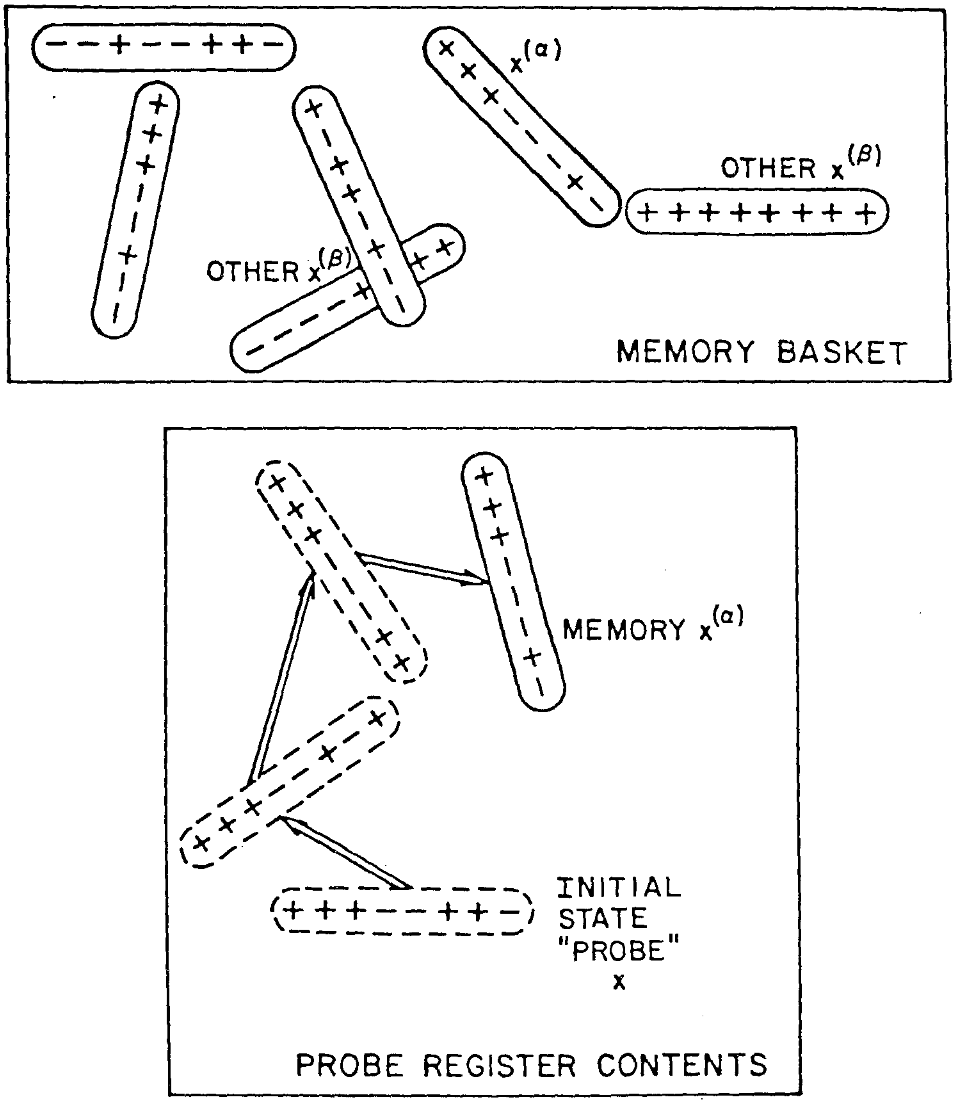

Fig. 5.2. Associative memory basket.

Thus, the two issues we focus on are the storage of prescribed fundamental memories within a neural network and their recovery. Requiring the fundamental memories to be fixed points ensures that they are recoverable (in a weak sense) under the threshold map. (Memory retrieval is error free if they are fixed points. We consider the effect of tolerating errors in retrieval in chapter VIII.) This constraint, by itself, scarce suffices for an associative memory, however, and we also require some error correction or "pull-in" capability.

In what follows, we will typically assume a "forced choice" model, in which unknown components are labelled -1 or 1 randomly, with a probability of half that they are labelled correctly. If particular components are known to be correct, it is tempting to "clamp" them to their true values, so that they do not undergo change. This, however, does not increase the storage capacity because it turns out that "right" components never change anyway. We will assume throughout that no component is clamped, and any or all components can change sign.

In probing any particular memory with a distorted version of the memory, we assume that at least $(1-\rho)\,n$ of the components are correct, so that ρn or fewer components are incorrect. (Here $0 \leq \rho < 1/2$ is the fractional Hamming distance of the probe from the memory.) Our requirement is that for a specified ρ, the probe is ultimately mapped onto the corresponding fundamental memory. We call ρ the (fractional) *radius of attraction.*

This property provides true associative capability. Given convergence to the correct memory for any given attraction radius $0 \leq \rho < 1/2$, all we require is that $(1-2\rho)\,n$ components of the memory be known. By forced choice, half of the remaining components, ρn, will be correct on average, so that convergence to the correct memory obtains.

There are at least three possibilities of convergence for the asynchronous case. First, the sphere of radius ρn may be directly or monotonically attracted to its fundamental memory center, meaning that every transition that is actually a change in a component is a change in the right direction, as in fig. 5.3 (a). (Alternatively, the synchronous version goes to its fundamental memory center in one step.) Second, with

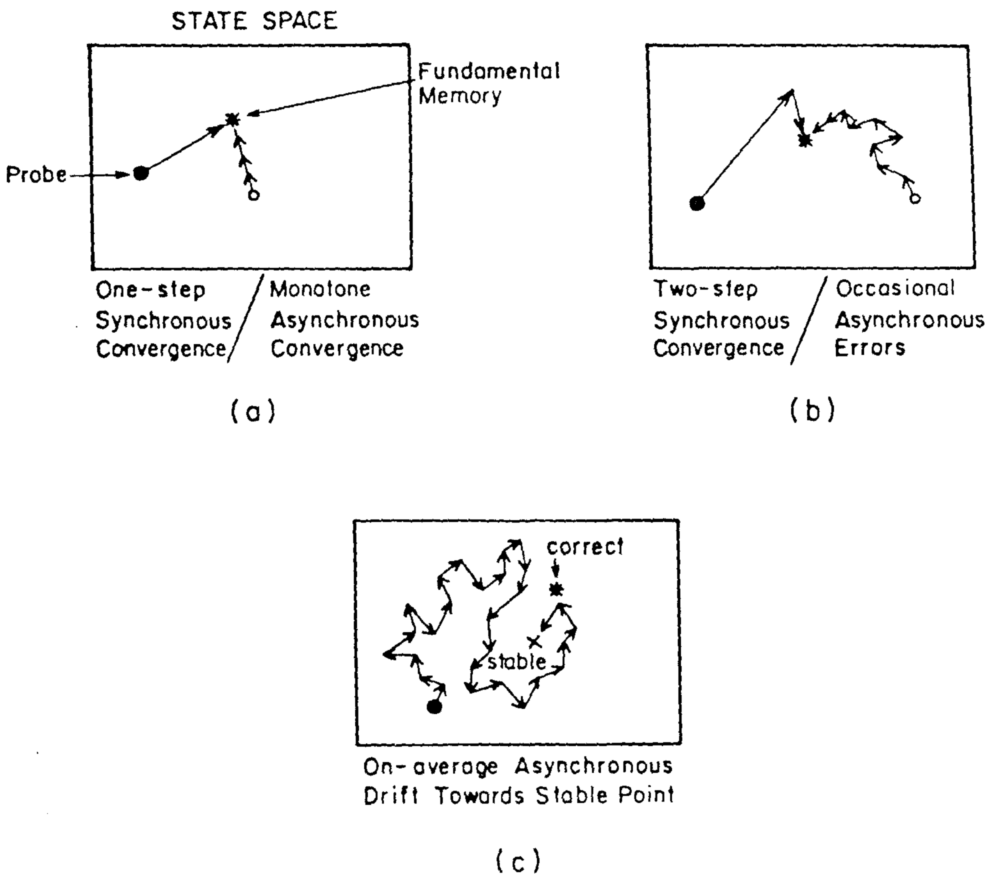
STATE SPACE
Fundamental Memory
Probe
One-step Synchronous Convergence
Monotone Asynchronous Convergence
(a)
Two-step Synchronous Convergence
Occasional Asynchronous Errors
(b)
correct
stable
On-average Asynchronous Drift Towards Stable Point
(c)

Fig. 5.3. Representation of the various types of convergence.

high enough probability but not probability one, a random step is in the right direction, as in fig. 5.3 (b). After enough steps, the probe has with high probability come very close to its fundamental memory center, so that then all subsequent changes are in the right direction, i.e., we are then directly attracted. (For the synchronous case, this implies two-iteration convergence in many algorithms.)

The third mode of convergence, does not correspond to anything obvious in the synchronous case. In this mode, components can change back and forth during their sojourn, but at least *on the average* get better, i.e., are more likely to be correct *after* a change than before. After a finite number of changes, the system settles down to a fixed point, as we know it must, and this fixed point is either the correct memory or not too far from it, say within ϵn . These situations are diagrammed in fig. 5.3 (c). We will be concerned mainly with the first two modes of convergence.

Another issue of practical importance is error tolerance. Requiring the memories to be fixed points enjoins no errors on retrieval. In practice, however, it may be permissible to allow a few components to be wrong in a retrieved memory. More formally, if we specify a *tolerance* of $0 \leq \epsilon < 1/2$, then we require that retrieved memories differ from the memories in at most ϵn components. The situation is illustrated in fig. 5.4, where a ball of attraction, $0 \leq \rho < 1/2$, and an error tolerance, $0 \leq \epsilon < \rho$, are specified. (It turns out for the outer product algorithm of the next chapter, that convergence is to the surface of the epsilon ball, as illustrated schematically.) We will return to the subject of error tolerance and how it affects storage capacity in chapter VIII. For the nonce we will require, in main, that the fundamental memories are fixed points.

The incorporation of sequences of associations and memory within the neural networks structure that we consider now naturally raises two issues: the nature of the memory encoding rule by means of which a desired structure of associations can be programmed into the network, and the capacity of the resultant system to recall the stored memories with some measure of error correction. Note that with the nature of the thresholding operations fixed, the only flexibility that we have to realise different neural networks is the choice of the synaptic weights of connections w_{ij} . The memory encoding rule is, in essence then, an algorithm for the appropriate choice of weights

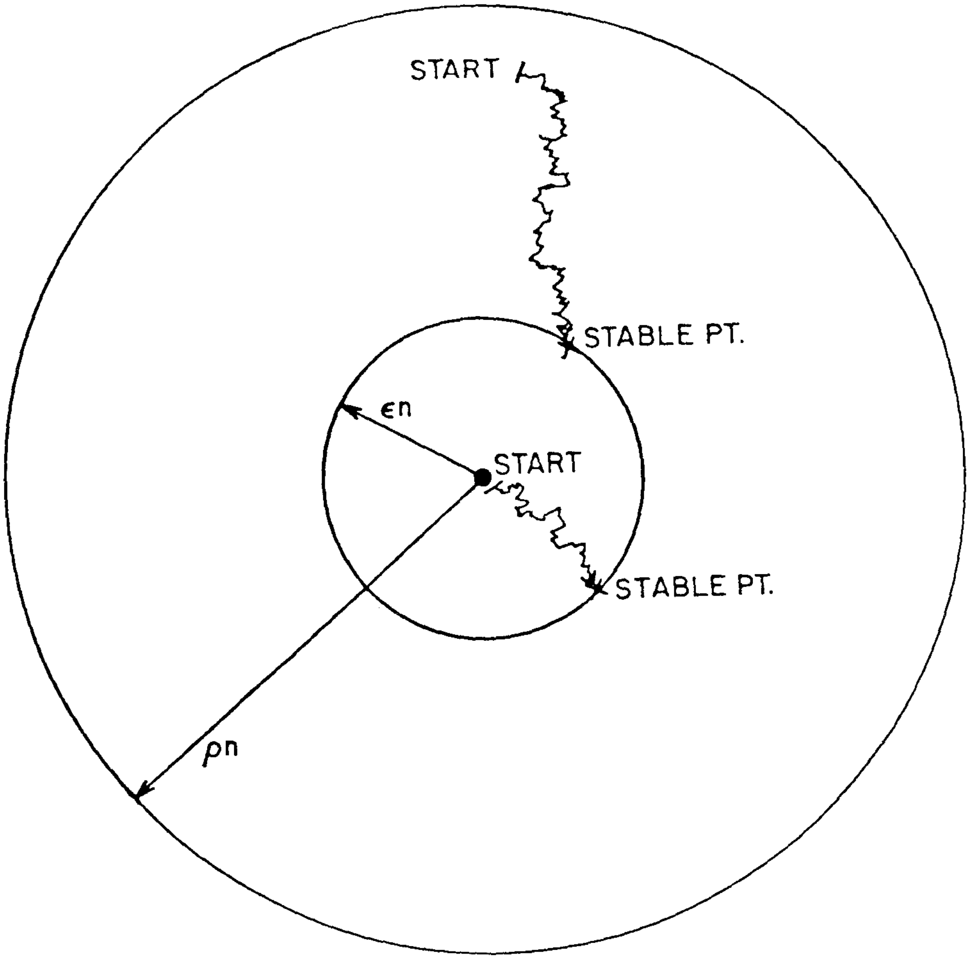

Fig. 5.4 Associative attraction with error tolerance (after McEliece, et al.)

w_{ij}. We will treat various algorithms for memory encoding in the next two chapters. We now proceed to a definiton of the storage capacity of an algorithm.

B. Capacity Definitions

Here we introduce formally the notion of capacity of any (algorithmically derived) associative neural network. We will restrict our attention to auto-association, and require that our associations be error free.

We consider networks comprised of n neurons. Let $\mathbf{U} = \{\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(m)}\} \subseteq \mathbb{B}^n$ be a randomly chosen m set of fundamental memories that we wish to store in a neural network by means of a particular algorithm for choice of interconnection weights (which depends on $\mathbf{U}$, in general). Loosely speaking, we will define the storage capacity of the particular algorithmic scheme under consideration to be the maximum rate of increase of the number of fundamental memories m with the number of neurons n. This is clearly a critical parameter determining the efficacy of that particular algorithm.

For error free associative maps we require that $u_i^{(\alpha)} = \text{sgn } (\sum w_{ij} u_j^{(\alpha)})$ for each component i, and for each memory. As pointed out earlier, the requirement of fixed points is independent of the particular mode of operation–synchronous or asynchronous.

We require that the m fundamental memories $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(m)} \in \mathbb{B}^n$ be stored as fixed points in the neural network. (We will require that *almost all* of the $\binom{2^n}{m}$ choices of fundamental memories be allowable candidates for storage.) The m-set of fundamental memories is assumed to be a randomly chosen set with the components $u_i^{(\alpha)}$, $i = 1,\ldots,n$, $\alpha = 1,\ldots,m$, chosen from a sequence of Bernoulli trials with

$$\mathbf{P}\left\{u_i^{(\alpha)} = -1\right\} = \mathbf{P}\left\{u_i^{(\alpha)} = 1\right\} = \frac{1}{2}\ .$$

Let algorithm X identify any particular algorithmic scheme for generating the neural interconnection weights. For the rest of this section we identify the matrix of weights

$\mathbf{W} = [w_{ij}]$ to have been generated by algorithm X, given the memories $\mathbf{u}^{(\alpha)}$, $\alpha = 1,...,m$, to be stored.

Definition. Let $\mathbf{w}_i \in \mathbb{R}^n$ be the vector of interconnection weights associated with neuron i, for each neuron $i = 1,...,n$. The event:

$$\operatorname{sgn}\left(\sum_{j=1}^{n} w_{ij}\, u_j^{(\alpha)}\right) = v_i^{(\alpha)} , \quad i = 1,...,n \ , \alpha = 1,...,m \ , \tag{5.2.1}$$

is described by saying that *the neural network stores m fundamental memories.*

For the algorithmically prescribed matrix of interconnection weights $[w_{ij}]$, the relations (5.2.1) describe an event whose occurrence depends on the particular values assumed by the randomly chosen fundamental memory components $u_i^{(\alpha)}$ only.

We now rigourously define the notion of capacity. Our definitions will subsume several commonly used notions of capacity. For the nonce, we consider only the storage of prescribed vectors as fixed points.

An issue of importance in defining capacity is that of *programmability.* It may well turn out that in the process of generating weights which store a given m-set of fundamental memories as fixed points, that extraneous fixed points are created incidentally. In fact, there is some evidence that there could be as much as an exponential number of these extraneous fixed points [14]. For any particular specification of m fundamental memories to be stored, we would like to create as few extra fixed points as possible, so that they do not interfere with system dynamics significantly. As an extreme example of how too many extraneous stable states can interfere with system dynamics, consider a matrix of weights which is just the identity matrix, where all the neurons are physically isolated from each other. For this case, *all* of the 2^n states of the system are fixed points, and there is no attraction behaviour whatsoever. In fact, it turns out that some restrictions have to be imposed on the choices of allowable weight matrices for auto-association in order to avoid this sort of situation. We will return to this issue in chapter VIII.

Our definitions of capacity will reflect, in some sense, the maximum number of arbitrarily *prescribed* memories that can be *programmed* into the network by specified algorithms. The extraneous fixed points that build up as a consequence of the storage of the prescribed memories themselves are not counted as contributing to the capacity. Formally, we will require that *almost all* m-sets of fundamental memories (with m within the capacity of the particular algorithm in use) be programmable as fixed points (or even as attractors) in the network by the specified algorithm. This will serve to eliminate the extraneous fixed points from consideration.

Definition. A sequence of integers $\{\underline{C}(n)\}_{n=1}^{\infty}$ is a *lower sequence of capacities* for algorithm X iff for each $\lambda \in (0,1)$, the neural network specified by the algorithm stores m fundamental memories with probability approaching one as $n \rightarrow \infty$ whenever $m \leq (1-\lambda)\underline{C}(n)$.

This is a lower estimate for the storage capacity; it tells us that for large n, if the number of associations is chosen to be less than the lower capacity, then with probability essentially one, we can find neural networks for almost all choices of fundamental memories $\mathbf{u}^{(\alpha)}$, $\alpha=1,\ldots,m$. Note that the requirement that almost all of the $\begin{pmatrix} 2^n \\ m \end{pmatrix}$ choices of m fundamental memories be programmable as fixed points in the network occurs naturally in the definition. This follows as a consequence of the random choice of memories, and the requirement that there is convergence with probability one.

An alternative approach to capacity overestimates the storage capacity.

Definition. A sequence of integers $\{\overline{C}(n)\}_{n=1}^{\infty}$ is an *upper sequence of capacities* for algorithm X iff for each $\lambda \in (0,1)$, the event that the neural network specified by the algorithm stores m fundamental memories has probability zero as $n \rightarrow \infty$ whenever $m \geq (1+\lambda)\overline{C}(n)$.

The above definition is an upper estimate for the storage capacity of the algorithm. If the number of memories, m, is chosen to be larger than the upper capacity, then for almost all choices of m memories, $\exists$ memories which are not fixed points. Figure 5.5 (a) illustrates the type of behaviour required by the above definitions.

The definitions are not as general as they might be. For instance, upper and lower tolerances $0 \leq \underline{\delta} < \overline{\delta} \leq 1$ could be specified instead of probability one and zero, respectively. The definitions as they stand will, however, suffice for our purposes. The only requirement we wish to impose for the definitions to be useful is that the probabilities of interest behave monotonically with the number of associations m, as illustrated schematically in figure 5.5 (a). Specifically, we would like to rule out oscillatory behaviour of the probabilities with increase in m –at least in the very high and very low probability regions. As we shall see in the following chapters, the probabilities are indeed monotonic in m.

The lower and upper capacity definitions do not result in unique sequences. Clearly, if $\{\underline{C}(n)\}$ is a lower sequence, then any smaller sequence is also a lower sequence. Similarly, if $\{\overline{C}(n)\}$ is an upper sequence, any larger sequence is also an upper sequence. The following result, however, indicates that from given lower sequences of capacity, we can actually create larger sequences (which do not differ too much from the original sequence) which are still lower sequences of capacity for algorithm X, and vice versa for upper sequences.

Proposition 5.2.1.

(a) If $\{\underline{C}(n)\}_{n=1}^{\infty}$ is a lower sequence of capacities, then so is $\{[1 \pm o(1)]\underline{C}(n)\}_{n=1}^{\infty}$.

(b) If $\{\overline{C}(n)\}_{n=1}^{\infty}$ is an upper sequence of capacities, then so is $\{[1 \pm o(1)]\overline{C}(n)\}_{n=1}^{\infty}$. ■

Proof. (a) Fix $\lambda \in (0,1)$ and consider $(1-\lambda)(1 \pm o(1))\underline{C}(n)$. For n large enough, $\exists$ $\lambda^* \in (0,1)$ such that $(1-\lambda)(1 \pm o(1)) \leq 1-\lambda^*$. Choose m small enough that $m \leq (1-\lambda)(1 \pm o(1))\underline{C}(n) \leq (1-\lambda^*)\underline{C}(n)$. By assumption of $\{\underline{C}(n)\}$ being a

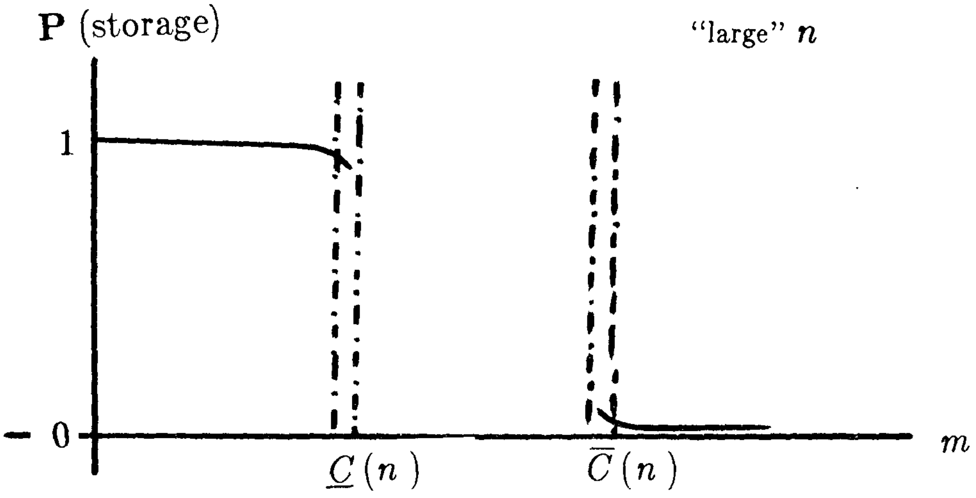

Fig. 5.5 (a). Upper and lower capacities.

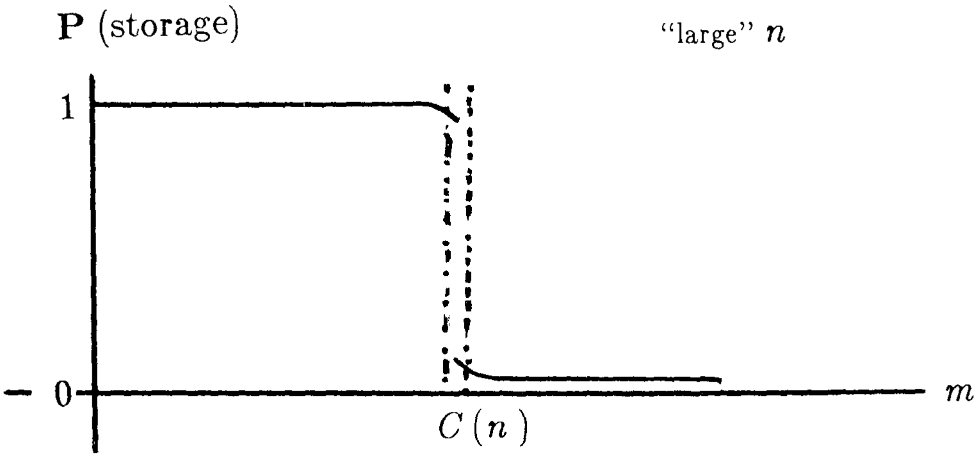

Fig. 5.5 (b). Capacity of neural networks.

lower sequence of capacities, it follows that the neural network specified by algorithm X stores m fundamental memories with probability approaching one as $n \to \infty$.
(b) The proof is similar. □

We now combine the two definitions to give us a sharper definition of capacity.

Definition. A sequence of integers $\{C(n)\}_{n=1}^{\infty}$ is a *sequence of capacities* for algorithm X iff it is both a lower sequence, and an upper sequence of capacities for algorithm X, i.e., $C(n) = \underline{C}(n) = \overline{C}(n)$.

The situation is schematically illustrated in fig. 5.5 (b). Here, the region between $\underline{C}(n)$ and $\overline{C}(n)$ in fig. 5.5 (a) is eliminated as $\underline{C}(n)$ and $\overline{C}(n)$ coincide. The situation is rather reminiscent of sphere hardening.

As in proposition (5.2.1), the following result demonstrates that if sequences of capacity do exist, then they are not very different from each other.

Proposition 5.2.2. If $\{C(n)\}$ is a sequence of capacities then so is $\{[1 \pm o(1)]C(n)\}$. Conversely, if $\{C(n)\}$ and $\{C(n)^*\}$ are any two sequences of capacities, then $C(n)^* \sim C(n)$ as $n \to \infty$.

Proof. Let $\{C(n)\}$ be a sequence of capacities. Let p denote the probability that the neural network specified by algorithm X stores m fundamental memories. Now, for every $\lambda \in (0,1)$ we can find $\lambda^* \in (0,1)$ such that for large enough n, $(1 \pm o(1))(1 - \lambda) \leq (1 - \lambda^*)$. Fix $\lambda \in (0,1)$, and for n large, choose $m \leq (1 - \lambda)(1 \pm o(1))C(n) \leq (1 - \lambda^*)C(n)$. As $\{C(n)\}$ is also a lower sequence, we have that $p \to 1$ as $n \to \infty$. $\{[1 \pm o(1)]C(n)\}$ is hence a lower sequence of capacities. In similar fashion it can be shown that $\{[1 \pm o(1)]C(n)\}$ is an upper sequence of capacities. Hence $\{[1 \pm o(1)]C(n)\}$ is also a sequence of capacities for algorithm X.

To prove the converse, let $\{C(n)\}$ and $\{C(n)^*\}$ be any two sequences of capacities. Without loss of generality, let $C(n)^* = [1 + \alpha_n]C(n)$. We must prove that $\alpha_n = \pm o(1)$. Fix $\lambda,\lambda^* \in (0,1)$. For $m \leq (1-\lambda)C(n)^* = (1-\lambda)(1+\alpha_n)C(n)$, we have $p \to 1$ as $n \to \infty$. Further, for $m \geq (1+\lambda^*)C(n)$, we have $p \to 0$ as $n \to \infty$. Hence, for *every* choice of scalars $\lambda,\lambda^* \in (0,1)$, we require that $(1-\lambda)(1+\alpha_n) < (1+\lambda^*)$ for large enough n; i.e., for every fixed choice of $\lambda,\lambda^* \in (0,1)$, we require that $(1+\alpha_n) < \frac{(1+\lambda^*)}{(1-\lambda)}$ for large enough n. It hence follows that $|\alpha_n| = o(1)$. □

The propositions establish that if sequences of capacity do exist, then: (1) they are not unique, and (2) they do not differ significantly from each other. In light of the above result, we define an equivalence class of sequences of capacities $[\{C(n)\}]$ with equivalence relation defined as follows: if $\{C(n)\}$ and $\{C(n)^*\}$ are members of this equivalence class of capacities, then they must satisfy the equivalence relation $C(n)^* \sim C(n)$. Henceforth, if a sequence of capacities $\{C(n)\}$ exists, then we shall say without elaboration that $C(n)$ is the *capacity* of algorithm X; by this we mean that $\{C(n)\}$ is a member of the equivalence class $[\{C(n)\}]$ of sequences of capacities.

The above definitions of capacity can be considered to be *strong sense* definitions as we require convergence of the requisite event–that *all* the prescribed memories be stored as fixed points–with probability one. This constraint is reasonably strong. An alternative scenario of interest could be where we require that *almost all* (but not necessarily all) of the memories be stored as fixed points. The situation here corresponds to it being permissible to "forget" a few memories as long as most of them are retained. This approach to memory storage leads to *weak sense* capacity definitions.

The weak sense capacity definitions that we utilise mirror the strong sense definitions above, except that we substitute on-average or expected behaviour for the probability one behaviour of the strong sense capacity definitions. In particular, for the definition of lower sequences of capacity (weak sense), we only require that the *expected number* of memories that are fixed points be $m - o(m)$ for m less than capacity instead of requiring that all the fundamental memories be fixed points with probability approaching one for large n. Similarly, for the definition of upper sequences of capacity (weak sense), we only require that the *expected number* of stored memories be $o(m)$ for m larger than capacity. Sequences of capacity (weak sense) are then defined as before as being simultaneously upper and lower sequences. The constraints imposed by the weak sense capacity definitions are not as stringent as those encountered for the strong sense definitions, so that we can expect capacities (weak sense) to be larger than capacities (strong sense).

For the zero error tolerance, associative structure that we are considering in this section, a necessary condition is that the fundamental memories themselves be fixed points. Hence, our definitions thus far have been geared toward characterising the capacities of algorithms for fixed point storage. The storage of fixed points, as noted earlier, is a static property of the system, and is independent of any particular mode of operation. For any specified mode of operation (synchronous one step, multiple synchronous steps, or asynchronous, for instance) the capacity definitions can be easily modified if we now wish to use the dynamics of the system to achieve error correction or attraction in addition to storing the memories as fixed points. The definitions of lower and upper sequences of capacities (either weak or strong sense) are modified by simply replacing the requirement that the fundamental memories be fixed points by the requirement that under the specified mode of operation the fundamental memories be attractors over the specified radius of attraction. The two propositions will continue to hold *in toto* for these modified definitions.

References

[1] G. E. Hinton and J. A. Anderson (eds.), *Parallel Models of Associative Memory.* Hillsdale, New Jersey: Lawrence-Erlbaum, 1981.

[2] F. Tanaka and S. P. Edwards, "Analytical theory of the ground state properties of a spin glass: I. Ising spin glass," *Jnl. Phys. F. Metal Phys.*, vol. 10, pp. 2769-2778, 1980.

[3] S. Wolfram, "Statistical mechanics of cellular automata," *Rev. Mod. Phys.*, vol. 55, pp. 601–644, 1983.

[4] D. O. Hebb, *The Organization of Behavior.* New York: Wiley, 1949.

[5] J. G. Eccles, *The Neurophysiological Basis of Mind.* Clarendon: Oxford, 1953.

[6] K. Nakano, "Associatron–a model of associative memory," *IEEE Trans. Sys., Man, and Cybern.*, vol. SMC-2, pp. 380–388, 1972.

[7] T. Kohonen, *Associative Memory: A System-Theoretic Approach.* Berlin: Springer-Verlag, 1977.

[8] S. Amari, "Neural theory of association and concept formation," *Biol. Cybern.*, vol. 26, pp. 175–185, 1977.

[9] W. A. Little, "The existence of persistent states in the brain," *Math. Biosci.*, vol. 19, pp. 101–120, 1974.

[10] G. Palm, "On associative memory," *Biol. Cybern.*, vol. 36, pp. 19–31, 1980.

[11] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Natl. Acad. Sci. USA*, vol. 79, pp. 2554–2558, 1982.

[12] J. J. Hopfield and D. W. Tank, "'Neural' computation of decisions in optimization problems," *Biol. Cybern.*, vol. XX, 1985, in press.

[13] D. W. Tank and J. J. Hopfield, "Simple optimization networks: an A/D converter and a linear programming circuit," submitted to *Biol. Cybern.*

[14] R. J. McEliece and E. C. Posner, "The number of stable points of an infinite-range spin glass memory," *Telecommunications and Data Acquisition Progress Report*, vol.

42–83, July-Sep. 1985, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California.

CHAPTER VI

OUTER PRODUCT NETWORKS

1. THE ENCODING ALGORITHM

With the nature of the threshold decision rule fixed, the only flexibility we have in storing information is in the specification of the neural interconnection weights $[w_{ij}]$. In this chapter we consider the quantitative behaviour of an algorithm for memory encoding which specifies the interconnection matrix according to an outer product formalism. We obtain precise estimates of the capacity of the outer product algorithm for storing memories. The capacity results hinge on two key lemmas which we prove in section 5. Alternate proofs for these results can also be formulated [1].

A. The Interconnection Matrix

Let $\mathbf{u}^{(\alpha)}$, $\alpha=1,\ldots,m$, be m (binary) state vectors of an n-neuron network that we wish to store as fundamental memories within the neural network. (In this chapter we consider specifically the case of auto-association; the extension of the analysis to hetero-associative storage is simple.) For each memory $\mathbf{u}^{(\alpha)}$ we form the $n \times n$ matrix

$$\mathbf{W}^{(\alpha)} = \mathbf{u}^{(\alpha)}\mathbf{u}^{(\alpha)^T} - I_n \ ,$$

where the superscript T denotes a transpose to a row vector, and I_n is the identity matrix. (For most of our results, we can subtract gI_n, where $0 \leq g \leq 1$.) Thus, $\mathbf{W}^{(\alpha)}$ is just the outer product of $\mathbf{u}^{(\alpha)}$ with itself, with the proviso that zeroes are placed on the diagonal. In the outer product construction the matrix of interconnection weights $\mathbf{W} = [w_{ij}]$ is formed as the sum of the outer product matrices $\mathbf{W}^{(\alpha)}$:

$$\mathbf{W} = \sum_{\alpha=1}^{m} \mathbf{W}^{(\alpha)} .$$

The directed interconnection strength linking neuron j to neuron i is hence given by

$$w_{ij} = \sum_{\alpha=1}^{m} u_i^{(\alpha)} u_j^{(\alpha)} - gm\,\delta_{ij} , \tag{6.1.1}$$

where $0 \leq g \leq 1$. ($g = 0$ implies that $w_{ii} = m$ for $i = 1,\ldots,n$, so that there is self-reinforcement for each neuron; $g = 1$ implies that $w_{ii} = 0$ so that we have a zero-diagonal weight matrix, and the neurons do not self link.) Thus, $\mathbf{W} = [w_{ij}]$ is a symmetric matrix of weights with a constant value along the diagonal of $(1 - g)m$.

The above algorithm for memory encoding is based on the sum of the outer products of the desired memories. The m vectors $\mathbf{u}^{(\alpha)}$ are what we refer to as the *fundamental memories.* An important point to keep in mind is that we assume that once the fundamental memories are specified, the parameters of the interconnection matrix are fixed; i.e., once $\mathbf{W}$ has been calculated, no other information about the chosen fundamental memories $\mathbf{u}^{(\alpha)}$ is available to the network. This is important when we wish to add memories to the list of things to be remembered, that is, when *learning* becomes an issue.

Information *retrieval* works as follows. Starting with an n dimensional ±1 vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$, which we call the *probe,* as our initial state, we require that the system dynamics flow in such a fashion as to terminate in the fundamental memory $\mathbf{u}^{(\alpha)}$ closest in Hamming distance to $\mathbf{x}$, provided that $\mathbf{x}$ is within the requisite error correction range of $\mathbf{u}^{(\alpha)}$. As before, we specify the Hamming distance as the natural similarity metric in the binary space we consider. The operation we require is a nearest neighbour search in Hamming space, hopefully terminating on the nearest designated fundamental memory. In all this, the system dynamics of state transitions is dictated purely by the choice of the connection matrix **W** according to the outer product algorithm (6.1.1), and the neural network iteration rules according to a threshold decision rule (threshold zero), and using either synchronous or asynchronous modes of operation.

The outer product scheme has been oft-proposed, and used in the literature. Hopfield [2] investigated a model with asynchronous dynamics, and demonstrated that the flow in state space was such as to minimise a bounded "energy" functional, and that associative recall of chosen memories was hence feasible with a measure of error correction. Nakano [3] coined the term "Associatron" for the technique, and demonstrated that with synchronous dynamics, a time-sequence of associations, with some ability for recall, and error correction could be obtained. The conditions under which long-term correlations can exist in memory have been investigated by Little [4], and Little and Shaw [5] utilising a synchronous model.

There are several seductive aspects of the algorithm that make it attractive. The algorithm is easily specified, and its implementation is relatively simple. Furthermore, additional memories can be tacked on by a simple incremental modification of the weights. In addition, the algorithm is robust and fault tolerant. These, and other aspects of the algorithm have been investigated in the references quoted above. The issue of the capacity of the algorithm, however, has remained an open question until recently [1]. The capacity as an objective measure of performance is a crucial parameter in determining the efficacy of the algorithm in memory storage, especially when large networks are to be built. We devote most of this chapter to providing rigourous answers to the question: What is the storage capacity of the outer product algorithm?

B. Memory Stability

We first sketch a plausibility argument to demonstrate that the memories are stable (at least in a probabilistic sense). Assume that one of the memories $\mathbf{u}^{(\alpha)}$ is the initial state of the neural network. For each $i=1,\ldots,n$, we have

$$\left(\mathbf{W}\mathbf{u}^{(\alpha)}\right)_i = \sum_{j=1}^{n} w_{ij}\, u_j^{(\alpha)}$$

$$= \sum_{\substack{j=1 \\ j\neq i}}^{n} \sum_{\beta=1}^{m} u_i^{(\beta)} u_j^{(\beta)} u_j^{(\alpha)}$$

$$= [n - 1 + (1-g)m]\, u_i^{(\alpha)} + \sum_{\beta\neq\alpha} \sum_{j\neq i} u_i^{(\beta)} u_j^{(\beta)} u_j^{(\alpha)} . \tag{6.1.2}$$

The algorithm can be viewed as a modification of a template matching (matched filtering) scheme for pattern classification. Here the probe (or unknown pattern) is matched against each of the m pattern classes (fundamental memories), and the resultant correlations are put together. Heuristically, we would expect the "signal" peak resulting from a proper match to dominate the sum of the "noisy" cross-correlations peaks resulting from improper matches if the number of memories, m , is small compared to the size, n , of the signal peak.

To quantify the above signal-to-noise ratio argument, assume for simplicity that $g = 1$. Now assuming that the memories are generated as a sequence of mn Bernoulli trials, we find that the second term (the double sum) of equation (6.1.2) has zero mean, and variance equal to $(n-1)(m-1)$, while the first term is simply $(n-1)$ times the sign of $u_i^{(\alpha)}$. The second term in equation (6.1.2) is comprised of a sum of independent random variables taking on values ± 1; it is hence asymptotically normal by the de Moivre-Laplace limit theorem. We hence have that the component $u_i^{(\alpha)}$ will be stable only if the mean to standard deviation given by $\frac{(n-1)^{1/2}}{(m-1)^{1/2}}$ is large. Thus, as long as the storage capacity of the system is not overloaded, in a way to be made precise, we expect the memories to be stable in some probabilistic sense. Note that the simple argument used above seems to require that $m = o(n)$. The outer product algorithm hence behaves well with regard to stability of the memories provided that the number of memories m is small enough compared to n the number of neurons in the system (or alternatively, the number of components in the memory vectors).

C. Examples

We illustrate the behaviour of the algorithm with a simple example. We consider a system of five neurons, and specify three fundamental memories as follows.

$$\mathbf{u}^{(1)} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad , \quad \mathbf{u}^{(2)} = \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \\ -1 \end{bmatrix} \quad , \quad \mathbf{u}^{(3)} = \begin{bmatrix} -1 \\ 1 \\ -1 \\ -1 \\ -1 \end{bmatrix} .$$

The 5 × 5 interconnection matrix of weights is then

$$\mathbf{W} = \begin{bmatrix} 0 & -1 & 1 & 3 & 1 \\ -1 & 0 & 1 & -1 & 1 \\ 1 & 1 & 0 & 1 & 3 \\ 3 & -1 & 1 & 0 & 1 \\ 1 & 1 & 3 & 1 & 0 \end{bmatrix} .$$

The matrix is, as expected, symmetric, zero diagonal, and requires an integral dynamic range of -1 to 3. (For the general outer product matrix of weights, the dynamic range requirement on the components is between $-m$ to m, where m is the number of memories.)

It is easy to verify in this instance that the three fundamental memories of choice are indeed fixed.

$$\mathbf{Wu}^{(1)} = (\,4 \;\; 0 \;\; 6 \;\; 4 \;\; 6)^T \;, \qquad \text{sgn } \mathbf{Wu}^{(1)} = (\,1 \;\; 1 \;\; 1 \;\; 1 \;\; 1)^T \; = \mathbf{u}^{(1)} \,,$$

$$\mathbf{Wu}^{(2)} = (\,2 \; -3 \; -2 \;\; 2 \; -2)^T \;, \qquad \text{sgn } \mathbf{Wu}^{(1)} = (\,1 \; -1 \; -1 \;\; 1 \; -1)^T \; = \mathbf{u}^{(2)} \,,$$

$$\mathbf{Wu}^{(3)} = (-6 \;\; 0 \; -4 \; -6 \; -4)^T \;, \qquad \text{sgn } \mathbf{Wu}^{(1)} = (-1 \;\; 1 \; -1 \; -1 \; -1)^T \; = \mathbf{u}^{(3)} \,.$$

It is difficult, however, to adduce consistent attraction behaviour from the present example because of the small size of n and m. Small sample behaviour can be expected to be quite prominent for this case, resulting in fluctuating behaviour in the attraction dynamics. These effects are smoothed out for large values of n, and with m lying within the capacity results to be derived.

2. ATTRACTION DYNAMICS

The outer product algorithm, as seen in the last section, clearly has the potential to store memories–at least in the weak sense that prescribed fundamental memories can be made fixed points of the system. If the algorithm is to be useful as an associative memory structure, however, we require the fundamental memories to be not merely fixed points, but to exhibit attraction, so that degraded or noisy memories can still be recognised. That the prescribed outer product algorithm exhibits such a desired error correcting capability can be demonstrated by constructing Lyapunov functions for the system. The precise argument changes somewhat depending on whether the synchronous or the asynchronous mode is in force. We hence consider the two cases separately.

A. Asynchronous Mode

We assume that state transitions are occasioned by a single randomly chosen neuron changing state at any given time, so that two states seen contiguously in time by the system can differ in at most one bit component. The mode of analysis is patterned after that of Hopfield [2], and presupposes that the fundamental memories are indeed fixed points, i.e., $\mathbf{u}^{(\alpha)} \mapsto \mathbf{u}^{(\alpha)}$ for each (or perhaps "almost all") of the fundamental memories $\alpha = 1,\ldots,m$. The essential result can be stated as follows:

Proposition 6.2.1. For any symmetric neural interconnection matrix $\mathbf{W}$ with non-negative diagonal elements, the asynchronous mode of operation always results in a fixed point, whatever the initial state of the system.

Proof. For every state $\mathbf{u}$ in the state space $\mathbb{B}^n$ define the quadratic form (an "energy functional")

$$E(\mathbf{u}) = -\frac{1}{2}\left\langle \mathbf{u}\,,\, \mathbf{W}\mathbf{u} \right\rangle$$

$$= -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\, u_i\, u_j \ .$$

The proof essentially consists of the demonstration that the energy functional E is non-increasing as $\mathbf{u}$ goes through a model trajectory. Assume that $\mathbf{u}$ represents the current state of the system, and that neuron i_0 updates its state next. (By assumption of asynchronous operation, all other neurons hold their current state fixed.) The proposition holds trivially if neuron i_0 retains the same state u_{i_0} after updating, i.e., if the potential $\sum_{j=1}^{n} w_{i_0 j}\, u_j$ seen by neuron i_0 has the same sign as its current state u_{i_0}; in this case the energy functional undergoes no change. Let us assume wlog then that neuron i_0 actually changes state on updating, so that component i_0 of the current probe vector $\mathbf{u}$ changes state to $-u_{i_0}$. Let $\mathbf{u}^*$ represent the new state vector. It follows that $u_i^* = u_i$ for $i \neq i_0$, and $u_{i_0}^* = -u_{i_0} = \mathrm{sgn}\left(\sum_{j=1}^{n} w_{i_0 j}\, u_j\right)$. The change in energy is given by

$$\Delta E = E(\mathbf{u}^*) - E(\mathbf{u}) = -\frac{1}{2}\left\{\sum_{j=1}^{n} w_{i_0 j}(\Delta u_{i_0})u_j + \sum_{i=1}^{n} w_{i i_0} u_i (\Delta u_{i_0}) + w_{i_0 i_0}(\Delta u_{i_0})^2\right\} ,$$

where $\Delta u_{i_0} = u_{i_0}^* - u_{i_0}$. (Note that it is not requisite that the diagonal elements w_{ii} be zero as in the original formulation of the outer product interconnection matrix; it suffices that the diagonal elements be non-negative, $w_{ii} \geq 0$.)

Using the symmetry of the matrix $\mathbf{W}$, and the fact $w_{i_0 i_0} \geq 0$, we have

$$\Delta E \leq - \Delta u_{i_0} \left\{ \sum_{j=1}^{n} w_{i_0 j} u_j \right\} .$$

We now exhaust the two possible cases. Assume $u_{i_0} = 1$. Then $u_{i_0}^* = -1$, and the potential seen by neuron i_0 must satisfy $\sum_{j=1}^{n} w_{i_0 j} u_j < 0$. Hence $\Delta E \leq \sum_{j=1}^{n} w_{i_0 j} u_j < 0$.

Now assume $u_{i_0} = -1$. Then $u_{i_0}^* = 1$, and the potential seen by neuron i_0 must satisfy $\sum_{j=1}^{n} w_{i_0 j} u_j \geq 0$. Hence $\Delta E \leq - \sum_{j=1}^{n} w_{i_0 j} u_j \leq 0$.

Thus in all cases the quadratic form E is non-increasing under asynchronous component changes. Now E is a continuous function on the compact (actually finite) set $\mathbb{B}^n$. Hence a finite minimum of E is reached on each trajectory. A state which is such an energy minimum does not necessarily have to be a fixed point, as it is possible that changes in sign $u_{i_0} = -1 \mapsto u_{i_0}^* = 1$ of a single neuron may result in no net change in energy, $\Delta E = 0$. But the only such changes involve a change from -1 to 1, so that after a finite number (at most n) of such changes, no more changes are possible. A fixed point is hence ultimately reached in the asynchronous case. □

Thus as long as the fundamental memories are fixed points, the gradient dynamics exhibited by the asynchronous algorithm indicate that we may expect to find a region of attraction around each fundamental memory if the operation is in the asynchronous mode.

B. Synchronous Mode

As seen before, the fixed points are the same for both asynchronous and synchronous procedures. Attraction behaviour for the synchronous case is slightly more complicated due to the possibility of limit cycling, with the result that fixed points need not always be reached. However, we can adduce an argument showing that the expected attraction flow is toward the closest fixed point.

Let $\mathbf{u}^{(\alpha)}$ represent a fundamental memory which is a fixed point. We have that

$$\mathbf{W}\mathbf{u}^{(\alpha)} = (n-1)\mathbf{u}^{(\alpha)} + \boldsymbol{\eta} \ ,$$

where η is a noise vector whose components have zero mean and standard deviation $\sqrt{(m-1)(n-1)} = o(1)$. The fundamental memories $\mathbf{u}^{(\alpha)}$, $\alpha = 1,\ldots,m$, (all assumed to be fixed points for simplicity) are hence *approximate eigenvectors* of $\mathbf{W}$ with the same *approximate eigenvalue* $(n-1)$. (This statement holds at least in a statistical sense: the memories are *eigenvectors-in-mean* of the linear map $\mathbf{W}$, but more on this later.) The spectrum of $\mathbf{W}$ is hence almost degenerate with the m fundamental memories all having roughly the same eigenvalue $(n-1)$. We now demonstrate that the maximum eigenvalue of $\mathbf{W}$ is essentially n, with its only other eigenvalue being $-m$; i.e., it is $(n-m)$ fold degenerate.

Consider any vector $\mathbf{x} \in \mathbb{R}^n$ in the space orthogonal to that spanned by the m memories. We have

$$\begin{aligned}[\mathbf{W}\mathbf{x}]_i &= \sum_{j=1}^{n} w_{ij}\, x_j = \sum_{\substack{j=1\\ j\neq i}} \sum_{\alpha=1}^{m} u_i^{(\alpha)} u_j^{(\alpha)} x_j \\ &= \sum_{\alpha=1}^{m} u_i^{(\alpha)} \sum_{j=1}^{n} u_j^{(\alpha)} x_j - \sum_{\alpha=1}^{m} u_i^{(\alpha)} u_i^{(\alpha)} x_i \\ &= -m x_i \ .\end{aligned}$$

Thus all vectors orthogonal to the fundamental memories are eigenvectors of $\mathbf{W}$ with degenerate eigenvalue $-m$.

Now, let $\mathbf{x} = \mathbf{u}^{(\alpha)} + \delta\mathbf{x}$ be a state vector such that $||\delta\mathbf{x}|| << ||\mathbf{u}^{(\alpha)}|| = \sqrt{n}$; i.e., $\mathbf{x}$ is close to $\mathbf{u}^{(\alpha)}$ in Hamming distance. We have $\mathbf{W}\mathbf{x} = \mathbf{W}\mathbf{u}^{(\alpha)} + \mathbf{W}\delta\mathbf{x} = (n-1)\mathbf{u}^{(\alpha)} + \boldsymbol{\eta} + \mathbf{W}\delta\mathbf{x}$. As $\mathbf{W}$ is a linear transformation, $\exists k$ such that

$||\mathbf{W}\delta\mathbf{x}|| \leq k \ ||\delta\mathbf{x}||$. The smallest such constant k is essentially n as $m = o(n)$, and the largest eigenvalue of $\mathbf{W}$ is approximately n. Hence, $||\mathbf{W}\delta\mathbf{x}|| \lesssim n \ ||\delta\mathbf{x}||$. Now $\mathbf{W}\mathbf{x}$ will approach $\mathbf{u}^{(\alpha)}$ if the contribution of the "noisy" term $\boldsymbol{\eta} + \mathbf{W}\delta\mathbf{x}$ is small compared to the "signal" term $(n - 1)\mathbf{u}^{(\alpha)}$. Define the signal-to-noise ratio, *SNR*, as the ratio of the Euclidean norms of the signal and the noise terms. Using the triangle inequality, we have that

$$SNR \geq \frac{(n - 1)||\mathbf{u}^{(\alpha)}||}{||\boldsymbol{\eta}|| + ||\mathbf{W}\delta\mathbf{x}||} .$$

Let d denote the Hamming distance between $\mathbf{x}$ and $\mathbf{u}^{(\alpha)}$ ($2\sqrt{d} << \sqrt{n}$ by assumption). Then $||\mathbf{W}\delta\mathbf{x}|| \leq n \ ||\delta\mathbf{x}|| = 2n\sqrt{d}$. Also, $||\mathbf{u}^{(\alpha)}|| = \sqrt{n}$. Finally, using the standard deviation of each component of the noise term $\boldsymbol{\eta}$ to estimate its norm, we have, $SNR \geq \frac{n\sqrt{n}}{n\sqrt{m} + 2n\sqrt{d}} = \frac{\sqrt{n}}{\sqrt{m} + 2\sqrt{d}}$. For small distances d, and for $m = o(n)$, we obtain a high signal-to-noise ratio. Consequently, the signal term tends to dominate the noisy perturbation terms, so that $\mathbf{u}^{(\alpha)}$ excercises a domain of attraction.

The above argument indicating the existence of basins of attraction around the fundamental memories used a quadratic energy functional. Recently, other Lyapunov functions based on the Manhattan norm and the max norm have been shown to exist for synchronous neural dynamics [6], thus validating the fact that gradient dynamics continue to hold for the synchronous case too.

The above implies that there is a domain or basin of attraction around each fundamental memory, with high probability. That is, most probe vectors lying in specified Hamming spheres surrounding the fundamental memories will reach the fundamental memory at the centre of the sphere as a stable point, in both the asynchronous and synchronous models, if there are not too many fundamental memories. This is not too surprising in retrospect; wrong components merely add to the noise, and we do not really expect them to significantly affect the qualitative behaviour of the algorithm. We will rigourise these observations in sections 4 and 5.

3. CAPACITY HEURISTICS

Before going to the formal proofs, we present two simplified heuristic derivations of capacity. The two procedures utilise rather different, but appealingly simple hypotheses: the first derivation assumes a jointly Gaussian spread of the potentials evinced at each neuron, while the second derivation assumes that the distribution of erroneous decisions is Poisson. It turns out from the rigourous analysis that while both simplifying assumptions lead to the correct answer, the second conjecture is nearer the mark.

A. A Gaussian Conjecture

Let $\mathbf{u}^{(\alpha)}$, $\alpha = 1,...,m$, be the m fundamental memories to be stored, and let the weights w_{ij} be given by equation (6.1.1). If each of the fundamental memories is to be a fixed point, we require that with high probability,

$$u_i^{(\alpha)} = \text{sgn}\ \left(\sum_{j=1}^{n} w_{ij}\, u_j^{(\alpha)} \right) \ , \ \text{for}\ i = 1,...,n \ \ , \ \alpha = 1,...,m \ . \tag{6.3.1}$$

Define the random variables $\{X_i^{(\alpha)}\}_{i=1,\alpha=1}^{n\ \ m}$ by

$$X_i^{(\alpha)} \triangleq u_i^{(\alpha)} \sum_{\substack{j=1 \\ j\neq i}}^{n} w_{ij}\, u_j^{(\alpha)} = (n-1) + \sum_{j\neq i} \sum_{s\neq r} u_i^{(\beta)} u_j^{(\beta)} u_i^{(\alpha)} u_j^{(\alpha)} \ . \tag{6.3.2}$$

Then

$$E\,(X_i^{(\alpha)}) = n-1 \triangleq \mu \ ,$$

and

$$E\left\{ (X_{i_1}^{(\alpha_1)} - \mu)(X_{i_2}^{(\alpha_2)} - \mu) \right\} = \begin{cases} 1 & ;\ \alpha_1 \neq \alpha_2 \ ,\ i_1 \neq i_2 \\ m-1 & ;\ \alpha_1 = \alpha_2 \ ,\ i_1 \neq i_2 \\ n-1 & ;\ \alpha_1 \neq \alpha_2 \ ,\ i_1 = i_2 \\ (m-1)(n-1) & ;\ \alpha_1 = \alpha_2 \ ,\ i_1 = i_2 \end{cases}$$

The random variables $X_i^{(\alpha)}$ are hence all pairwise correlated. Considering the $(m \times n)$ matrix of random variables

$$\begin{bmatrix} X_1^{(1)} & X_2^{(1)} & \cdots & X_n^{(1)} \\ X_1^{(2)} & X_2^{(2)} & \cdots & X_n^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ X_1^{(m)} & X_2^{(m)} & \cdots & X_n^{(m)} \end{bmatrix},$$

we see that each random variable $X_i^{(\alpha)}$ has a strong correlation with random variables $X_i^{(\beta)}$ on the same column (correlation coefficient $\frac{1}{m-1}$), a slightly weaker correlation with random variables $X_j^{(\alpha)}$ on the same row (correlation coefficient $\frac{1}{n-1}$), and weakest correlation with all other off-row, off-column random variables $X_j^{(\beta)}$ (correlation coefficient $\frac{1}{(m-1)(n-1)}$).

The requirement that each of the memories $\mathbf{u}^{(\alpha)}$ is a fixed point implies that for each $\alpha = 1,\ldots,m$, and $i = 1,\ldots,n$, the random variables $X_i^{(\alpha)}$ satisfy sgn $X_i^{(\alpha)} = 1$, i.e., $X_i^{(\alpha)} \geq 0$. It is easily seen that each random variable $X_i^{(\alpha)}$ individually exhibits central tendency as a consequence of the DeMoivre-Laplace Limit Theorem. It is hence tempting to conjecture that *all* the random variables $X_i^{(\alpha)}$ are jointly Gaussian–perhaps in some asymptotic sense. We will adopt this Gaussian hypothesis in the following.

Conjecture: For n large enough, the random variables $[X_i^{(\alpha)}]$ are jointly normal, with second order statistics given by equation (6.3.3).

Now let $\{Y_i^{(\alpha)}\}_{i=0,\alpha=0}^{n\ \ m}$ be an i.i.d. set of Gaussian random variables with zero mean, and unit variance. We construct the normal random variables $\{Z_i^{(\alpha)}\}_{i=1,\alpha=1}^{n\ \ m}$ as follows:

$$Z_i^{(\alpha)} \triangleq (n-1) - Y_0^{(0)} - (m-2)^{1/2} Y_0^{(\alpha)} - (n-2)^{1/2} Y_i^{(0)} + [(m-2)(n-2)]^{1/2} Y_i^{(\alpha)}.$$

Asymptotically the Gaussian random variables $Z_i^{(\alpha)}$ have the same statistics as the random variables $X_i^{(\alpha)}$ by the Gaussian hypothesis. Define

$$f_n(\alpha,\beta,\gamma) \triangleq \prod_{r=1}^{m} \Phi\left(\frac{(n-1)-\gamma-\beta_r(m-2)^{1/2}-\alpha(n-2)^{1/2}}{[(m-2)(n-2)]^{1/2}}\right)$$

and

$$I_n(\beta,\gamma) \triangleq (2\pi)^{-1/2}\int_{\alpha=-\infty}^{\infty} f_n(\alpha,\beta,\gamma)e^{-\frac{\alpha^2}{2}}\,d\alpha\,, \qquad (6.3.4)$$

where Φ is the cumulative Gaussian distribution function

$$\Phi(x) \triangleq (2\pi)^{-}1/2\int_{-\infty}^{x} e^{-\frac{\alpha^2}{2}}\,d\alpha \quad ,\forall\, x \in \mathbb{R}\,.$$

Let $P_s(n)$ denote the probability that each of the fundamental memories is a fixed point, i.e., equation (6.3.1) holds. Then, under the Gaussian hypothesis for the $X_i^{(\alpha)}$'s, we have

$$\mathbf{P}_S(n) = \mathbf{P}\left\{\, Z_i^{(\alpha)} \geq 0\,,\ i=1,\ldots,n\ ,\ \alpha=1,\ldots,m \,\right\}$$

$$= \mathbf{P}\left\{ Y_i^{(\alpha)} \geq -\left[\frac{(n-1)-Y_0^{(0)}-(n-2)^{½}Y_i^{(0)}-(m-2)^{½}Y_0^{(\alpha)}}{[(m-2)(n-2)]^{½}}\right],\ i=1,\ldots,n\ ,\ \alpha=1,\ldots,m\right\}.$$

The events

$$\left\{Y_i^{(\alpha)} \geq -\left[\frac{(n-1)-Y_0^{(0)}-(n-1)^{½}Y_i^{(0)}-(m-1)^{½}Y_0^{(\alpha)}}{[nm-2(n+m)+2)]^{½}}\right]\right\}$$

are conditionally independent given the random variables $Y_0^{(0)}$, $Y_0^{(\alpha)}$, and $Y_i^{(0)}$. Utilising the above result we can show with some manipulation of integrals that

$$P_S(n) = (2\pi)^{-\frac{m+1}{2}}\int_{\gamma=-\infty}^{\infty}\int_{\beta_m=-\infty}^{\infty}\cdots\int_{\beta_1=-\infty}^{\infty}[I_n(\boldsymbol{\beta},\gamma)]^n\, e^{-\frac{1}{2}(\sum_{r=1}^{m}\beta_r^2+\gamma^2)}\,d\boldsymbol{\beta}\,d\gamma\,,$$

(6.3.5)

where $I_n\,(\beta,\gamma)$ is as defined in equation (6.3.4).

Equation (6.3.5) is essentially the reduction of an $(nm + n + m + 1)$-dimensional Gaussian integral into an $(m + 2)$-dimensional integral. Asymptotic estimates for equation (6.3.5) can be found by partitioning the regions of integration with some care, and using the behaviour of the tails of the cumulative Gaussian distribution function Φ. It can then be shown that as $n \rightarrow \infty$,

$$P_S(n) \sim \left[1 - \left(\frac{m}{2\pi n} \right)^{1/2} e^{-\frac{n}{2m}} \right]^{mn} .$$

For $m \sim \dfrac{n}{4 \log n}$, we will have $P_S(n) \rightarrow 1$ as $n \rightarrow \infty$. This corresponds to a strong sense fixed point storage capacity. If we just require that *most* of the fundamental memories are fixed points, it turns out that the capacity estimate doubles.

B. A Poisson Conjecture

Consider the random variables $X_i^{(\alpha)}$ defined in equation (6.3.2). The i-th component of memory $\mathbf{u}^{(\alpha)}$ will be in error if the random variable $X_i^{(\alpha)}$ is negative. Now, the terms of the double sum constituting $X_i^{(\alpha)}$ can be shown to be i.i.d. ±1 random variables (proved as a special case of lemma (6.4.5)) so that, by the DeMoivre-Laplace Limit Theorem, the probability that a single component of one of the memories is in error is asymptotically given by

$$\mathbf{P}\left\{X_i^{(\alpha)} < 0\right\} \sim \Phi\left(- \frac{n^{1/2}}{m^{1/2}} \right) .$$

Thus, the *expected number of component errors* is asymptotically $mn\ \Phi\left(\dfrac{n^{1/2}}{m^{1/2}} \right)$.

Thus far the analysis is fairly rigourous. We now utilise the following hypothesis.

Conjecture: The distribution of errors is asymptotically Poisson.

It will turn out in lemma (6.5.2) that under suitable restrictions, this statement holds. Assuming it for the moment, we see that the probability that each of the fundamental memories is a fixed point will then be given asymptotically by the expression

$$P_S(n) \sim \exp\left\{-mn\,\Phi\left(\frac{n^{1/2}}{m^{1/2}}\right)\right\}.$$

We then obtain that $P_S(n) \to 1$ as $n \to \infty$ if the number of memories is restricted to $m \sim \frac{n}{4\log n}$. This is the same result that we came up with using the Gaussian hypothesis. If we now require only that most of the fundamental memories be fixed points, then we get a doubling in capacity as before.

4. PRELIMINARY LEMMAS

In this section we present technical results needed to rigourously estimate the capacity of the outer product scheme. Lemma (6.4.1) indicates a good uniform estimate for the probability that the sum of N independent (1,0) random variables takes on integral values whose deviation from the mean is at most $o(N^{3/4})$. The estimate is just the probability of the approximating normal over an interval of length one centred on the targeted deviation from the mean. A good uniform asymptotic expansion for the cumulative distribution of a sum of N independent ± 1 random variables, valid for the same large deviations as lemma (6.4.1), is demonstrated in lemma (6.4.2). The approximation is the usual normal distribution valid for *small* deviations. Lemma (6.4.3) is the strong form of the large deviation Central Limit Theorem. Lemma (6.4.4) is a special case of Bonferroni's Inequalities [7], but is proved here for completeness. Finally, a result on independence of products of symmetric ± 1 random variables is stated and proved in lemma (6.4.5).

Lemma 6.4.1. Let $\{X_j\}_{j=1}^{\infty}$ be an i.i.d. sequence of random variables drawn from a sequence of Bernoulli trials, with

$$X_j = \begin{cases} 1 \text{ , with probability } p \\ 0 \text{ , with probability } q = 1-p \text{ ,} \end{cases}$$

where $0 < p < 1$. Fix $N \in \mathbb{Z}^+$, and consider the sum

$$Y_N = \sum_{j=1}^{N} X_j \ .$$

As $N \to \infty$, let the integer k vary so that

$$| \, k - Np \, | < B(N) = \begin{cases} o(N^{2/3}) \text{ if } p \neq q \\ o(N^{3/4}) \text{ if } p = q = \frac{1}{2} \ . \end{cases} \tag{6.4.1}$$

Then

$$\mathbf{P}\left\{Y_N = k\right\} \sim \frac{1}{\sqrt{2\pi pqN}} \int_{k-Np-\frac{1}{2}}^{k-Np+\frac{1}{2}} \exp\left(\frac{-t^2}{2pqN}\right) dt \tag{6.4.2}$$

as $N \to \infty$, uniformly for all k satisfying (6.4.1).

Proof. cf. Ref. [7], chp. VII, Sec. 6.

Lemma 6.4.2. Under the hypothesis of lemma (6.4.1), if the real number x varies as $N \to \infty$ so that

$$| \, x - Np \, | < B(N) = \begin{cases} o(N^{2/3}) \text{ if } p \neq q \\ o(N^{3/4}) \text{ if } p = q = \frac{1}{2} \text{ ,} \end{cases}$$

then

$$\mathbf{P}\left\{Y_N \leq x\right\} \sim \frac{1}{\sqrt{2\pi pqN}} \int_{t=-\infty}^{x-Np} \exp\left(\frac{-t^2}{2pqN}\right) dt \ . \tag{6.4.3}$$

Proof. cf. Ref. [7] as for the previous lemma, together with prob. 14, pg. 195.

Lemma 6.4.3. If X_N is the sum of N i.i.d. random variables, each ±1 with probability 1/2, and $v = o(N^{3/4})$, then as $N \to \infty$,

$$\mathbf{P}\left\{X_N \leq v\right\} \sim \frac{1}{\sqrt{2\pi}} \int_{t=-\infty}^{v/\sqrt{N}} \exp\left(\frac{-t^2}{2}\right) dt = \Phi\left(\frac{v}{\sqrt{N}}\right) .$$

If in addition, $\frac{v}{\sqrt{Npq}} \to -\infty$ as $n \to \infty$, then

$$\mathbf{P}\left\{X_N \leq v\right\} \sim \frac{1}{\sqrt{2\pi}} \frac{\sqrt{N}}{v} e^{-\frac{v^2}{2N}} .$$

Proof. Lemma (6.4.2) applies here, with $Y_N = (X_N + N)/2$, $p = q = 1/2$, and $x = (v + N)/2$. Hence

$$\mathbf{P}\left\{S_N \leq v\right\} \sim \left(\frac{2}{\pi N}\right)^{½} \int_{t=-\infty}^{v/2} \exp\left(-\frac{2t^2}{N}\right) dt \ .$$

The rest of the lemma follows from the asymptotic formula for the error function. □

Lemma 6.4.4. Let $E_1,\ldots,E_N$ be measurable subsets of a probability space. For $1 \leq k \leq N$, let σ_k be the sum of the probabilities of all sets formed by intersecting k of the events $E_1,\ldots,E_N$:

$$\sigma_k = \sum_{j_1<j_2<\dots<j_k} \mathbf{P}\left\{\bigcap_{l=1}^{k} E_{j_l}\right\} .$$

Then for every K, with $1 \leq K \leq N$,

$$\mathbf{P}\left\{\bigcup_{j=1}^{N} E_j\right\} = \sum_{k=1}^{K} (-1)^{k-1} \sigma_k + (-1)^K a_K , \tag{6.4.4}$$

where $a_K \geq 0$.

Proof. Consider a point which lies in exactly L of the events E_j, $1 \leq L \leq N$. On the left, this point is counted only once. On the right, it is counted exactly $\binom{L}{k}$ times in each σ_k with $k \leq L$, for a total contribution of

$$\sum_{k=1}^{\min(K,L)} (-1)^{k-1}\binom{L}{k} = \begin{cases} 1-(1-1)^L = 1 \text{ , for } K \geq L \text{ ,} \\ 1-(-1)^K \binom{L-1}{K} \text{ , for } 1 \leq K < L \text{ .} \end{cases}$$

The latter equality is proved by induction on k using $\binom{L-1}{K-1} + \binom{L-1}{K} = \binom{L}{K}$, starting from $k = 1$, for which $L = 1-(-1)(L-1)$. Hence if we define the random variable X by

$$X = \begin{cases} 0 \text{ , } & \text{if } L \leq K \text{ ,} \\ \binom{L-1}{K} \text{ , } & \text{if } L > K \text{ ,} \end{cases}$$

then equation (6.4.4) holds with

$$a_K = \mathbf{E}(X) \geq 0 . \qquad \square$$

Lemma 6.4.5. Let K be a fixed, positive integer, and let $\{X_1, \dots, X_K\}$ be a set of i.i.d. random variables taking on values -1 and 1 only, each with probability half. Let $\{C_1, \ldots, C_K\}$ be any set of ± 1 random variables independent of the X_js. Then the random variables $\{C_j X_j\}_{j=1}^{K}$ are i.i.d., and take on values -1 and 1, each with probability half.

Proof. Let $Y_j = C_j X_j$, $j=1,\ldots,K$. Clearly the random variables Y_j take on values -1 and 1 only. Now

$$\mathbf{P}\{Y_j = 1\} = \mathbf{P}\{X_j = -1, C_j = -1\} + \mathbf{P}\{X_j = 1, C_j = 1\}$$

$$= \frac{1}{2}\left[\mathbf{P}\{C_j = -1\} + \mathbf{P}\{C_j = 1\}\right] = \frac{1}{2},$$

and

$$\mathbf{P}\{Y_j = -1\} = 1 - \mathbf{P}\{Y_j = 1\} = \frac{1}{2},$$

so that the Y_js are symmetric ± 1 random variables.

Let $y_1, \ldots, y_K \in \mathbb{B}$. Then

$$\mathbf{P}\{Y_j = y_j, j = 1,\ldots,K\}$$

$$= \mathbf{P}\{Y_1 = y_1 \mid Y_j = y_j, j = 2,\ldots,K\}\, \mathbf{P}\{Y_j = y_j, j = 2,\ldots,K\}$$

$$= \Big[\mathbf{P}\{X_1 = x_1 \mid C_1 = c_1, Y_j = y_j, j = 2,\ldots,K\}\, \mathbf{P}\{C_1 = c_1 \mid Y_j = y_j, j = 2,\ldots,K\}$$

$$+ \mathbf{P}\{X_1 = -x_1 \mid C_1 = -c_1, Y_j = y_j, j = 2,\ldots,K\}\, \mathbf{P}\{C_1 = -c_1 \mid Y_j = y_j, j = 2,\ldots,K\}\Big]$$

$$\times \mathbf{P}\{Y_j = y_j, j = 2,\ldots,K\},$$

where $c_1, x_1 \in \mathbb{B}$, and $c_1 x_1 = y_1$. Then we have

$$\mathbf{P}\left\{Y_j = y_j\,,\ j = 1,\ldots,K\right\}$$

$$= \frac{1}{2}\left[\mathbf{P}\left\{C_1 = c_1 \mid Y_j = y_j\,,\ j = 2,\ldots,K\right\} + \mathbf{P}\left\{C_1 = -c_1 \mid Y_j = y_j\,,\ j = 2,\ldots,K\right\}\right]$$

$$\times\, \mathbf{P}\left\{Y_j = y_j\,,\ j = 2,\ldots,K\right\}$$

$$= \frac{1}{2}\,\mathbf{P}\left\{Y_2 = y_2, \ldots, Y_K = y_K\right\} .$$

Proceeding by mathematical induction, we get $\mathbf{P}\left\{Y_1 = y_1, \ldots, Y_K = y_K\right\} = 2^{-K}$.

□

5. CAPACITY: A TALE OF TWO LEMMAS

Let $\mathbf{U} = \left\{\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(m)}\right\} \subseteq \mathbb{B}^n$ be the set of specified fundamental memories to be stored in the n-neuron network. The fundamental memories $\mathbf{u}^{(\alpha)} = (u_1^{(\alpha)}, \ldots, u_n^{(\alpha)})^T$ are assumed to have been independently drawn from a symmetric binomial distribution; specifically, the mn components $\left\{u_i^{(\alpha)}\right\}_{i=1,\alpha=1}^{n\ \ m}$ of the fundamental memories are i.i.d. random variables with

$$\mathbf{P}\left\{u_i^{(\alpha)} = 1\right\} = \mathbf{P}\left\{u_i^{(\alpha)} = -1\right\} = \frac{1}{2} .$$

The interconnection matrix $\mathbf{W} = [w_{ij}]$ is formed as the sum of outer products of the fundamental memories as before

$$w_{ij} = \sum_{\alpha=1}^{m} u_i^{(\alpha)} u_j^{(\alpha)} - gm\,\delta_{ij} ,$$

where $0 \leq g \leq 1$. The case $g = 1$ corresponds to the original outer product matrix construction, where the matrix is constrained to have zeroes along the diagonal so that there is no self-reinforcement at each neural site. Retaining a non-zero diagonal, however, does not materially affect the analysis.

Our approach will be to first obtain a rigourous estimate for the capacity of the outer product algorithm under the (fairly weak) condition that the fundamental memories are required to be just fixed points of the system. The analysis is then extended rigourously to estimate capacity for attraction over a specified radius in a single synchronous step. Capacities are then derived for multiple synchronous step and asynchronous error correction over specified radii of attraction. Finally, cases of attraction with error tolerance are considered.

A. Fixed Points

For the set of fundamental memories to be a set of fixed points of the system we require that for each $i=1,\dots,n$, and each $\alpha=1,\dots,m$,

$$u_i^{(\alpha)} = \text{sgn}\ \left(\sum_{j=1}^{n} w_{ij}\, u_j^{(\alpha)}\right) .$$

For large enough n, the probability that the term inside the sum is zero is very small. Hence, the probability that there is a row sum violation on the i-th component of the α-th fundamental memory $\mathbf{u}^{(\alpha)}$ is given by $\mathbf{P}\left\{\text{sgn}\ \left(\sum_{j=1}^{n} u_i^{(\alpha)} w_{ij}\, u_j^{(\alpha)}\right) = -1\right\} + o(1)$.

Form the sums

$$X_i^{(\alpha)} = \sum_{j=1}^{n} w_{ij}\, u_j^{(\alpha)} u_i^{(\alpha)}$$

$$= \sum_{j\neq i,} \sum_{\beta=1}^{m} u_i^{(\alpha)} u_j^{(\alpha)} u_i^{(\beta)} u_j^{(\beta)}$$

$$= n + (1-g)m - 1 + \sum_{j\neq i} \sum_{\beta\neq\alpha} u_i^{(\alpha)} u_j^{(\alpha)} u_i^{(\beta)} u_j^{(\beta)}$$

$$= n + (1-g)m - 1 + Z_i^{(\alpha)} , \qquad (6.5.1)$$

where we use the random variable $Z_i^{(\alpha)}$ to denote the double sum. For large enough n, it follows that **U** is a set of fixed points of the system if each of the random variables $X_i^{(\alpha)}$, $i=1,...,n$, $\alpha=1,...,m$, is positive. Now let τ represent the probability of a row sum violation on the ith component of the αth fundamental memory $\mathbf{u}^{(\alpha)}$; i.e., τ is the probability that the random variable $X_i^{(\alpha)}$ is negative. We have

$$\tau = \mathbf{P} \left\{ Z_i^{(\alpha)} < -(n + (1-g)m - 1) \right\} + O(e^{-n}) .$$

The following assertion is a result on the uniformity of the distribution of row sum violations. This remark will be useful in proving a sort of 0-1 law for storage capacity.

Proposition 6.5.1. Let τ and τ^* be the probabilities of row sum violations when the number of fundamental memories is m and m^*, respectively. Then $m^* > m \Longleftrightarrow \tau^* \geq \tau$.

Proof. The random variable $Z_i^{(\alpha)}$ has independent summands by virtue of lemma (6.4.5). Further, for larger m, $Z_i^{(\alpha)}$ has more independent summands for m^*, than for m if $m^* > m$, and hence is more likely to be large negative. In fact, the distribution of the number of row sum violations for m^* lies below that for m, so that more violations are likely to occur. The converse also follows from the uniformity of the binomial distribution. □

Lemma (6.5.1) below is a particular application of the Large Deviation Lemma to the situation we now face. The result is an asymptotic expression for τ, the probability that a particular row sum is violated. The result agrees with what would be obtained by a naive application of the Central Limit Theorem.

Lemma 6.5.1. As $n \to \infty$, if m satisfies:

(1) $m = o(n)$, and

(2) $m \geq M(n)$, where $\dfrac{M(n)}{\sqrt{n}} \to \infty$,

then

$$\tau \sim \frac{m^{\frac{1}{2}}}{(2\pi n)^{\frac{1}{2}}} \exp\left\{ - \left(\frac{n}{2m} + 1 - g \right) \right\} . \tag{6.5.2}$$

Proof. By lemma (6.4.5) the random variable $Z_i^{(\alpha)}$ in (6.5.1) has independent summands. Lemma (6.4.3) now applies to the random variable $Z_i^{(\alpha)}$, with

$$N = (m - 1)(n - 1) ,$$

$$v = -[n + (1-g)m - 1] .$$

The hypotheses of the lemma are satisfied as $m > M(n)$. Hence

$$\tau \sim \Phi\left(- \frac{n + (1-g)m - 1}{\sqrt{(m - 1)(n - 1)}} \right) .$$

Since

$$\frac{n + (1-g)m - 1}{\sqrt{(m - 1)(n - 1)}} \sim \left(\frac{n}{m} \right)^{\frac{1}{2}} \to \infty$$

as $n \to \infty$, we have by lemma (6.4.3) that

$$\tau \sim \frac{m^{\frac{1}{2}}}{(2\pi n)^{\frac{1}{2}}} \exp\left\{ - \frac{(n + (1-g)m - 1)^2}{2(n - 1)(m - 1)} \right\} .$$

Using $m/n \to 0$, and $m/\sqrt{n} \to \infty$, we have

$$\frac{(n+(1-g)m-1)^2}{2(n-1)(m-1)} = \frac{n}{2m} + 1 - g + o(1) .$$

Hence (6.5.2) follows. □

The next lemma concerns the joint distribution of q sums, $X_{i_h}^{(\alpha_h)}$, $h = 1,\dots,q$. This is the key result leading to the capacity, and demonstrates that asymptotically as $n \to \infty$, the number of row sum violations obeys a Poisson distribution.

Lemma 6.5.2. Let q be any fixed, positive integer, and let $\{(i_h, \alpha_h) \in \{1,\dots,n\} \times \{1,\dots,m\} : h = 1,\dots,q\}$ be q distinct pairs of integers. Then, under the hypothesis of lemma (6.5.1), if $M(n) = n^{\kappa}$ with $3/4 < \kappa < 1$, and as $n \to \infty$,

$$\mathbf{P}\left\{X_{i_h}^{(\alpha_h)} < 0 \,,\, h = 1,\dots,q\right\} \sim \tau^q . \tag{6.5.3}$$

Proof. Without loss of generality, assume i_h, $\alpha_h \le q$ for $h = 1,\dots,q$. Have

$$\begin{aligned} X_{i_h}^{(\alpha_h)} &= n + (1-g)m - 1 + \sum_{j \ne i_h} \sum_{\beta \ne \alpha_h} u_{i_h}^{(\alpha_h)} u_j^{(\alpha_h)} u_{i_h}^{(\beta)} u_j^{(\beta)} \\ &= n + (1-g)m - 1 + S_1^{(h)} + S_2^{(h)} \end{aligned} \tag{6.5.4}$$

where

$$S_2^{(h)} = \sum_{j > q} \sum_{\beta > q} u_{i_h}^{(\alpha_h)} u_j^{(\alpha_h)} u_{i_h}^{(\beta)} u_j^{(\beta)} \tag{6.5.5}$$

and $S_1^{(h)}$ is the sum of the remaining terms; ($S_1^{(h)}$ is the sum of $(n-1)(m-1) - (n-q)(m-q) = (q-1)(n+m-q-1)$ independent, ± 1 random variables.) Now, $\forall\, \epsilon \in (0\,,\,1/4)$,

$$\frac{n^{\frac{1}{2}+\epsilon}}{(q-1)^{3/4}(n+m-q-1)^{3/4}} \to 0 ,$$

and

$$\frac{n^{\frac{1}{2}+\epsilon}}{(q-1)^{1/2}(n+m-q-1)^{1/2}} \to 0 ,$$

as $n \to \infty$. Hence by lemma (6.4.3)

$$\mathbf{P}\left\{ | S_1^{(h)} | \geq n^{\frac{1}{2}+\epsilon} \right\} \sim 2\,\Phi\left(-\frac{n^{\frac{1}{2}+\epsilon}}{(q-1)^{1/2}(n+m-q-1)^{1/2}} \right)$$

$$= O\left(e^{-C_1 n^{2\epsilon}} \right) , \qquad (6.5.6)$$

where C_1 is a positive constant.

Let P_q be a partition of $\{1,...,q\}$ into $| P_q | \leq q$ disjoint subsets with each subset containing those $h \in \{1,...,q\}$ which identify the same memory α_h. Let $\theta_1 , \cdots , \theta_{| P_q |} \in \{1,...,q\}$ identify $| P_q |$ distinct memories. We identify the memories α_h by means of the function $\gamma_q(h) : \{1,...,q\} \to \{1,..., | P_q |\}$, where

$$\gamma_q(h_1) = \gamma_q(h_2) \Leftrightarrow \alpha_{h_1} = \alpha_{h_2} .$$

Specifically, $\alpha_h = \theta_{\gamma_q(h)}$.

Similarly, let Q_q be a partition of $\{1,...,q\}$ into $| Q_q | \leq q$ disjoint subsets, with each subset containing those $h \in \{1,...,q\}$ which identify the same neuron (component position) i_h. Let $\psi_1 , \cdots , \psi_{| Q_q |} \in \{1,...,q\}$ identify $| Q_q |$ distinct component positions. The component positions i_h can now be identified by means of the function $\delta_q : \{1,...,q\} \to \{1,..., | Q_q |\}$, where

$$\delta_q(h_1) = \delta_q(h_2) \Leftrightarrow i_{h_1} = i_{h_2} .$$

Specifically, $i_h = \psi_{\delta_q(h)}$.

Note that

$$\delta_q(h_1) = \delta_q(h_2) \Longrightarrow \gamma_q(h_1) \neq \gamma_q(h_2) ,$$

and

$$\gamma_q(h_1) = \gamma_q(h_2) \Longrightarrow \delta_q(h_1) \neq \delta_q(h_2)$$

as a consequence of choosing distinct pairs (i_h, α_h); i.e., for any two pairs (i_{h_1}, α_{h_1}), and (i_{h_2}, α_{h_2}), either the component i_h, or the memory α_h, could be the same, but not both. Also note that

$$1 \leq |P_q| , |Q_q| \leq q ,$$

and

$$2\lfloor\sqrt{q}\rfloor \leq |P_q| + |Q_q| \leq 2q .$$

Now rewrite equation (6.5.5) for $S_2^{(h)}$ as

$$S_2^{(h)} = u_{i_h}^{(\alpha_h)} \sum_{j>q} u_j^{(\alpha_h)} \sum_{\beta>q} u_{i_h}^{(\beta)} u_{j_h}^{(\beta_h)} .$$

As h runs through $1,...,q$, we can group the chosen memories α_h into $|P_q|$ memories, with $\alpha_h = \theta_{\gamma_q(h)} \in \{1,...,q\}$, and where $\gamma_q(h) \in \{1,..., |P_q|\}$, and $\theta_{t_1} \neq \theta_{t_2}$ if $t_1 \neq t_2$. Thus, the term $u_j^{(\alpha_h)} = u_j^{(\theta_{\gamma_q(h)})}$ runs through the components of a $|P_q|$-vector as h varies. Let $\{\mathbf{v}_f\}_{f=1}^{2^{|P_q|}} = \mathbb{B}^{|P_q|}$ be the set of vertices of the binary hyper-cube in $|P_q|$-space. For each f, define the set M_f by

$$M_f = \{ j > q : u_j^{(\theta_{\gamma_q(h)})} = v_f^{(\gamma_q(h))} , h = 1,...,q \} ,$$

where we use the convention that $v_f^{(\gamma_q(h))}$ represents the $\gamma_q(h)$-th component of the binary $|P_q|$-tuple $\mathbf{v}_f$. We then have

$$S_2^{(h)} = u_{i_h}^{(\alpha_h)} \sum_{f=1}^{2^{|P_q|}} v_f^{(\gamma_q(h))} \sum_{j \in M_f} \sum_{\beta > q} u_{i_h}^{(\beta)} u_j^{(\beta)} .$$

As before, as h runs through $1,\ldots,q$, we can group the chosen neurons (component positions) i_h into $|Q_q|$ component positions, with $i_h = \psi_{\delta_q(h)} \in \{1,\ldots,q\}$, and where $\delta_q(h) \in \{1,\ldots,|Q_q|\}$, and $\psi_{t_1} \neq \psi_{t_2}$ if $t_1 \neq t_2$. Thus, the term $u_{i_h}^{(\beta)} = u_{\psi_{\delta_q(h)}}^{(\beta)}$ runs through the components of a $|Q_q|$-vector as h varies. Let $\{\mathbf{w}_g\}_{g=1}^{2^{|Q_q|}} = \mathbb{B}^{|Q_q|}$ be the set of vertices of the binary hyper-cube in $|Q_q|$-space. For each g, define the set N_g by

$$N_g = \{\beta > q : u_{\psi_{\delta_q(h)}}^{(\beta)} = w_g^{(\delta_q(h))} , h = 1,\ldots,q\} ,$$

where we use the convention that $w_g^{(\delta_q(h))}$ represents the $\delta_q(h)$-th component of the binary $|Q_q|$-tuple $\mathbf{w}_g$. We then have

$$S_2^{(h)} = u_{i_h}^{(\alpha_h)} \sum_{f=1}^{2^{|P_q|}} \sum_{g=1}^{2^{|Q_q|}} v_f^{(\gamma_q(h))} w_g^{(\delta_q(h))} \sum_{j \in M_f} \sum_{\beta \in N_g} u_j^{(\beta)} . \tag{6.5.7}$$

Let $\lambda_f = |M_f|$, and $\omega_g = |N_g|$. For a given f, and a given g, we define the events $J_f(j)$, $j > q$, and $B_g(\beta)$, $\beta > q$, by the following attributes. $J_f(j)$ is the set of points in the ensemble on which

$$j \in M_f \quad , \quad \text{or equivalently} \; , \quad u_j^{(\theta_{\gamma_q(h)})} = v_f^{(\gamma_q(h))} \, , h = 1,\ldots,q \; .$$

Similarly, $B_g(\beta)$ is the set of points in the ensemble on which

$$\beta \in N_g \ , \quad \text{or equivalently} \ , \quad u_{\psi_{\delta_q(h)}}^{(\beta)} = w_g^{(\delta_q(h))} \ , \ h = 1,\ldots,q \ .$$

Claim 1. The events $J_f(j)$, $j > q$, and $B_g(\beta)$, $\beta > q$, are jointly independent for each fixed f, g.

Proof of claim 1: Consider the events $J_f(j_1), \ldots, J_f(j_k)$, and the events $B_g(\beta_1), \ldots, B_g(\beta_l)$, for some k, l. Then

$$\mathbf{P}\left\{J_f(j_s)\,,\, B_g(\beta_t)\,,\ s = 1,\ldots,k\,,\ t = 1,\ldots,l\right\}$$

$$= \mathbf{P}\left\{u_{j_s}^{(\theta_{\gamma_q(h)})} = v_f^{(\gamma_q(h))}\,,\ u_{\psi_{\delta_q(h)}}^{(\beta_t)} = w_g^{(\delta_q(h))}\,,\ s = 1,\ldots,k\,,\ t = 1,\ldots,l\,,\ h = 1,\ldots,q\right\}$$

$$\triangleq p_{f,g}(k,l) \ .$$

Note that $\theta_{\gamma_q(h)} \leq q$, $\psi_{\delta_q(h)} \leq q$ for $h = 1,\ldots,q$, and $j_s > q$, $s = 1,\ldots,k$, and $\beta_t > q$ for $t = 1,\ldots,l$. The components $u_i^{(\alpha)}$ are chosen independently from a sequence of Bernoulli trials. Hence

$$p_{f,g}(k,l) = \prod_{s=1}^{k} \mathbf{P}\left\{u_{j_s}^{(\theta_{\gamma_q(h)})} = v_f^{(\gamma_q(h))}\,,\ h = 1,\ldots,q\right\}$$

$$\times \prod_{t=1}^{l} \mathbf{P}\left\{u_{\psi_{\delta_q(h)}}^{(\beta_t)} = w_g^{(\delta_q(h))}\,,\ h = 1,\ldots,q\right\}$$

$$= \prod_{s=1}^{k} \mathbf{P}\left\{J_f(j_s)\right\} \prod_{t=1}^{l} \mathbf{P}\left\{B_g(\beta_t)\right\} \ .$$

As the index sets $\{j_1, \ldots, j_k\}$, and $\{\beta_1, \ldots, \beta_l\}$ are arbitrary, it follows that the events $J_f(j)$, $j > q$, and $B_g(\beta)$, $\beta > q$, are jointly independent for fixed f and g.

Note that for $j > q$, and $\beta > q$,

$$\mathbf{P}\left\{J_f(j)\right\} = \mathbf{P}\left\{u_j^{(\theta_{\gamma_q(h)})} = v_f^{(\gamma_q(h))},\ h = 1,\ldots,q\right\}$$

$$= 2^{-|P_q|}, \tag{6.5.8}$$

and

$$\mathbf{P}\left\{B_g(\beta)\right\} = \mathbf{P}\left\{u_{\psi_{\delta_q(h)}}^{(\beta)} = w_g^{(\delta_q(h))},\ h = 1,\ldots,q\right\}$$

$$= 2^{-|Q_q|}. \tag{6.5.9}$$

For each f and g, define the random variables $\left\{\varsigma_f(j)\right\}_{j>q}$, and $\left\{\nu_g(\beta)\right\}_{\beta>q}$ by

$$\varsigma_f(j) = \begin{cases} 1 \text{ if } j \in M_f \\ 0 \text{ if } j \notin M_f \end{cases},$$

and

$$\nu_g(\beta) = \begin{cases} 1 \text{ if } \beta \in N_g \\ 0 \text{ if } \beta \notin N_g \end{cases}.$$

By claim 1 it follows that the random variables $\left\{\varsigma_f(j)\right\}$ and $\left\{\nu_g(\beta)\right\}$ are mutually independent; further, the random variables $\left\{\varsigma_f(j)\right\}$ are i.i.d. with

$$\mathbf{P}\left\{\varsigma_f(j) = 1\right\} = 2^{-|P_q|},$$

and

$$\mathbf{P}\left\{\varsigma_f(j) = 0\right\} = 1 - 2^{-|P_q|}, \tag{6.5.10}$$

from claim 1 and equation (6.5.8). Similarly, having recourse to the claim and equation (6.5.9), we have that the random variables $\left\{\nu_g(\beta)\right\}$ are i.i.d. with

$$\mathbf{P}\left\{\nu_g(\beta)=1\right\}=2^{-|Q_q|},$$

and

$$\mathbf{P}\left\{\nu_g(\beta)=0\right\}=1-2^{-|Q_q|}. \qquad (6.5.11)$$

We now estimate the cardinality of the sets M_f and N_g using Large Deviation Theory for a given f , and a given g :

For the cardinality of M_f , we have:

$$\lambda_f = |M_f| = \sum_{j>q} \varsigma_f(j), \qquad (6.5.12)$$

and so from claim 1 and equation (6.5.10),

$$\bar{\lambda} = E(\lambda_f) = (n-q)\,E\left\{\varsigma_f(j)\right\}$$

$$= 2^{-|P_q|}(n-q). \qquad (6.5.13)$$

The random variables $\left\{\varsigma_f(j)\right\}$ are i.i.d., and hence λ_f is the sum of $(n-q)$ independent (1,0) random variables. Let $0 < \epsilon < 1/8$. As $n \rightarrow \infty$, we have

$$\frac{n^{\epsilon}\sqrt{n}}{(n-q)^{2/3}} \sim n^{\epsilon-\frac{1}{6}} \rightarrow 0,$$

and

$$\frac{n^{\epsilon}\sqrt{n}}{(n-q)^{1/2}} \sim n^{\epsilon} \rightarrow \infty.$$

Hence

$$\mathbf{P}\left\{\,|\lambda_f - \bar{\lambda}| > n^{\epsilon}\sqrt{n}\,\right\} \sim O\left(e^{-C_2 n^{2\epsilon}}\right) \qquad (6.5.14)$$

where C_2 is a positive constant.

Now consider the cardinality of the index set N_g . Have

$$\omega_g = | N_g | = \sum_{\beta > q} \nu_g(\beta) , \qquad (6.5.15)$$

and following claim 1 and (6.5.11), we have

$$\bar{\omega} = E(\omega_g) = (m - q) E\left\{\nu_g(\beta)\right\}$$

$$= 2^{-|Q_q|}(m - q) . \qquad (6.5.16)$$

The random variables $\left\{\nu_g(\beta)\right\}$ are i.i.d., and hence ω_g is the sum of $(m - q)$ independent (1,0) random variables. Let $0 < \epsilon < 1/8$. As $n \rightarrow \infty$, we have

$$\frac{n^{\epsilon}\sqrt{m}}{(m - q)^{2/3}} \sim \frac{n^{\epsilon}}{m^{1/6}} \leq \frac{n^{\epsilon}}{M(n)^{1/6}} = n^{\epsilon - \kappa/6} \leq n^{\epsilon - 1/8} \rightarrow 0 ,$$

and

$$\frac{n^{\epsilon}\sqrt{m}}{(m - q)^{1/2}} \sim n^{\epsilon} \rightarrow \infty .$$

Hence

$$\mathbf{P}\left\{ | \omega_g - \bar{\omega} | > n^{\epsilon}\sqrt{m} \right\} \sim O\left(e^{-C_3 n^{2\epsilon}}\right) \qquad (6.5.17)$$

where C_3 is a positive constant.

We now prove the following assertion:

Claim 2. The random variables λ_f and ω_g are independent.

Proof of Claim 2: From equation (6.5.12), λ_f depends solely on the $(n - q)$ random variables $\left\{\varsigma_f(j)\right\}_{j > q}$, while from (6.5.15), ω_g depends solely upon the $(m - q)$ random variables $\left\{\nu_g(\beta)\right\}_{\beta > q}$. From claim 1, $\left\{\varsigma_f(j)\right\}$ and $\left\{\nu_g(\beta)\right\}$ are mutually independent random variables, and hence it follows that λ_f and ω_g are independent.

Now let the random variables $x_{f,g}$ be defined by

$$x_{f,g} = \sum_{j \in M_f} \sum_{\beta \in N_g} u_j^{(\beta)} . \tag{6.5.18}$$

$x_{f,g}$ is hence the sum of $\lambda_f \, \omega_g$ ± 1, independent random variables. Let $0 < \epsilon < 1/8$. Then

$$\mathbf{P}\left\{ \, | x_{f,g} | > n^{\epsilon}\sqrt{mn} \, \right\}$$

$$= \mathbf{P}\left\{ \, | x_{f,g} | > n^{\epsilon}\sqrt{mn} \;\middle|\; | \lambda_f - \overline{\lambda} | \leq n^{\epsilon}\sqrt{n} \, , \, | \omega_g - \overline{\omega} | \leq n^{\epsilon}\sqrt{m} \, \right\}$$

$$\times \mathbf{P}\left\{ \, | \lambda_f - \overline{\lambda} | \leq n^{\epsilon}\sqrt{n} \, , \, | \omega_g - \overline{\omega} | \leq n^{\epsilon}\sqrt{m} \, \right\}$$

$$+ \mathbf{P}\left\{ \, | x_{f,g} | > n^{\epsilon}\sqrt{mn} \;\middle|\; | \lambda_f - \overline{\lambda} | \leq n^{\epsilon}\sqrt{n} \, , \, | \omega_g - \overline{\omega} | > n^{\epsilon}\sqrt{m} \, \right\}$$

$$\times \mathbf{P}\left\{ \, | \lambda_f - \overline{\lambda} | \leq n^{\epsilon}\sqrt{n} \, , \, | \omega_g - \overline{\omega} | > n^{\epsilon}\sqrt{m} \, \right\}$$

$$+ \mathbf{P}\left\{ \, | x_{f,g} | > n^{\epsilon}\sqrt{mn} \;\middle|\; | \lambda_f - \overline{\lambda} | > n^{\epsilon}\sqrt{n} \, , \, | \omega_g - \overline{\omega} | \leq n^{\epsilon}\sqrt{m} \, \right\}$$

$$\times \mathbf{P}\left\{ \, | \lambda_f - \overline{\lambda} | > n^{\epsilon}\sqrt{n} \, , \, | \omega_g - \overline{\omega} | \leq n^{\epsilon}\sqrt{m} \, \right\}$$

$$+ \mathbf{P}\left\{ \, | x_{f,g} | > n^{\epsilon}\sqrt{mn} \;\middle|\; | \lambda_f - \overline{\lambda} | > n^{\epsilon}\sqrt{n} \, , \, | \omega_g - \overline{\omega} | > n^{\epsilon}\sqrt{m} \, \right\}$$

$$\times \mathbf{P}\left\{ \, | \lambda_f - \overline{\lambda} | > n^{\epsilon}\sqrt{n} \, , \, | \omega_g - \overline{\omega} | > n^{\epsilon}\sqrt{m} \, \right\}$$

$$= A + B$$

where A is the first term in the sum, and B the remaining terms. Now,

$$B \leq 3 \max \left\{ \mathbf{P}\left\{ \, | \lambda_f - \overline{\lambda} | > n^{\epsilon}\sqrt{n} \, \right\} , \mathbf{P}\left\{ \, | \omega_g - \overline{\omega} | > n^{\epsilon}\sqrt{m} \, \right\} \right\}$$

$$\sim O\left(e^{-(C_2+C_3)n^{2\epsilon}}\right),$$

which follows from claim 2, and equations (6.5.14), and (6.5.17). Hence,

$$\mathbf{P}\left\{ |\lambda_f - \bar{\lambda}| \le n^\epsilon\sqrt{n}\ ,\ |\omega_g - \bar{\omega}| \le n^\epsilon\sqrt{m} \right\}$$

$$\sim \left(1 - O\left(e^{-C_2 n^{2\epsilon}}\right)\right)\left(1 - O\left(e^{-C_3 n^{2\epsilon}}\right)\right) \sim 1 .$$

We now restrict our attention to those sample points for which the events $|\lambda_f - \bar{\lambda}| \le n^\epsilon\sqrt{n}$, and $|\omega_g - \bar{\omega}| \le n^\epsilon\sqrt{m}$ occur. As $n \to \infty$, we have from equations (6.5.13), and (6.5.16) that

$$\frac{n^\epsilon\sqrt{mn}}{\lambda_f^{3/4}\,\omega_g^{3/4}} \le \frac{n^\epsilon\sqrt{mn}}{(\bar{\lambda} - n^\epsilon\sqrt{n}\,)^{3/4}\,(\bar{\omega} - n^\epsilon\sqrt{m}\,)^{3/4}}$$

$$\sim \frac{n^\epsilon\sqrt{mn}}{2^{-\frac{3}{4}|P_q|}\,n^{3/4}\,2^{-\frac{3}{4}|Q_q|}\,m^{3/4}}$$

$$= 2^{\frac{3}{4}(|P_q| + |Q_q|)}\,\frac{n^{\epsilon - 1/4}}{m^{1/4}} \to 0 ,$$

and

$$\frac{n^\epsilon\sqrt{mn}}{\lambda_f^{1/2}\,\omega_g^{1/2}} \sim 2^{\frac{1}{2}(|P_q| + |Q_q|)}\,n^\epsilon \to \infty .$$

Hence $A \sim O\left(e^{-C_4^* n^{2\epsilon}}\right)$, where C_4^* is a positive constant. Thus

$$\mathbf{P}\left\{ |x_{f,g}| > n^\epsilon\sqrt{mn} \right\} \sim O\left(e^{-C_4 n^{2\epsilon}}\right),$$

where $C_4 = \min\,(C_4^*\ ,\ C_2 + C_3)$.

Let S be the set of sample points on which the following hold jointly: For $0 < \epsilon < 1/8$

$$| S_1^{(h)} | \leq n^{1/2+\epsilon}, \; h = 1,\ldots,q,$$

$$| \lambda_f - \overline{\lambda} | \leq n^{1/2+\epsilon}, \; f = 1,\ldots,2^{|P_q|},$$

$$| \omega_g - \overline{\omega} | \leq m^{1/2} n^{\epsilon}, \; g = 1,\ldots,2^{|Q_q|},$$

$$| x_{f,g} | \leq m^{1/2} n^{1/2+\epsilon}, \; f = 1,\ldots,2^{|P_q|}, \; g = 1,\ldots,2^{|Q_q|} . \quad (6.5.19)$$

Using (6.5.6), (6.5.14), (6.5.17), and , we now obtain

$$\mathbf{P}\left\{\neg S\right\} = O\left(e^{-C_5 n^{2\epsilon}}\right), \quad (6.5.20)$$

where $C_5 = \min\,(C_1, C_2, C_3, C_4)$. Combining (6.5.7) and (6.5.18), we have

$$\frac{X_{i_h}^{(\alpha_h)}}{\sqrt{\overline{\lambda}\overline{\omega}}} = u_{i_h}^{(\alpha_h)} \sum_{f=1}^{2^{|P_q|}} \sum_{g=1}^{2^{|Q_q|}} v_f^{(\gamma_q(h))} w_g^{(\delta_q(h))} \frac{x_{f,g}}{\sqrt{\lambda_f\,\omega_g}}$$

$$+ u_{i_h}^{(\alpha_h)} \sum_{f=1}^{2^{|P_q|}} \sum_{g=1}^{2^{|Q_q|}} v_f^{(\gamma_q(h))} w_g^{(\delta_q(h))} x_{f,g} \left(\frac{1}{\sqrt{\overline{\lambda}\overline{\omega}}} - \frac{1}{\sqrt{\lambda_f\,\omega_g}} \right)$$

$$+ \frac{n + (1-g)m - 1}{\sqrt{\overline{\lambda}\overline{\omega}}} + \frac{S_1^{(h)}}{\sqrt{\overline{\lambda}\overline{\omega}}} .$$

From (6.5.13), (6.5.16) and (6.5.19), we have that in S,

$$\frac{S_1^{(h)}}{\sqrt{\overline{\lambda}\overline{\omega}}} = O\left(\frac{n^{1/2+\epsilon}}{\sqrt{mn}} \right) = O\left(\frac{n^{\epsilon}}{\sqrt{m}} \right) .$$

Denote the second sum by T. In S we have

$$| T | \le \sum_{f=1}^{2^{|P_q|}} \sum_{g=1}^{2^{|Q_q|}} | x_{f,g} | \frac{| \sqrt{\lambda_f \omega_g} - \sqrt{\bar{\lambda}\bar{\omega}} |}{\sqrt{\bar{\lambda}\bar{\omega}\lambda_f \omega_g}}$$

$$\le \sum_{f,g} | x_{f,g} | \frac{| \lambda_f \omega_g - \bar{\lambda}\bar{\omega} |}{\sqrt{\bar{\lambda}\bar{\omega}\lambda_f \omega_g} \, (\sqrt{\lambda_f \omega_g} + \sqrt{\bar{\lambda}\bar{\omega}})}$$

$$\le \sum_{f,g} | x_{f,g} | \left[\frac{| \omega_g (\lambda_f - \bar{\lambda}) | + | \bar{\lambda}(\omega_g - \bar{\omega}) |}{\sqrt{\bar{\lambda}\bar{\omega}\lambda_f \omega_g} \, (\sqrt{\lambda_f \omega_g} + \sqrt{\bar{\lambda}\bar{\omega}})} \right]$$

$$\lesssim \frac{2^{(|P_q| + |Q_q|)} n^{\epsilon} \sqrt{mn} \, (2^{-|Q_q|} mn^{1/2+\epsilon} + 2^{-|P_q|} n^{1+\epsilon} m^{1/2})}{2^{-\frac{3}{2}(|P_q| + |Q_q|)} m^{3/2} n^{3/2}}$$

$$\sim 2^{(\frac{3}{2} |P_q| + \frac{5}{2} |Q_q|)} \frac{n^{2\epsilon}}{\sqrt{m}}$$

$$= O\left(\frac{n^{2\epsilon}}{\sqrt{m}} \right)$$

by equations (6.5.13), (6.5.16), and (6.5.19). Hence in S we have

$$\frac{X_{i_h}^{(\alpha_h)}}{\sqrt{\bar{\lambda}\bar{\omega}}} = u_{i_h}^{(\alpha_h)} \sum_{f=1}^{2^{|P_q|}} \sum_{g=1}^{2^{|Q_q|}} v_f^{(\gamma_q(h))} w_g^{(\delta_q(h))} \frac{x_{f,g}}{\sqrt{\lambda_f \omega_g}}$$

$$+ \frac{n + (1-g)m - 1}{\sqrt{\bar{\lambda}\bar{\omega}}} + O\left(\frac{n^{2\epsilon}}{\sqrt{m}} \right) . \qquad (6.5.21)$$

Note that $\frac{n^{2\epsilon}}{\sqrt{m}} \le \frac{n^{2\epsilon}}{M(n)^{1/2}} = n^{2\epsilon - \kappa/2} \le n^{2\epsilon - 3/8}$. For ϵ taking values in the open interval $0 < \epsilon < 1/8$, we hence have $\frac{n^{2\epsilon}}{\sqrt{m}} \to 0$ as $n \to \infty$. Now define

$$\begin{aligned} f_1(b) = \mathbf{P} \Bigg\{ & S \ , \ u_{i_h}^{(\alpha_h)} \sum_{f=1}^{2^{|P_q|}} \sum_{g=1}^{2^{|Q_q|}} v_f^{(\gamma_q(h))} w_g^{(\delta_q(h))} \frac{x_{f,g}}{\sqrt{\lambda_f \omega_g}} \\ & < -\frac{n + (1-g)m - 1}{\sqrt{\bar{\lambda}\bar{\omega}}} + b \ , \ h = 1,\ldots,q \Bigg\} . \end{aligned} \tag{6.5.22}$$

Using equations (6.5.19) through (6.5.22), we have

$$\begin{aligned} f_1(-K\ n^{\epsilon - 1/4}) + O(e^{-C_5 n^{2\epsilon}}) &< \mathbf{P}\left\{X_{i_h}^{(\alpha_h)} < 0 \, , h = 1,\ldots,q\right\} \\ &< f_1(K\ n^{\epsilon - 1/4}) + O(e^{-C_5 n^{2\epsilon}}) \end{aligned} \tag{6.5.23}$$

for a positive constant K.

Let $\mathbf{\Lambda}$ and $\mathbf{\Omega}$ be the set of values of $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_{2^{|P_q|}})$, and $\omega = (\omega_1, \ldots, \omega_{2^{|Q_q|}})$, respectively, in S. Let $\mathbf{\Gamma}$ and $\mathbf{\Upsilon}$ denote the allowed partitions $\mathbf{M} = (M_1, \ldots, M_{2^{|P_q|}})$, and $\mathbf{N} = (N_1, \ldots, N_{2^{|Q_q|}})$, respectively, of $\{(k,r) : k,r > q\}$ in S. Denoting binary q-tuples by $\mathbf{y} = (y_1,\ldots,y_q) \in \mathbb{B}^q$, we have

$$f_1(b) = \sum_{\substack{\mathbf{y} \in \mathbb{B}^q \\ \boldsymbol{\lambda} \in \mathbf{\Lambda} \, , \, \boldsymbol{\omega} \in \mathbf{\Omega} \\ \mathbf{M} \in \mathbf{\Gamma} \, , \, \mathbf{N} \in \mathbf{\Upsilon}}} f_2(b ; \mathbf{y},\boldsymbol{\lambda},\boldsymbol{\omega},\mathbf{M},\mathbf{N}) \ \mathbf{P}\left\{ \mathbf{y},\boldsymbol{\lambda},\boldsymbol{\omega},\mathbf{M},\mathbf{N} \right\} , \tag{6.5.24}$$

where

$$f_2(b ; \mathbf{y},\boldsymbol{\lambda},\omega,\mathbf{M},\mathbf{N}) = \mathbf{P} \Bigg\{ S \ , \ y_h \sum_{f=1}^{2^{|P_q|}} \sum_{g=1}^{2^{|Q_q|}} v_f^{(\gamma_q(h))} w_g^{(\delta_q(h))} \frac{x_{f,g}}{\sqrt{\lambda_f \omega_g}}$$

$$< -\frac{n+(1-g)m-1}{\sqrt{\bar{\lambda}\bar{\omega}}} + b \ , \ h = 1,\ldots,q \ \Big| \ \mathbf{y},\boldsymbol{\lambda},\boldsymbol{\omega},\mathbf{M},\mathbf{N} \Big\} . \tag{6.5.25}$$

Equation (6.5.25) is the sum of probability masses over the set of lattice points of allowable values in the region $D(b)$ in $2^{(|P_q|+|Q_q|)}$–space defined by the inequalities

$$y_h \sum_{f=1}^{2^{|P_q|}} \sum_{g=1}^{2^{|Q_q|}} v_f^{(\gamma_q(h))} w_g^{(\delta_q(h))} \frac{x_{f,g}}{\sqrt{\lambda_f \omega_g}} < -\frac{n+(1-g)m-1}{\sqrt{\bar{\lambda}\bar{\omega}}} + b \ , \ h = 1,\ldots,q \ , \tag{6.5.26}$$

and

$$| x_{f,g} | \le n^{\epsilon}\sqrt{mn} \ , \ f = 1,\ldots,2^{|P_q|} \ , \ g = 1,\ldots,2^{|Q_q|} \ . \tag{6.5.27}$$

Now note from the defining equation (6.5.18) for $x_{f,g}$, that the $x_{f,g}$s are sums of independent, ±1 random variables over disjoint sets. The random variables $x_{f,g}$ are hence jointly independent, with each point probability being the product of $2^{(|P_q|+|Q_q|)}$ probabilities, one for each $x_{f,g}$. By virtue of lemma (6.4.1), each point probability can be replaced by a $2^{(|P_q|+|Q_q|)}$–dimensional integral over a box. The union of these rectangles is a region $\Delta(b)$ in $2^{(|P_q|+|Q_q|)}$–dimensional space, which differs from $D(b)$ only in the addition or deletion of points on the boundary. Hence

$$f_2(b ; \mathbf{y},\boldsymbol{\lambda},\boldsymbol{\omega},\mathbf{M},\mathbf{N}) \sim F_2(b ; \mathbf{y},\boldsymbol{\lambda},\boldsymbol{\omega},\mathbf{M},\mathbf{N}) \tag{6.5.28}$$

where

$$F_2(b ; \mathbf{y},\boldsymbol{\lambda},\boldsymbol{\omega},\mathbf{M},\mathbf{N}) = \int \underset{\Delta(b)}{\cdots} \int \prod_{f=1}^{2^{|P_q|}} \prod_{g=1}^{2^{|Q_q|}} \frac{1}{\sqrt{2\pi\lambda_f \, \omega_g}} \exp \left[- \frac{t_{f}{}^{2}{}_{,g}}{2\lambda_f \, \omega_g} \right] dt_{f\ ,g} \ . \tag{6.5.29}$$

Claim 3. $D(b)$ is a monotonic function of b; specifically, $D(b)$ satisfies the folowing inclusions: $b_1 > b_2 \implies D(b_2) \subset D(b_1)$.

Proof of Claim 3: From (6.5.26) and (6.5.27), we see that as b increases, more and more sample points $\mathbf{x} = (x_{1,1},\ldots,x_{2^{|P_q|},2^{|Q_q|}})^T$ in the rectangle $[-n^{\epsilon}\sqrt{mn}\ ,n^{\epsilon}\sqrt{mn}\ ]^{2^{(|P_q| + |Q_q|)}}$ are included in $D(b) \subset [-n^{\epsilon}\sqrt{mn}\ ,n^{\epsilon}\sqrt{mn}\ ]^{2^{(|P_q| + |Q_q|)}}$ as the inequalities (6.5.26) are satisfied for more sample points.

Now, traversing across each small rectangle centred at a sample point results in varying each $x_{f\ ,g}$ by two. The sums in (6.5.26) hence vary by $O(\frac{1}{\sqrt{mn}})$ across each small rectangle. Hence, we can find a suitable constant C_6 depending solely on q, which in conjunction with claim 3 yields the following inclusion relations:

$$D(b - \frac{C_6}{\sqrt{mn}}) - E_1 \subset \Delta(b) \subset D(b + \frac{C_6}{\sqrt{mn}}) + E_2 \,,$$

where the sets E_1, and E_2, contain only points which are within distance two of at least one of the planes defined by (6.5.27). Again having recourse to Large Deviation Theory (lemma (6.4.2)), and using (6.5.19), and (6.5.20), we have

$$\mathbf{P} \left\{ \mathbf{x} \in E_i \ \mid \ \mathbf{y},\boldsymbol{\lambda},\boldsymbol{\omega},\mathbf{M},\mathbf{N} \right\} = O\left(e^{-C_7 n^{2\epsilon}} \right) \,, \ i = 1,2 \,, \tag{6.5.30}$$

for a positive constant C_7.

Now define

$$F_3(\,b\,;\mathbf{y},\boldsymbol{\lambda},\omega,\mathbf{M},\mathbf{N}\,) = \int \underset{D(b)}{\cdots} \int \prod_{f=1}^{2^{|P_q|}} \prod_{g=1}^{2^{|Q_q|}} \frac{1}{\sqrt{2\pi\lambda_f\,\omega_g}} \exp\left[-\frac{t_{f,g}^2}{2\lambda_f\,\omega_g}\right] dt_{f,g}\ ; \tag{6.5.31}$$

then

$$F_3\left(\,b - \frac{C_6}{\sqrt{mn}}\,;\mathbf{y},\boldsymbol{\lambda},\omega,\mathbf{M},\mathbf{N}\,\right) + O(e^{-C_7 n^{2\epsilon}}) < F_2(b\,;\mathbf{y},\boldsymbol{\lambda},\omega,\mathbf{M},\mathbf{N})$$

$$< F_3\left(\,b + \frac{C_6}{\sqrt{mn}}\,;\mathbf{y},\boldsymbol{\lambda},\omega,\mathbf{M},\mathbf{N}\,\right) + O(e^{-C_7 n^{2\epsilon}})\ . \tag{6.5.32}$$

Let $\left\{\xi_{f,g}\right\}_{f=1}^{2^{|P_q|}}\,{}_{,\,g=1}^{2^{|Q_q|}}$ be i.i.d. Gaussian random variables with

$$E\left\{\xi_{f,g}\right\} = 0\ ,$$

and

$$\mathrm{Var}\left\{\xi_{f,g}\right\} = \lambda_f\,\omega_g\ .$$

Let $\boldsymbol{\xi} = (\xi_{1,1}, \ldots, \xi_{2^{|P_q|},2^{|Q_q|}})$ be a vector in $2^{|P_q|+|Q_q|}$–dimensional space. Then

$$F_3(\,b\ ;\mathbf{y},\boldsymbol{\lambda},\omega,\mathbf{M},\mathbf{N}\,) = \mathbf{P}\left\{\,\boldsymbol{\xi} \in D(b)\right\}\ .$$

Have

$$\mathbf{P}\left\{\,|\,\xi_{f,g}\,| > n^{\epsilon}\sqrt{mn}\ ,\ f = 1,\ldots,2^{|P_q|}\ ,\ g = 1,\ldots,2^{|Q_q|}\right\} = O\left(e^{-C_8 n^{2\epsilon}}\right)$$

for a positive constant C_8. Set

$$\eta_h = y_h \sum_{f=1}^{2^{|P_q|}} \sum_{g=1}^{2^{|Q_q|}} v_f^{(\gamma_q(h))} w_g^{(\delta_q(h))} \frac{\xi_{f,g}}{\sqrt{\lambda_f\,\omega_g}}\ ,\ \ h = 1,\ldots,q\ . \tag{6.5.33}$$

Defining the region $D_0(b)$ in $2^{|P_q|+|Q_q|}$–space by the inequalities

$$\eta_h < - \frac{n + (1-g)m - 1}{\sqrt{\lambda\bar{\omega}}} + b \ , h = 1,...,q \ , \tag{6.5.34}$$

we get

$$F_3(b \ ; \mathbf{y},\boldsymbol{\lambda},\boldsymbol{\omega},\mathbf{M},\mathbf{N}) = \mathbf{P} \left\{ \boldsymbol{\xi} \in D_0(b) \right\} + O(e^{-C_8 n^{2\epsilon}})$$

$$= \mathbf{P} \left\{ \eta_h < - \frac{n + (1-g)m - 1}{\sqrt{\lambda\bar{\omega}}} + b \ , h = 1,...,q \right\} + O(e^{-C_8 n^{2\epsilon}}) \ . \tag{6.5.35}$$

From (6.5.33), the random q-vector $\boldsymbol{\eta} = (\eta_1, \ . \ . \ . \ , \eta_q)$ is obtained by means of a linear transformation from the jointly normal random variables $\xi_{f\ ,g}$, $f = 1,...,2^{|P_q|}$, $g = 1,...,2^{|Q_q|}$. The random variables η_h are hence jointly normal. Now, we have

$$E(\eta_h) = y_h \sum_{f=1}^{2^{|P_q|}} \sum_{g=1}^{2^{|Q_q|}} v_f^{(\gamma_q(h))} w_g^{(\delta_q(h))} \frac{E(\xi_{f\ ,g})}{\sqrt{\lambda_f \ \omega_g}} = 0 \ ,$$

and

$$E(\eta_h \eta_{h^*}) = y_h \ y_{h^*} \sum_{f=1}^{2^{|P_q|}} \sum_{g=1}^{2^{|Q_q|}} v_f^{(\gamma_q(h))} w_g^{(\delta_q(h))} v_f^{(\gamma_q(h^*))} w_g^{(\delta_q(h^*))}$$

$$= y_h \ y_{h^*} \left[\sum_{f=1}^{2^{|P_q|}} v_f^{(\gamma_q(h))} v_f^{(\gamma_q(h^*))} \right] \left[\sum_{g=1}^{2^{|Q_q|}} w_g^{(\delta_q(h))} w_g^{(\delta_q(h^*))} \right] .$$

If $h \neq h^*$, then by choice of distinct pairs (i_h, α_h), at least one following must hold: $\gamma_q(h) \neq \gamma_q(h^*)$ or $\delta_q(h) \neq \delta_q(h^*)$. In either case, at least one of the two sums in the above equation must be zero. Hence

$$E(\eta_h \eta_{h^*}) = 2^{(|P_q| + |Q_q|)} \delta_{hh^*} \ .$$

The random variables $\{\eta_h\}_{h=1}^{q}$ are hence i.i.d., and normal, with zero mean, and variance $2^{(|P_q| + |Q_q|)}$. Hence

$$p_h \triangleq \mathbf{P}\left\{ \eta_h < - \frac{n + (1-g)m - 1}{\sqrt{\bar{\lambda}\bar{\omega}}} + b \ , h = 1,\ldots,q \right\}$$

$$= \Phi\left[2^{-\frac{1}{2}(|P_q| + |Q_q|)} \left(- \frac{n + (1-g)m - 1}{\sqrt{\bar{\lambda}\bar{\omega}}} + b \right) \right]^q . \qquad (6.5.36)$$

From (6.5.23), we need to estimate (6.5.36) for $b = O(n^{\epsilon - \frac{1}{4}})$. Set

$$z = 2^{-\frac{1}{2}(|P_q| + |Q_q|)} \left(- \frac{n + (1-g)m - 1}{\sqrt{\bar{\lambda}\bar{\omega}}} + O(n^{\epsilon - \frac{1}{4}}) \right)$$

$$= - \frac{n + (1-g)m - 1}{\sqrt{(n-q)(m-q)}} + O(n^{\epsilon - 1/4}) ,$$

which we obtain by substituting for $\bar{\lambda}$ and $\bar{\omega}$ from equations (6.5.13) and (6.5.16). Then we have

$$z \sim - \sqrt{\frac{n}{m}} \quad \text{as } n \rightarrow \infty .$$

Also

$$z^2 = \frac{[n + (1-g)m - 1]^2}{(n-q)(m-q)} + O\left(\frac{n^{2\epsilon}}{\sqrt{m}} \right)$$

$$= \frac{n}{m} + 2(1-g) + O\left(\frac{m}{n} + \frac{n}{m^2} + \frac{n^{2\epsilon}}{\sqrt{m}} \right) .$$

By hypotheses, we have $\frac{m}{n} \rightarrow 0$, $\frac{n}{m^2} \leq \frac{n}{M(n)^2} \rightarrow 0$, and $\frac{n^{2\epsilon}}{\sqrt{m}} \leq n^{\epsilon - 1/4} \rightarrow 0$ as $n \rightarrow \infty$. Hence

$$z^2 \sim \frac{n}{m} + 2(1-g)\,.$$

Harking back to (6.5.36), we have as $n \rightarrow \infty$,

$$p_h = [\Phi(z)]^q \sim \frac{m^{1/2}}{(2\pi n)^{1/2}} \exp\left\{ - \left(\frac{n}{2m} + 1 - g \right) \right\} \sim \tau$$

by lemma (6.5.1). Retracing our path through equations (6.5.35), (6.5.32), (6.5.28), and (6.5.24), we obtain

$$f_1(O(n^{\epsilon - \frac{1}{4}})) \sim \sum_{\substack{\mathbf{y} \in \mathbf{B}^q \\ \boldsymbol{\lambda} \in \boldsymbol{\Lambda},\, \boldsymbol{\omega} \in \boldsymbol{\Omega} \\ \mathbf{M} \in \boldsymbol{\Gamma},\, \mathbf{N} \in \mathbf{T}}} \tau^q \,\mathbf{P}\left\{ \mathbf{y},\boldsymbol{\lambda},\omega,\mathbf{M},\mathbf{N} \right\} + O\left(e^{-C_7 n^{2\epsilon}} + e^{-C_8 n^{2\epsilon}} \right)$$

$$= \tau^q + O\left(e^{-C_7 n^{2\epsilon}} + e^{-C_8 n^{2\epsilon}} \right).$$

Substituting in (6.5.23) we finally obtain

$$\mathbf{P}\left\{ X_{i_h}^{(\alpha_h)} < 0 \,,\, h = 1,\ldots,q \right\} \sim \tau^q + O\left(e^{-C n^{2\epsilon}} \right), \qquad (6.5.37)$$

where $C = \min\,(C_5, C_7, C_8)$.

From lemma (6.5.1), we have as $n \rightarrow \infty$,

$$\tau^q \sim \frac{1}{(2\pi)^{q/2}} \left(\frac{m}{n} \right)^{q/2} e^{-\frac{qn}{2m}}$$

$$\geq \frac{1}{(2\pi)^{q/2}} \left(\frac{m}{n} \right)^{q/2} e^{-\frac{qn}{2M(n)}}$$

$$\geq \frac{1}{(2\pi)^{q/2}} \left(\frac{m}{n} \right)^{q/2} e^{-\frac{q}{2} n^{(1-\kappa)}} .$$

τ^q will be the dominant term in (6.5.37) if $2\epsilon > 1 - \kappa$. We also require that $\epsilon < \kappa/6$ for (6.5.17) to hold. Hence, we require κ and ϵ to satisfy $\frac{1-\kappa}{2} < \epsilon < \frac{\kappa}{6}$. These inequalities are satisfied for the prescribed choices of $3/4 < \kappa < 1$, and $0 < \epsilon < 1/8$. Under these conditions then, as $n \rightarrow \infty$,

$$\mathbf{P} \left\{ X_{i_h}^{(\alpha_h)} < 0 \, , h = 1,\ldots,q \right\} \sim \tau^q \, . \qquad \square$$

Lemma (6.5.2) coupled with estimates to be derived in the following theorem proves that the number of row sum violations is asymptotically Poisson as $n \rightarrow \infty$. Specifically, for every fixed $N \geq 0$, the probability of exactly N row sum violations is asymptotic as $n \rightarrow \infty$ to $\frac{\epsilon^N e^{-N}}{N!}$, where $\epsilon = n\,\tau$ is the expected number of row sum violations, held essentially constant in the Theorem by proper choice of m as a function of n.

We now encapsulate the above lemmas in the following theorem, which is the main result of this chapter.

Theorem 6.5.1. Let $\delta > 0$ be fixed. Then, as $n \rightarrow \infty$, if:

(1) If $m = \frac{n}{2 \log n} \left[1 + \frac{\frac{1}{2} \log \log n + 1 - g + \log (\delta \sqrt{4\pi})}{\log n} + o\left(\frac{1}{\log n} \right) \right]$,(6.5.38)

(6.5.38)

then the expected number of fixed vectors $\mathbf{u}^{(\alpha)}$ is asymptotically $m \; e^{-\delta}$.

(2) If $$m = \frac{n}{4\log n}\left[1 + \frac{\frac{3}{4}\log\log n + \frac{1}{2}(1-g) + \frac{1}{2}\log(8\delta\sqrt{2\pi})}{\log n} + o\left(\frac{1}{\log n}\right)\right],$$

(6.5.39)

then the probability that all m vectors are fixed is asymptotically $e^{-\delta}$.

Proof. (1) Consider the n events $E_i = \{X_i^{(1)} < 0\}$, $i = 1,\ldots,n$. For every fixed N, we have by lemma (6.5.2) that as $n \to \infty$,

$$\mathbf{P}\left\{E_{i_1} \cap E_{i_2} \cap \cdots \cap E_{i_N}\right\} \sim \tau^N .$$

Applying the above result to lemma (6.4.4), and taking cognisance of the fact that σ_N contains $\binom{n}{N}$ terms, we obtain

$$\sigma_N \sim \binom{n}{N}\tau^N \sim \frac{1}{N!}(n\,\tau)^N .$$

Using (6.5.38), and (6.5.2),

$$n\,\tau \sim \frac{n}{\sqrt{4\pi}\log n}\exp\left\{-\log n + \frac{1}{2}\log\log n + \log(\delta\sqrt{4\pi}) + o(1)\right\},$$

so that

$$n\,\tau \sim \delta .$$

Choosing K even in lemma (6.4.4), we have

$$\sum_{k=1}^{2K}(-1)^{k-1}\frac{\delta^k}{k!} \lesssim \mathbf{P}\left\{E_1 \cup E_2 \cup \cdots \cup E_n\right\} \lesssim \sum_{k=1}^{2K-1}(-1)^{k-1}\frac{\delta^k}{k!} .$$

For large K, both sums are arbitrarily close to $1 - e^{-\delta}$. Hence

$$\mathbf{P}\left\{E_1 \bigcup E_2 \bigcup \cdots \bigcup E_n\right\} \sim 1 - e^{-\delta}.$$

The above is the probability that the chosen memory $\mathbf{u}^{(1)}$ is not a fixed point. The expected number of vectors $\mathbf{u}^{(\alpha)}$ that are fixed, which by symmetry is m times the probability that $\mathbf{u}^{(1)}$ is a fixed point, is hence $me^{-\delta}$.

(2) We consider the complement of the probability $\mathbf{P}\left\{E_1 \bigcup E_2 \bigcup \cdots \bigcup E_N\right\}$, with $N = mn$, and the E_ks being the mn events $\left\{X_i^{(\alpha)} < 0\right\}$, $i = 1,...,n$, $\alpha = 1,...,m$. The argument unfolds in exactly the same fashion as the proof for part (1). □

Corollary 6.5.1. The storage capacity (weak sense) of fixed points in the outer product algorithm is $\frac{n}{2 \log n}$; specifically, as $n \to \infty$:

(1) For every fixed $0 \leq \epsilon < 1$, the expected number of fundamental memories that are fixed points in the algorithm is $m - o(m)$ if $m = \frac{n}{2 \log n}(1 - \epsilon)$.

(2) For every fixed $\epsilon > 0$, the expected number of fundamental memories that are fixed points in the algorithm is $o(m)$ if $m = \frac{n}{2 \log n}(1 + \epsilon)$.

Proof. (1) Fix $0 \leq \epsilon < 1$, and set $m = \frac{n}{2 \log n}(1 - \epsilon)$. Set $\delta = \frac{1}{\log n}$ in the theorem, and let m^* be the corresponding value for the number of memories. (In fact, all we require is that δ is a decreasing function of n that approaches zero when n becomes infinite.) Let τ and τ^* be the probabilities of row sum violations for m and m^*, respectively. Clearly, $m < m^*$. Hence, by proposition (6.5.1), it follows that $\tau \leq \tau^*$. By the proof of the theorem it follows that the expected number of memories that are fixed points is at least $m\ e^{-\frac{1}{\log n}} = m - o(m)$.

(2) Fix $\epsilon > 0$, and set $m = \frac{n}{2 \log n}(1 + \epsilon)$. Set $\delta = \log n$ in the theorem, and let m^* be the corresponding value for the number of memories. (In fact, all we require is that δ is an increasing function of n that becomes infinite when n becomes infinite.)

Let τ and τ^* be the probabilities of row sum violations for m and m^*, respectively. For large enough n, we have $m > m^*$. Hence, by proposition (6.5.1), it follows that $\tau \geq \tau^*$. By the proof of the theorem it follows that the expected number of memories that are fixed points is at most $m\ e^{-\log n} = o(m)$. □

Note that the asymptotic behaviour is stronger than was required by the definition of capacity. In fact, setting $\delta = \epsilon$ in the theorem, and using the fact that $e^{-\epsilon} > 1 - \epsilon$, we see that the expected number of memories that are fixed points in the algorithm is asymptotically at least $m\,(1 - \epsilon)$ if

$$m = \frac{n}{2\log n}\left[1 + \frac{\log\log n}{2\log n} + O_\epsilon\left(\frac{1}{\log n}\right)\right].$$

In particular, $m = \dfrac{n}{2\log n}$ works.

Corollary 6.5.2. The storage capacity (strong sense) of fixed points in the outer product algorithm is $\dfrac{n}{4\log n}$; specifically, as $n \to \infty$:

(1) For every fixed $0 \leq \epsilon < 1$, the probability that all the memories are fixed points approaches one as n approaches infinity if $m = \dfrac{n}{4\log n}(1 - \epsilon)$.

(2) For every fixed $\epsilon > 0$, the probability that there is at least one fundamental memory that is not fixed approaches one as n approaches infinity if $m = \dfrac{n}{4\log n}(1 + \epsilon)$.

Proof. The proofs are the same as for Corollary (6.5.1). □

Again, the asymptotic behaviour is stronger than was required by the definition of capacity. In fact, fór every fixed $0 \leq \epsilon < 1$, the probability of all the memories being fixed is at least $1 - \epsilon$ as n approaches infinity if

$$m = \frac{n}{4 \log n} \left[1 + \frac{3 \log \log n}{4 \log n} + O_\epsilon \left(\frac{1}{\log n} \right) \right].$$

Again, $m = \dfrac{n}{4 \log n}$ works.

B. One-Step Synchronous Attraction

We now consider capacity under the constraint that the fundamental memories are not merely fixed points, but that they are also attractors over a specified radius of attraction ρn, with $0 \leq \rho < 1/2$. (We will frequently refer to ρ as the radius of attraction, where no confusion can ensue; ρ is, of course, simply the fractional radius of attraction.) We first consider the case of one-step synchronous attraction where we require that almost all state vectors in a Hamming ball of radius ρn surrounding the fundamental memories are mapped to the corresponding memories in one synchronous step.

We denote by $\mathbf{u}[\alpha] \in \overline{B}(\mathbf{u}^{(\alpha)}, \rho n)$ a state within a Hamming radius ρn of fundamental memory $\mathbf{u}^{(\alpha)}$. We assume that the probe vectors $\mathbf{u}[\alpha]$ are chosen independently, and with uniform probability from the states within the Hamming ball of radius ρn, $0 \leq \rho < 1/2$, centred at the fundamental memory $\mathbf{u}^{(\alpha)}$. For one-step synchronous convergence, we hence require that under a synchronous algorithm, $\mathbf{u}[\alpha] \mapsto \mathbf{u}^{(\alpha)}$, $\alpha = 1,...,m$. Hence, for $i = 1,...,n$, and $\alpha = 1,...,m$, we require that the mn sums

$$X_i^{(\alpha)} = \sum_{j \neq i} \sum_{\beta=1}^{m} u_i^{(\alpha)} u_j [\alpha] \, u_i^{(\beta)} u_j^{(\beta)}$$

are positive. Now, form the sums

$$Y_i^{(\alpha)} = \sum_{j \neq i}{}' u_j [\alpha] \, u_j^{(\alpha)} , \tag{6.5.40}$$

and

$$Z_i^{(\alpha)} = \sum_{j \neq i} \sum_{\beta \neq \alpha} u_i^{(\alpha)} u_j\,[\alpha]\; u_i^{(\beta)} u_j^{(\beta)} \,. \tag{6.5.41}$$

Then, we have

$$X_i^{(\alpha)} = (1-g\,)m \;+\; Y_i^{(\alpha)} + Z_i^{(\alpha)} \,. \tag{6.5.42}$$

Note that the random variables $Z_i^{(\alpha)}$ and $Y_i^{(\alpha)}$ are independent. We will need the following assertion, which is essentially a statement of the fact that the probability mass in Hamming spheres is concentrated on the surface of the spheres, with the interior having very small probability.

Lemma 6.5.3. For every fixed $0 < \delta < 1$, as $n \rightarrow \infty$,

$$\mathbf{P}\left\{ \;|\; Y_i^{(\alpha)} - n\,(1-2\rho)\;|\; > 2n^{\delta} \right\} = O\left(n \left(\frac{1-\rho}{\rho} \right)^{-n^{\delta}} \right) .$$

Proof. Let $E^{(\alpha)}$ be a random variable denoting the total number of component errors in $\mathbf{u}[\alpha]$, i.e., $E^{(\alpha)}$ denotes the Hamming distance between the fundamental memory $\mathbf{u}^{(\alpha)}$, and the probe $\mathbf{u}[\alpha]$. Let $E_i^{(\alpha)}$ denote the number of errors in components $u_j\,[\alpha]$, $j \neq i$. We have $0 \leq E_i^{(\alpha)} \leq \rho n$, and $E^{(\alpha)} - 1 \leq E_i^{(\alpha)} \leq E^{(\alpha)}$. Also,

$$Y_i^{(\alpha)} = n\; - 1 - 2E_i^{(\alpha)} \,.$$

The probability that $Y_i^{(\alpha)}$ takes on value $n\; - 1 - 2e$ is the probability that there were e component errors in the probe with the i-th component being correct, summed with the probability that there were $e\, + 1$ component errors in the probe with the i-th component being in error. Hence,

$$\mathbf{P}\left\{Y_i^{(\alpha)} = n - 1 - 2e\right\} = \sum_{d=0}^{\rho n} \mathbf{P}\left\{E_i^{(\alpha)} = e \mid E^{(\alpha)} = d\right\} \mathbf{P}\left\{E^{(\alpha)} = d\right\}$$

$$= \mathbf{P}\left\{E_i^{(\alpha)} = e \mid E^{(\alpha)} = e\right\} \mathbf{P}\left\{E^{(\alpha)} = e\right\}$$

$$+ \mathbf{P}\left\{E_i^{(\alpha)} = e \mid E^{(\alpha)} = e + 1\right\} \mathbf{P}\left\{E^{(\alpha)} = e + 1\right\} .$$

For $0 \leq e \leq \rho n - 1$,

$$\mathbf{P}\left\{Y_i^{(\alpha)} = n - 1 - 2e\right\} = \frac{n - e}{n} \frac{\binom{n}{e}}{\sum_{\nu=0}^{\rho n} \binom{n}{\nu}} + \frac{e + 1}{n} \frac{\binom{n}{e+1}}{\sum_{\nu=0}^{\rho n} \binom{n}{\nu}}$$

$$= 2 \frac{\binom{n-1}{e}}{\sum_{\nu=0}^{\rho n} \binom{n}{\nu}} .$$

And for $e = \rho n$,

$$\mathbf{P}\left\{Y_i^{(\alpha)} = n - 1 - 2\rho n\right\} = \frac{\binom{n-1}{e}}{\sum_{\nu=0}^{\rho n} \binom{n}{\nu}} .$$

Hence

$$\mathbf{P}\left\{ \left| Y_i^{(\alpha)} - n(1 - 2\rho) \right| > 2n^{\delta}\right\} = \mathbf{P}\left\{E_i^{(\alpha)} < \rho n - n^{\delta} - \tfrac{1}{2}\right\}$$

$$= \frac{2 \sum_{e=0}^{\lfloor \rho n - n^{\delta} \rfloor} \binom{n-1}{e}}{\sum_{\nu=0}^{\rho n} \binom{n}{\nu}}$$

$$\leq 2\rho n \frac{\binom{n-1}{\rho n - n^{\delta}}}{\binom{n}{\rho n}}$$

$$\leq 2\rho^2 n \left(\frac{\rho}{1-\rho}\right)^{n^{\delta}-1}$$

$$= O\left(n \left(\frac{1-\rho}{\rho}\right)^{-n^{\delta}}\right) .$$

□

Thus, as $n \to \infty$, with probability approaching one the probes are chosen from the surface of the Hamming balls of radii ρn surrounding the fundamental memories. Rewriting equation (6.5.42), we will have a row sum violation only if

$$Z_i^{(\alpha)} < -n(1-2\rho) - (1-g)m - [Y_i^{(\alpha)} - n(1-2\rho)] .$$

Set $n_\rho = n(1-2\rho)$. Using lemma (6.5.3), it is now easy to demonstrate that lemmas (6.5.1), and (6.5.2) concerning the distribution of row sum violations continue to hold with n_ρ being substituted for every occurrence of n in the lemmas. The following theorem then follows as before.

Theorem 6.5.2. Let $\delta > 0$ be fixed, and let $0 \leq \rho < 1/2$ be any given radius of attraction. Then, as $n \to \infty$, if:

(1) If $m = \dfrac{n(1-2\rho)^2}{2\log n}\left[1 + \dfrac{\frac{1}{2}\log\log n + 1 - g + \log(\delta\sqrt{4\pi})}{\log n} + o\left(\dfrac{1}{\log n}\right)\right]$,

then the expected number of vectors $\mathbf{u}^{(\alpha)}$ whose entire Hamming sphere of radius ρn is directly attracted to $\mathbf{u}^{(\alpha)}$ is asymptotically $m\ e^{-\delta}$.

(2) If $m = \frac{n\,(1-2\rho)^2}{4\log n}\left[1 + \frac{\frac{3}{4}\log\log n + \frac{1}{2}(1-g) + \frac{1}{2}\log(8\delta\sqrt{2\pi})}{\log n} + o\left(\frac{1}{\log n}\right)\right]$,

then the probability that all m vectors are fixed is asymptotically $e^{-\delta}$.

Corollary 6.5.3. The storage capacity (weak sense) of the outer product algorithm is $\frac{(1-2\rho)^2 n}{2\log n}$ for one-step synchronous attraction over a radius $0 \le \rho < \frac{1}{2}$; specifically, as $n \to \infty$:

(1) For every fixed $0 \le \epsilon < 1$, the expected number of fundamental memories that are attractors over a radius ρ is $m - o(m)$ if $m = \frac{(1-2\rho)^2 n}{2\log n}(1-\epsilon)$.

(2) For every fixed $\epsilon > 0$, the expected number of fundamental memories that are attractors over a radius ρ is $o(m)$ if $m = \frac{(1-2\rho)^2 n}{2\log n}(1+\epsilon)$.

Corollary 6.5.4. The storage capacity (strong sense) of the outer product algorithm is $\frac{(1-2\rho)^2 n}{4\log n}$ for one-step synchronous attraction over a radius $0 \le \rho < \frac{1}{2}$; specifically, as $n \to \infty$:

(1) For every fixed $0 \le \epsilon < 1$, the probability that all the memories are attractors over a radius ρ approaches one as n approaches infinity if $m = \frac{(1-2\rho)^2 n}{4\log n}(1-\epsilon)$.

(2) For every fixed $\epsilon > 0$, the probability that there is at least one fundamental memory which is not an attractor over a radius ρ approaches one as n approaches infinity if $m = \frac{(1-2\rho)^2 n}{4\log n}(1+\epsilon)$.

C. Non-Direct Convergence

There is a loss of a factor of $(1-2\rho)^2$ in the storage capacity of the outer product algorithm, in requiring the fundamental memories to be not only fixed points, but in addition, attract over a ball of radius $0 \le \rho < \frac{1}{2}$ directly, in a single synchronous step, or alternatively, in asynchronous fashion monotonically correcting the erroneous components. Direct convergence places rather stringent constraints upon the system, and we expect that improvements in storage capacity could be effected if multiple synchronous steps (or alternatively, non-direct asynchronous convergence, where occasional erroneous steps are taken) were to be allowed. In point of fact, it turns out that allowing non-direct convergence of this sort results in capacity gains that countervail the losses in capacity that accrued for single-step synchronous attraction.

Fix a small $\rho^* > 0$. If the number of fundamental memories is now chosen to be $m = (1 - 2\rho^*)^2 \frac{n}{4 \log n}$, then by theorem (6.5.2), each fundamental memory attracts directly (one synchronous step, or monotonic asynchronous convergence) over a Hamming sphere of radius $\rho^* n$ surrounding the memory. Let ρ close to, but less than a half, be the desired radius of attraction. Extending lemma (6.5.1) for the one-step synchronous convergence case (i.e., replacing each occurrence of n in equation (6.5.2) by $n_\rho = (1 - 2\rho)^2 n$), we obtain that the asymptotic probability that a single component is erroneously labeled is

$$\tau \sim \frac{1}{\sqrt{2\pi}} \frac{1 - 2\rho^*}{1 - 2\rho} e^{-(1 - 2\rho)(1 - g)} \frac{1}{\sqrt{\log n}} n^{-\frac{(1 - 2\rho)^2}{(1 - 2\rho^*)^2}} \to 0 \text{ as } n \to \infty . \qquad (6.5.43)$$

Consider first the multiple step synchronous case. The probe vector has essentially ρn components incorrectly specified. The first synchronous state transition will map the probe vector to a state where essentially $n \tau$ components are wrong, with high probability. For any fixed ρ^*, however small, we can choose n large enough so that the probability of component misclassification τ from equation (6.5.43) becomes smaller still. Thus, for large enough n, the probe vector will be mapped within the confines of a Hamming sphere of (small) radius ρ^* surrounding the memory. By choice of m as dictated by theorem (6.5.2), we have that the next state transition converges directly to the fundamental memory from almost all states in the Hamming sphere of radius ρ^*, with very high probability. Thus, for every fixed (small) ρ^*, and every choice of attraction radius $\rho < \frac{1}{2}$, however large, we can find n large enough so that states in the Hamming balls of radii ρ surrounding the memories will converge to the corresponding fundamental memories within two synchronous state transitions. Now, keeping ρ fixed we allow ρ^* to approach zero. Thus, for every fixed attraction radius $0 \leq \rho < \frac{1}{2}$, the $(1 - 2\rho)^2$ term can be dropped from the capacity expression for large enough n.

The above result also holds for the asynchronous case. In this case, however, it is possible for state transitions to result in wandering outside the ρn-sphere, where the expression for τ may not be valid. To compensate for this possibly deleterious effect, we resort to the following artifice to ensure that the state transitions are confined within the ρn-sphere with high probability. Fix a small positive quantity $\eta > 0$. Now consider state vectors in the smaller ball of radius $\rho(1 - \eta)n$. For n large enough, we can ensure that τ is smaller than η. Now, starting from the smaller ball, we will be confined within the larger ρn-ball with high probability. The estimate for τ holds good here, so that we can expect ultimate convergence as above.

In fine, the capacity (in the strong sense) of the outer product algorithm for non-direct attraction over any specified radius $0 \leq \rho < \frac{1}{2}$, is $\frac{n}{4 \log n}$. Relaxing the capacity definition to the weak sense, effects a doubling in capacity to $\frac{n}{2 \log n}$.

To summarise, for any fixed radius of attraction $0 \leq \rho < \frac{1}{2}$, and any fixed $0 < \epsilon < 1$, for large enough n:

(1) Almost all of the ρn-sphere around almost all the m fundamental memories may be ultimately expected to be attracted to the correct fundamental memory if the number of memories are less than or equal to $(1 - \epsilon)\frac{n}{2 \log n}$. If the number of memories is further constrained to be less than or equal to $(1 - \epsilon)\frac{n}{4 \log n}$, then with high probability almost all of the ρn-spheres surrounding each fundamental memory are ultimately attracted.

(2) At most an asymptotically negligible fraction of the fundamental memories will even be fixed points if the number of memories exceeds $(1 + \epsilon)\frac{n}{2 \log n}$. If the number

of memories is larger than $(1+\epsilon)\frac{n}{4\log n}$, then with high probability there will be at least one fundamental memory that is not a fixed point.

An important feature of these results is that they are not dependent on the specified attraction radius. We can specify an attraction radius ρ as close to a half as we wish, but the asymptotic capacity results for the case of non-direct convergence continue to hold *in toto*. Of course, the closer ρ is to a half, the larger n will have to be before the predicted behaviour begins to emerge. This is in sharp contradistinction to the nature of the capacity results for direct one-step synchronous attraction. For this case, as we saw, the capacity deteriorates rapidly as the attraction radius ρ increases toward a half. The relative behaviour of the capacities under these different operating conditions gives rise to some interesting consequences.

Let us assume that $(1-2\rho)n$ components of a probe are known to be correct, while the remaining $2\rho n$ components are unreliably known, and are treated as "don't cares." We will assume a forced choice scenario, wherein, values are assigned randomly to the "don't care" components, so that we end up on average with ρn incorrect components, and $(1-\rho)n$ correct components in the probe. A viable procedure would be to proceed by clamping the accurately known components at their known values. This, however, does not really improve performance, as from the preceeding discussion, convergence is pretty rapid in any case.

An alternate approach would be to disable the unknown $2\rho n$ components initially, and then assign them values based upon computations involving solely the correct components. It turns out that this procedure does not affect behaviour for non-direct convergence, as the asymptotic behaviour does not depend upon the actual specified attraction radius. For the case of direct convergence, however, the picture changes somewhat. The capacity where we assume a forced choice for the "don't care" components is $(1-2\rho)^2\frac{n}{4\log n}$. (We adopt the strong sense definition of capacity for simplicity. The results hold equally well for the weak sense definition.) This corresponds to the potential seen by any one neuron having a mean-to-standard deviation ratio of asymptotically $(1-2\rho)\frac{n^{1/2}}{m^{1/2}}$. If the "don't care" components are disabled initially, however, the mean-to-standard deviation ratio goes up to $(1-2\rho)^{1/2}\frac{n^{1/2}}{m^{1/2}}$. Disabling the unreliable input components hence results in an increase in the capacity of the outer product algorithm to $(1-2\rho)\frac{n}{4\log n}$ for attraction over a radius ρ in a single synchronous step.

D. Error Tolerance

The order $\frac{n}{\log n}$ capacities that we have obtained for the outer product algorithm are not too niggardly, especially as they carry with them considerable error correction and attraction capability. From the viewpoint of the number of degrees of freedom used–the n^2 weights–however, the result gives us pause, as the storage does not appear to be very efficient.

One answer to the problem consists in seeking alternative algorithms which pack information much more densely into the neural structure, and in the next chapter we present an algorithm which stores essentially a constant number of bits per interconnection. An alternate way to improve the capacity of the outer product algorithm is to make storage conditions a little less stringent. The culprit in this instance is our requirement that the fundamental memories be fixed points of the system. If we are willing to tolerate errors in recall of the fundamental memories, then the capacity of the outer product algorithm can be greatly increased [8]. We pursue this idea briefly, in this section, and outline the expected capacity under error tolerant conditions. We go into the issue of error tolerant associations in much greater detail in chapter IX.

Let an attraction radius $0 \leq \rho < \frac{1}{2}$ be specified, and let $0 < \epsilon \leq \rho$ be the specified error tolerance. The error tolerance ϵ prescribes (small) Hamming balls of radii ϵ surrounding each memory, and we require that states lying in the (large) Hamming spheres of radius ρ surrounding the memories be mapped into the epsilon ball, and *remain* there. This just formalises the fact that we are willing to tolerate a few component errors in the retrieved memories, and the maximum amount of allowed component errors is ϵn .

Now, as in lemma (6.5.1), the probability that a component is in error with the probe a distance ρn from a memory is given asymptotically with n by

$$\tau \sim \Phi \left\{ -(1-2\rho) \frac{n^{1/2}}{m^{1/2}} \right\} .$$

We assume for simplicity that a synchronous mode is in force, and that the outer product matrix of weights is zero diagonal. The expected number of component errors in each memory per synchronous iteration is hence $n \, \Phi \left\{ -(1-2\rho) \frac{n^{1/2}}{m^{1/2}} \right\}$. We denote this quantity by ρ^*. If $\rho > \rho^*$, then the algorithm is moving in the direction of collapsing the balls of attraction into smaller balls. If $\rho < \rho^*$ on the other hand, the algorithm is proceeding in the direction of stretching the balls of attraction outward. By lemma (6.5.3), we had seen that most of the volume of Hamming balls lie on their surface, with their interiors contributing very little of the volume. Hence, if the attraction balls surrounding the fundamental memories are to be collapsed into a smaller tolerance ball of radius ϵ, we will require equilibrium between the diverse forces stretching and compressing the ball. We will hence require that

$$\epsilon = \Phi \left\{ -(1-2\epsilon) \frac{n^{½}}{m^{½}} \right\} .$$

If a number of memories linear in n, with $m = \kappa n$, is admissibile, and is to result in storage of the m fundamental memories with no more than ϵn errors in components, then for equilibrium we need

$$\epsilon = \Phi \left\{ - \frac{1-2\epsilon}{\sqrt{\kappa}} \right\} .$$

For a given error tolerance ϵ, the above equation can be solved uniquely for κ.

Thus, through the expedient of introducing error tolerance, we can actually increase the storage capacity of the outer product algorithm until it is linear in the number of neurons. The preceeding constant κ is very small, however, and diminishes very quickly as ϵ becomes small.

A note of discord in the happy performance of the algorithm is struck by the fact that, while states in the large ball of attraction are typically mapped to the surface of the small ball of error tolerance, states within the small ball of tolerance are also mapped to the surface of the ball. This is a consequence of the lopsided geometry of Hamming space, where the surfaces of spheres contain essentially all the mass of the sphere. This is illustrated schematically in fig. 5.4 (after Ref. [1]).

References

[1] R. J. McEliece, E. C. Posner, E. R. Rodemich, and S. S. Venkatesh, "The capacity of the Hopfield associative memory," *Conf. on Neural Network Models for Computing*, Santa Barbara, California, April 1985; submitted to *IEEE Trans. Inform. Theory*.

[2] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Natl. Acad. Sci. USA*, vol. 79, pp. 2554–2558, 1982.

[3] K. Nakano, "Associatron-a model of associative memory," *IEEE Trans. Sys., Man, and Cybern.*, vol. SMC-2, pp. 380–388, 1972.

[4] W. A. Little, "The existence of persistent states in the brain," *Math. Biosci.*, vol. 19, pp. 101–120, 1974.

[5] W. A. Little and G. L. Shaw, "Analytic study of the memory storage capacity of a neural network," *Math. Biosci.*, vol. 39, pp. 281–290, 1978.

[6] G. Vichniac, "Exploiting the computational power of Boolean nets," *Conf. on Neural Networks for Computing*, Snowbird, Utah, April 1986.

[7] W. Feller, *An Introduction to Probability Theory and its Applications*, vol. I, 3rd Edition. New York: Wiley, 1968.

[8] E. C. Posner and E. R. Rodemich, "Linear capacity in the Hopfield model," *Conf. on Neural Networks for Computing*, Snowbird, Utah, April 1986.

CHAPTER VII

SPECTRAL APPROACHES AND COMPARISONS

1. TAILORED SPECTRA FOR MEMORY ENCODING

The outer product approach for storing memories is simple, and robust. The memory capacity that we derive from the scheme is, however, not quite as much as we could have hoped. Specifically, in the case of strong attraction where we require to store the fundamental memories as fixed points, the storage capacity of the outer product scheme was seen to be of the order of $\frac{n}{\log n}$ memories, or $\frac{n^2}{\log n}$ bits. (Since each memory consists of n constituent bits.) On the other hand, there are potentially n^2 degrees of freedom available in choosing n^2 weights for a fully interconnected system. The outer product scheme hence has a capacity of the order of $\frac{1}{\log n}$ bits per interconnection. An ever decreasing amount of information (in bits per interconnection) is stored in the interconnections, as the number of interconnections increase. By intuitive degrees of freedom arguments, on the other hand, we might hope to be able to store at least a constant amount of information per interconnection, so that the system is cost effective. (This is particularly important from the point of semiconductor implementations of these networks; as has been long known, the major cost component in large planar VLSI systems is the interconnections [1].) It is hence important from the viewpoint of the cost of additional interconnections that more information be stored in the interconnections than the $O(\frac{1}{\log n})$ bits per interconnection that was obtained for the outer product algorithm for memory encoding.

In this chapter we examine an algorithm reported previously by Venkatesh and Psaltis [2] and Personnaz, et al., [3], for encoding memories, wherein we achieve information storage of the order of 1 bit per interconnection in the neural network by suitably shaping the spectrum of the interconnection matrix of weights.

A. A New Perspective of the Outer Product Scheme

We again assume that m memories $\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \ldots, \mathbf{u}^{(m)} \in \mathbb{B}^n$ have been chosen randomly. For strict stabillity, we require that $(\Delta \circ \mathbf{W})(\mathbf{u}^{(\alpha)}) = \mathbf{u}^{(\alpha)}$ for $\alpha=1,\ldots,m$. Specifically, if $\mathbf{W}\mathbf{u}^{(\alpha)} = \mathbf{v}^{(\alpha)} \in \mathbb{R}^n$, we require that $\operatorname{sgn}(v_i{}^{(\alpha)}) = u_i{}^{(\alpha)}$ for each $i=1,\ldots,n$.

For the outer product scheme for generating the elements of the weight matrix, we have from equation (6.1.1)

$$\left(\mathbf{W}_{op}\,\mathbf{u}^{(\alpha)}\right)_i = \sum_{j=1}^{n} \left(\mathbf{W}_{op}\right)_{ij} u_j{}^{(\alpha)} = \sum_{\substack{j=1\\ j\neq i}}^{n} \sum_{s=1}^{m} u_i{}^{(\beta)} u_j{}^{(\beta)} u_j{}^{(\alpha)}$$

$$= (n-1)u_i{}^{(\alpha)} + \sum_{s\neq r} \sum_{j\neq i} u_i{}^{(\beta)} u_j{}^{(\beta)} u_j{}^{(\alpha)}$$

$$= (n-1)u_i{}^{(\alpha)} + \delta u_i{}^{(\alpha)}\,, \tag{7.1.1}$$

where $\mathbf{E}(\delta u_i{}^{(\alpha)}) = 0$, $\operatorname{Var}(\delta u_i{}^{(\alpha)}) = (n-1)(m-1)$. Hence

$$\frac{\mathbf{E}(\,|\,(n-1)u_i{}^{(\alpha)}\,|\,)}{\left(\operatorname{Var}(\delta u_i{}^{(\alpha)})\right)^{1/2}} = \left(\frac{n-1}{m-1}\right)^{1/2} \rightarrow \infty \text{ as } n \rightarrow \infty\,,$$

where we require that $m = o(n)$ from theorem (6.5.1) so that the memories are stable with high probability. We can hence write

$$\mathbf{W}_{op}\,\mathbf{u}^{(\alpha)} = (n-1)\mathbf{u}^{(\alpha)} + \delta\mathbf{u}^{(\alpha)}$$

where $\delta\mathbf{u}^{(\alpha)}$ has components $\delta u_i{}^{(\alpha)}$ whose contributions are small compared to $u_i{}^{(\alpha)}$, at least in a probabilistic sense. In essence, the memories $\mathbf{u}^{(\alpha)}$ are "*eigenvectors-in-mean*" or "*pseudo-eigenvectors*" of the linear operator $\mathbf{W}_{op}$, with "*pseudo-eigenvalues*" $n-1$.

B. Constructive Spectral Approaches

In this section we demonstrate constructive schemes for the generation of the weight matrix which yield a larger capacity than the outer product scheme at the cost of increased complexity in the construction of the weight matrix, and possible difficulty in updating it if new memories are desired to be stored. (If a small amount of extraneous storage of partial results from the previous stage is present, however, Greville's algorithm [4] allows of a systematic and easy updating of the weight-matrix when a new memory is to be stored.) This construction ensures that the given set of memories is stable under the algorithm; specifically, we obtain linear operators $\mathbf{W}_s$ which ensure that the conditions sgn $\left(\mathbf{W}_s\,\mathbf{u}^{(\alpha)}\right)_i = u_i{}^{(\alpha)}$, $i=1,\ldots,n$, $r=1,\ldots,m$ are satisfied for $m \leq n$. The construction entails an extension of the approach outlined in the previous section so that the memories $\mathbf{u}^{(\alpha)}$ are *true eigenvectors* of the linear operator $\mathbf{W}_s$. Related approaches that have been considered before include those of Kohonen [5], who considers a purely linear mapping which is optimal in the mean-square sense, and Poggio's polynomial mapping technique [6]. Another scheme which seems to be formally related to our approach is the interesting orthogonalization technique proposed by Amari [7].

We now utilize a result due to J. Komlós on binary n-tuples, to establish two results which have a direct bearing on the construction of the weight matrix.

Theorem 7.1.1.

(1) For all randomly chosen binary (-1,1) n-tuples $\mathbf{u}^{(1)},\mathbf{u}^{(2)},\ldots,\mathbf{u}^{(m)} \in \mathbb{B}^n$ with $m \leq n$, define the $n \times m\,(-1,1)$ matrix $\mathbf{U} = \left[\mathbf{u}^{(1)}\mathbf{u}^{(2)}\cdots\mathbf{u}^{(m)}\right]$. Then

$\mathbf{P}\left\{\text{rank}(\mathbf{U}) = m\right\} \to 1$ as $n \to \infty$.

(2) Let Ξ_n be the family of bases for $\mathbb{R}^n$ with all basis elements constrained to be binary n-tuples; (i.e., $E = \left\{\mathbf{e}_1,\mathbf{e}_2, \ldots, \mathbf{e}_n\right\} \in \Xi_n$ iff $\mathbf{e}_1,\mathbf{e}_2, \ldots, \mathbf{e}_n \in \mathbb{B}^n$ are linearly independent). Then asymptotically as $n \to \infty$, *almost all* vectors $\mathbf{u} \in \mathbb{B}^n$ have a representation of the form

$$\mathbf{u} = \sum_{j=1}^{n} \alpha_j \mathbf{e}_j \ , \quad \alpha_j \neq 0 \text{ for each } j=1,\ldots,n \ , \tag{7.1.2}$$

for *almost all* bases E in Ξ_n.

Proof.

(1) This is essentially Komlós' result [8]. Let A_n denote the number of singular $n \times n$ matrices with binary elements (-1,1). Then Komlós demonstrated that

$$\lim_{n \to \infty} \frac{A_n}{2^{n^2}} = 0 \ . \tag{7.1.3}$$

(Komlós' result was for $n \times n$ (0,1) matrices, but it holds equally well for $n \times n$ (-1,1) matrices.) Let $A_{n,m}$ denote the number of $n \times m$ (-1,1) matrices with rank strictly less than m. Then we have that $A_{n,m}\ 2^{n(n-m)} \leq A_n$, so that from equation (7.1.3) we have that $A_{n,m}\ 2^{-nm} \to 0$ as $n \to \infty$. It then follows that asymptotically as $n \to \infty$, almost all $n \times m$ (-1,1) matrices with $m \leq n$ are full rank. This proves the first part of the theorem.

(2) We first estimate the cardinality of Ξ_n :

Let $\Theta_n = \left\{T = \left\{\mathbf{d}_1,\mathbf{d}_2, \ldots, \mathbf{d}_n\right\} \subset \mathbb{B}^n : T \text{ is a linearly dependent set}\right\}$. We have

$$|\Xi_n| = \binom{2^n}{n} - |\Theta_n|$$

$$= \binom{2^n}{n} \left[1 - \frac{n! \ | \Theta_n |}{2^{n^2} \left(1 - \frac{1}{2^n}\right)\left(1 - \frac{2}{2^n}\right) \cdots \left(1 - \frac{n-1}{2^n}\right)} \right] . \tag{7.1.4}$$

Let $T = \{ \mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_n \} \in \Theta_n$. Then $\left[\mathbf{d}_1 \mathbf{d}_2 \cdots \mathbf{d}_n \right]$ is a singular matrix. Each permutation of the column vector $\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_n$ yields another distinct singular matrix, so that $n! \ | \Theta_n | \le A_n$. Using this result with equation (7.1.4) we get

$$1 - \frac{A_n}{2^{n^2}} \frac{1}{\left(1 - \frac{n(n-1)}{2^{n-1}}\right)} \le \frac{| \Xi_n |}{\binom{2^n}{n}} \le 1 .$$

Define the sequence $\{ \kappa_n \}$ by

$$\kappa_n = 1 - \frac{A_n}{2^{n^2}} \frac{1}{\left(1 - \frac{n(n-1)}{2^{n-1}}\right)} . \tag{7.1.5}$$

Then from equation (7.1.3) we have that $\kappa_n \to 1$ as $n \to \infty$.

Define a sequence of random variables $\{ S_n \}_{n=1}^{\infty}$ such that S_n takes on the value zero if a randomly chosen binary n-tuple $\mathbf{u} \in \mathbb{B}^n$ has the representation in a randomly chosen basis $E \in \Xi_n$, and one otherwise. To complete the proof it suffices to show that $E \{ S_n \} = \mathrm{P} \{ S_n = 1 \} \to 0$ as $n \to \infty$.

Fix $\mathbf{u} \in \mathbb{B}^n$, $E \in \Xi_n$, and assume that does not hold. Then $\exists \, j \in \{ 1,2,\ldots,n \}$ s.t. $\alpha_j = 0$. Assume without loss of generality that $\alpha_n = 0$. Then

$$\mathbf{u} = \sum_{j=1}^{n-1} \alpha_j \mathbf{e}_j \ , \quad \alpha_j \ge 0 . \tag{7.1.6}$$

We hence have that $\{\mathbf{e}_1,\mathbf{e}_2, \ldots, \mathbf{e}_{n-1},\mathbf{u}\} \in \Theta_n$. An overestimate for the number of choices of $\mathbf{u}$ and E such that (7.1.6) holds is $\binom{2^n}{n-1}2^n$. Also, the total number of ways that we can choose $E \in \Xi_n$, and $\mathbf{u} \in \mathbb{B}^n$ is $|\Xi_n| \, 2^n$. Hence, from this and equation (7.1.5), we have

$$\mathbf{P}\left\{S_n = 1\right\} \leq \frac{\binom{2^n}{n-1}}{|\Xi_n|} \leq \frac{\binom{2^n}{n-1}}{\binom{2^n}{n}} \frac{1}{\kappa_n} \leq \frac{n\, 2^{-n}}{\kappa_n \left(1 - \frac{n-1}{2^n}\right)} .$$

By definition of κ_n, we now have that $\mathbf{P}\left\{S_n = 1\right\} \to 0$ as $n \to \infty$. □

We use these results to establish the validity of the following schemes for constructing the weight matrix $\mathbf{W}_s$.

Fix $m \leq n$, and let $\lambda^{(1)},\lambda^{(2)}, \ldots, \lambda^{(m)} \in \mathbb{R}^+$ be fixed (but arbitrary) positive real numbers. Let $\mathbf{u}^{(1)},\mathbf{u}^{(2)}, \ldots, \mathbf{u}^{(m)} \in \mathbb{B}^n$ be the m randomly chosen memories to be stored in the memory. Assume $\mathbf{u}^{(1)},\mathbf{u}^{(2)}, \ldots, \mathbf{u}^{(m)}$ are linearly independent (over $\mathbb{R}$).

STRATEGY 1: Define the $m \times m$ diagonal matrix $\mathbf{\Lambda} = \mathbf{dg}[\lambda^{(1)},\lambda^{(2)}, \ldots, \lambda^{(m)}]$, and the $n \times m\,(-1,1)$ matrix of memories $\mathbf{U} = [\mathbf{u}^{(1)}\mathbf{u}^{(2)} \cdots \mathbf{u}^{(m)}]$. Set $\mathbf{W}_s = \mathbf{U\Lambda}\left(\mathbf{U}^T\mathbf{U}\right)^{-1}\mathbf{U}^T$.

STRATEGY 2: Choose any $(n-m)$ vectors $\mathbf{u}^{(m+1)},\mathbf{u}^{(m+2)}, \ldots, \mathbf{u}^{(n)} \in \mathbb{B}^n$ such that the vectors $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(m)},\mathbf{u}^{(m+1)}, \ldots, \mathbf{u}^{(n)}$ are linearly independent. Define the augmented $n \times n$ diagonal matrix $\mathbf{\Lambda}_a$, and the augmented $n \times n\,(-1,1)$ matrix $\mathbf{U}_a$ by $\mathbf{\Lambda}_a = \mathbf{dg}[\lambda^{(1)}, \ldots, \lambda^{(m)},0,\ldots,0]$, and $\mathbf{U}_a = [\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(m)},\mathbf{u}^{(m+1)}, \ldots, \mathbf{u}^{(n)}]$. Set $\mathbf{W}_s = \mathbf{U}_a\mathbf{\Lambda}_a\mathbf{U}_a^{-1}$.

Note that in both strategies, $\left\{\lambda^{(1)},\lambda^{(2)}, \ldots, \lambda^{(m)}\right\}$ is the *spectrum* of the linear operator $\mathbf{W}_s$, and the memories $\mathbf{u}^{(1)},\mathbf{u}^{(2)}, \ldots, \mathbf{u}^{(m)}$ are the corresponding eigenvectors. Furthermore, there is considerable flexibiity in the choice of the $(n-m)$ linearly independent vectors $\mathbf{u}^{(m+1)},\mathbf{u}^{(m+2)}, \ldots, \mathbf{u}^{(n)}$ of strategy 2. (In fact, from theorem (7.1.1), almost all choices of $(n-m)$ vectors will satisfy linear independence asymptotically.) Alternative schemes can also be obtained by combining the two strategies, viz., by choosing fewer than $(n-m)$ additional linearly independent vectors and then using the *pseudo-inverse* scheme of strategy 1 on the augmented matrix of memories. In fact, for $m=n$, the two strategies are identical.

The crucial assumption of linear independence of the memories is vindicated by theorem (7.1.1). Specifically, $\text{rank}(\mathrm{U}) = m$, and $\text{rank}(\mathrm{U}_a) = n$ for almost any choice of memories, so that the inverses are well defined.

C. Examples

We consider the same example as we did for the outer product algorithm. Again $n = 5$, $m = 3$, and the fundamental memories are chosen to be

$$\mathbf{u}^{(1)} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad \mathbf{u}^{(2)} = \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \\ -1 \end{bmatrix}, \quad \mathbf{u}^{(3)} = \begin{bmatrix} -1 \\ 1 \\ -1 \\ -1 \\ -1 \end{bmatrix}.$$

We choose $\mathbf{W}$ such that $\mathbf{u}^{(1)}$, $\mathbf{u}^{(2)}$, and $\mathbf{u}^{(3)}$ are each eigenvectors of $\mathbf{W}$, with common positive eigenvalue $\lambda = 2$. We form $\mathbf{W}$ according to the pseudo-inverse technique of strategy 1. Then

$$\mathbf{W} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

Note that the matrix is symmetric, and that unlike the outer product algorithm, the weight matrix is not necessarily zero diagonal.

It is simple to verify that $\mathbf{u}^{(1)}$, $\mathbf{u}^{(2)}$, and $\mathbf{u}^{(3)}$ are indeed eigenvectors of $\mathbf{W}$ with eigenvalue $\lambda = 2$, so that the three fundamental memories are indeed fixed points. If the diagonal elements are replaced by zeroes to enhance the attraction dynamics, it can be seen that $\mathbf{u}^{(1)}$ and $\mathbf{u}^{(3)}$ remain fixed points. Fundamental memory $\mathbf{u}^{(2)}$ is, however, no more a fixed point.

2. ALGORITHM CHARACTERISATION

A. Features

(1) The memories are fixed points for all spectral strategies.

For strategy 1 we have for each α=1,2,...,m ,

$$(\mathbf{\Delta} \circ \ \mathbf{W})\mathbf{u}^{(\alpha)} = (\mathbf{\Delta} \circ \ (\mathbf{U}\mathbf{\Lambda}(\mathbf{U}^T\,\mathbf{U})^{-1}\mathbf{U}^T\,))\mathbf{u}^{(\alpha)} = \mathbf{\Delta}(\lambda^{(\alpha)}\mathbf{u}^{(\alpha)}) = \mathbf{u}^{(\alpha)}\;,$$

as $\lambda^{(\alpha)}>0$ so that sgn $(\lambda^{(\alpha)}u_i{}^{(\alpha)}) = u_i{}^{(\alpha)}$. Similarly, for strategy 2 we have

$$(\mathbf{\Delta} \circ \ \mathbf{W})\mathbf{u}^{(\alpha)} = (\mathbf{\Delta} \circ \ (\mathbf{U}_a\,\mathbf{\Lambda}_a\,\mathbf{U}_a^{-1}))\mathbf{u}^{(\alpha)} = \mathbf{\Delta}(\lambda^{(\alpha)}\mathbf{u}^{(\alpha)}) = \mathbf{u}^{(\alpha)}\;.$$

Thus the memories are stable whichever strategy is adopted.

(2) The storage capacity of the scheme is n for all strategies.

This follows immediately, as a linear transformation can have at most n linearly independent eigenvectors in an n-dimensional space.

(3) A small number of additional stable states are created by both strategies.

Let us, for simplicity, consider the eigenvalues $\lambda^{(\alpha)}$ to be equal to some value $\lambda > 0$. Let $\Gamma = \mathrm{span}\{\mathbf{u}^{(1)},\mathbf{u}^{(2)}, \ldots , \mathbf{u}^{(m)}\} \subset \mathbb{R}^n$. Clearly, if $\mathbf{u}$ belongs to the restriction of Γ to $\mathbb{B}^n$, then $\mathbf{u}$ is also stable for both strategies. (In fact, $\mathbf{u}$ is also an eigenvector of $\mathbf{W}_s$, with eigenvalue λ.) Since the number of binary vectors that can be constructed as a linear combination of the memories is the same for both strategies, it follows that the number of additional stable states created in this fashion will also be the same. Furthermore, by theorem (7.1.1), there will not be many such stable states created if $m < n$. Note, however, that the number of additional stable states, while small compared to the total number of states, may be quite numerous compared to m. In addition there will be some more stable states created in more or less random fashion in both strategies. Such stable states satisfy the more general stability requirement: sgn $(\mathbf{Wu})_i = u_i$ for each $i=1,...,n$, and are not eigenvectors of the linear operator $\mathbf{W}$.

(4) Both strategies have some capacity for positive recognition of unfamiliar starting states.

Let $\Phi \subset \mathbb{R}^n$ denote the null space of $\mathbf{W}$. For strategy 1, Φ is the orthogonal subspace to Γ, while for strategy 2, $\Phi = \text{span}\{\mathbf{u}^{(m+1)},\mathbf{u}^{(m+2)}, \ldots, \mathbf{u}^{(n)}\}$. If $\mathbf{u} \in \Phi$, we have $\mathbf{Wu} = 0$. Consequently, at least for a synchronous algorithm, $(\mathbf{\Delta} \circ \mathbf{W}_s)$ will iteratively map $\mathbf{u}$ to some vector $\mathbf{u}_n \in \mathbb{B}^n$ for all $\mathbf{u} \in \Phi$ (or else go into a limit cycle). The vector $\mathbf{u}_n$ in this case represents a positive indication that the starting state was not familiar. Again, for both cases, the restriction of Φ to $\mathbb{B}^n$ will have small probability of occurrence as a consequence of theorem (7.1.1).

The storage capacity of all spectral strategies is clearly the same, and in light of the previous remarks we expect them to have similar attraction behaviour. Note, however, that there is a computational advantage in choosing strategy 1 as it involves just a $m \times m$ matrix inversion as opposed to the $n \times n$ matrix inversion required in strategy 2. Further, the choice of the null space in strategy 2 involves careful selection of the additional $(n-m)$ vectors which span the null space; care should be taken to ensure that these vectors are well separated (in a Hamming distance sense) from the memories. Vectors in the null space in strategy 1 are, however, guaranteed to be maximally separated from the memories as they are orthogonal to them. In what follows we assume that we construct $\mathbf{W}$ according to the prescription of strategy 1.

B. Attraction

Having established that storing the memories as eigenstates of the linear operator can increase the storage capacity to n , we now probe the question of whether there exists a region of attraction around each memory, so that the neuronal network functions as a content-addressable memory.

We first examine the behaviour of the system operating in the synchronous mode. Let $\mathbf{u}^{(\alpha)}$ be a stable state of the system, and let $\mathbf{u} = \mathbf{u}^{(\alpha)} + \delta\mathbf{u}$ be a vector such that $||\delta\mathbf{u}|| << ||\mathbf{u}^{(\alpha)}|| = \sqrt{n}$. Then we have $\mathbf{W}\mathbf{u} = \mathbf{W}\mathbf{u}^{(\alpha)} + \mathbf{W}\delta\mathbf{u}$. As $\mathbf{W}$ is a linear transformation, $\exists k$ s.t. $||\mathbf{W}\delta\mathbf{u}|| \leq k\,||\delta\mathbf{u}||$, so that for $||\delta\mathbf{u}||$ sufficiently small, the perturbation caused by the term is small compared to $||\mathbf{W}\mathbf{u}^{(\alpha)}|| = \lambda^{(\alpha)}\sqrt{n}$. Thus, for small enough $\delta\mathbf{u}$, we expect the vector $\mathbf{u}$ to be mapped onto $\mathbf{u}^{(\alpha)}$ by the algorithm. We would then anticipate that there exists a small region of attraction around the stable state $\mathbf{u}^{(\alpha)}$.

To expand on this theme, we assume that we construct a weight-matrix using the pseudo-inverse scheme of strategy 1. Let us also assume that the spectrum of $\mathbf{W}$ is chosen to be m-fold degenerate, so that $\lambda^{(1)} = \lambda^{(2)} = \cdots = \lambda^{(m)} = \lambda > 0$. Then we claim that $||\mathbf{W}\mathbf{x}|| \leq \lambda||\mathbf{x}||$ for all $\mathbf{x} \in \mathbb{R}^n$. To see this we write $\mathbf{x}$ in the form $\mathbf{x} = \mathbf{x}_1 + \mathbf{x}_2$, where $\mathbf{x}_1 \in \Gamma$, and $\mathbf{x}_2 \in \Phi$. (Recall that we defined $\Gamma = \mathrm{span}\{\mathbf{u}^{(1)},\mathbf{u}^{(2)}, \ldots , \mathbf{u}^{(m)}\}$, and Φ was the orthogonal subspace to Γ.) Then $\mathbf{W}\mathbf{x} = \mathbf{W}\mathbf{x}_1 = \lambda\mathbf{x}_1$. Also $||\mathbf{x}||^2 = ||\mathbf{x}_1||^2 + ||\mathbf{x}_2||^2 \geq ||\mathbf{x}_1||^2$. Hence, $||\mathbf{W}\mathbf{x}|| = \lambda||\mathbf{x}_1|| \leq \lambda||\mathbf{x}||$.

Now, if $\mathbf{u}^{(\alpha)}$ is a stable state of the system, and $\mathbf{u} = \mathbf{u}^{(\alpha)} + \delta\mathbf{u}$, then $\mathbf{W}\mathbf{u} = \lambda\mathbf{u}^{(\alpha)} + \mathbf{W}\delta\mathbf{u}$, so that $\mathbf{u}$ will be mapped onto $\mathbf{u}^{(\alpha)}$ by the adaptation algorithm only if the perturbation term $\mathbf{W}\delta\mathbf{u}$ is sufficiently small. As a measure of the strength of the perturbation, we define the *signal-to-noise ratio* (SNR) by $\dfrac{||\mathbf{W}\mathbf{u}^{(\alpha)}||}{||\mathbf{W}\delta\mathbf{u}||} = \dfrac{\lambda\sqrt{n}}{||\mathbf{W}\delta\mathbf{u}||}$; a high SNR implies that the perturbation term is weak, and conversely. From the discussion in the preceding paragraph, we have that the SNR $\geq \dfrac{\sqrt{n}}{||\delta\mathbf{u}||}$. If d denotes the Hamming distance between $\mathbf{u}$ and $\mathbf{u}^{(\alpha)}$, then $||\delta\mathbf{u}|| = 2\sqrt{d}$. For vectors $\mathbf{u}$ in the immediate neighbourhood of $\mathbf{u}^{(\alpha)}$, we have

$d << n$. We hence obtain a large SNR which is lower bounded by $\frac{\sqrt{n}}{2\sqrt{d}}$, which is indicative of a small perturbation term (compared to the "signal" term). It hence follows, insofar as we accept the SNR to be an accurate barometer of attraction behaviour, that the stable state $\mathbf{u}^{(\alpha)}$ exercises a region of attraction around it.

For asynchronous operation, a more direct argument can be supplied for the existence of a flow in the state space towards stable states. We will work with variants of the matrix $\mathbf{W}$ chosen according to strategy 1 from the last section, and utilize a mode of analysis patterned after that of Hopfield [9]. The energy functional corresponding to a particular state $\mathbf{u}$ is

$$E = -\frac{1}{2}\left\langle \mathbf{u} , \mathbf{W}\mathbf{u} \right\rangle = -\frac{1}{2} \sum_{i,j=1}^{n} w_{ij}\, u_i\, u_j \ .$$

The change in energy corresponding to a single bit change in the k-th position, $u_k \rightarrow u_k + \delta u_k$ where $\delta u_k \in \left\{-2,2\right\}$ is

$$\Delta E = -\frac{1}{2}\, \delta u_k \sum_{j=1}^{n} w_{kj}\, u_j - \frac{1}{2}\, \delta u_k \sum_{i=1}^{n} w_{ik}\, u_i - 2w_{kk} \ . \tag{7.2.1}$$

We first consider the case where the spectrum of $\mathbf{W}$ is m-fold degenerate, so that $\lambda^{(1)} = \lambda^{(2)} = \cdots = \lambda^{(m)} = \lambda > 0$. In this case we have $\mathbf{W}^T = \mathbf{U}(\mathbf{U}^T\mathbf{U})^{-1}\mathbf{\Lambda}\mathbf{U}^T = \mathbf{U}\mathbf{\Lambda}(\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T = \mathbf{W}$, so that $\mathbf{W}$ is symmetric.

Let us first consider the case where we render the diagonal elements of $\mathbf{W}$ to be zero so that $w_{kk} = 0$. The first and second terms for ΔE are identical; further, the algorithm for changing states ensures that δu_k and $\sum w_{kj}\, u_j$ have the same sign, so that ΔE is non-positive, and the asynchronous algorithm hence proceeds in a direction towards decreasing the energy. As the energy is a bounded functional, we have convergence to a stable state.

Relaxing the zero-diagonal restriction we see that the flow is again towards the local energy minima centered at the memories, but with an added perturbation term due to w_{kk}. Let $(\mathbf{U}^T\mathbf{U})^{-1} = [a_{rs}]$. Then $w_{kk} = \lambda \sum_{r=1}^{n} a_{rr} + \lambda \sum_{r=1}^{n} \sum_{\substack{s=1 \\ s \neq r}}^{n} a_{rs}\, u_k{}^{(\alpha)} u_k{}^{(\beta)}$. The diagonal terms, a_{rr}, of $(\mathbf{U}^T\mathbf{U})^{-1}$ are positive because $(\mathbf{U}^T\mathbf{U})^{-1}$ is symmetric positive definite. Hence w_{kk} consists of a strong positive term and a perturbation term with zero mean. Consequently ΔE is typically less than zero. Also note that $\mathbf{W}$ is non-negative definite so that the energy is always non-positive. All vectors in the null space of $\mathbf{W}$ have zero energy so that the flow in state space is away from these vectors. Vectors in the null space hence constitute *repellor states.*

The above argument demonstrates that the asynchronous algorithm will typically generate flows in state space that minimize the energy functional. By an argument similar to that for the outer product scheme, we can show that the energy attains (local) minima at stable memories, so that a region of attraction around the memory is established. In the general case, however, this does not preclude the possibility of lower energy stable states being incidentally created close to a memory, so that the attractive flow in the region is dominated by the extraneous stable state. For the case of the m-fold degenerate spectral scheme, however, this does not happen, and, in fact, *global energy minima are formed at the memories.* We demonstrate this in what follows.

For each memory $\mathbf{u}^{(\alpha)}$, the energy is given by

$$E \triangleq E(\mathbf{u}^{(\alpha)}) = -\frac{1}{2}\left\langle \mathbf{u}^{(\alpha)}, \mathbf{W}\mathbf{u}^{(\alpha)} \right\rangle = -\frac{\lambda n}{2}.$$

Let $\mathbf{u} \in \mathbb{B}^n$ be arbitrary. We can write $\mathbf{u}$ in the form $\mathbf{u} = \sum_{r=1}^{m} \alpha^{(\alpha)}\mathbf{u}^{(\alpha)} + \mathbf{u}_0$, where $\mathbf{u}_0$ is a vector in the orthogonal subspace to the space spanned by the m memories $\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \ldots, \mathbf{u}^{(m)}$, and $\alpha^{(\alpha)}$ are real scalars. Then the energy is given by

$$E(\mathbf{u}) = -\frac{1}{2}\left\langle \mathbf{u}, \mathbf{W}\mathbf{u} \right\rangle = -\frac{1}{2}\left\langle \sum_{r=1}^{m} \alpha^{(\alpha)}\mathbf{u}^{(\alpha)} + \mathbf{u}_0 , \sum_{r=1}^{m} \lambda\alpha^{(\alpha)}\mathbf{u}^{(\alpha)} \right\rangle$$

$$= - \lambda || \sum_{r=1}^{m} \alpha^{(\alpha)} \mathbf{u}^{(\alpha)} ||^2$$

$$\geq - \frac{1}{2} \lambda ||\mathbf{u}||^2 = - \frac{\lambda n}{2} ,$$

since $||\mathbf{u}||^2 = || \sum_{r=1}^{m} \alpha^{(\alpha)} \mathbf{u}^{(\alpha)} ||^2 + ||\mathbf{u}_0||^2$ by the Pythagorean theorem. We've hence established that $E(\mathbf{u}) \geq E(\mathbf{u}^{(\alpha)})$ for any choice of vector in $\mathbb{B}^n$ so that the contention that the memories form global energy minima in the spectral scheme is established when the spectrum is m-fold degenerate. This is not true in general for the outer product scheme.

C. Modifications and Spectral Choice

We saw in the last section that, with a choice of constant, positive eigenvalues, we obtain the desired flow in state space towards the memories. However, as m approaches n, the number of extraneous stable states created also increases. For $m < n$ the number of these stable states is still small compared to the total number of states–though perhaps large compared to m–by virtue of theorem (7.1.1); for $m = n$ however, *all* the states become stable, and $\mathbf{W}$ is exactly the identity matrix.

It is thus advantageous (even when $m < n$) to put in some scatter in the eigenvalues so as to reduce the number of additional stable states created. (Linear combinations of eigenvectors will no longer be eigenvectors as the spectrum is no longer degenerate.) The choice of eigenvalues affects the radius of attraction of the individual stored memories, and an optimal choice of eigenvalues involves a trade-off between storage capacity and the radius of attraction around each memory in order to satisfy the requirements of a specific problem. In the next section, we investigate certain *ad hoc* methods for selecting the eigenvalues which yield good experimental results.

When the scatter of the eigenvalues is small compared to their mean value, λ, we expect the flow in state space to be still towards the minimization of energy. Essentially, if $| \lambda^{(\alpha)} - \lambda | \leq \epsilon$, where ϵ is small compared to λ, then there is an additional perturbation term in equation (7.2.1) for the change in energy, and this term is no larger than $(\epsilon/\lambda) \sum | w_{kj} |$. For small enough perturbation of the eigenvalues (small ϵ), we do not expect a substantial change in the overall flow in state space towards minimizing energy. At the same time, however, an improvement in the attraction radius is effected as the number of additional stable states created is expected to decrease. In the following we argue heuristically why we expect this to be true.

The additional stable states that are incidentally created will be scattered around the state space of the system, and regions of attraction will be consequently formed around them. If such a state is formed close to one of the memories, the region of attraction centered on this incidental stable state will compete with the region of attraction centered on the memory, with a consequent decrease in the attraction radius of that memory. Decreasing the number of additional stable states created by introducing a small amount of scatter in the eigenvalues results in a smaller probability that additional stable states are created close to the memories. Thus, we expect that *introduction of a small amount of scatter in the eigenvalues improves overall performance.*

For large perturbations of the eigenvalues around their mean, the energy functional is no longer appropriate for the description of the flow in state space. The memories, however, are still stable, and exercise a small region of attraction around them. If $\mathbf{u} \in \mathbb{B}^n$, we again write $\mathbf{u} = \sum_{r=1}^{m} \alpha_r \mathbf{u}^{(\alpha)} + \mathbf{u}_0$, where $\mathbf{u}_0$ lies in the null space of $\mathbf{W}$. We then have $\mathbf{W}\mathbf{u} = \sum_{r=1}^{m} \alpha_r \lambda^{(\alpha)} \mathbf{u}^{(\alpha)}$. It can be seen from the above expression that the memories corresponding to larger eigenvalues tend to dominate the flow in state space. To quantify this a little, we rewrite $\mathbf{u}$ as $\mathbf{u}^{(\alpha)} + \delta\mathbf{u}$. Let the Hamming distance between $\mathbf{u}^{(\alpha)}$ and $\mathbf{u}$ be d. Again, as $\mathbf{W}$ is a linear operator, $\exists$ k s.t. $||\mathbf{W}\mathbf{x}|| \leq k \, ||x|| \; \forall \, \mathbf{x} \in \mathbb{R}^n$. This yields a lower bound for the signal-to-noise ratio: $\mathrm{SNR} = \frac{||\mathbf{W}\mathbf{u}^{(\alpha)}||}{||\mathbf{W}\delta\mathbf{u}||} \geq \frac{\lambda^{(\alpha)}\sqrt{n}}{2k\sqrt{d}}$. The signal-to-noise ratio is lower bounded by a quantity that is inversely proportional to the square root of the Hamming distance, and directly proportional to the eigenvalue. Hence, for a given Hamming distance, increasing the eigenvalue improves the SNR, and hence improves the attraction radius. Thus, in general, we have that *the radius of attraction increases as the corresponding eigenvalue increases.*

In the next section we consider a simple *ad hoc* technique for introducing a small degree of scatter in the eigenvalues. The spread in the eigenvalues is obtained by using the correlations (inner products) between the memories. Let ρ_{rs} denote the inner product between the memories $\mathbf{u}^{(\alpha)}$ and $\mathbf{u}^{(\beta)}$, i.e., $\rho_{rs} = \left\langle \mathbf{u}^{(\alpha)}, \mathbf{u}^{(\beta)} \right\rangle = \sum_{i=1}^{n} u_i^{(\alpha)} u_i^{(\beta)}$. We then choose the eigenvalues $\lambda^{(\alpha)}$ according to the prescription

$$\lambda^{(\alpha)} = n - \frac{\sum_{s \neq r} \rho_{rs}}{m-1} \quad , r = 1, \ldots, m \ . \tag{7.2.2}$$

The rationale behind the above scheme is as follows: ρ_{rs} is a measure of the distance between the memories $\mathbf{u}^{(\alpha)}$, and $\mathbf{u}^{(\beta)}$. Specifically, ρ_{rs} achieves its maximum value of n when the memories are identical, and its minimum value of $-n$ when the memories are negations of one another; a value of $\rho_{rs} = 0$ indicates that the memories are $n/2$ apart. If two of the memories are close to each other, (i.e., ρ_{rs} is large and positive), we have from equation (7.2.2) that the corresponding eigenvalues would be roughly equal. As a consequence, neither memory will dominate the attractive flow in state space, so that both memories would have comparable radii of attraction; i.e., one memory will not poach upon the region of attraction of the other. (Conversely, if the memories are far apart, a relatively large disparity in the corresponding eigenvalues is possible, but this would not seriously affect the attraction radii as the memories are well separated.) The choice of eigenvalues according to the above "correlation" method tends to decrease the eigenvalues corresponding to those memories that are close (in Hamming distance) to many other memories, and to increase the eigenvalues corresponding to those memories which are remote from most of the other memories. (Note that there exists a one-to-one correspondence between the Hamming distance, d_{rs}, and the inner product, ρ_{rs}, between two binary vectors, $\rho_{rs} = n - 2d_{rs}$, so that equation (7.2.2) can be formulated equally well in terms of the Hamming distance between the memories.)

We now demonstrate that the scatter in the eigenvalues introduced by the method of equation (7.2.2) is small. We have for r -1,..,m ,

$$E\left\{\lambda^{(\alpha)}\right\} = n - \frac{1}{m-1}\sum_{s\neq r} E\left\{\rho_{rs}\right\} = n \ ,$$

and

$$\mathrm{Var}\left\{\lambda^{(\alpha)}\right\} = \frac{1}{(m-1)^2} E\left\{\sum_{s\neq r}\sum_{t\neq r} \rho_{rs}\,\rho_{rt}\right\} = \frac{n}{m-1} \ .$$

(Recall that the memories are assumed to be samples taken from independent sequences of Bernoulli trials.) The eigenvalues are identically distributed random variables, with mean n , and variance $\frac{n}{m-1}$. The mean-to-standard deviation is hence given by $\sqrt{n\,(m-1)}$, so that the expected scatter is small for large n . Note that the scatter in the eigenvalues decreases as the number of memories stored increases. As very small perturbations in the eigenvalues would not affect the behaviour, the rate of decrease of the scatter of the eigenvalues could be reduced as m becomes large by, for example, replacing $(m-1)$ by $\sqrt{m-1}$ in equation (7.2.2).

Equation (7.2.2) suggests a simple method of introducing scatter into the eigenvalues. Other methods are of course clearly possible, whereby we could pay more attention to high correlation terms as these are potentially more damaging.

3. COMPUTER SIMULATIONS

From the results of the previous sections, the storage capacity of the spectral algorithm is seen to be considerably more than that of the outer product scheme. A question at issue in determining their relative performance is what the attraction radius is for the two schemes, and how rapidly it dwindles with increases in the number of memories stored. Sharp analytical bounds have been difficult to arrive at in the spectral case–in part because of the difficulty of appropriate statistical modeling.

If a Gaussian conjecture holds for the weight matrix in this case, results similar to that for the outer product scheme can be derived; the larger storage capacity of the spectral scheme would then imply that the attraction radius would tail off more slowly with increase in the number of memories stored.

Trends observed in computer simulations have bolstered the above intuitive supposition that the increased storage capacity of the spectral approach (vis-à-vis the outer product scheme) results in significantly improved performance as an associative memory. Systems with state vectors of 32 and 64 bits were considered for simulations on a digital computer, and the algorithms were run in both synchronous and asynchronous fashion. (In order to expedite processing time the asynchronous algorithms were not run by updating individual bits in random fashion; rather, state changes were made by altering the state of that bit for which the gradient of the energy function $-\ \delta E$ in equation (7.2.1)–was a maximum. This ensured that the asynchronous procedure converged quickly along the contour of steepest descent.) The memories were chosen using a binomial pseudo-random number generator. Vectors at any specified Hamming distance, d_H, from any given memories were obtained by randomly choosing d_H bits out of the n bits comprising the memory, and reversing their sign. Weight matrices for the spectral scheme were generated for each given set of memories by using the pseudo-inverse strategy, while for the outer products scheme the sum of Kronecker products expression of equation was used. We encapsulate some of the trends observed in the computer simulations in the following discussion. Some figures are also presented to complement the discussion.

We utilize as our performance measure the Hamming radius of attraction corresponding to a given number of memories. All the plots were generated for a typical memory, and a typical set of "error" vectors chosen in a random fashion at the various Hamming distances indicated on the plots. As the plots represent a typical sample rather than a statistical average, some deviations from monotonicity are visible in the figures. However, simulations on a variety of memories with different choices of error vectors indicate that the plots (sans the fluctuations) are quite representative of the average attraction behaviour of memories under the algorithm.

For comparison with the outer product scheme, a pseudo-inverse spectral strategy with an m-fold degenerate spectrum was used; the eigenvalues, $\lambda^{(\alpha)}$, were all chosen equal to n. The subsequent results were found to be largely independent of which memory was considered, and what the original choice of memories was. The memories in all cases exhibited strong qualitative and quantitative similarity in their attraction behaviour, with well-nigh the same radii of attraction.

For a small number of memories, m, the performance of the two schemes is roughly the same. Almost invariably, though, the spectral strategy showed a slightly larger radius of attraction, but this was not significant for the range of n considered between 32 and 64. The observed radius of attraction for the case of small m was a sizeable fraction of n (approximately $n / 2$ for the range of n considered).

As m increases, the performance of the outer product scheme deteriorates much more rapidly than that of the spectral scheme. For m large enough, (about $m=6$ for $n=32$, and $m=12$ for $n=64$), the outer product scheme becomes overloaded, and the memories themselves are not stable any longer; at this point the spectral scheme still shows a sizeable radius of attraction around each memory. For the range of n between 32 and 64 considered, memories stored using the spectral method exhibited some attraction behaviour for m up to the order of $n/2$. Of course, in all cases the memories themselves were stable up to $m = n$ for the spectral scheme.

These results are illustrated for a typical memory in figures 7.1-7.4. In figures 7.1 and 7.2, comparative plots of radii of attraction versus the number of memories are made between the two schemes for synchronous and asynchronous modes of operation for the case of $n = 32$. A similar plot is made in fig. 7.3 for the case of $n = 64$ for the synchronous mode. For these plots the number of memories, m, is plotted along the x-axis, and the radius of attraction plus one is plotted along the y-axis; a value of $y = 0$ indicates that the memory was not stable, a value of $y = 1$ indicates that the memory was stable but there was no attraction of nearest-neighours, and a value of $y = j$ indicates attraction within a Hamming radius of $j-1$. In fig. 7.4 the largest number of memories for which there was at least a unit radius of attraction was plotted against n, for n lying in the range 32 to 64. The linear nature of the relationship, with $m = n/2$ is clear, for this range of n. We conjecture that this

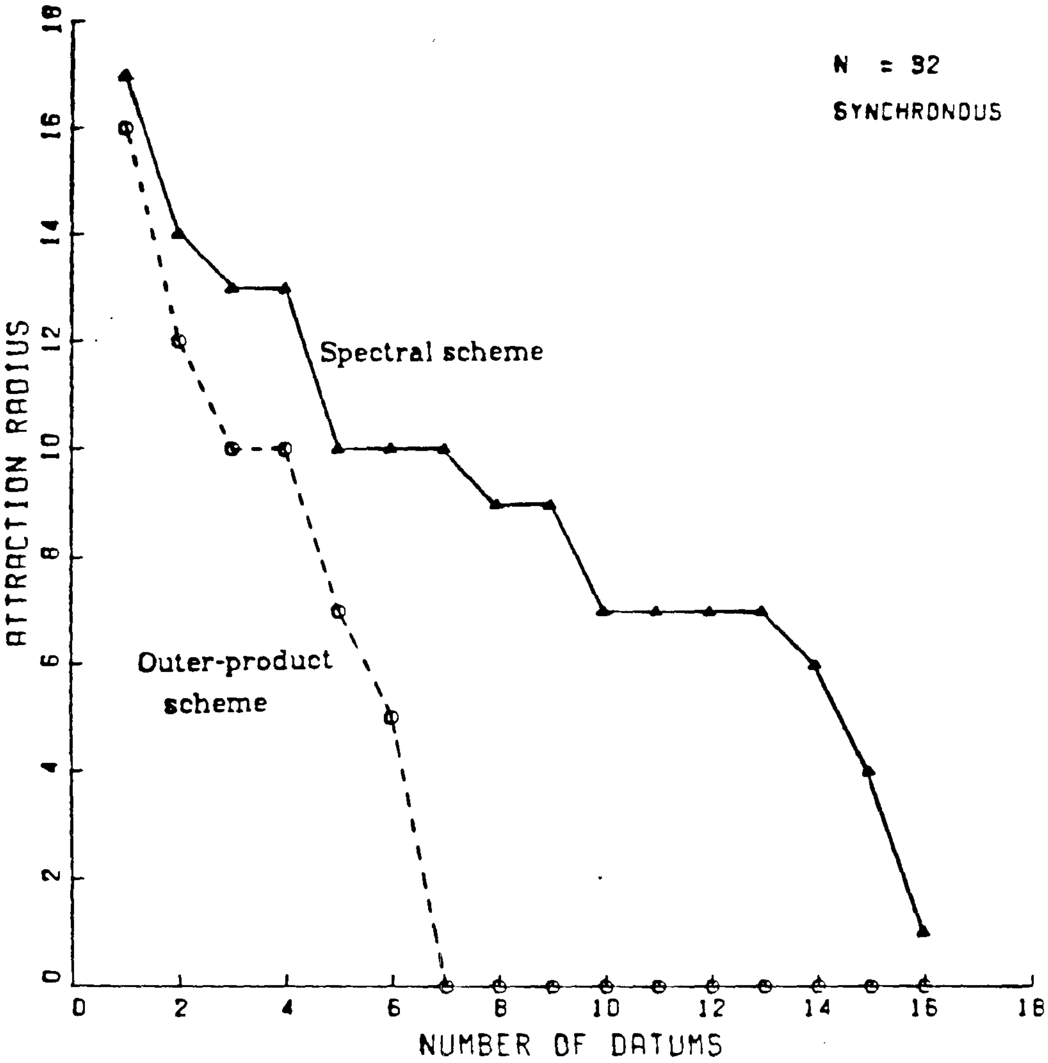

Fig. 7.1. Outer product scheme compared with spectral scheme using equal eigenvalues.

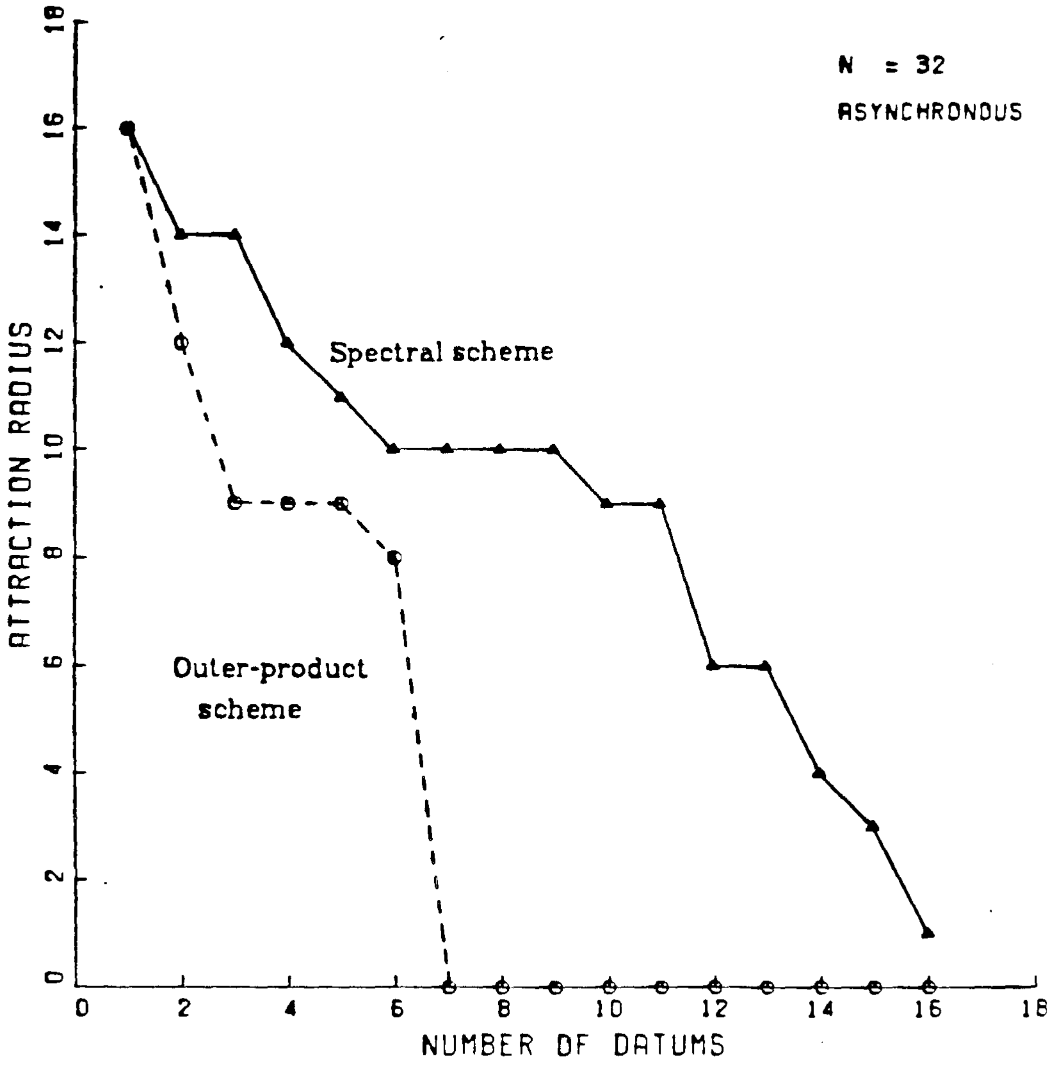

Fig. 7.2. Outer product scheme compared with spectral scheme using equal eigenvalues.

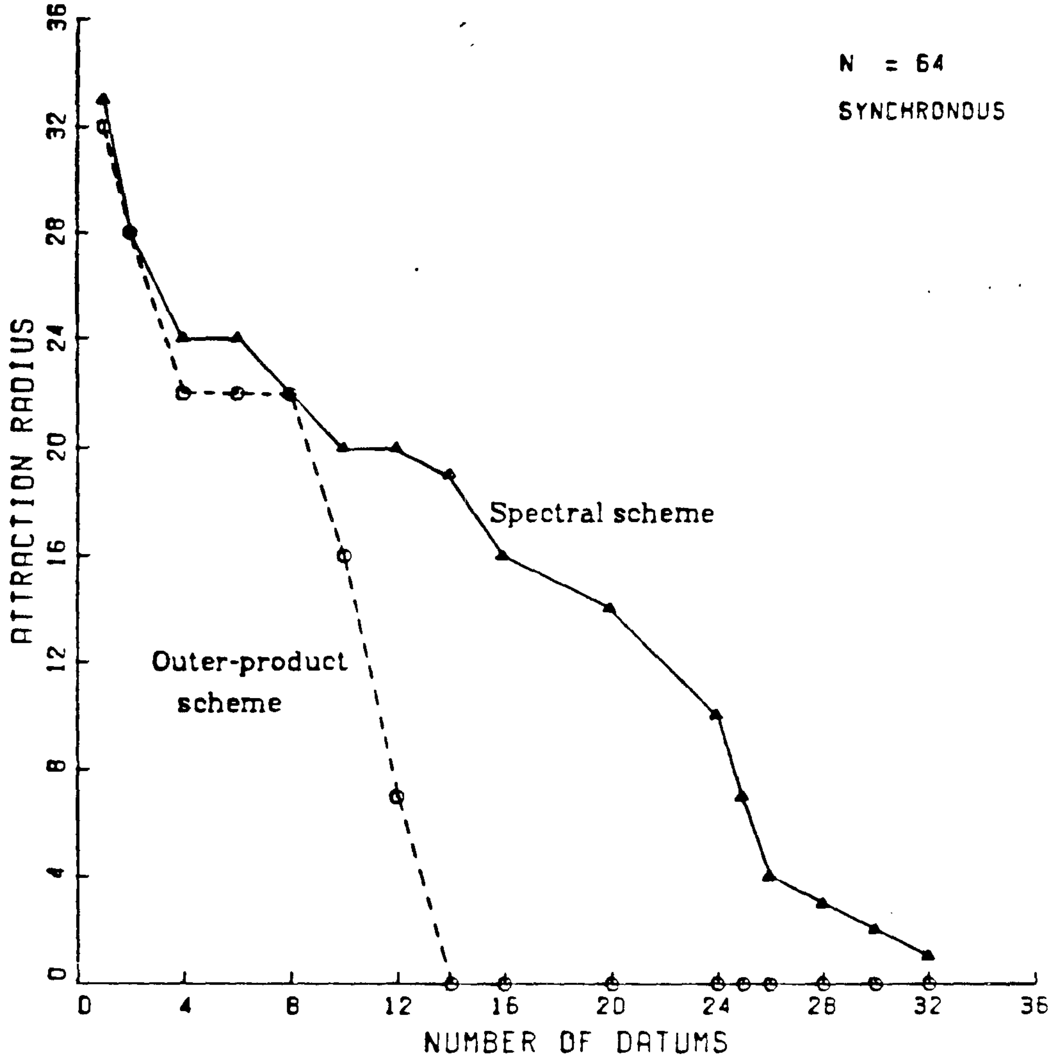

Fig. 7.3. Outer product scheme compared with spectral scheme using equal eigenvalues.

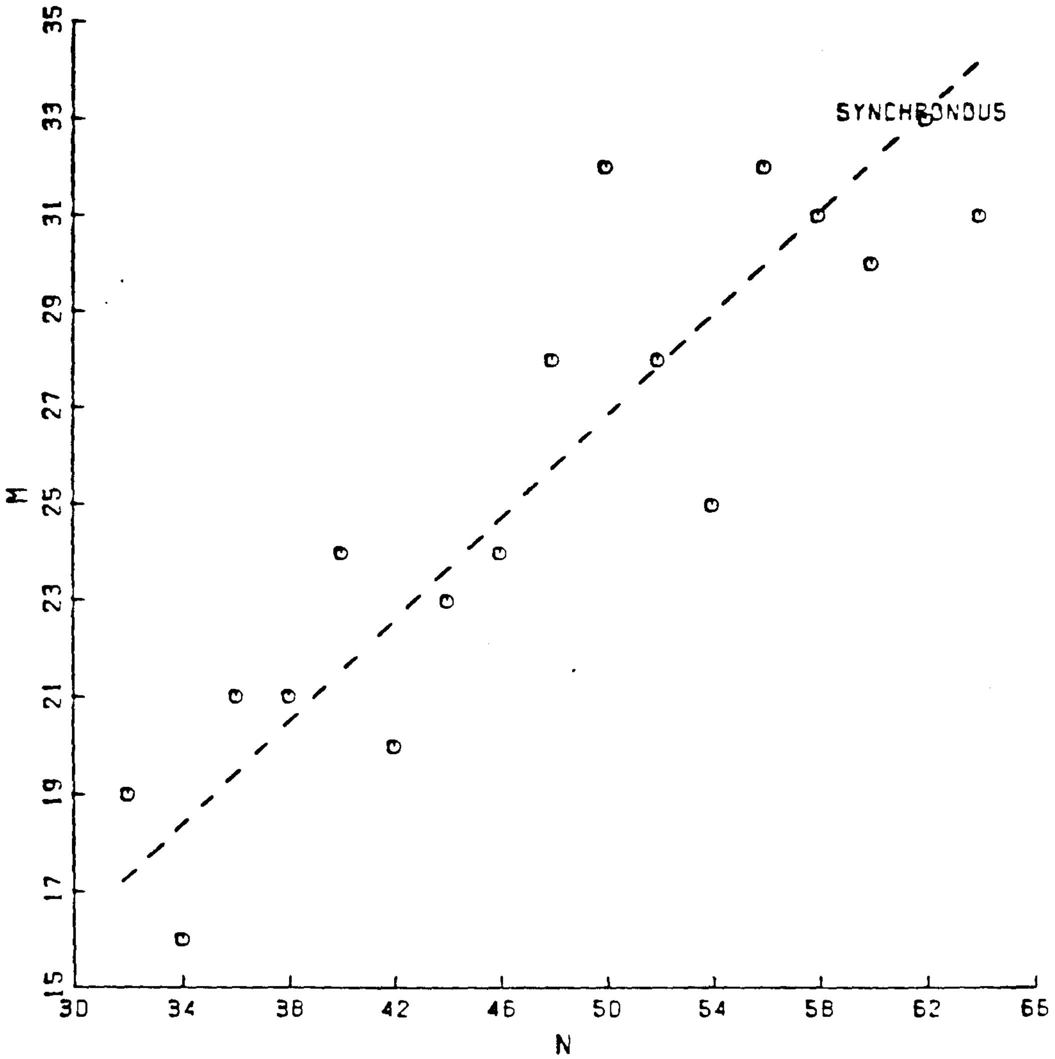

Fig. 7.4. Number of memories ,m , that can be stored in the spectral scheme with attraction over unit distance plotted vs. the number of dimensions, n .

result holds for all n .

Surprisingly enough, the performance of the synchronous and the asynchronous algorithms was virtually identical for both schemes, as illustrated for the typical example of figures 7.1 and 7.2. In general, it was found that the asynchronous procedure enjoyed about a single bit of advantage in attraction radius over the synchronous procedure for the range of n considered. For synchronous processing, convergence of the state vector adaptation process was, in general, seen to be more rapid in almost every case for the spectral scheme. In what follows we will not explicitly differentiate between the results obtained by synchronous processing, and asynchronous processing, as both procedures were seen to yield essentially the same attraction radius for both the spectral scheme, and the outer product scheme.

The radius of attraction decreases monotonically with the number of memories stored for both schemes. Specifically, converging states which were furthest removed from the memories were the first to cease to converge as additional memories were stored. Weak stability (where convergence is to states close to the memories rather than the memories themselves) was observed when the number of memories stored was large for both schemes. Regular trends were not observed in the attraction behaviour of weakly stable states. In particular, the monotonic decrease in the radius of convergence with increase in the number of memories stored was not seen; storage of new memories was sometimes seen to wipe out whole blocks of weakly stable states, and to create new weakly stable states. Overloading the outer product scheme (so that the memories themselves are not stable) was seen to result in the creation of strongly stable states which were not in general close to the memories.

Ringing in the form of state cycles $A \rightarrow B \rightarrow A$, and similar, more complicated state cycles, was observed on occasion for the outer product scheme, but never for the spectral scheme. The instance of ringing in the outer product scheme was found to be relatiyely more frequent for the case of synchronous operation than for asynchronous operation. In general, the number of cases of ringing became more frequent as the number of memories stored increased.

In order to test the robustness of the scheme to changes in the weight matrix, we considered a modified spectral weight matrix whose elements were thresholded to have binary values. Even for this extreme distortion of the weight matrix, the scheme was essentially still functional. The storage capacity was seen to decrease, but memories could still be stored–up to the (diminished) storage capacity–as stable states with attractor-like behaviour. The radius of attraction corresponding to a particular number of stored memories was also seen to decrease. Ringing or state space oscillation was noted in many cases.

Comparisons with thresholded versions of the outer product algorithm showed some superiority in attraction radius for the spectral algorithm in the cases considered, with qualitative similarity to the behaviour for the unthresholded case. The net effect of thresholding in both schemes was evidenced in a (small) decrease in the storage capacity of the respective algorithms, and creation of considerably more locally stable states than in the original algorithms. In all cases simulated, the thresholded spectral strategy was seen to perform considerably better than the thresholded outer product scheme. Comparative plots are shown for the two schemes in figures 7.5, 7.6, and 7.7 for a typical memory, and a random choice of "error" vectors around the memory. Note that there is considerably more fluctuation in the behaviour of the curve than was seen in figures 7.1 and 7.2 for the unthresholded case. The fluctuations are indicative of the fact that many more incidental stable states are created in the thresholded versions of the two algorithms, with some of them being quite close to the memories. Any particular choice of memory hence displays certain preferred directions of attraction directed away from incidentally stable states close to the memory. Vectors which differ from the memory along these preferred distances will hence be attracted at larger Hamming distances than vectors which lie between the memory and another locally stable state which is close to the memory.

We also considered some variants in the spectral approach with a view to improving performance: using a zero-diagonal spectral strategy, and considering the use of non-equal eigenvaules.

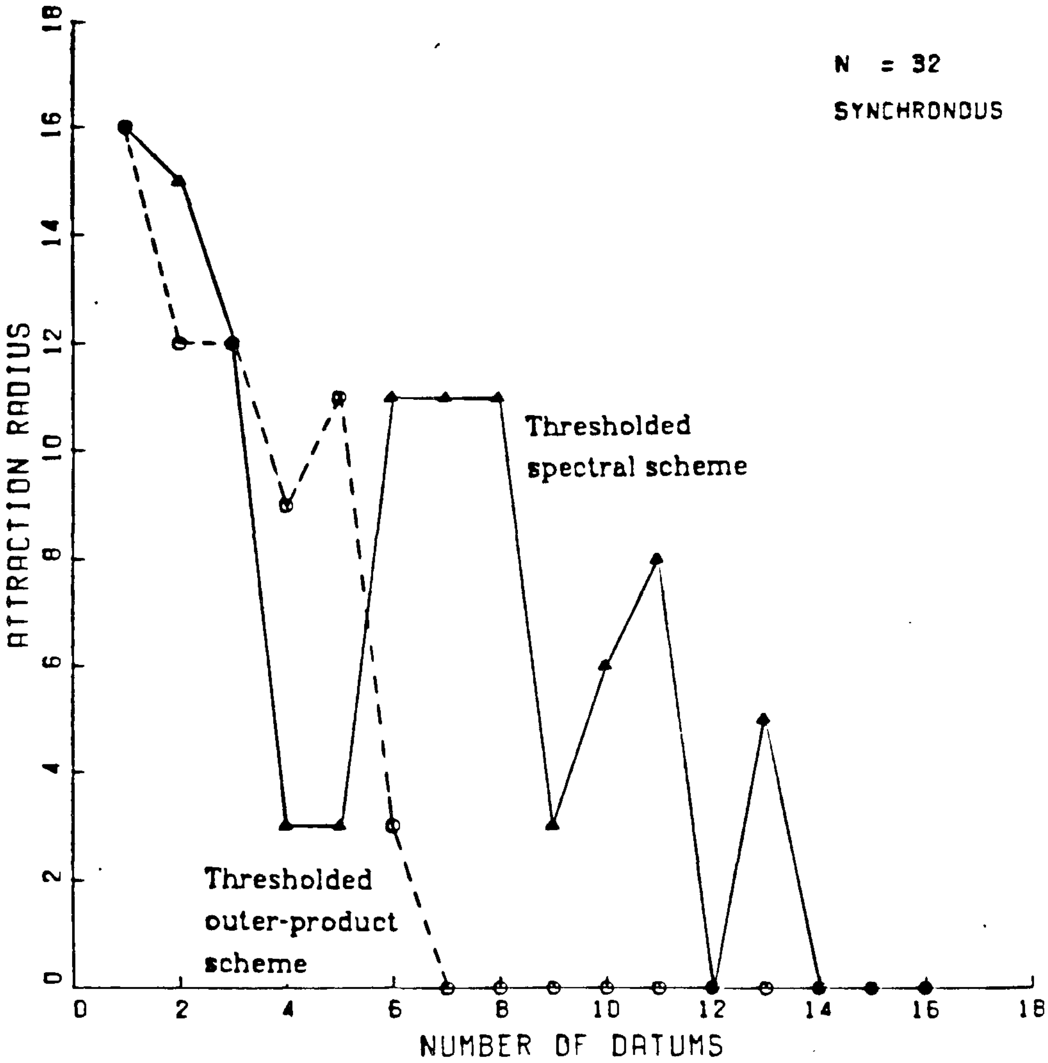

Fig. 7.5. Thresholded outer product scheme compared with thresholded spectral scheme using equal eigenvalues.

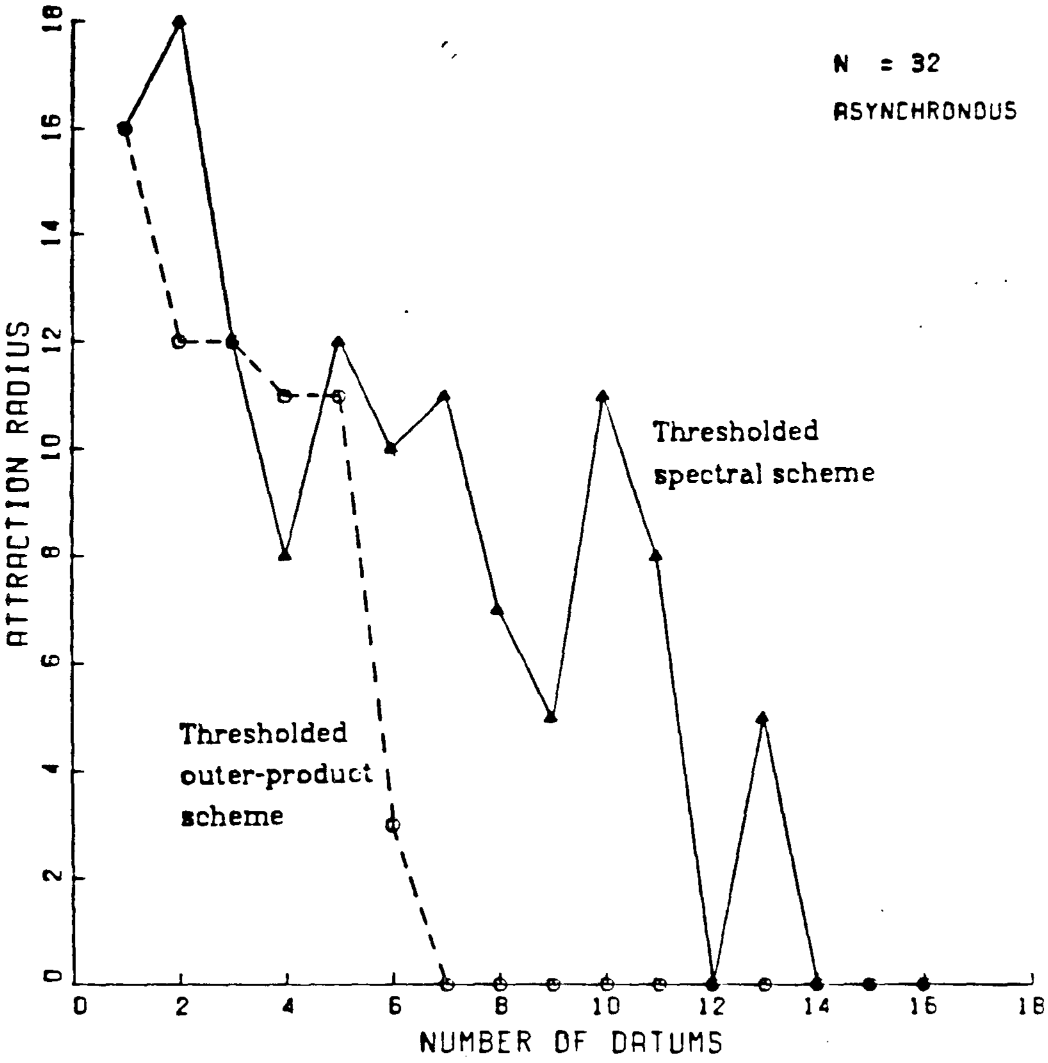

Fig. 7.6. Thresholded outer product scheme compared with thresholded spectral scheme using equal eigenvalues.

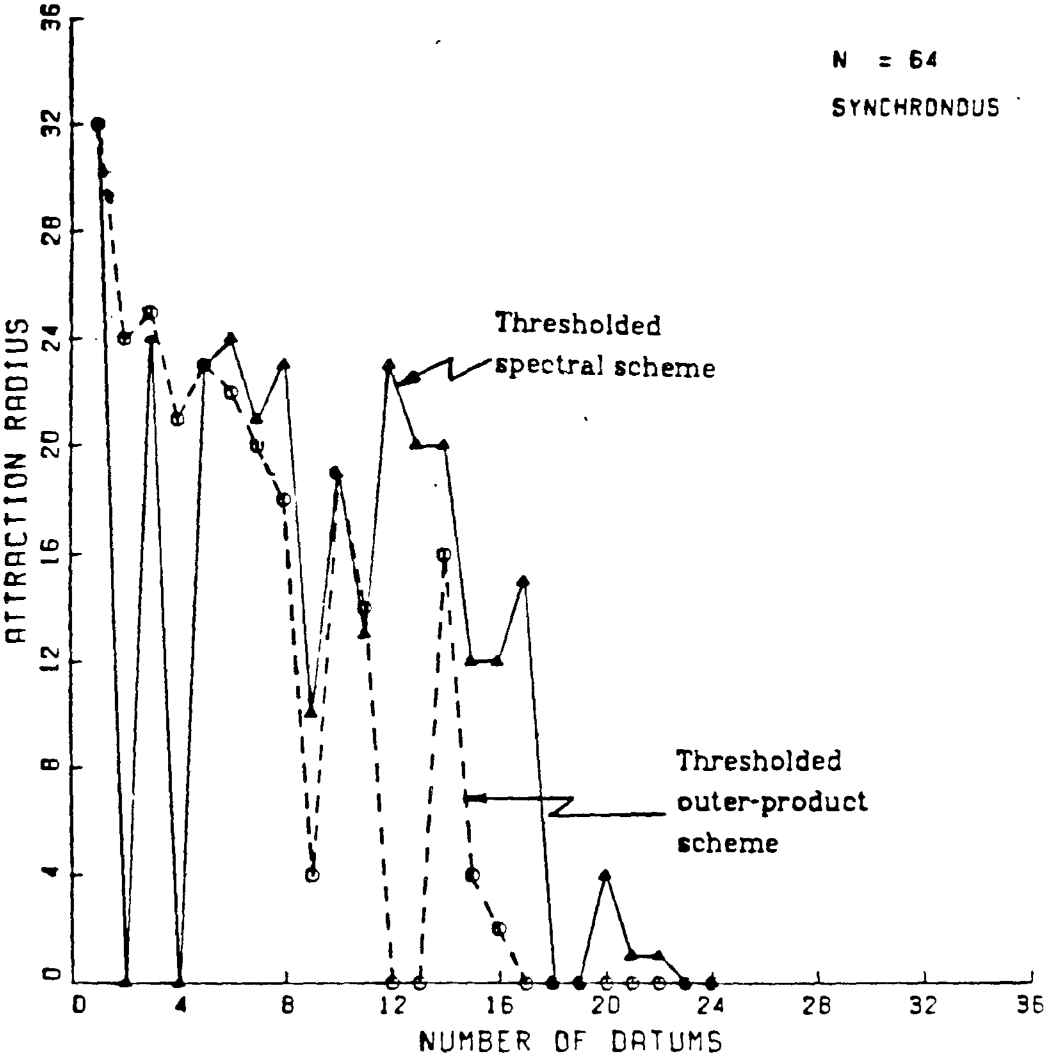

Fig. 7.7. Thresholded outer product scheme compared with thresholded spectral scheme using equal eigenvalues.

Rendering the diagonal of the spectral weight matrix zero: For small m, the performance was found to be largely unaffected in that the attraction radius remained essentially the same. Slight increases in the attraction radii were generally seen in most cases. As the number of memories increased, the improvement in performance of the zero-diagonal spectral scheme was seen to become more marked, especially for the asynchronous algorithm. Again, as the number of memories increased, the radius of convergence decreased. State space oscillations were noted in some cases, the synchronous algorithm exhibiting considerably more cases of ringing than the asynchronous algorithm.

As espoused at the end of the last section, non-equal eigenvalues may be used with advantage in the spectral scheme. We considered first the "correlation method" for choosing the eigenvalues as given in equation (7.2.2). This corresponds to small perturbations of the eigenvalues around their mean. Finally, in order to determine the effect of the eigenvalues on the attraction radius corresponding to each memory, we considered larger perturbations of the eigenvalues, around their mean value.

Correlation method for choice of eigenvalues: The eigenvalues $\lambda^{(\alpha)}$ were chosen according to the prescription of equation (7.2.2) as $\lambda^{(\alpha)} = n - \frac{\sum\limits_{s \neq r} \rho_{rs}}{m-1}$, where ρ_{rs} is the inner product between the memories $\mathbf{u}^{(\alpha)}$, and $\mathbf{u}^{(\beta)}$. For such a choice of eigenvalues, the attraction radius was seen to increase slightly in most of the cases simulated. Again, no state space oscillations were detected, and the asynchronous algorithm functioned marginally better than the synchronous algorithm. Of all the methods implemented on the computer, combinations of the above technique with the diagonal of the weight matrix restricted to zero were seen to yield the best performance. Comparative plots for this variant of the spectral strategy and the spectral strategy with degenerate spectrum are shown in figures 7.8, 7.9 and 7.10 for a typical memory.

Perturbations of the eigenvalues: Small, random perturbations did not have a significant effect on performance. Decreasing a memory's eigenvalue (relative to the mean), in general, caused a decrease in the corresponding radius of attraction. When the eigenvalue was decreased sufficiently, the radius of attraction shrank to zero, and

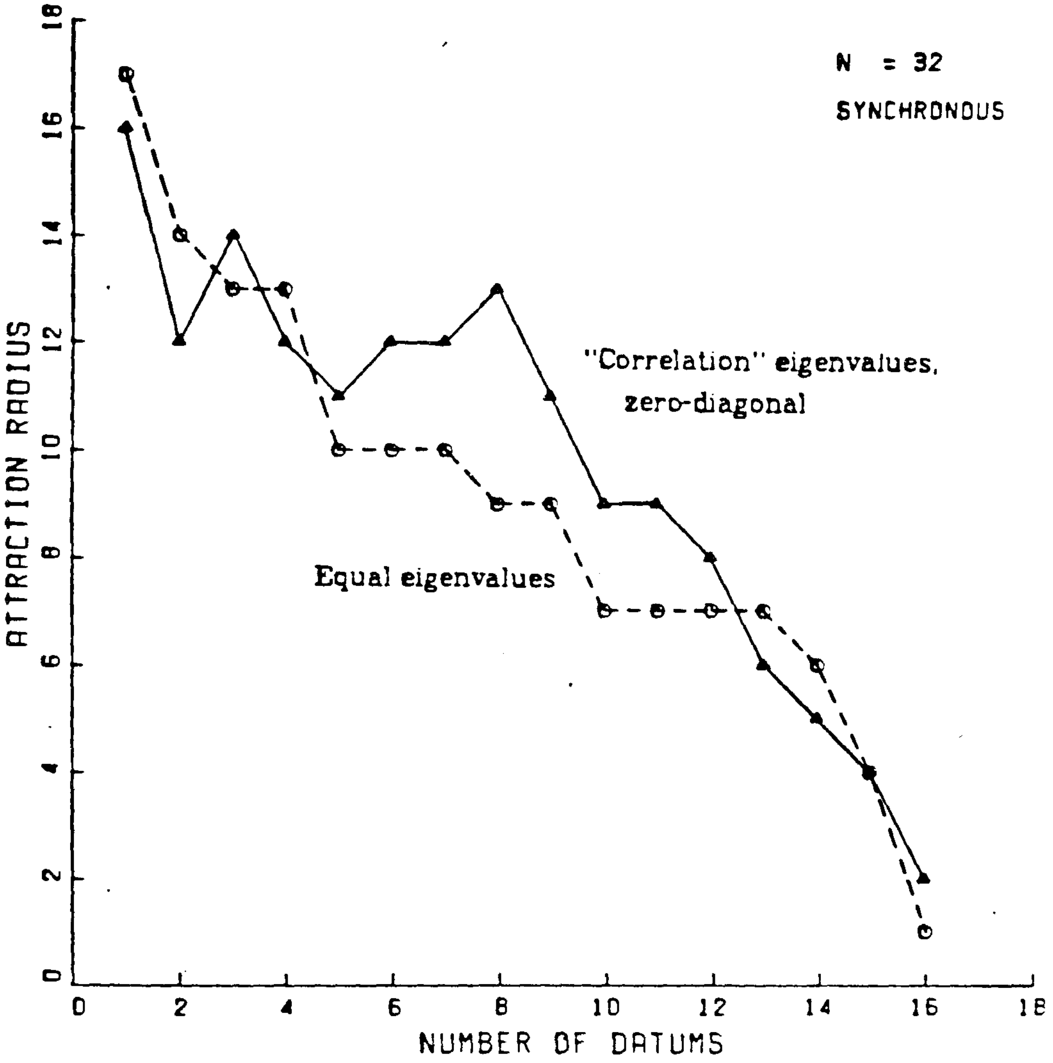

Fig. 7.8. Comparison of two spectral schemes: Usage of equal eigenvalues compared with a "correlation" choice of eigenvalues combined with a zero-diagonal.

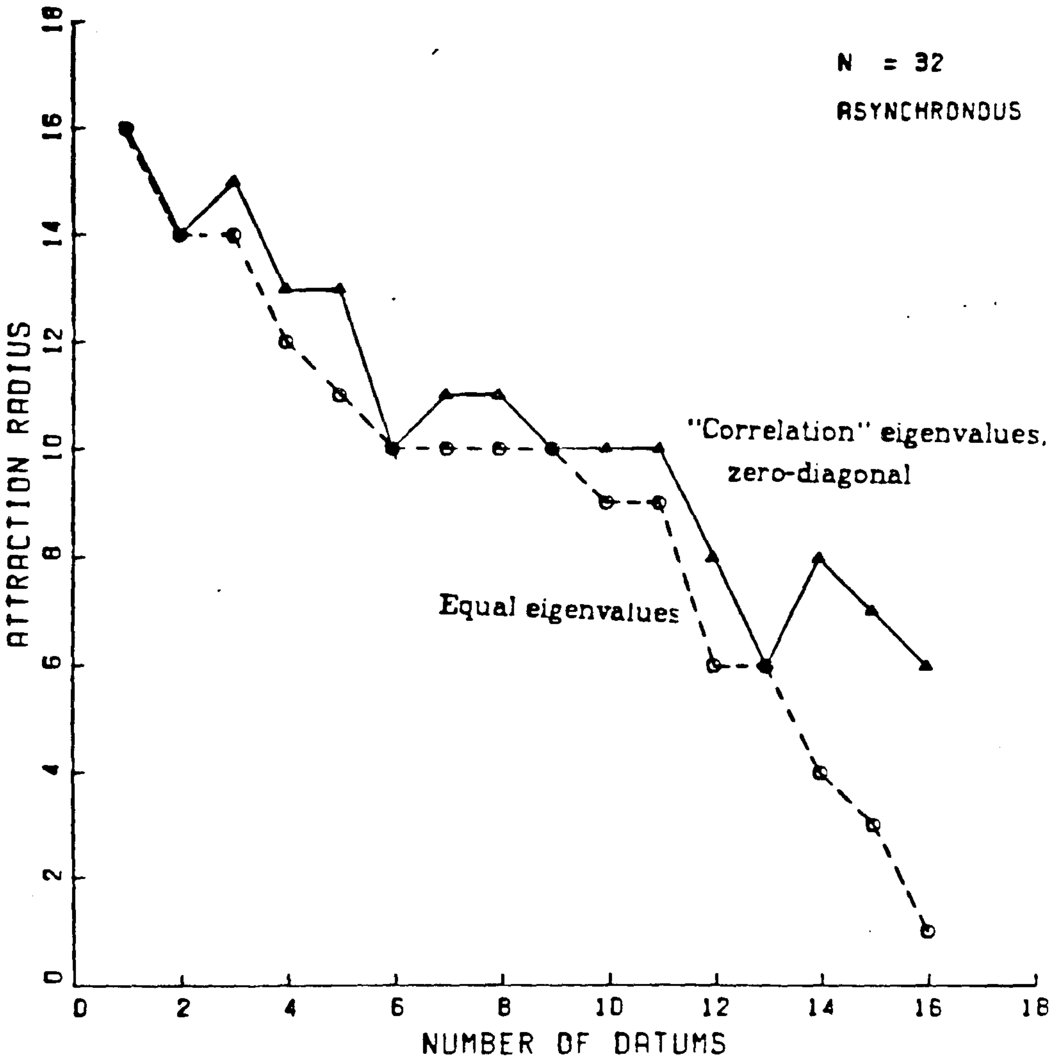

Fig. 7.9. Comparison of two spectral schemes: Usage of equal eigenvalues compared with a "correlation" choice of eigenvalues combined with a zero-diagonal.

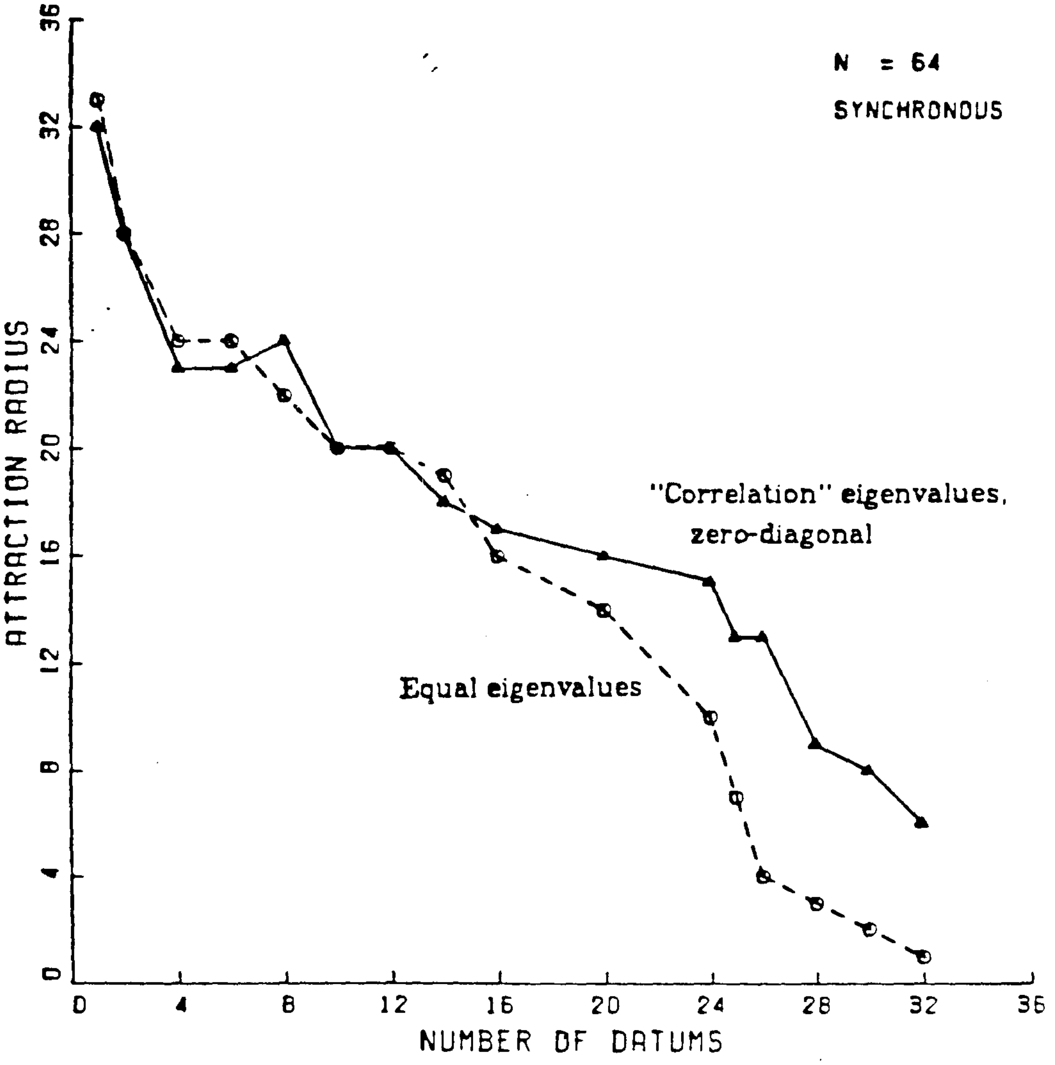

Fig. 7.10. Comparison of two spectral schemes: Usage of equal eigenvalues compared with a "correlation" choice of eigenvalues combined with a zero-diagonal.

thereafter until the eigenvalue reached zero the memory was stable, but was not an attractor. Increasing a memory's eigenvalue sufficiently was seen to increase the radius of attraction in general. (Again, small changes in eigenvalue did not affect performance.) Beyond a certain point the increase in attraction radius with increase in the corresponding eigenvalue was seen to saturate. It was also seen that if all other memories were far away from the test memory, then good performance could be achieved even with a small eigenvalue, and the deterioration in performance with decrease of the eigenvalue was slow. Conversely, if other memories were close to the test memory, the deterioration in performance with decrease in eigenvalue was more precipitate. These effects are in accordance with our expectations as seen in the previous section. Figures 7.11 and 7.12 illustrate the effect of changing the eigenvalue upon the attraction radius of a single memory.

References

[1] I. E. Sutherland and C. A. Mead, "Microelectronics and computer science," *Scientific American*, vol. 237, pp. 210–228, 1977.

[2] S. S. Venkatesh and D. Psaltis, "Information storage and retrieval in two associative nets," *Conf. on Neural Network Models for Computing*, Santa Barbara, California, April 1985; submitted to *IEEE Trans. Inform. Theory.*

[3] L. Personnaz, I. Guyon, and G. Dreyfus, "Information storage and retrieval in spin-glass like neural networks," *Jnl. Physique Lett.*, vol. 46, pp. L359–L365, 1985.

[4] T. N. E. Greville, "Some applications of the pseudoinverse of a matrix," *SIAM Rev.*, vol. 2, pp. 15–22, 1960.

[5] T. Kohonen, *Associative Memory: A System-Theoretical Approach.* Berlin, Heidelberg: Springer Verlag, 1977.

[6] T. Poggio, "On optimal nonlinear associative recall", *Biol. Cybern.*, vol. 19, pp.

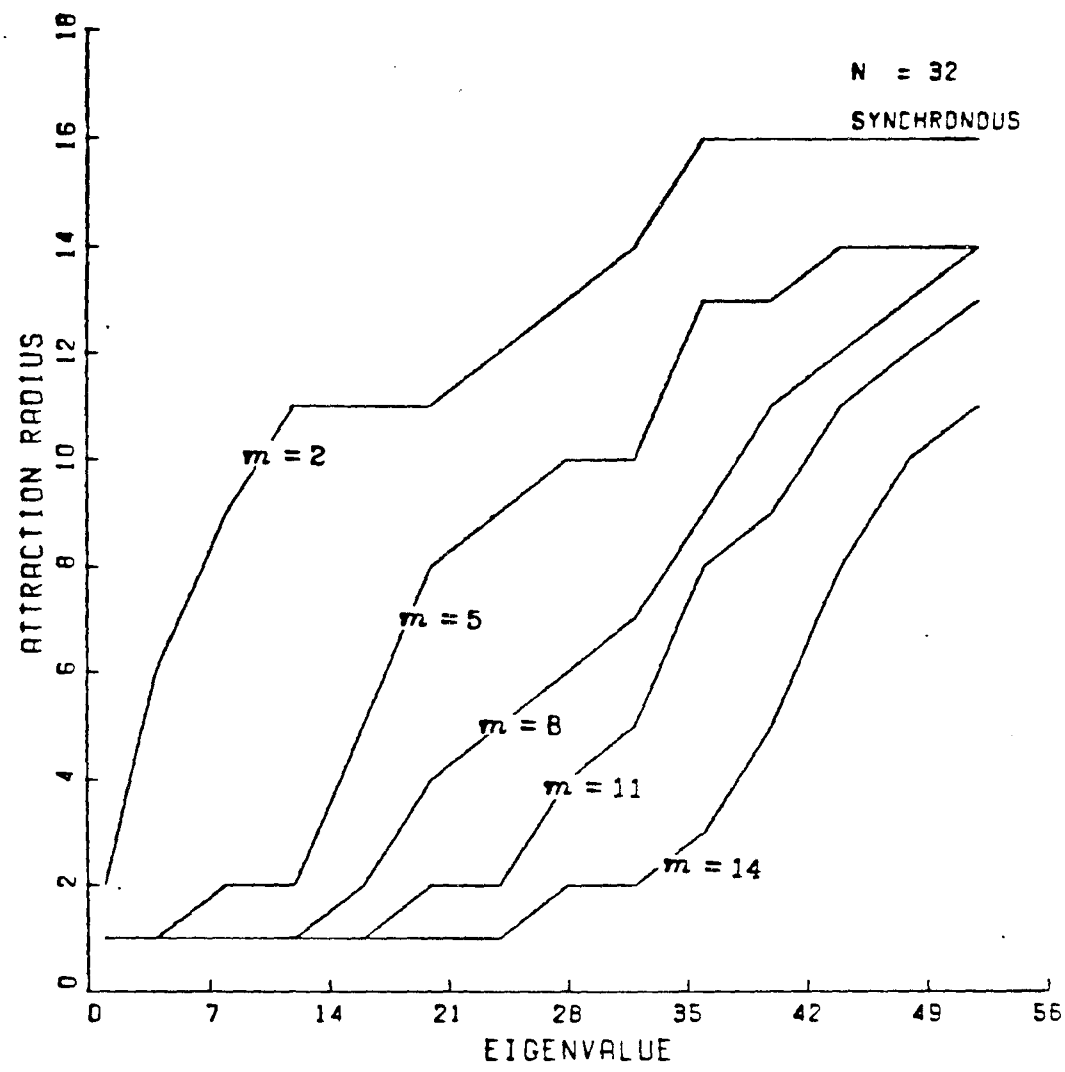

Fig. 7.11. Attraction radius of a typical memory plotted as a function of the eigenvalue. The number of memories is a parameter, and the eigenvalues of all other memories are equal to n .

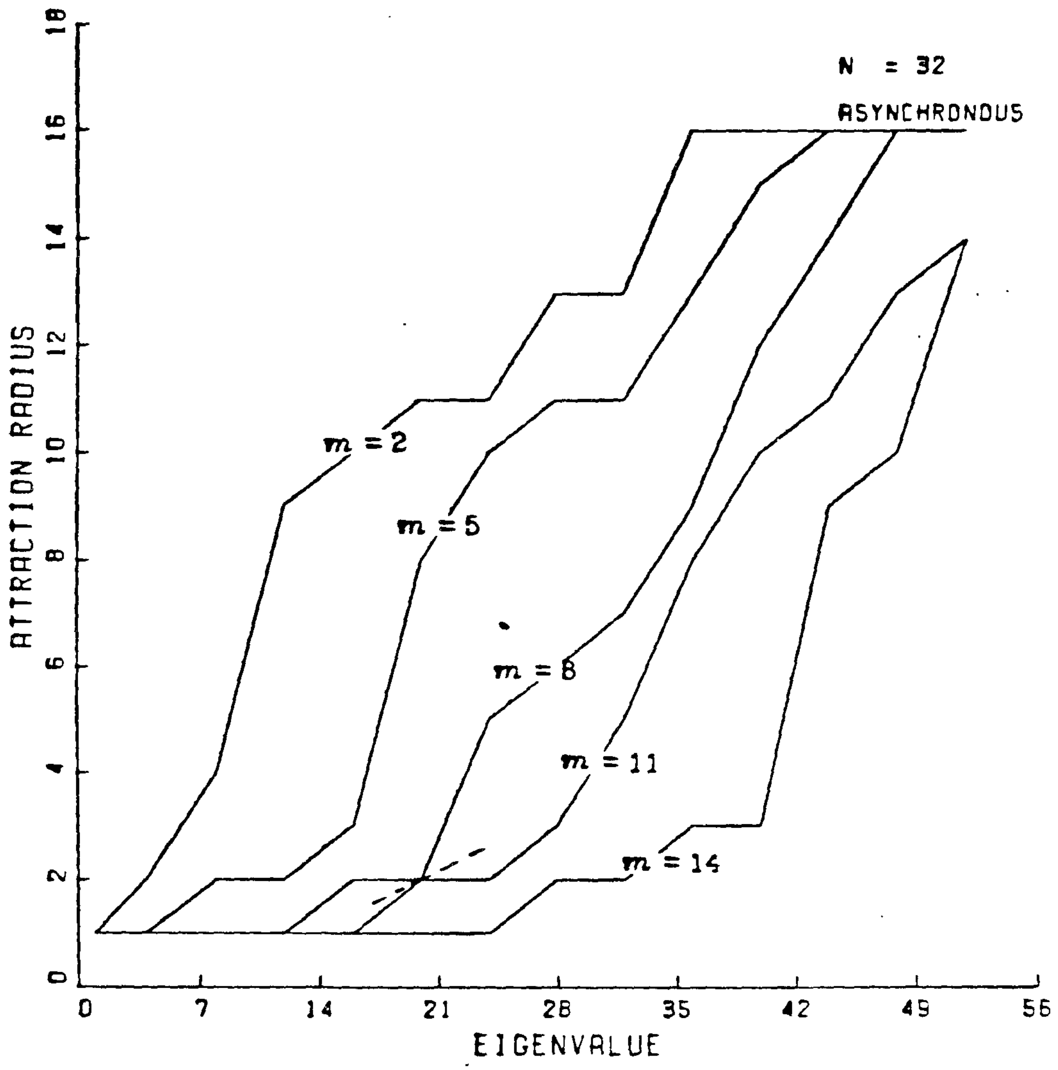

Fig. 7.12. Attraction radius of a typical memory plotted as a function of the eigenvalue. The number of memories is a parameter, and the eigenvalues of all other memories are equal to n .

201–209, 1975.

[7] S. Amari, "Neural theory of association and concept formation," *Biol. Cybern.*, vol. 26, pp. 175–185, 1977.

[8] J. Komlós, "On the determinant of (0,1) matrices," *Studia Scientarum Mathematicarum Hungarica*, vol. 2, pp. 7–21, 1967.

[9] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Natl. Acad. Sci. USA*, vol. 79, pp. 2554–2558, 1982.

CHAPTER VIII

MAXIMAL EPSILON CAPACITY

1. REDUCED MODEL FOR ERROR TOLERANT ASSOCIATIONS

Thus far, we have been concerned with characterising the memory storage capacity of particular algorithms. The relative efficacy of various algorithms, however, can best be gauged if the ultimate storage capacity of the neural network model itself is determined. This will be our focus in this chapter. In particular, we will provide answers for questions such as: What is the maximum number of associations that can be stored when all possible (McCulloch-Pitts) neural networks are allowed for consideration? What gains can be achieved in capacity if there is some tolerance to errors?

The rest of this section describes the reduced neural network model under consideration, and sets up the framework of error tolerance. A precise definition of capacity, consistent with the earlier definitions, is also provided.

The capacity results are quoted in section 3 (see [1] also). The distribution of errors, and the universality of the capacity results are treated in section 4. Finally, section 5 treats the issue of how optimal networks can be found by iterative techniques that converge reasonably quickly.

A. One-Step Synchronous Associations

We again consider a network of n labelled neurons. Recall that each formal neuron (modelled after McCulloch-Pitts) is a threshold gate characterised by a vector of real weights $\mathbf{w}_i = (w_{i1}, \ldots, w_{in})$, and a real threshold (which we take to be zero). The neurons accept real n-tuples $\mathbf{u} \in \mathbb{R}^n$ as input, and return binary scalars $v_i \in \mathbb{B}$ as output according to the threshold rule $v_i = \operatorname{sgn}\,(\sum_{j=1}^{n} w_{ij}\, u_j)$. Given an input $\mathbf{u} \in \mathbb{R}^n$, in a single synchronous transition the neural network under advisement yields as output a binary n-tuple $\mathbf{v} \in \mathbb{B}^n$, whose components v_i are the outputs of each of the individual neurons.

Our concern in this chapter will be mainly with hetero-associative storage within the neural network structure. (We indicate how the results apply to auto-associative storage in section 3.) Specifically, we require to store prescribed associations of the form $\mathbf{u} \mapsto \mathbf{v}$ in the neural network by suitable choice of weights w_{ij}.

We tacitly assume a synchronous mode of operation for simplicity; the inputs to the network are presented simultaneously to each neuron, and each neuron returns an output binary variable in concert. We will further restrict our attention to single step synchronous transitions of the form $\mathbf{u} \mapsto \mathbf{v}$. The reduced network model that we consider is illustrated in fig. 8.1. An input pattern (a real n-vector) is simultaneously presented to n threshold gates (neurons), which act in concert to produce a binary n vector as output. The components of the output binary n vector are the various decisions provided by each of the threshold gates.

The reduced model that we consider in this chapter clearly eschews the neural feedback mechanism, and allows consideration of only a single state transition at a time. Furthermore, the input patterns are now assumed to be drawn from real n space. The reduced system under consideration, imposes much fewer constraints than the fully interconnected neural systems that we had considered in the previous chapters, at the cost of potentially losing the capability to do more complex tasks such as error correction. The general capacity results derived in this chapter will hence bound cases where more complex associative behaviour (such as soft error correction in

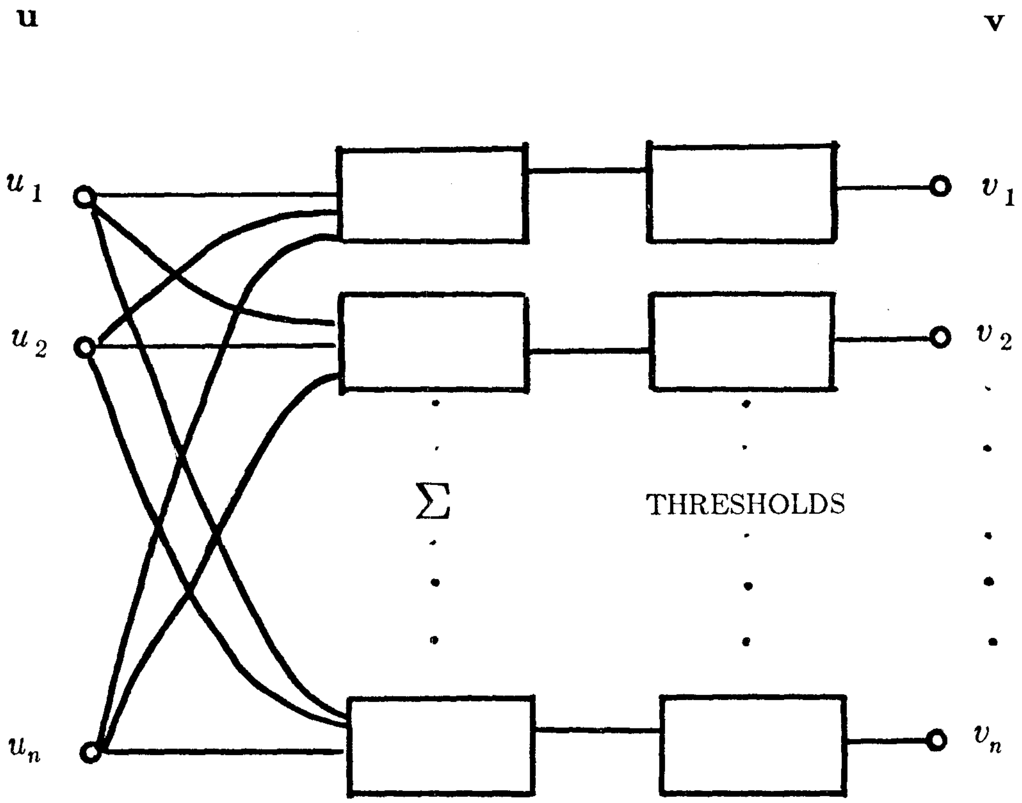

Fig. 8.1. Single state transitions in reduced network model.

distorted memories) is achieved through the medium of feedback and dense neuronal interconnection.

Let m denote the number of associations of the form $\mathbf{u} \mapsto \mathbf{v}$ to be stored in the network. Specifically, we require to store m associations $\mathbf{u}^{(\alpha)} \mapsto \mathbf{v}^{(\alpha)}$, $\alpha=1,\ldots,m$. We call the input probe vectors $\mathbf{u}^{(\alpha)}$ the *fundamental memories*, and the desired resultant vectors $\mathbf{v}^{(\alpha)}$ the *associated memories*. The specified m-set of fundamental memories $\{\mathbf{u}^{(1)},\mathbf{u}^{(2)}, \ldots, \mathbf{u}^{(m)}\} \subseteq \mathbb{R}^n$ is assumed to be chosen independently from any probability distribution invariant to reflection of coordinates in real n-space. The corresponding m-set of associated memories $\{\mathbf{v}^{(1)},\mathbf{v}^{(2)}, \ldots, \mathbf{v}^{(m)}\} \subseteq \mathbb{B}^n$ is also a randomly specified set, with components $v_i^{(\alpha)} \in \{-1,1\}$, $i=1,\ldots,n$, $\alpha=1,\ldots,m$, chosen from a sequence of Bernoulli trials with equal probabilities of success and failure.

We will also refer to the components $v_i^{(\alpha)}$ of the associated memories $\mathbf{v}^{(\alpha)}$ as *decisions* (made by neuron i w.r.t. fundamental memory $\mathbf{u}^{(\alpha)}$). This is motivated by the fact that if we consider neuron i as an isolated threshold gate, then $v_i^{(\alpha)}$ is simply the decision made by the threshold gate when pattern $\mathbf{u}^{(\alpha)}$ is the input.

B. Error Tolerance

For error free associative maps we require that $v_i^{(\alpha)} = \text{sgn}\ (\sum w_{ij}\, u_j^{(\alpha)})$ for each component i and for each corresponding pair of memories. Under error tolerant conditions, however, some of the components of the retrieved states could be allowed to be in error. We now prescribe a mechanism which determines the allowed error distribution in the components of the retrieved states.

It is clear that if for a neuron (threshold gate) we specify a certain number of "don't care" decisions, then the number of "don't care" decisions determines the maximum number of decision errors made by the neuron. Similarly, if we specify "don't care" components for each associated memory, then the number of "don't care" components in each associated memory determines the maximum number of errors made in retrieving each associated memory from the corresponding fundamental memory. Our approach to introducing error tolerance is hence to specify "don't care" decisions in the associated memories by a suitable distribution of choice such that the expected number of "don't care" decisions coincides with (twice) the allowable fraction of errors. Specifically, we perform a sequence of mn independent and identical experiments to determine whether each (random) decision $v_i^{(\alpha)}$, $i=1,...,n$, $\alpha=1,...,m$, is to be labelled a "don't care" decision, with the probability that any particular decision $v_i^{(\alpha)}$ be labelled a "don't care" decision given by twice the allowable fraction of errors.

Formally, let $0 \leq \epsilon < 1/2$ denote the allowable fraction of decision errors made by a neuron for the case of threshold gates, and let ϵ also denote the allowable fraction of component errors in the retrieved associated memories for the case of neural networks. Let $V_i^{(\alpha)}$, $i=1,...,n$, $\alpha=1,...,m$, be the outcomes of mn identical, and independent experiments whose outcomes are subsets of $\{-1,1\}$ such that

$$V_i^{(\alpha)} = \begin{cases} \{v_i^{(\alpha)}\} & \text{with probability } 1-2\epsilon \\ \mathbb{B} & \text{with probability } 2\epsilon \,. \end{cases} \tag{8.1.1}$$

If the outcome $V_i^{(\alpha)} = \{v_i^{(\alpha)}\}$, then we will require that neuron i produce decision $v_i^{(\alpha)}$ as output whenever it receives fundamental memory $\mathbf{u}^{(\alpha)}$ as input. If, however, the outcome $V_i^{(\alpha)} = \mathbb{B}$, then we associate a "don't care" decision with component $v_i^{(\alpha)}$ of the associated memory $\mathbf{v}^{(\alpha)}$, so that neuron i can result in either -1 or 1 as output when $\mathbf{u}^{(\alpha)}$ is input. For obvious reasons we call $V_i^{(\alpha)}$ the *decision set* associated with decision $v_i^{(\alpha)}$. Furthermore, we shall say that $V_i^{(\alpha)}$ is *normal* if $V_i^{(\alpha)} = \{v_i^{(\alpha)}\}$ (i.e., the decision has to be accurate), and we shall say that $V_i^{(\alpha)}$ is *exceptional* if $V_i^{(\alpha)} = \mathbb{B}$ (i.e., the decision is "don't care"). Clearly, once the fundamental memories $\mathbf{u}^{(\alpha)}$, the associated memories $\mathbf{v}^{(\alpha)}$, and the decision sets $V_i^{(\alpha)}$ have been specified, we need to find neural networks for which the neurons yield correct decisions only for the restricted set of decision sets which are normal. The definitions of chapter V now generalise naturally.

Definition. Let $\mathbf{w}_i \in \mathbb{R}^n$ be the vector of interconnection weights associated with neuron i, for each neuron $i=1,...,n$ of a neural network. The event:

$$\operatorname{sgn}\left\{\sum_{j=1}^{n} w_{ij}\, u_j^{(\alpha)}\right\} \in V_i^{(\alpha)}, \quad i=1,\ldots,n \ , \alpha=1,\ldots,m \ , \tag{8.1.2}$$

is described by saying that *the neural network stores m associations with tolerance epsilon.*

This is the analogue of the previous definition from chapter V for error free associations, and generalises the definition to allow errors. The case $\epsilon = 0$ reverts to the requirement of perfect recall.

Note that by virtue of the random decision sets $V_i^{(\alpha)}$ being drawn from independent, and identical experiments, the actual distribution of errors is spread independently across the components $v_i^{(\alpha)}$ of the associated memories. Furthermore, the conditional distribution of the decision sets $V_i^{(\alpha)}$ given the associated memories $\mathbf{v}^{(\alpha)}$ is binomial. Hence, the expected number of "don't care" decisions attributed to each neuron is $2\epsilon m$, and the expected number of errors in each of the associated memories is ϵn. Thus the modified definitions reflect (at least in an average sense) a tolerance of up to a fraction ϵ of errors in the decisions. We demonstrate in section 4 that the number of errors evinced in the recall of each associated memory actually does approach the prescribed tolerance ϵn.

C. Definition of Capacity

The previous definitions of capacity that we had utilised enabled us to characterise the storage capacity of *specific* algorithms for generating the neural interconnection weights. In this chapter our focus is not on any particular algorithm for generating interconnection weights, but on the universe of McCulloch-Pitts neural networks obtained by allowing *all* choices of real interconnection weights for consideration. Specifically, we want to specify the maximum number of associations that can be stored when all possible neural networks are allowed for consideration. We hence apply the capacity definitions of chapter V to all possible neural network realisations rather than to specific network realisations obtained by the application of a specified algorithm.

Let a tolerance $0 \leq \epsilon < 1/2$ be fixed. In querying whether there exists a choice of interconnection weights for which the neural network maps the fundamental memories to the associated memories with at most ϵn errors (on average), the event of interest is described by the following attribute:

Event E: "$\exists$ a neural network which stores m associations with tolerance ϵ."

Note that the event E is defined on the sample space obtained as the product of the probability spaces over which the components of the fundamental memories $u_i^{(\alpha)}$, the components of the associated memories $v_i^{(\alpha)}$, and the decision sets $V_i^{(\alpha)}$, ($i=1,...,n$, and $\alpha=1,...,m$,) are defined. In consonance with our earlier definitions of capacity, we now have the following definitions.

Definition. A sequence of integers $\{\underline{C_\epsilon}(n)\}_{n=1}^{\infty}$ is a *lower sequence of epsilon capacities* for neural networks iff for each $\lambda \in (0,1)$, event E occurs with probability approaching one as $n \to \infty$ whenever $m \leq (1-\lambda)\underline{C_\epsilon}(n)$.

Again, this is a lower estimate for the storage capacity as it tells us that for large n, if the number of associations is chosen to be less than the lower capacity, then with probability essentially one, we can find neural networks for almost all choices of associations $\mathbf{u}^{(\alpha)} \mapsto \mathbf{v}^{(\alpha)}$, $\alpha=1,...,m$, such that the errors number at most a fraction ϵ. The following definition overestimates the storage capacity.

Definition. A sequence of integers $\{\overline{C_\epsilon}(n)\}_{n=1}^{\infty}$ is an *upper sequence of epsilon capacities* for neural networks iff for each $\lambda \in (0,1)$, event E occurs with probability approaching zero as $n \to \infty$ whenever $m \geq (1+\lambda)\overline{C_\epsilon}(n)$.

The above definition is an upper estimate for storage capacity; if the number of associations is chosen to be larger than the upper capacity, then for almost every choice of m associations, there will exist particular associations that cannot be stored within the described error tolerance in any neural network.

Again, the only requirement that we would wish to impose so that the definitions are useful is that the probabilities of interest behave monotonically with the number of associations m, as is illustrated schematically in fig. 6.5. Specifically, we would like to rule out oscillatory behaviour of the probabilities with increase in m. As we shall see in section 3, the probabilities are indeed monotonic in m.

Definition. A sequence of integers $\left\{C_\epsilon(n)\right\}_{n=1}^{\infty}$ is a *sequence of epsilon capacities* for neural networks iff it is both a lower sequence, and an upper sequence of epsilon capacities, i.e., $C_\epsilon(n) = \underline{C_\epsilon}(n) = \overline{C_\epsilon}(n)$.

Note that we are utilising the strong form of the capacity definitions, where convergence is required with probability one. Further, attraction behaviour is not considered–the definition is analogous to the capacity definition for fixed point storage. This capacity is an upper bound for capacity if error correction is desired in addition. Note that in our definition, (zero)-capacity corresponds to the maximum number of associations that can be stored under conditions of perfect recall, i.e., with the tolerance epsilon being identically zero.

Proposition (5.2.1) and (5.2.2) hold *in toto* for the above definitions of epsilon capacity, as do the comments following the definitions of capacity in chapter V. (The propositions themselves do not depend on the exact event, nor yet on the probability space under consideration, but are a consequence solely of the nature of asymptotic behaviour desired.) In particular, proposition (5.2.2) establishes that if sequences of epsilon capacity do exist, then: (1) they are not unique, and (2) they do not differ significantly from each other. Thus, we define an equivalence class of sequences of epsilon capacities $[\left\{C_\epsilon(n)\right\}]$ with equivalence relation defined as follows: if $\left\{C_\epsilon(n)\right\}$ and $\left\{C_\epsilon(n)'\right\}$ are members of this equivalence class of epsilon capacities, then they must satisfy the equivalence relation $C_\epsilon(n)' \sim C_\epsilon(n)$. Henceforth, if a sequence of epsilon capacities $\left\{C_\epsilon(n)\right\}$ exists, then we shall say without elaboration that $C_\epsilon(n)$ is the *epsilon capacity* of neural networks; by this we mean that $\left\{C_\epsilon(n)\right\}$ is a member of the equivalence class $[\left\{C_\epsilon(n)\right\}]$ of sequences of epsilon capacities.

2. PRELIMINARY RESULTS

Before getting to the main theorems on capacity, we will need some technical results. In the first part of this section we quote seven preliminary results that will be needed in the proofs. The second part of this section is devoted to proving the main lemmas which are the basis of the theorem proofs. We shall use the convention that all logarithms are to base e unless explicitly stated otherwise.

A. Technical Lemmas

Lemma 8.2.1. $\log(1-x) \geq -2x \quad \forall\, x \in [0,1/2]$.

Proof. The Taylor series expansion for $\log(1-x)$ converges uniformly in the interval $-1 \leq x \leq 1$. Hence

$$\log(1-x) = -x - \frac{x^2}{2} - \frac{x^3}{3} - \cdots$$

$$= -x \left(1 + \frac{x}{2} + \frac{x^2}{3} + \cdots \right)$$

$$\geq -x \; (1 + x + x^2 + \cdots)$$

$$= \frac{-x}{1-x} .$$

Finally, $\frac{-x}{1-x} \geq -2x$ whenever $0 \leq x \leq 1/2$. □

Lemma 8.2.2. $\forall$ integers j and k,

$$\binom{k}{j} - \binom{k-1}{j} = \binom{k-1}{j-1} .$$

Proof. Consider $0 \leq j \leq k-1$ wlog. Then

$$\binom{k}{j} - \binom{k-1}{j} = \frac{k!}{j!(k-j)!} - \frac{(k-1)!}{j!(k-j-1)!}$$

$$= \frac{(k-1)!}{j!(k-j)!}(k-(k-j))$$

$$= \binom{k-1}{j-1} . \qquad \square$$

Lemma 8.2.3. (De Moivre - Laplace Theorem for Large Deviations)

Let $\{X_j\}_{j=1}^{\infty}$ be an infinite sequence of Bernoulli trials

$$X_j = 1 \quad \text{with probability } p$$

$$= 0 \quad \text{with probability } q = 1-p \ ,$$

and where $0 < p < 1$. Form the sums $S_N = \sum_{j=1}^{N} X_j$, and let $\{v_N\}$ be a sequence such that

$$| v_N - Np | \leq K(N) = o([Npq]^{2/3}) \quad \text{if } p \neq q$$

$$= o(N^{3/4}) \quad \text{if } p = q = 1/2 .$$

Then

$$\mathbf{P}\left\{S_N \le v_N\right\} = \sum_{k=0}^{v_N} \binom{N}{k} p_k \, q^{N-k} \sim \Phi\left(\frac{v_N - Np}{\sqrt{Npq}}\right)$$

as $N \to \infty$. If in addition, $\dfrac{(v_N - Np)}{\sqrt{Npq}} \to -\infty$, then

$$\mathbf{P}\left\{S_N < v_N\right\} \sim \frac{1}{\sqrt{2\pi}} \frac{\sqrt{Npq}}{\mid v_N - Np \mid} \exp\left\{-\left[\frac{(v_N - Np)^2}{2Npq}\right]\right\}.$$

Proof. cf. Ref. [2], pg. 193, and prob. 14, pg. 195.

Lemma 8.2.4. Define $\mu : [0,1] \to \mathbb{R}$ by

$$\mu(\alpha) = 2^{-(1-H(\alpha))} \quad \forall \, \alpha \in [0,1] \, ,$$

and where

$$H(\alpha) = -\, \alpha \log_2(\alpha) - (1-\alpha) \log_2(1-\alpha)$$

is the entropy function (in bits). Let N be a fixed positive integer. Then, for every fixed $\alpha \in [1/2 \, , 1]$, we have

$$2^{-N} \sum_{j=0}^{\lfloor \alpha N - 1 \rfloor} \binom{N}{j} \ge 1 - [\mu(\alpha)]^N \, .$$

Proof. The proof follows as a special case of theorem 1 in Ref. [3].

Lemma (8.2.3) gives a large deviation estimate for the binomial distribution, where the deviation from the mean is $o(N^{2/3})$ (or $o(N^{3/4})$ if $p = q = 1/2$). Lemma (8.2.4) illustrates that for *very* large deviations (of the order of n) the tail of the binomial no more shows a limiting central tendency.

The following result is a fundamental theorem in combinatorial probability due originally to Schläfli.

Definition. m points in real n space are in *general position* (in n space) iff any subset of n points or fewer is linearly independent.

Lemma 8.2.5. (Function Counting Theorem).

The number of dichotomies of m points in real n space that can be separated by a hyperplane through the origin (i.e., homogeneously linearly separable dichotomies of m points in n space) is at most $2\sum_{j=0}^{n-1}\binom{m-1}{j}$. The upper bound is achieved if the m points are in general position.

Proof. cf. Refs. [4], and [5].

Lemma 8.2.6. The probability that the m-set of fundamental memories $\{\mathbf{u}^{(1)},\mathbf{u}^{(2)}, \ldots, \mathbf{u}^{(m)}\} \subset \mathbb{B}^n$ is linearly independent (over the field of the reals) approaches one as $n \rightarrow \infty$ provided m is chosen less than or equal to n .

Proof. The result follows from a result due to Komlós [6] who demonstrated that almost all $n \times n$ (0,1)-matrices have non-zero determinant.

Lemma 8.2.7. (Borel-Cantelli Lemma)

Let $\{n_j\}_{j=1}^{\infty}$ be an increasing sequence of integers. Let $\{E_j\}_{j=1}^{\infty}$ be an infinite sequence of events defined on the sample space of an infinite sequence of Bernoulli trials and such that each E_j depends solely on the outcome of the first n_j Bernoulli trials. If $\sum \mathrm{P}\{E_j\}$ cọnverges, then with probability one only finitely many events E_j occur.

Proof. cf. Ref. [2], pg. 201.

B. Main Lemmas

We now obtain estimates of the probability of the event E that was crucial to our definitions of capacity in the last section. The next lemma estimates the required probability for the case of perfect association: $\epsilon = 0$.

Lemma 8.2.8. Fix the tolerance ϵ to be identically zero. Define

$$P_0(k,n) \triangleq 2^{-(k-1)} \sum_{j=0}^{n-1} \binom{k-1}{j} . \tag{8.2.1}$$

Then

$$\mathbf{P}\left\{E\right\} = Q_0(m,n) \triangleq [P_0(m,n)]^n . \tag{8.2.2}$$

Proof. Fix $i \in \left\{1,...,n\right\}$. The particular dichotomy $\left\{U_i^+, U_i^-\right\}$ of the m-set of fundamental memories that is of interest is defined by:

$$U_i^+ = \left\{\mathbf{u}^{(\alpha)} : u_i^{(\alpha)} = 1\right\} ,$$

and

$$U_i^- = \left\{\mathbf{u}^{(\alpha)} : u_i^{(\alpha)} = -1\right\} .$$

Event E is realised if there exists a hyperplane through the origin which separates the dichotomy $\left\{U_i^+, U_i^-\right\}$ for each $i=1,...,n$. From Schläfli's fundamental function counting theorem, the number of dichotomies of the m fundamental memories $\mathbf{u}^{(\alpha)}$, $\alpha=1,...,m$, that can be separated in n space is at most $2\sum_{j=0}^{n-1}\binom{m-1}{j}$ out of a total of 2^m possible dichotomies. The fundamental memories $\mathbf{u}^{(\alpha)}$ are chosen independently from real n space, so that all dichotomies are separable with equal probability. Hence, denoting the probability that there exists a choice of weights $\mathbf{w}_i = (w_{i1} \cdots w_{in})$ for which the i-th neuron makes no errors in m decisions by $p_{i,0}$, we have

$$p_{i,0} = 2^{-(m-1)} \sum_{j=0}^{n-1} \binom{m-1}{j} = P_0(m,n) .$$

Now for each $i=1,...,n$, the choices of the dichotomies $\{U_i^+, U_i^-\}$ are mutually independent events as the components $u_i^{(\alpha)}$ of the fundamental memories are chosen independently. Hence,

$$\mathbf{P}\{E\} = \prod_{i=1}^{n} p_{i,0}$$

$$= Q_0(m,n) \triangleq [P_0(m,n)]^n \ . \qquad \square$$

Note that for $m \leq n$, we have $Q_0(m,n) = 1$. This, coupled with lemma (8.2.6), yields an immediate lower bound on capacity. Upper bounds on capacity will be arrived at in the next section by examining the asymptotic behaviour of $Q_0(m,n)$.

Lemma 8.2.9. Let $\epsilon \in [0, 1/2)$ be a given tolerance. Then

$$\mathbf{P}\{E\} = Q_\epsilon(m,n) \triangleq \left[\sum_{k=0}^{m} \binom{m}{k} (1-2\epsilon)^k (2\epsilon)^k \ P_0(k,n) \right]^n . \qquad (8.2.3)$$

Proof. Fix $\epsilon \in [0,1/2)$. Let $p_{i,\epsilon}$ denote the probability that there exists a choice of weights $\mathbf{w}_i = (w_{i1} \cdots w_{in})$ for which the i-th neuron makes m decisions with tolerance ϵ. For $i=1,...,n$, we have

$p_{i,\epsilon} = \sum_{k=0}^{m} \sum_{0 \leq \alpha_1 < ... < \alpha_k \leq m} \mathbf{P}\{\exists$ weight vector $\mathbf{w}_i \in \mathbb{R}^n$ such that neuron i yields correct decisions $v_i^{(\alpha_1)}, \ldots, v_i^{(\alpha_k)}$ | the decision sets $V_i^{(\alpha_1)}, \ldots, V_i^{(\alpha_k)}$ are normal, and the decision sets $V_i^{(\alpha)}$, $\alpha \neq \alpha_j$, $j=1,...,k$, are exceptional $\}$ $\times$ $\mathbf{P}\{$ the decision sets $V_i^{(\alpha_1)}, \ldots, V_i^{(\alpha_k)}$ are normal, and the decision sets $V_i^{(\alpha)}$, $\alpha \neq \alpha_j$, $j=1,...,k$, are exceptional $\}$

$= \sum_{k=0}^{m} \mathbf{P}\{\exists$ weight vector $\mathbf{w}_i$ for which neuron i yields at least

k correct decisions $\} \times \binom{m}{k}(1-2\epsilon)^k\,(2\epsilon)^{m-k}$,

as we require only k correct decisions given that $m-k$ are "don't cares", and the "don't care" decision sets follow the binomial distribution. From the proof of lemma (8.2.8) we now have

$$p_{i,\epsilon} = \sum_{k=0}^{m} \binom{m}{k}(1-2\epsilon)^k\,(2\epsilon)^{m-k}\; P_0(k,n)\;.$$

The decisions $v_i^{(\alpha)}$ are jointly independent, as are the decision sets $V_i^{(\alpha)}$. Hence

$$\mathrm{P}\,\{E\} = \prod_{i=1}^{n} p_{i,\epsilon}$$

$$= \left[\; \sum_{k=0}^{m} \binom{m}{k}(1-2\epsilon)^k\,(2\epsilon)^{m-k}\; P_0(k,n) \;\right]^n$$

$$\triangleq Q_\epsilon(m,n)\;.$$

□

Lemma 8.2.10. For each positive integer n, $P_0(k,n)$ and $Q_0(k,n)$ are monotone non-increasing functions of k.

Proof. We will demonstrate that the difference $P_0(k,n) - P_0(k+1,n)$ is non-negative for every choice of k and n. From the defining equation (8.2.1), and lemma (8.2.2) we have

$$P_0(k,n) - P_0(k+1,n) = 2^{-k}\left[\,2\sum_{j=0}^{n-1}\binom{k-1}{j} - \sum_{j=0}^{n-1}\binom{k}{j}\,\right]$$

$$= 2^{-k}\left[\,\sum_{j=1}^{n-1}\binom{k-1}{j} - \sum_{j=1}^{n-1}\binom{k-1}{j-1}\,\right]$$

$$= 2^{-k} \binom{k-1}{n-1} \geq 0 .$$

Hence, for each fixed n, $P_0(k,n)$ is a monotone non-decreasing function of k. From the defining equation (8.2.2), we have $Q_0(k,n) = [P_0(k,n)]^n$, so that $Q_0(k,n)$ has the same monotone character as $P_0(k,n)$. □

3. EPSILON CAPACITY OF NEURAL NETWORKS

We are now in a position to encapsulate all the prior results into a statement on capacity. We start with proving a result on capacity under error-free conditions. Using this result we extend our analysis to the case where errors are permitted, and obtain rigourous results on the epsilon capacity of neural networks.

A. (Zero)-Capacity

Theorem 8.3.1. For every fixed $\lambda \in (0,1)$,

(a) $P_0(\lfloor 2n(1-\lambda)\rfloor, n)$, $Q_0(\lfloor 2n(1-\lambda)\rfloor, n) \to 1$ as $n \to \infty$,

(b) $P_0(2n,n) = 1/2$, $Q_0(2n,n) = 2^{-n}$,

(c) $P_0(\lceil 2n(1+\lambda)\rceil, n)$, $Q_0(\lceil 2n(1+\lambda)\rceil, n) \to 0$ as $n \to \infty$.

Proof. We consider the case where the tolerance $\epsilon = 0$ in the defining equation (8.2.3). Fix λ arbitrarily in the open interval (0,1). Now part (b) of the theorem follows immediately as

$$P_0(2n,n) = 2^{-(2n-1)} \sum_{j=0}^{n-1} \binom{2n-1}{j} = 1/2 ,$$

while from the defining equation (8.2.2) in lemma (8.2.8), we have

$$Q_0(2n\,,n\,) = [P_0(2n\,,n\,)]^n \;= 2^{-n}\ .$$

We now prove part (c) of the theorem. Note that it suffices to show that $P_0(\lfloor 2n\,(1+\lambda)\rfloor,n\,) \to 0$ as $n \to \infty$, as $0 \leq Q_0(m\,,n\,) = [P_0(m\,,n\,)]^n \leq P_0(m\,,n\,)$ for every choice of m and n. Now, set $N = \lfloor 2n\,(1+\lambda)\rfloor - 1$. Then, using the defining equation for P_0 from lemma (8.2.10), we have

$$P_0(\lfloor 2n\,(1+\lambda)\rfloor,n\,) = 2^{-\lfloor 2n\,(1+\lambda)\rfloor+1}\sum_{k=0}^{n-1}\binom{\lfloor 2n\,(1+\lambda)\rfloor - 1}{k}$$

$$= 2^{-N}\sum_{k=0}^{N/2-n\lambda-\delta}\binom{N}{k}$$

where $|\,\delta\,| \leq 2$. Clearly we are working with the extreme tails of the binomial distribution, so that we expect the above sum to approach zero for large n. More formally, choose a sequence of positive integers $\{t_N\}$ such that $t_N = o(N^{\frac{3}{4}})$, and $\frac{t_N}{\sqrt{N}} \to \infty$ as $n \to \infty$. (Note that $n\,\lambda = \Omega(N)$, so that trivially $n\,\lambda+\delta \geq t_N$). Then

$$P_0(\lfloor 2n\,(1+\lambda)\rfloor,n\,) \leq 2^{-N}\sum_{k=0}^{N/2-t_N}\binom{N}{k}\ .$$

By lemma (8.2.3) we have that the righthand side of the above inequality approaches zero as $n \to \infty$. This completes the proof of part (c).

To prove part (a) of the theorem it suffices to show that $Q_0(\lfloor 2n\,(1-\lambda)\rfloor,n\,) \to 1$ as $n \to \infty$. Fix $0 < \lambda < 1/2$. Set $M = \lfloor 2n\,(1-\lambda)\rfloor - 1$, and $\alpha = \dfrac{n}{\lfloor 2n\,(1-\lambda)\rfloor - 1}$. We have $n = \alpha M$, and $M \to \infty$ as $n \to \infty$. Further

$$\frac{1}{2 - (2\lambda + \frac{1}{n})} \leq \alpha < \frac{1}{2 - (2\lambda + \frac{2}{n})} .$$

Define the positive integer $n_0 = | \left\lceil \frac{2}{1 - 2\lambda} \right\rceil | + 1$. It is then easy to verify that $n \geq n_0 \Longrightarrow 1/2 < \alpha < 1$ with inequality being strict on both sides. Henceforth we assume $n \geq n_0$. From lemma (8.2.8) we have

$$Q_0(\lfloor 2n\,(1\text{-}\lambda) \rfloor, n\,) = \left[2^{-\lfloor 2n\,(1\text{-}\lambda)\rfloor + 1} \sum_{k=0}^{n-1} \binom{\lfloor 2n\,(1\text{-}\lambda)\rfloor - 1}{k} \right]^n$$

$$= \left[2^{-M} \sum_{k=0}^{\alpha M - 1} \binom{M}{k} \right]^{\alpha M} .$$

By lemma (8.2.4) we then have

$$Q_0(\lfloor 2n\,(1\text{-}\lambda) \rfloor, n\,) \geq \left[1 - \{\mu(\alpha)\}^M \right]^{\alpha M} , \qquad (8.3.1)$$

where $\mu(\alpha) = 2^{-(1 - H(\alpha))}$.

Now $1/2 < \alpha < 1$, so that $H(\alpha)$ is well-defined, and $0 < H(\alpha) < 1$, again with strict inequality. Hence, $1/2 < \mu(\alpha) < 1$, so that we have $0 < \mu(\alpha)^M < 1$. Taking the logarithm of both sides of the inequality (8.3.1), we have

$$\log Q_0(\lfloor 2n\,(1\text{-}\lambda) \rfloor, n\,) \geq \alpha M \, \log\,(1 - \mu(\alpha)^M\,)$$

$$= -\,\alpha M \left(\mu(\alpha)^M + \frac{\mu(\alpha)^{2M}}{2} + \frac{\mu(\alpha)^{3M}}{3} + \cdots \right)$$

where the Taylor series expansion for the logarithm converges as $0 < \mu(\alpha)^M < 1$. Now $\mu(\alpha)^M \rightarrow 0$ as $n \rightarrow \infty$. Hence

$$\log Q_0(\lfloor 2n\,(1-\lambda)\rfloor ,n\,) \geq -\,\alpha m\ \left(\mu(\alpha)^M + O(\mu(\alpha)^{2M}\,)\right)$$

As $Q_0(\lfloor 2n\,(1-\lambda)\rfloor ,n\,)$ is a probability, we have

$$0 \geq \log Q_0(\lfloor 2n\,(1-\lambda)\rfloor ,n\,) \geq -\,\alpha M\ \ \mu(\alpha)^M\,(1 + o(1))\ . \qquad (8.3.2)$$

Now consider

$$\log \alpha M\, \mu(\alpha)^M \ = \log \alpha + \log M\ + M\ \log \mu(\alpha)$$

$$= \log \alpha + \log M\ - (1 - H\,(\alpha))\, M\ \log 2\ .$$

Now $\log \alpha < \log 1 = 0$. So

$$\log \alpha M\, \mu(\alpha)^M \ < \log M\ - (1 - H\,(\alpha))\, M\ \log 2\ .$$

Define $\delta(\lambda) = 1 - H\,(\frac{1}{2-2\lambda})$. ($\delta(\lambda)$ is well-defined as $0 < \lambda < \frac{1}{2}$, so that $H\,(\frac{1}{2-2\lambda})$ is well-defined.) Have $0 < H\,(\frac{1}{2-2\lambda}) < 1$ so that $\delta(\lambda) > 0$ strictly. Consequently, to every fixed $\lambda \in (0,\frac{1}{2})$ we can associate the fixed, positive real number $\delta(\lambda)$ which depends solely on λ. We then have

$$1 - H\,(\alpha) = 1 - H\left(\frac{n}{\lfloor 2n\,(1-\lambda)\rfloor - 1}\right)$$

$$> 1 - H\left(\frac{1}{2 - 2\lambda}\right)$$

$$='\delta(\lambda) > 0\ .$$

Hence

$$\log \alpha M\, \mu(\alpha)^M \ < \log M\ - M\ \ \delta(\lambda)\,\log 2 \rightarrow -\,\infty \text{ as } n\ \rightarrow \infty\ .$$

So $\alpha M\,\mu(\alpha)^M \rightarrow 0^-$ as $n \rightarrow \infty$. Referring back to the double-inequality (8.3.2), we see that $\log Q_0(\lfloor 2n(1-\lambda)\rfloor, n)$ is bounded from above by 0, and bounded from below by a negative quantity that approaches zero as n approaches infinity. Hence

$$\log Q_0(\lfloor 2n(1-\lambda)\rfloor, n) \rightarrow 0^- \text{ as } n \rightarrow \infty .$$

So

$$Q_0(\lfloor 2n(1-\lambda)\rfloor, n) \rightarrow 1 \text{ as } n \rightarrow \infty . \qquad \square$$

Corollary 8.3.1. $C_0(n) = 2n$ is the (zero)-capacity of neural networks.

Proof. From part (c) of the theorem it follows that $\{2n\}$ is a sequence of upper (zero)-capacities, and from part (a), likewise, it is also a sequence of lower (zero)-capacities. Hence $C_0(n) = 2n$ is the (zero)-capacity of neural networks. $\square$

The results of theorem (8.3.1) may seem somewhat surprising at first sight. Recall from equation (8.2.1) that $P_0(m,n) = 1$ for $m \leq n$, while for $m > n$, we have $P_0(n) < 1$ strictly. Further, $Q_0(m,n) = [P_0(m,n)]^n$. Thus, for large n, in the range $n < m < 2n$, $Q_0(m,n)$ is the result of taking a quantity less than one to a large power. A naive expectation would then be that $Q_0(m,n) = 1$ for $m \leq n$, but that $Q_0(m,n)$ approaches zero rapidly when $m > n$ for large n. In actual fact, however, theorem (8.3.1) asserts that the rate of fall of $P_0(m,n)$ in the range $n < m < 2n$ is sufficiently small so that the asymptotic behaviour of $Q_0(m,n)$ is virtually identical to that of $P_0(m,n)$.

An insight into this asymptotic behaviour may be obtained by recalling that $P_0(m,n)$ is the probability that we can find a solution vector which makes a given m-set of decisions, while $Q_0(m,n)$ is the probability that we can find n such solution vectors. Now, we can form 2^m such m-sets of decisions. For $m > n$, the n m-sets of decisions required by the neural network form an asymptotically negligible fraction of the total number (2^m) of decision m-sets. Hence, if we can find one solution n vector of weights $\mathbf{w} = (w_{i1},\ldots,w_{in})$ with high enough probability for $m \approx 2n$, then we should also be able to find n $(= o(2^m))$ such solution vectors with high probability. Theorem (8.3.1) essentially echoes this.

The provable capacity results are somewhat looser if we restrict the fundamental memories to be binary n-tuples only, instead of allowing them to be chosen from real n-space. In this case, while $\{2n\}$ is certainly a sequence of upper (zero)-capacities, it is not immediately clear that it is also a sequence of lower (zero)-capacities. Clearly, lower (zero)-capacity cannot exceed $2n(1 - o(n))$. The problem in exactly reconciling the lower (zero)-capacity with this upper bound arises because we restrict our choice of fundamental memories to binary n-space. From lemma (8.2.6), it follows that almost all m-sets of fundamental memories with $m \leq n$ are linearly independent. This, however, does not appear to extend to the fundamental memories being in general position for $m > n$ – and specifically for $m \approx 2n$ – so that equality in lemma (8.2.9) does not neccessarily obtain. In general it appears somewhat difficult to characterise those choices of fundamental memories that are not in general position, but which can still be stored. There is some experimental evidence, however, indicating that the lower (zero)-capacity does achieve its upper bound [7] of $2n$. We conjecture that $2n$ is actually a sequence of lower (zero)-capacities, so that the (zero)-capacity of neural networks (or the capacity of perfect recall) is also $2n$. (If (zero)-capacity exists at all for the case of fundamental memories chosen from binary n-tuples only, then it must be $2n$, as it has to be simultaneously a sequence of upper and lower (zero)-capacities.) The result is, however, not yet fully rigourous. Note that, in any case, we expect the lower (zero)-capacity to be at least n, as lemma (8.2.6) assures us that for $m \leq n$, almost all choices of m fundamental memories chosen from the vertices of the binary n cube are linearly independent as $n \to \infty$.

A capacity result due to Abu-Mostafa and St. Jacques [8] gives n as an upper bound for the capacity of neural networks, while we anticipate a capacity of as much as twice n. The capacity result of [8], however, required that for *every* choice of m fundamental memories with $m <$ capacity there must exist at least one neural network in which the chosen m-set of vectors can be stored as memories. This leads to the upper bound of n on capacity.

The requirement that every choice of m fundamental memories be stored is, however, a bit too strict, and results in an upper bound $m < 3$. To see this, note that two vectors differing only in one component can never be stored simultaneously using a zero-diagonal matrix of interconnections. In order to avoid such pathological cases in our definition of capacity, we require that asymptotically as $n \to \infty$, for *almost all* choices of m fundamental memories with $m <$ capacity, there exist some neural network in which the chosen m-set of vectors can be stored as memories. This leads to the upper bound of $2n$.

Thus, as long as we are willing to eschew the requirement that *all* of the $\binom{2^n}{m}$ possible choices of m fundamental memories be allowable choices in favour of the requirement that *most* (but not neccessarily all) of these choices be allowable candidates for storage, then we can potentially gain by as much as a factor of two in capacity. Another consequence of the above two statements on capacity is that if $2n > m > n$, then there are guaranteed to be choices of m fundamental memories which cannot be stored in *any* neural network; such choices of fundamental memories will however constitute an asymptotically negligible proportion of the total number of choices $\binom{2^n}{m}$.

B. Epsilon Capacity

We now extend the results of theorem (8.3.1) to the case where a degree of error tolerance is permitted.

Theorem 8.3.2. Let ϵ be a given error tolerance, $0 \leq \epsilon < 1/2$. Then, for every fixed λ in the open interval $0 < \lambda < 1$, the following implications hold:

(a) Let ν be fixed, but arbitrary, in the open interval $1/2 < \nu < 2/3$. Then $Q_\epsilon(m,n) \to 1$ as $n \to \infty$ if

$$m \leq \frac{2n(1-\lambda)}{(1-2\epsilon)} - \frac{[4\epsilon(1-\lambda)]^\nu n^\nu}{(1-2\epsilon)} + \frac{\nu[4\epsilon(1-\lambda)]^{2\nu} n^{2\nu-1}}{2(1-2\epsilon)(1-\lambda)} + O(n^{3\nu-2}) .$$

(b) $Q_\epsilon(m,n) \rightarrow 0$ as $n \rightarrow \infty$ if

$$m \geq \frac{2n(1+\lambda)}{(1-2\epsilon)} .$$

Proof. Let $\epsilon \in [0, 1/2]$ be the given tolerance. Fix λ arbitrarily in the open interval $0 < \lambda < 1$. We first prove part (b) of the theorem.

Let $m \geq \frac{2n(1+\lambda)}{(1-2\epsilon)}$. From the defining equation for $Q_\epsilon(m,n)$ in lemma (8.2.9) we have

$$Q_\epsilon(m,n) = \left[\sum_{k=0}^{m} \binom{m}{k} (1-2\epsilon)^k (2\epsilon)^{m-k} P_0(k,n) \right]^n$$

$$= \left[\sum_{k=0}^{\lfloor 2n(1+\lambda/2) \rfloor} \binom{m}{k} (1-2\epsilon)^k (2\epsilon)^{m-k} P_0(k,n) + \sum_{k=\lfloor 2n(1+\lambda/2) \rfloor+1}^{m} \binom{m}{k} (1-2\epsilon)^k (2\epsilon)^{m-k} P_0(k,n) \right]^n$$

$$\leq \left[\sum_{k=0}^{\lfloor 2n(1+\lambda/2) \rfloor} \binom{m}{k} (1-2\epsilon)^k (2\epsilon)^{m-k} + P_0(\lfloor 2n(1+\lambda/2) \rfloor, n) \right]^n .$$

The last inequality follows from lemma (8.2.10); the probability $P_0(k,n)$ has a monotone non-increasing character so that $P_0(k,n) \leq P_0(\lfloor 2n(1+\lambda/2) \rfloor, n)$ for $k > \lfloor 2n(1+\lambda/2) \rfloor$, and in the range $0 \leq k \leq \lfloor 2n(1+\lambda/2) \rfloor$, we have $P_0(k,n) \leq P_0(0,n) = 1$.

Now, if $\epsilon = 0$, we have $Q_\epsilon(m,n) = Q_0(m,n)$. Part (b) of the theorem then holds as a consequence of theorem (8.3.1) (c). Assume $\epsilon > 0$. Form the sequence $\{x_n\}$ with $x_n = \lfloor 2n(1+\lambda/2) \rfloor$. Then

$$Q_\epsilon(m,n) \leq \left[\sum_{k=0}^{x_n} \binom{m}{k} (1-2\epsilon)^k (2\epsilon)^{m-k} + P_0(x_n, n) \right]^n . \qquad (8.3.3)$$

Now:

$$\frac{x_n - m(1-2\epsilon)}{[m(2\epsilon)(1-2\epsilon)]^{2/3}} = \frac{x_n}{[m(2\epsilon)(1-2\epsilon)]^{2/3}} - \frac{[m(1-2\epsilon)]^{1/3}}{(2\epsilon)^{2/3}}$$

is a monotonically decreasing function of m. Choose n large enough so that $4n\epsilon > 1$. Then

$$\frac{x_n - m(1-2\epsilon)}{[m(2\epsilon)(1-2\epsilon)]^{2/3}} \leq \frac{2n(1+\lambda/2) - 2n(1+\lambda)}{[4n\,\epsilon(1+\lambda)]^{2/3}}$$

$$= \frac{-n\lambda}{[4n\,\epsilon(1+\lambda)]^{2/3}}$$

$$= -n^{1/3}\left(\frac{\lambda}{[4\epsilon(1+\lambda)]^{2/3}}\right)$$

$$\rightarrow -\infty \text{ as } n \rightarrow \infty .$$

From the above, and lemma (8.2.3), we see that the sum in (8.3.3) corresponds to the extreme tails of the binomial distribution. Hence, as $n \rightarrow \infty$,

$$\sum_{k=0}^{x_n} \binom{m}{k} (1-2\epsilon)^k (2\epsilon)^{m-k} = o(1) .$$

Also, $x_n = \lfloor 2n(1+\lambda/2) \rfloor$, so that by theorem (8.3.1), we have as $n \rightarrow \infty$ that

$$P_0(x_n, n) = o(1) .$$

Thus, $Q_\epsilon(m, n) \rightarrow 0$ as $n \rightarrow \infty$. This concludes the proof of part (b) of the theorem. We now prove part (a):

Fix ν in the open interval $1/2 < \nu < 2/3$, and choose n large enough so that the term $\frac{2n(1-\lambda)}{(1-2\epsilon)}$ dominates all the other terms in the upper bound for m in (a).

Now, from lemma (8.2.9) we have

$$Q_\epsilon(m,n) = \left[\sum_{k=0}^{m} \binom{m}{k} (1-2\epsilon)^k \, (2\epsilon)^{m-k} P_0(k,n) \right]^n .$$

If $k \le n$, we have $P_0(k,n) = 1$ as can be seen from equation (8.2.1); also

$$\sum_{k=0}^{m} \binom{m}{k} (1-2\epsilon)^k \, (2\epsilon)^{m-k} = 1 \quad \forall\, m \le n \,,$$

so that

$$Q_\epsilon(m,n) = 1 \;\; \forall \;\; m \le n \,. \tag{8.3.4}$$

Now, if $\lambda \ge \frac{1+2\epsilon}{2}$, we have $m \le \frac{2n(1-\lambda)}{(1-2\epsilon)} \le n$, so that part (a) of the theorem holds as a consequence of equation (8.3.4). Henceforth for part (a) we take $0 < \lambda < \frac{1+2\epsilon}{2}$.

Further, if $\epsilon = 0$, we have

$$n \; < m \; \le 2n(1-\lambda) + O(n^{3\nu-2})$$

$$= 2n(1-\lambda) + o(1)$$

as $\nu < 2/3 \implies n^{3\nu-2} \to 0$ as $n \to \infty$. Thus, for $\epsilon = 0$, theorem (8.3.1) (a) applies so that part (a) of theorem (8.3.2) holds trivially. So without loss of generality let $0 < \epsilon < 1/2$. Set

$$v_m = m(1-2\epsilon) + [m(2\epsilon)(1-2\epsilon)]^\nu \,. \tag{8.3.5}$$

Have

$$n \; < m \; \le \frac{2n(1-\lambda)}{(1-2\epsilon)} - \frac{[4\epsilon(1-\lambda)]^\nu n^\nu}{(1-2\epsilon)} + \frac{\nu[4\epsilon(1-\lambda)]^{2\nu} n^{2\nu-1}}{2(1-2\epsilon)(1-\lambda)} + o(1) \,.$$

Hence, as $n \rightarrow \infty$, we have

$$v_m < m \ ,$$

and

$$v_m \leq 2n\,(1-\lambda) + o(1) \ ,$$

as can be verified from equation (8.3.5). Hence

$$Q_\epsilon(m\,,n\,) = \left[\sum_{k=0}^{v_m} P_0(k\,,n\,) \binom{m}{k} (1\text{-}2\epsilon)^k \ (2\epsilon)^{m-k} + \sum_{k=v_m+1}^{m} P_0(k\,,n\,) \binom{m}{k} (1\text{-}2\epsilon)^k \ (2\epsilon)^{m-k} \right]^n$$

$$\geq \left[\sum_{k=0}^{v_m} P_0(k\,,n\,) \binom{m}{k} (1\text{-}2\epsilon)^k \ (2\epsilon)^{m-k} \right]^n$$

$$\geq [P_0(v_m\,,n\,)]^n \left[\sum_{k=0}^{v_m} \binom{m}{k} (1\text{-}2\epsilon)^k \ (2\epsilon)^{m-k} \right]^n \ ,$$

the last inequality following from lemma (8.2.10). Hence

$$Q_\epsilon(m\,,n\,) \geq [P_0(\, \lfloor 2n\,(1\text{-}\lambda) + o(1) \rfloor, n\,)]^n \left[\sum_{k=0}^{v_m} \binom{m}{k} (1\text{-}2\epsilon)^k \ (2\epsilon)^{m-k} \right]^n$$

$$= Q_0(\, \lfloor 2n\,(1\text{-}\lambda) + o(1) \rfloor, n\,) \left[\sum_{k=0}^{v_m} \binom{m}{k} (1\text{-}2\epsilon)^k \ (2\epsilon)^{m-k} \right]^n . \qquad (8.3.6)$$

By theorem (8.3.1) (a),

$$Q_0(\, \lfloor 2n\,(1\text{-}\lambda) + o(1) \rfloor, n\,) \rightarrow 1 \quad \text{as} \quad n \rightarrow \infty \ . \qquad (8.3.7)$$

Further, setting $p = (1\text{-}2\epsilon)$, and $m = N$, we find the hypotheses of lemma (8.2.3) are satisfied; viz.,

$$\frac{v_m - m(1-2\epsilon)}{[m(2\epsilon)(1-2\epsilon)]^{2/3}} = [m(2\epsilon)(1-2\epsilon)]^{\nu - 2/3} \to 0 \text{ as } n \to \infty ,$$

and

$$\frac{v_m - m(1-2\epsilon)}{\sqrt{m(2\epsilon)(1-2\epsilon)]}} = [m(2\epsilon)(1-2\epsilon)]^{\nu - 1/2} \to \infty \text{ as } n \to \infty ,$$

for our choice of $1/2 < \nu < 2/3$. Let

$$A \triangleq \sum_{k=0}^{v_m} \binom{m}{k} (1-2\epsilon)^k \, (2\epsilon)^{m-k} \sim \Phi\left(\frac{v_m - m(1-2\epsilon)}{\sqrt{m(2\epsilon)(1-2\epsilon)}} \right) .$$

Hence, as $n \to \infty$,

$$A^n \sim \left[1 - \Phi\left(- \frac{v_m - m(1-2\epsilon)}{\sqrt{m(2\epsilon)(1-2\epsilon)}} \right) \right]^n$$

$$\sim \left[1 - \frac{1}{\sqrt{2\pi}\,[m(2\epsilon)(1-\epsilon)]^{\nu - 1/2}} \exp\left\{ - \frac{[m(2\epsilon)(1-2\epsilon)]^{2\nu-1}}{2} \right\} \right]^n$$

$$\triangleq (1-x)^n ,$$

where

$$x = \frac{1}{\sqrt{2\pi}\,[m(2\epsilon)(1-2\epsilon)]^{\nu-1/2}} \exp\left\{ - \frac{[m(2\epsilon)(1-2\epsilon)]^{2\nu-1}}{2} \right\} .$$

So

$$0 \geq n \log A \sim n \log(1-x)$$

$$\geq -2nx .$$

The upper bound of zero follows because A is a probability, so that A, $A^n \leq 1$. The asymptotic lower bound of $-2nx$ follows from lemma (8.2.1), as for large enough m, the exponential term x is bounded between 0 and 1/2. Now,

$$\log 2nx = \log n + \log x + \log 2$$

$$= \log n - \left(\nu - 1/2\right) \log \left[m\,(2\epsilon)(1-2\epsilon)\right] - \frac{[m\,(2\epsilon)(1-2\epsilon)]^{2\nu-1}}{2} + 1/2 \log \frac{2}{\pi}$$

$$\leq \log n - n^{2\nu-1} \frac{[2\epsilon(1-2\epsilon)]^{2\nu-1}}{2} .$$

The last inequality follows because $\nu > ½$, and $m > n$. Now, we have $2\nu-1 > 0$ strictly, and as $\epsilon > 0$ is also strict, the term $n^{2\nu-1} \frac{[2\epsilon(1-2\epsilon)]^{2\nu-1}}{2}$ dominates the $\log n$ term for n large enough. Hence

$$\log n - n^{2\nu-1} \frac{[2\epsilon(1-2\epsilon)]^{2\nu-1}}{2} \rightarrow -\infty \text{ as } n \rightarrow \infty .$$

Thus, $\log 2nx \rightarrow -\infty$, so that $2nx \rightarrow 0^+$ as $n \rightarrow \infty$. It then follows that

$$0 \geq n \log A \gtrsim -2nx \rightarrow 0^- \text{ as } n \rightarrow \infty .$$

Hence

$$A^n = \left[\sum_{k=0}^{v_m} \binom{m}{k} (1-2\epsilon)^k \; (2\epsilon)^{m-k} \right]^n \rightarrow 1 \text{ as } n \rightarrow \infty . \qquad (8.3.8)$$

Rewriting equation (8.3.6) we now have

$$Q_0(\lfloor 2n\,(1-\lambda)+o(1)\rfloor, n\,) \times \left[\sum_{k=0}^{v_m} \binom{m}{k} (1-2\epsilon)^k \; (2\epsilon)^{m-k} \right]^n \leq Q_\epsilon(m\,,n\,) \leq 1 .$$

From equations (8.3.7) and (8.3.8), we see that the lower bound for $Q_\epsilon(m\,,n\,) \rightarrow 1$ as $n \rightarrow \infty$. Hence $Q_\epsilon(m\,,n\,) \rightarrow 1$ as $n \rightarrow \infty$. □

Corollary 8.3.2. $C_\epsilon(n) = \frac{2n}{1-2\epsilon}$ is the epsilon capacity of neural networks.

Proof. From theorem (8.3.2) (b) it follows that $\frac{2n}{1-2\epsilon}$ is a sequence of upper epsilon capacities. Now, let $0 < \lambda^* < 1$ be arbitrary, and choose $m \leq \frac{2n(1-\lambda)^*}{1-2\epsilon}$. For every choice of λ^*, we can find $\lambda = \lambda^*(1 + o(1))$ such that for every choice of $1/2 < \nu < 2/3$, and for large enough n, $0 < \lambda < 1$, and m satisfies the inequality in part (a) of theorem (8.3.2). (This follows because we can write the inequality for m in theorem (8.3.2) (a) as $m \leq \frac{2n(1-\lambda)(1-o(1))}{1-2\epsilon}$.) Hence, for every choice of $0 < \lambda^* < 1$, we have $Q_\epsilon(m,n) \to 1$ as $n \to \infty$ if $m \leq \frac{2n(1-\lambda^*)}{1-2\epsilon}$, so that $\frac{2n}{1-2\epsilon}$ is also a sequence of lower epsilon capacities. □

As for the theorem (8.3.1), the provable epsilon capacity results are somewhat looser if we restrict the fundamental memories to be binary n-tuples only, instead of allowing them to be chosen from real n-space. In this case, while $\left\{\frac{2n}{1-2\epsilon}\right\}$ is a sequence of upper epsilon capacities, the lower epsilon capacity cannot exceed $2n(1 + o(n))$. We conjecture that $\frac{2n}{1-2\epsilon}$ is actually a sequence of lower epsilon capacities, so that the epsilon capacity of neural networks (under the proviso that the fundamental memories are also binary n-tuples) may also be conjectured to be $\frac{2n}{1-2\epsilon}$. The result is, however, not yet fully rigourous. Note that, $\frac{n}{1-2\epsilon}$ is a sequence of lower epsilon capacities, as a consequence of lemma (8.2.6).

Our capacity results, so far, are for the case of associative mappings of the form $\mathbf{u}^{(\alpha)} \mapsto \mathbf{v}^{(\alpha)}$, where $\mathbf{u}^{(\alpha)}$ and $\mathbf{v}^{(\alpha)}$ are randomly specified fundamental and associated memories, respectively. This nominally corresponds to the case of hetero-association. An allied form of association is auto-association where the fundamental and the associated memories are the same, i.e., $\mathbf{u}^{(\alpha)} = \mathbf{v}^{(\alpha)}$. For autoassociation we essentially require that the fundamental memories be fixed points of the neural network, at least for the case where we require perfect recall. While the capacity results above hold *in toto* for autoassociation, some strictures do apply on the allowable choice of linear transformations. That restrictions of some form are required can be easily seen: a choice of interconnection weights corresponding to the identity matrix, $w_{ij} = \delta_{ij}$, results in *all* the 2^n possible state vectors being fixed points! It is clearly ridiculous to claim that the capacity is hence 2^n. The point of essence here is *programmability*; we would like to be able to specify linear transformations dependent upon the actual choice of fundamental memories, and in such a way that not too many extraneous stable points (corresponding to spurious memories) are created. With a choice such as the identity transformation, for example, the element of programmability is lost, and the choice of the transformation has become independent of the chosen memories themselves. One consequence is that all states become stable, and there can clearly be no basin or well of associative attraction. To avoid this type of problem, it suffices to discard from consideration all "identity type" transformations, as made precise in the following constraint.

Sufficient condition for autoassociative storage: Let $\mathbf{w}_i = (w_{i1}, \ldots, w_{in})$, represent the vector of interconnection weights corresponding to neuron i, and let $\mathbf{e}_i$ be a standard basis n-vector with a one in the i-th position, and zeroes everywhere else. For purposes of autoassociation, it suffices to restrict consideration to choices of interconnection weights satisfying the following inequality:

$$| w_{ii} | < \left(1 - \frac{1}{n} \right)^{½} ||\mathbf{w}_i|| \,.$$

The above inequality is fully equivalent to requiring that

$$\cos^{-1} \left(\frac{\langle \mathbf{w}_i , \mathbf{e}_i \rangle}{||\mathbf{w}_i||} \right) + \cos^{-1} \left(\frac{1}{\sqrt{n}} \right) > \frac{\pi}{2} \,.$$

Note that the argument of the first inverse cosine on the left hand side is $\dfrac{w_{ii}}{\left\{ \sum_{j=1}^{n} w_{ij}^2 \right\}^{1/2}}$; this is just the relative strength of the diagonal term w_{ii} *vis-à-vis* the length of the vector $\mathbf{w}_i$. The right hand side is just the modulus of the angle between any axis and any vertex of the hypercube. The inequality simply specifies that the angle between the weight vector $\mathbf{w}_i$ and the unit vector $\mathbf{e}_i$ pointing along the i-th axis be at least as large as the angle between the axis and any vertex. Choices of weights satisfying the above constraint will ensure that the diagonal term will not dominate the non-diagonal terms. A simpler requirement than the above is simply to require that the weight matrix is zero-diagonal, $w_{ii} = 0$, i.e., no neuron talks to itself. With such restrictions the capacity results continue to hold for auto-association.

4. DISTRIBUTION OF ERRORS

Recall that the distribution of errors (more accurately, the distribution of "don't care" decisions) is determined by those random decision sets $V_i^{(\alpha)}$ which are exceptional, i.e., which specify a "don't care" decision. Because of the binomial distribution of choice of "don't care" decisions, each associated memory has an average of $2\epsilon n$ "don't care" components–half of these on average resulting in errors, and the other half resulting in correct decisions. However, we might query whether, when all the associated memories are considered jointly, there is a significant probability that there are much more or much less than $2\epsilon n$ "don't care" decisions for many of the memories.

Our concern stems from an alternate, but appealing, deterministic definiton of epsilon capacity, wherin we require that each of the associated memories has exactly $2\epsilon n$ fixed "don't care" decisions. It would be meet if our probabilistic choice of "don't care" decisions from a binomial distribution resulted in essentially the number of "don't care" decisions prescribed by some deterministic choice.

An alternate problem of some importance is whether the capacity results are strongly tied to the choice of a binomial distribution for "don't care" decisions–natural though it be in some sense.

A. Strong Convergence to Mean Error Rate

By the Strong Law of Large Numbers, *each* individual memory has $2\epsilon n$ "don't care" decisions attributed to it with probability one. However, if our definition of epsilon capacity is to jell with the deterministic definition, we require that the above attribute holds, not just individually, but jointly for each of a very large number, m, of associations. As m increases very rapidly with n, it is not clear whether the joint probability of essentially $2\epsilon n$ "don't care" decisions for each associated memory also approaches one. In fact, we will prove the following rather stronger assertion after developing some notation.

Assertion. As $n \to \infty$, the number of "don't care" decisions attributed to each of the m associated memories approaches $2\epsilon n$; further, for each neural site $i=1,\ldots,n$, the number of exceptional decison sets $V_i^{(\alpha)} = \mathbb{B}$ approaches $2\epsilon m$.

Let $\{m_n\}_{n=1}^{\infty}$ denote the sequence of the number of associations to be stored indicated explicitly as a function of n. We require that m_n satisfy $\frac{n}{1-2\epsilon} \leq m_n \leq \frac{2n}{1-2\epsilon}$, for some fixed error tolerance $0 \leq \epsilon < 1/2$.

Let $X_i^{(\alpha)}$ be the indicator of exceptional decision sets $V_i^{(\alpha)}$, i.e.,

$$X_i^{(\alpha)} = \begin{cases} 1 & \text{if } V_i^{(\alpha)} \text{ is exceptional} \\ 0 & \text{if } V_i^{(\alpha)} \text{ is normal .} \end{cases}$$

We form the random sums

$$S^{(\alpha)}(n) = \sum_{i=1}^{n} X_i^{(\alpha)} \quad , \quad \alpha=1,\ldots,m_n \quad ,$$

$$S_i(n) = \sum_{\alpha=1}^{m_n} X_i^{(\alpha)} \quad , \quad i=1,\ldots,n \quad .$$

$S^{(\alpha)}(n)$ is clearly the number of "don't care" decisions attributed to associated memory $\mathbf{v}^{(\alpha)}$, and $S_i(n)$ is the corresponding number of "don't care" decisions attributed to neural site i. The assertion can now be seen to be equivalent to the following statement:

$$S^{(\alpha)}(n) \rightarrow 2\epsilon n \quad , \quad \alpha=1,\ldots,m_n \ ,$$

and

$$S_i(n) \rightarrow 2\epsilon m_n \quad , \quad i=1,\ldots,n \quad , \text{ as } n \rightarrow \infty \ . \tag{8.4.1}$$

We now prove that the assertion holds in the sense that with probability one $S^{(\alpha)}(n) - 2\epsilon n$, and $S_i(n) - 2\epsilon m_n$ become, and *remain* small for *each* $\alpha=1,\ldots,m_n$, and $i=1,\ldots,n$.

Theorem 8.4.1. The assertion (8.4.1) holds with probability one.

Proof. Let $\{\varsigma_n\}$, and $\{\eta_n\}$ be given sequences of positive real numbers. For $\alpha=1,\ldots,m_n$, let $E^{(\alpha)}(n,\varsigma_n)$ be the event

$$| \ S^{(\alpha)}(n) - 2\epsilon n \ | \ > \varsigma_n \ , \tag{8.4.2}$$

and for $i=1,\ldots,n$, let $E_i(n,\eta_n)$ be the event

$$| \ S_i(n) - 2\epsilon m_n \ | \ > \eta_n \ . \tag{8.4.3}$$

Define the composite event

$$E(n,\varsigma_n,\eta_n) = \bigcup_{i=1}^{n} E_i(n,\eta_n) \bigcup_{\alpha=1}^{m_n} E^{(\alpha)}(n,\varsigma_n) \ .$$

To prove the theorem it suffices to show that for every fixed $\varsigma > 0$, and every fixed $\eta > 0$, with probability one, there occur only finitely many of the events $E(n, n\varsigma, n\eta)$. Fix a constant $c > 2$. In (8.4.2) and (8.4.3), set

$$\varsigma_n = \left[\left(2c \log n + 2\log\left(\frac{4}{1-2\epsilon}\right)\right) n(2\epsilon)(1-2\epsilon)\right]^{1/2}, \tag{8.4.4}$$

and

$$\eta_n = \left[\left(2c \log n + 2\log\left(\frac{4}{1-2\epsilon}\right)\right) m_n(2\epsilon)(1-2\epsilon)\right]^{1/2}, \tag{8.4.5}$$

and consider the events $E^{(\alpha)}(n, \varsigma_n)$, $\alpha=1,\ldots,m_n$, and $E_i(n, \eta_n)$, $i=1,\ldots,n$. Have $\frac{n}{1-2\epsilon} \le m_n \le \frac{2n}{1-2\epsilon}$. Hence

$$\frac{\varsigma_n}{[n(2\epsilon)(1-2\epsilon)]^{2/3}}, \quad \frac{\eta_n}{[m_n(2\epsilon)(1-2\epsilon)]^{2/3}} \to 0 \text{ as } n \to \infty,$$

and

$$\frac{\varsigma_n}{\sqrt{n(2\epsilon)(1-2\epsilon)}}, \quad \frac{\eta_n}{\sqrt{m_n(2\epsilon)(1-2\epsilon)}} \to \infty \text{ as } n \to \infty.$$

The hypotheses of lemma (8.2.3) are satisfied, and hence

$$\mathbf{P}\left\{E^{(\alpha)}(n, \varsigma_n)\right\} \sim \frac{1}{\sqrt{2\pi}\,\varsigma_n} \exp\left\{-[c \log n + \log(\frac{4}{1-2\epsilon})]\right\},$$

$$\mathbf{P}\left\{E_i(n, \eta_n)\right\} \sim \frac{1}{\sqrt{2\pi}\,\eta_n} \exp\left\{-[c \log n + \log(\frac{4}{1-2\epsilon})]\right\}.$$

Thus, at least for large enough n,

$$\mathbf{P}\left\{E(n, \varsigma_n, \eta_n)\right\} \le \sum_{\alpha=1}^{m_n} \mathbf{P}\left\{E^{(\alpha)}(n, \varsigma_n)\right\} + \sum_{i=1}^{n} \mathbf{P}\left\{E_i(n, \eta_n)\right\}$$

$$< (m_n + n) \exp \left\{-[c \log n + \log (\frac{4}{1-2\epsilon})]\right\}$$

$$\leq 2m_n \exp \left\{-[c \log n + \log (\frac{4}{1-2\epsilon})]\right\}$$

$$\leq \left(\frac{4n}{1-2\epsilon}\right) \exp \left\{-[c \log n + \log (\frac{4}{1-2\epsilon})]\right\}$$

$$= \frac{1}{n^{c-1}}, \; c > 2 .$$

Hence $\sum \mathbf{P} \left\{E(n, \varsigma_n, \eta_n)\right\}$ converges. By lemma (8.2.7) we then have that with probability one only finitely many of the events $E(n, \varsigma_n, \eta_n)$ occur with ς_n and η_n given by equations (8.4.4) and (8.4.5), respectively.

Now assume the event $E(n, n\varsigma, n\eta)$ occurs for fixed $\varsigma > 0$, $\eta > 0$. Have

$$n\varsigma > \varsigma_n = \left[\left(2c \log n + 2 \log (\frac{4}{1-2\epsilon})\right) n (2\epsilon)(1-2\epsilon)\right]^{1/2},$$

and

$$n\eta > \eta_n = \left[\left(2c \log n + 2 \log (\frac{4}{1-2\epsilon})\right) m_n (2\epsilon)(1-2\epsilon)\right]^{1/2},$$

for n sufficiently large. Hence, if the event $E(n, n\varsigma, n\eta)$ occurs, then so does the event $E(n, \varsigma_n, \eta_n)$. Thus, if infinitely many of the events $E(n, n\varsigma, n\eta)$ occur, then so do infinitely many of the events $E(n, \varsigma_n, \eta_n)$, and this has probability zero.

Clearly the above argument holds for every fixed $\varsigma > 0$, $\eta > 0$. Thus, for every fixed $\varsigma > 0$, $\eta > 0$, with probability one there occur only finitely many of the events $E(n, n\varsigma, n\eta)$. □

The number of components of each retrieved memory that are treated as "don't care" is hence essentially $2\epsilon n$. As the number of components in error will be half this number with high probability for sufficiently large n , it follows that the number of components in error in each of the retrieved memories is essentially ϵn . Similarly, the number of erroneous decisions made at each neural site is essentially ϵm .

B. Universality of Capacity Bounds

Thus far we have considered a particular mechanism for the introduction of error tolerance into the decision process. The specification of the decisions to be labeled "don't cares" was through the agency of the random decision sets $V_i^{(\alpha)}$ assumed to be drawn from a binomial distribution corresponding to a sequence of mn Bernoulli trials, each with probability 2ϵ of resulting in a "don't care" decision.

This particular mode of choice of "don't care" decisions is natural and intuitively appealing as it outlines a method of specifying "don't care" decisions in a random and independent fashion. Further, for any two particular choices of sequences of "don't care" decisions, the choice with fewer "don't cares" is more likely in accordance with our preference for more accurate recall. However, as we saw from the theorem of the last section, *typical* sequences of "don't cares" are essentially $2\epsilon n$ in number for each associated memory. It is hence reasonable to say that the resultant upper epsilon capacity of $\frac{2n}{1-2\epsilon}$ indeed specifies a tolerance of up to $2\epsilon n$ errors in decision for each associated memory.

The notion of introducing the random decision sets $V_i^{(\alpha)}$ to specify decisions which we treat as "don't cares" is much more general, however. In the general case we could specify the values taken by $V_i^{(\alpha)}$, $i=1,\ldots,n$, $\alpha=1,\ldots,m$, to be taken from some product space, with probabilities of points in th joint ensemble given by some suitable distribution

$$\mathbf{P}\left\{V_i^{(\alpha)} = d_i^{(\alpha)} \, , \, i=1,\ldots,n \, , \, \alpha=1,\ldots,m\right\} = p\left(d_i^{(\alpha)}\right) , \qquad (8.4.6)$$

which need not, in general, correspond to the binomial distribution. The only requirement we impose on the distribution of choice is that (at least in an asymptotic sense) the number of "don't care" decisions approaches $2\epsilon n$ for each associated memory with high probability, where $\epsilon \in [0,1/2)$ is the prescribed error tolerance.

Thus, if for the chosen distribution we denote by $f_\epsilon(m,n)$ the probability that for each associated memory $\mathbf{v}^{(\alpha)}$, the number of "don't care" decisions is less than or equal to a quantity $\rho(\epsilon,n) \sim 2\epsilon n$, then we require that

$$f_\epsilon(m,n) \rightarrow 1 \text{ as } n \rightarrow \infty \ \forall\, m \leq cenu \ . \tag{8.4.7}$$

Note that distributions of the form (8.4.6) also include deterministic choices of exactly $2\epsilon n$ "don't care" decisions for each associated memory. (For such cases equation (8.4.7) is of course trivially satisfied.) For deterministic choices, the distribution (8.4.6) has the entire probability mass concentrated at a single point, i.e., $p(d_i^{(\alpha)})$ assumes the value one if $d_i^{(\alpha)} = \mathbb{B}$ for $2\epsilon mn$ pairs $(i,\alpha) = (i_1,\alpha_1),\ldots,(i_{2\epsilon mn},\alpha_{2\epsilon mn})$, and $d_i^{(\alpha)} = \{v_i^{(\alpha)}\}$, $(i,\alpha) \neq (i_j,\alpha_j)$, and assumes the value zero otherwise.

Example 1. (Deterministic choice)

$V_i^{(\alpha)}$ is exceptional if $i \leq 2\epsilon n$ for each α, and $V_i^{(\alpha)}$ is normal if $i \geq 2\epsilon n+1$ for each α.

By means of this particular (deterministic) distribution of choice of "don't care" decisions, we label the first $2\epsilon n$ decisions for each associated memory as "don't cares," and require correct classification for all the remaining decisions. (We might suspect that choosing "don't care" decisions in such an uninspired fashion, with many neural sites not benefiting at all, will not gain us in capacity. The suspicion turns out to be well founded.) □

Example 2. (Deterministic choice)

$V_i^{(\alpha)}$ is exceptional if i and α jointly satisfy the inequalities $k(2\epsilon n)+1 \leq i \leq (k+1)2\epsilon n$, and $k(2\epsilon m)+1 \leq \alpha \leq (k+1)2\epsilon m$, for $k=0,1,...,1/2\epsilon$, and $V_i^{(\alpha)}$ is normal otherwise. (For simplicity we assume that $1/2\epsilon$ is an integer.)

Consider an $n \times m$ matrix with rows corresponding to the n neural sites i, and with columns corresponding to the m associated memories $\mathbf{v}^{(\alpha)}$, which has entries 1 at positions (i,α) where the corresponding decision sets $V_i^{(\alpha)}$ is exceptional, and entries 0 where the corresponding decision sets $V_i^{(\alpha)}$ are normal. The above fixed choice of "don't cares" represents a block diagonal matrix with each block being an $2\epsilon n \times 2\epsilon m$ submatrix with all entries being 1. (In the case of example 1, the corresponding matrix has 1's in its first $2\epsilon n$ rows, and 0's in all the other rows.)

It is clear from the structure of the "don't care" matrix that all neural sites benefit from "don't care" decisions to exactly the same extent, so that, in light of theorem (8.4.1), we might expect some rewards in capacity. This indeed turns out to be the case. □

Example 3. (Markovian choice)

We now consider a probabilistic choice of distribution of "don't cares." Instead of having errors distributed randomly, and independently, as in the binomial distribution, we now require that the errors tend to cluster together (but not quite to the same extent as in examples 1 and 2.) We specify a Markovian distribution of "don't cares" as follows:

The decison sets $V_i^{(\alpha)}$ are the outcomes of a sequence of mn experiments such that for every $\alpha \neq \beta$, the outcomes $V_1^{(\alpha)}, \ldots, V_n^{(\alpha)}$ are jointly independent of the outcomes $V_1^{(\beta)}, \ldots, V_n^{(\beta)}$. Each $V_i^{(\alpha)}$ has an *a priori* distribution

$$V_i^{(\alpha)} = \{v_i^{(\alpha)}\} \text{ with probability } 1\text{–}2\epsilon$$

$$= \mathbb{B} \quad \text{with probability } 2\epsilon\ .$$

For each α, the outcomes $V_i^{(\alpha)}$ form a Markov chain with

$$\mathbf{P}\left\{V_{i+1}^{(\alpha)} = \mathbb{B}\ ,\ V_i^{(\alpha)} = \mathbb{B}\right\} = \theta/2\ ,$$

$$\mathbf{P}\left\{V_{i+1}^{(\alpha)} = \mathbb{B}\ ,\ V_i^{(\alpha)} = \left\{v_i^{(\alpha)}\right\}\right\} = (1-\theta)/2\ ,$$

$$\mathbf{P}\left\{V_{i+1}^{(\alpha)} = \left\{v_{i+1}^{(\alpha)}\ ,\ V_i^{(\alpha)} = \mathbb{B}\right\} = (1-\theta)/2\ ,$$

$$\mathbf{P}\left\{V_{i+1}^{(\alpha)} = \left\{v_{i+1}^{(\alpha)}\ ,\ V_i^{(\alpha)} = \left\{v_i^{(\alpha)}\right\}\right\} = \theta/2\ ,$$

for $i=1,\ldots,n-1$, and where $\theta > \frac{1}{2}$. The joint distribution of (8.4.6) is then given by

$$p\,(d_i^{(\alpha)}) = \prod_{\alpha=1}^{m} p\,(d_n^{(\alpha)} \mid d_{n-1}^{(\alpha)})\, p\,(d_{n-1}^{(\alpha)} \mid d_{n-2}^{(\alpha)}) \cdots p\,(d_2^{(\alpha)} \mid d_1^{(\alpha)})\, p\,(d_1^{(\alpha)})\ ,$$

with the conditional probabilities given by the Markov chain transition probabilities.

□

With a plethora of possible distributions of "don't care" decisions confronting us, we might wonder if by some suitable choice of distribution (8.4.6) we could obtain a significant increase in capacity over the $\frac{2n}{1-2\epsilon}$ result obtained earlier using a binomial distribution of choice. The answer is furbished by the following:

Theorem 8.4.2. For any mode of choice of "don't care" decisions determined by a probability distribution (8.4.6), the upper epsilon capacity is bounded from above by $\frac{2n}{1-2\epsilon}$.

Before we prove the above statement, a digression:

Lemma 8.4.1. Let $2\epsilon mn$ exceptional decision sets $V_i^{(\alpha)}$ be given. For each $i=1,\ldots,n$, let e_i be the number of "don't care" decisions corresponding to neural site i , $e_1+e_2+\ldots+e_n = 2\epsilon mn$. Then:

(a) If $\gamma n = \min(e_i) < 2\epsilon m$, then for each $\lambda > 0$, and m such that $\frac{2n}{1-\gamma} \leq m \leq \frac{2n\,(1-\lambda)}{1-2\epsilon}$, we have

$$\prod_{i=1}^{n} P_0(m-e_i, n) < [P_0(m-2\epsilon m, n)]^n \tag{8.4.8}$$

for n large enough. Equality holds in (8.4.8) $\forall\ m$ if $\gamma = 2\epsilon$.

(b) If $\delta m = \min\left\{e_i : e_i \neq 0\right\} < m$, then for each $\lambda > 0$, and m such that $2n \leq m \leq \frac{2n\,(1-\lambda)}{1-\delta}$, we have

$$\prod_{i=1}^{n} P_0(m-e_i, n) > [P_0(m, n)]^{n-2\epsilon n} \tag{8.4.9}$$

for n large enough. Equality holds in (8.4.9) $\forall\ m$ if $\delta = 1$.

Proof. To avoid technicalities, we assume $2\epsilon n$ and $2\epsilon m$ are integers. In part (a), if $\gamma = 2\epsilon$, then $e_i = 2\epsilon m$ for each i so that (8.4.8) holds for this case. Assume $\gamma < 2\epsilon$. Have

$$\prod_{i=1}^{n} P_0(m-e_i, n) < P_0(m-\gamma m, n) \leq 1/2$$

whenever $m-\gamma m \geq 2n$ by theorem (8.3.1) (b) and lemma (8.4.1). Moreover, by theorem (8.3.1) (c), $[P_0(m-2\epsilon m, n)]^n \rightarrow 1$ as $n \rightarrow \infty$ $\forall\ m$ such that $m-2\epsilon m \leq 2n\,(1-\lambda)$ for each $\lambda > 0$. This completes the proof of part (a).

For part (b), equality in (8.4.9) is obvious when $\delta = 1$. Now assume $\delta < 1$. Without loss of generality we can assume that $e_i > 0$ for each i, as otherwise we can cancel the corresponding terms $P_0(m, n)$ from both sides of (8.4.9). Hence, by theorem (8.3.1) (c)

$$\prod_{i=1}^{n} P_0(m-e_i, n) \geq [P_0(m-\delta m, n)]^n \to 1 \text{ as } n \to \infty$$

whenever $m-\delta m \leq 2n(1-\lambda)$ for each $\lambda > 0$, and

$$[P_0(m, n)]^{n-2\epsilon n} \leq P_0(m, n) \to 0 \text{ as } n \to \infty$$

whenever $m \geq 2n$. □

$\prod P_0(m-e_i, n)$ is simply the maximum probability that we can find a neural network with the desired distribution of "don't care" decisions. Note that the probability upper bound in (8.4.8) corresponds to the distribution of example 2, while the lower bound in (8.4.9) corresponds to the distribution of example 1. Lemma (8.4.1) in conjunction with theorem (8.3.1) yields the following:

Proposition 8.4.1. For any particular choice of exactly $2\epsilon mn$ "don't care" decisions, the following hold:

(a) The maximum upper epsilon capacity of $\frac{2n}{1-2\epsilon}$ is achieved iff each associated memory has $2\epsilon n$ "don't care" decisions attributed to it, and each neural site has $2\epsilon m$ "don't care" decisions attributed to it.

(b) The maximum achievable upper epsilon capacity is $2n$ iff there is at least one neural site which has no "don't care" decisions attributed to it.

(c) The maximum achievable upper epsilon capacity lies between $2n$ and $\frac{2n}{1-2\epsilon}$ for any choice of $2\epsilon mn$ "don't care" decisions.

Thus, for any deterministic choice of $2\epsilon mn$ "don't care" decisions, the capacity is bounded from above by $\frac{2n}{1-2\epsilon}$. In light of constraint (8.4.7), hence, it is not surprising that theorem (8.4.2) holds.

Proof of theorem (8.4.2). As before, let E denote the event that $\exists$ a neural network with tolerance ϵ. Let (8.4.6) denote the distribution of choice, subject only to constraint (8.4.7). By lemma (8.4.1),

$$\mathbf{P}\left\{E\right\} \leq f_{\epsilon}(m,n)\,[P_0(m(1-2\epsilon),n)]^n + 1 - f_{\epsilon}(m,n)\,.$$

By theorem (8.3.1), and equation (8.4.7), the upper bound converges to one as $n \to \infty$, whenever $(1-2\epsilon)m \leq 2n(1-\lambda)$ for each $\lambda > 0$, and converges to zero whenever $(1-2\epsilon)m \geq 2n(1+\lambda)$ for each $\lambda > 0$. Hence the upper epsilon capacity is bounded from above by $\frac{2n}{1-2\epsilon}$ for all distributions (8.4.6) satisfying constraint (8.4.7). □

Recapitualting, we started out with a definition of epsilon capacity based on a binomial distribution of errors. As a consequence of theorem (8.4.1) it turned out that the original definition of epsilon capacity approached the deterministic definition of epsilon capacity in a rather strong sense, with the epsilon capacity being $\frac{2n}{1-2\epsilon}$ for a binomial distribution of errors. Note, however, that the capacity bound may not be tight for arbitrary distributions of choice (as in example 1, for instance).

5. OPTIMAL WEIGHT MATRICES

The capacity results of the previous sections indicate that as long as the number of associations to be stored is within the capacity, then with high probability, there exist weight matrices which store the required associations within the required error tolerance. While it is gratifying to know that there exist networks which can store the prescribed associations, practical interest centres on whether these optimal weight matrices can be found. Fortunately, there exist iterative techniques which yield the desired interconnection weights.

We fruitfully employ the formal analogy between McCuloch-Pitts neurons and Rosenblatt's *perceptroñ* [9] to devise an iterative technique to obtain optimal weight matrices. The procedure is based upon *reinforcement learning* which utilises single-sample correction, and can be used to "learn" the rows of the weight matrix one by one.

Let $\mathbf{w}_i[0] = (w_{i1}[0],...,w_{in}[0])$ be some arbitrary initial choice of weight vector corresponding to the i-th row of the weight matrix $\mathbf{W}$. The following iterative scheme prescribes a sequence of weight vectors $\{\mathbf{w}_i[k]\}$ which converges to an optimal weight vector in a finite number of steps. (By an optimal weight vector $\mathbf{w}_i$, we mean a vector whose components form the i-th row of some optimal matrix for storing $\frac{2n}{1-2\epsilon}$ associations with tolerance ϵ.)

We assume that for the prescribed choice of "don't care" components, there exists a neural network which stores $\frac{2n}{1-2\epsilon}$ associations with the prescribed error distribution. (We know that this is ensured with high probability.)

Let the i-th components of the associated memories which have been specified to be accurately retrieved be denoted by $v_i^{(\alpha_1)}, \ldots, v_i^{(\alpha_{j_i})}$. The fundamental memories $\mathbf{u}^{(\alpha_k)}$, $k = 1,...,j_i$, are presented cyclically to neuron i, and the weight vector $\mathbf{w}_i$ is modified if an incorrect decision $(-v_i^{(\alpha_k)})$ is returned. Correction stops only when all the specified memories $\mathbf{u}^{(\alpha_k)}$, $k = 1,...,j_i$, are classified correctly by neuron i. As the weight vector is modified only when there is a misclassification, we may just as well consider only the sequence of misclassified memories. Let $\mathbf{u}[k]$ denote the sequence of misclassified fundamental memories, and let $\mathbf{v}[k]$ denote the corresponding sequence of associated memories. The incremental rule for generating a sequence of weight vectors corresponding to the i-th row of the weight matrix $\mathbf{W}$ is given by

$$\mathbf{w}_i[0] \text{ arbitrary}$$

$$\mathbf{w}_i[k+1] = \mathbf{w}_i[k] + \rho_k \, v_i[k] \, \mathbf{u}[k],$$

where ρ_k is a positive, incremental sequence.

It is easily seen that at each correction, the new weight vector tends to diminish the error in the previous misclassification by adding a term $\rho_k \, v_i[k] \left\langle \mathbf{u}[k], \mathbf{u}[k] \right\rangle = \rho_k n \, v_i[k]$ to the previous potential seen by the i-th neuron. The following result now assures us that the procedure actually converges.

Theorem 8.5.1. (Perceptron Convergence Theorem)
If the incremental sequence $\{\rho_k\}$ satisfies

$$\rho_k \geq 0 ,$$

$$\lim_{K \to \infty} \sum_{k=1}^{K} \rho_k = \infty ,$$

and

$$\lim_{K \to \infty} \frac{\sum_{k=1}^{K} \rho_k^2}{\left(\sum_{k=1}^{K} \rho_k \right)^2} = 0 ,$$

then the sequence $\mathbf{w}_i\ [\ k\ ]$ converges to an optimal solution vector $\hat{\mathbf{w}}_i$ (if it exists) in a finite number of steps.

Proof. cf. [10].

Particular choices of increments that can be useful are the fixed increment, $\rho_k = \text{constant} > 0$, and the harmonic series increment, $\rho_k = \frac{1}{k}$. A factor that can be fruitfully employed where attraction is desired, is the usage of margins. Here, a margin $M > 0$ is specified, and a fundamental memory is deemed misclassified at the i-th position if $\left\{ \sum_{j=1}^{n} w_{ij}[\ k\ ]\ u_j[\ k\ ] \right\} v_i[\ k\ ] - M < 0$. The Perceptron Convergence theorem applies for this case also.

Thus, if an optimal solution matrix exists, we can apply the reinforcement learning scheme to sequentially obtain each of the constituent optimal weight (row) vectors $\hat{\mathbf{w}}_1, \ldots, \hat{\mathbf{w}}_n$, of an optimal weight matrix $\hat{\mathbf{W}}$. Other techniques exist for learning the optimal weights, such as the *relaxation procedure* and the *Ho-Kashyap* methods (cf. [11], for instance). All these techniques are descent procedures, which utilise incremental learning in some form or the other.

References

[1] S. S. Venkatesh, "Epsilon capacity of neural networks," *Conf. on Neural Networks for Computing,* Snowbird, Utah, May 1986.

[2] W. Feller, *An Introduction to Probability Theory and its Applications*, vol. I, 3rd Edition. New York: Wiley, 1968.

[3] H. Chernoff, "A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations," *Ann. Math. Stat.*, vol. 23, pp. 493–507, 1952.

[4] J. G. Wendel, "A problem in geometric probability," *Mathematica Scandivica*, vol. 11, pp. 109–111, 1962.

[5] L. Schlafli, *Gesammelte Mathematische Abhandlungen I.* Basel, Switzerland: Verlag Birkhäuser, pp. 209–212, 1950.

[6] J. Komlós, "On the determinant of (0,1) matrices," *Studia Scientarum Mathematicarum Hungarica*, vol. 2, pp. 7–21, 1967.

[7] B. Widrow, "Generalization and information storage in networks of adaline 'neurons,'" *Self Organizing Systems.* Washington: Spartan Books, pp. 442, 459, 1962.

[8] Y. S. Abu-Mostafa and J. S. Jacques, "Information capacity of the Hopfield model," *IEEE Trans. Inform. Theory,* vol. IT-31, pp. 461–464, 1985.

[9] F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms.* Washington: Spartan Books, 1962.

[10] M. Minsky and S. Papert, *Perceptrons: An Introduction to Computational*

Geometry. Cambridge, Massachusetts: MIT Press, 1969.

[11] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis.* New York: Wiley, 1973.

Extensions

CHAPTER IX

SELECTED PROBLEMS

1. DISCUSSION

Our dealing with associative structures based on networks of densely interconnected neurons lends a natural interpretation to the role of the linear map, and the point rule in computation. To echo our statement in the introduction, the linear map represents *communication* of information between the various processing nodes of the system (the neurons), and the point rule provides the necessary non-linear adjunct to *logical computation.* The neural networks we have been analysing thus far, for instance, can be thought of as a host of simple threshold gates communicating massively with each other.

As we have determined, for simple structures determined by linear maps with point rules, computational capability is circumscribed fairly tightly by the number of available degrees of freedom. Natural questions that arise at this juncture are: Is it possible to significantly increase computational capability by improving communication and/or elemental computational power within the network? Alternatively, can we sacrifice some communication, and still obtain satisfactory performance?

The first issue deals with the design and specification of more complex processing structures which result in powerful computation, while the second problem deals with specifying very low complexity structures which do not sacrifice much in performance. A central issue here is the characterisation of processor capability in terms of the available communication, and the "raw" or elemental computing power available at each processing node. While beggaring an answer to either question–is a network of locally connected CRAY III's computationally superior to a network of densely interconnected microprocessors?–we will examine a few selected problems impinging on these issues as natural generalisations of the structures we have been considering. There are, of course, issues here that we do not take into consideration: questions on representation and uniformity, the issue of programmability, and the amount of "preprocessing" required. All of these factors will contribute to the determination of the ultimate worth of any processing scheme.

We will discuss four problems in this chapter, all dealing with modifications to the structure of neural networks that we have been considering. The first problem deals with the issue of using binary interconnections between neurons, thus saving considerably in implementation cost. We next consider neurons with many states, and determine whether they can be utilised in conjunction with suitable multiple threshold decision rules, and linear maps to achieve similar associative storage to the kind we have been analysing. The third issue we consider is the modification of system architecture to compensate for deterministic distortions in input patterns; we use the creation of shift invariant associative memories as an illustrative example in this regard. Finally, we discuss certain natural generalisations of neural network architecture to higher order systems which increase capacity tremendously.

2. BINARY INTERCONNECTIONS

A. Introduction

Practical implementations of the sort of neural network structure that we have been discussing hinge upon the realisability of dense synaptic interconnections. Clearly, providing of the order of n^2 interconnections provides particular problems in design, even for networks of moderate size. Additional problems arise from the fact that the synaptic interconnections may evince considerable dynamic range requirements. As we saw, synaptic interconnections realised by the outer product algorithm required a dynamic range linear in n; for more efficient structures which approach the computational limit of these structures, the dynamic range required in the synaptic strengths is exponential in n [1]. A practical question of import is whether good performance can still be obtained while placing sanctions on the allowed dynamic range for the synaptic weights.

The restriction of the synaptic weights to binary (on-off) interconnections was considered by Hopfield [2], who concludes that reasonable performance still attains if the weights corresponding to the outer product algorithm are binarised. The results of chapter IV indicate that this is indeed a reasonable approach, wherein it appears that hardlimiting the weights should not drastically alter the behaviour of the network. We shall adopt a slightly different tack.

B. Majority Rule Based Interconnections

Let $\mathbf{U} = \left\{\mathbf{u}^{(1)} \cdots \mathbf{u}^{(m)}\right\} \subseteq \mathbb{B}^n$ be an m-set of fundamental memories as before. We again assume that the random components, $u_i^{(\alpha)}$, $1 = 1,...,n$, $\alpha = 1,...,m$, of the memories are drawn from a sequence of Bernoulli trials with equal probabilities of success and failure:

$$\mathbf{P}\left\{u_i^{(\alpha)} = 1\right\} = \mathbf{P}\left\{u_i^{(\alpha)} = -1\right\} = \frac{1}{2} .$$

For each pair $(i,j) \in \left\{1,...,n\right\} \times \left\{1,...,n\right\}$, let $\left\{U_{ij}^{+}, U_{ij}^{-}\right\}$ be a dichotomy of $\mathbf{U}$ defined by

$$U_{ij}^{+} = \left\{\mathbf{u}^{(\alpha)} \in \mathbf{U} : u_i^{(\alpha)} u_j^{(\alpha)} = 1\right\} ,$$

and

$$U_{ij}^{-} = \left\{\mathbf{u}^{(\alpha)} \in \mathbf{U} : u_i^{(\alpha)} u_j^{(\alpha)} = -1\right\} . \qquad (9.2.1)$$

Majority Rule for Interconnection Weights:

$$w_{ij} = \begin{cases} 1 & \text{if} \quad | U_{ij}^+ | \geq | U_{ij}^- | \\ -1 & \text{if} \quad | U_{ij}^+ | < | U_{ij}^- | \end{cases} . \tag{9.2.2}$$

(Note that $w_{ii} = 1$, $i = 1,...,n$; this follows directly from (9.2.1), as $U_{ii}^+ = \mathbf{U}$, and $U_{ii}^- = \emptyset$ for this case.)

Form the random variables, $X_i^{(\alpha)}$, $i = 1,...n$, $\alpha = 1,...,m$, by

$$X_i^{(\alpha)} = \sum_{j=1}^{n} w_{ij}\, u_i^{(\alpha)} u_j^{(\alpha)}$$

$$= 1 + \sum_{j \neq i} w_{ij}\, u_i^{(\alpha)} u_j^{(\alpha)} . \tag{9.2.3}$$

The random variable $X_i^{(\alpha)}$ is just the potential seen by the i-th neuron when $\mathbf{u}^{(\alpha)}$ is the current state, multiplied by the binary component $u_i^{(\alpha)}$. As before, the requirement that the fundamental memories be fixed points is equivalent to the constraint that the random variables $X_i^{(\alpha)}$ are each non-negative.

We first develop a brief, ad hoc rationale for the majority rule algorithm. Define the probabilities p and q by

$$p = \mathbf{P}\left\{\mathbf{u}^{(\alpha)} \in U_{ij}^+ ,\ | U_{ij}^+ | \geq | U_{ij}^- | \right\} + \mathbf{P}\left\{\mathbf{u}^{(\alpha)} \in U_{ij}^- ,\ | U_{ij}^+ | < | U_{ij}^- | \right\} ,$$

and

$$q = 1 - p . \tag{9.2.4}$$

By virtue of the random choice of memories, the above probabilities are independent of α, i, and j. Now, it is clear that p is the probability that the component product $u_i^{(\alpha)} u_j^{(\alpha)}$ of the memory $\mathbf{u}^{(\alpha)}$ takes on the majority sign. As the components are chosen independently, it follows that the event $\{u_i^{(\alpha)} u_j^{(\alpha)}$ has majority sign$\}$ is more probable than the event $\{u_i^{(\alpha)}u_j^{(\alpha)}$ has minority sign$\}$; hence $p > 1/2$, and $q < 1/2$. From equation (9.2.2) we then have

$$\mathbf{P}\left\{w_{ij}\, u_i^{(\alpha)}u_j^{(\alpha)} = 1\right\} = p > 1/2 ,$$

and

$$\mathbf{P}\left\{w_{ij}\, u_i^{(\alpha)}u_j^{(\alpha)} = -1\right\} = q < 1/2 . \tag{9.2.5}$$

It then follows from equation (9.2.3) that

$$\mathbf{E}\,(X_i^{(\alpha)}) = 1 + (n - 1)(p - q) > 0 . \tag{9.2.6}$$

If the signal-to-noise ratio, i.e., the ratio of the square of the mean of $X_i^{(\alpha)}$ to the variance of $X_i^{(\alpha)}$, is large, then we expect that $X_i^{(\alpha)} \geq 0$ with high probability.

The following assertion is easily seen to hold.

Proposition 9.2.1. The majority rule neural network can be equivalently obtained by homogeneously thresholding the interconnection weights of an outer product neural network; specifically,

$$w_{ij} = \mathrm{sgn}\left\{\sum_{\beta=1}^{m} u_i^{(\beta)}u_j^{(\beta)} - gm\,\delta_{ij}\right\} , \tag{9.2.7}$$

where $g = 0$ or 1. As before,

$$w_{ii} = \mathrm{sgn}\left\{(1 - g)m\right\} = 1 .$$

Using equation (9.2.3) in conjunction with equation (9.2.7), we have

$$X_i^{(\alpha)} = 1 + \sum_{j \neq i} \mathrm{sgn}\left\{\sum_{\beta=1}^{m} u_i^{(\alpha)}u_j^{(\alpha)}u_i^{(\beta)}u_j^{(\beta)}\right\}$$

$$= 1 + \sum_{j \neq i} \operatorname{sgn} \left\{ 1 + \sum_{\beta \neq \alpha} u_i^{(\alpha)} u_j^{(\alpha)} u_i^{(\beta)} u_j^{(\beta)} \right\} . \tag{9.2.8}$$

In analogy with lemma (6.5.1), the following result holds:

Lemma 9.2.1. As $n \to \infty$, let m satisfy the following:
(1) $m = o(n)$, and
(2) $m \geq M(n)$, where $\frac{M(n)}{n^{2/3}} \to \infty$.

Then $\tau = \mathbf{P}\left\{X_i^{(\alpha)} < 0\right\} \sim \frac{\sqrt{m}}{2\sqrt{n}} e^{-\frac{n}{\pi m}}$.

Proof. Define the ±1 random variables

$$X_{ij}^{(\alpha)} = \operatorname{sgn} \left\{ 1 + \sum_{\beta \neq \alpha} u_i^{(\alpha)} u_j^{(\alpha)} u_i^{(\beta)} u_j^{(\beta)} \right\} .$$

Clearly,

$$X_i^{(\alpha)} = 1 + \sum_{j \neq i} X_{ij}^{(\alpha)} .$$

The random variables $X_{ij}^{(\alpha)}$ are i.i.d. as each of the terms in the β-sum are independent by lemma (6.4.5). Now,

$$p = \mathbf{P}\left\{X_{ij}^{(\alpha)} = 1\right\}$$

$$= \mathbf{P}\left\{\sum_{\beta \neq \alpha} u_i^{(\alpha)} u_j^{(\alpha)} u_i^{(\beta)} u_j^{(\beta)} \geq -1\right\} .$$

Set $v_{ij}^{(\alpha,\beta)} = u_i^{(\alpha)} u_j^{(\alpha)} u_i^{(\beta)} u_j^{(\beta)}$. Then

$$p = \mathbf{P}\left\{\sum_{\beta \neq \alpha} v_{ij}^{(\alpha,\beta)} \geq -1\right\} .$$

The sum of the i.i.d. random variables $v_{ij}^{(\alpha,\beta)}$ is a random variable governed by a symmetric binomial distribution. For the case m even, we have

$$p = \frac{1}{2} + \frac{1}{2}\,\mathbf{P}\left\{\sum_{\beta \neq \alpha} v_{ij}^{(\alpha,\beta)} = -1\right\} ;$$

while for the case m odd, we have

$$p = \frac{1}{2} + \frac{1}{2}\,\mathbf{P}\left\{\sum_{\beta \neq \alpha} v_{ij}^{(\alpha,\beta)} = 0\right\} .$$

In either case, through the application of Sterling's formula for large enogh n, we obtain

$$p \sim \frac{1}{2} + \frac{1}{\sqrt{2\pi m}} ,$$

and

$$q \sim \frac{1}{2} - \frac{1}{\sqrt{2\pi m}} .$$

Hence, as $n \rightarrow \infty$,

$$p - q \sim \left(\frac{2}{\pi m}\right)^{1/2} ,$$

and

$$pq \sim \frac{1}{4} .$$

It is easy to verify that the Large Deviation Central Limit Theorem (lemma (6.4.3)) continues to hold for the sum of the random variables $X_{ij}^{(\alpha)}$, so that

$$\tau = \mathbf{P}\left\{\sum_{\beta \neq \alpha} X_{ij}^{(\alpha)} < -1\right\}$$

$$\sim \Phi\left(\frac{-1-(n-1)\sqrt{\frac{2}{\pi m}}}{\sqrt{n-1}} \right) \sim \Phi\left(-\frac{\sqrt{2n}}{\sqrt{\pi m}} \right).$$

By choice of m, we have $\frac{\sqrt{2n}}{\sqrt{\pi m}} \rightarrow \infty$ as $n \rightarrow \infty$. The lemma follows by the asymptotic formula for the error function. □

The above result gives us the probability that a particular bit of memory is not fixed. This can be used to obtain very crude bounds on capacity. In fact, if $\epsilon \in (0,1)$ is the maximum allowable probability that any particular bit of memory is not fixed, then for small ϵ, as $n \rightarrow \infty$, $\tau \lesssim \epsilon$ if $m \leq \frac{n}{\pi \log (\frac{1}{2\epsilon})}$. The crude analysis thus gives us an upper bound on storage capacity which is linear with n (though with a small constant). To estimate the actual capacity, we utilise the Poisson conjecture for the distribution of errors, and obtain analogously with the results of chapter VI that the capacity, $C(n)$, of the majority rule algorithm for storing fixed points is

$$C(n) = \frac{n}{2\pi \log n}.$$

The result indicates that the loss in capacity resulting from the binarisation of the interconnection weights of the outer product scheme is surprisingly small. The outer product algorithm is hence robust in this sense to changes in system parameters.

We conjecture that the maximal storage capacity of networks with binary interconnections is $n - \log^* n$.

3. MULTIPLE NEURAL STATES

A. Multiple Threshold Point Rules

Our usage of a binary n-tuple as state-vector presupposes that data coming in from the real world is suitably encoded as a binary bit stream (except in those cases where the bit stream itself constitutes the datum). While this suffices for many applications, at times, however, we might wish to avail of the facility of having neurons with multiple states with each neural state, for example, being the direct encoding of some state of nature. For instance, the English alphabet together with delimiters could be directly encoded into 37 neural states. Of course, if each neuron can take on one of many possible states, then simple single threshold point rules will no more suffice to specify changes in neural state. We will investigate a particular case where inter-neural communication is through a linear map, $\mathbf{W}$, as before, but each neuron utilises a multiple-threshold point rule.

For some fixed integer κ, we assume that each neuron can take on one of $2\kappa + 1$ states in $K = \{-\kappa,...,-1,0,1,...,\kappa\}$, symmetrically about zero. The n-tuple of neural states determines the state of the network. Clearly, there are $(2\kappa + 1)^n$ possible states of the network. Interneuron communication is through the medium of a linear matrix of interconnection weights $\mathbf{W} = [w_{ij}]$ as before. Specifically, if $\mathbf{u} \in K^n$ is the present state of the system, each neuron $i = 1,...,n$ perceives a potential $\sum_{j=1}^{n} w_{ij} \; u_j$. State transitions are determined by a multiple-threshold comparison rule at each neuron. Let 2κ real scalar thresholds

$$-\infty < t_{-\kappa} < t_{-\kappa+1} < \cdots < t_{\kappa-1} < \infty$$

be fixed. Form the intervals

$$J_l = \begin{cases} [t_{l-1}, t_l) & \text{if } -\kappa+1 \leq l \leq \kappa-1 \\ (-\infty,\, t_{-\kappa}) & \text{if } l = -\kappa \\ [t_{\kappa-1}, \infty) & \text{if } l = \kappa \end{cases} .$$

Clearly $\bigcup_{l=-\kappa}^{\kappa} J_l = \mathbb{R}$, so that the intervals J_l form a partition of the real line. The decision rule at each neuron is specified as follows: if $\mathbf{u} \in K^n$ is the present state of the system, then the updated state, $u_i{}'$, of the i-ith neuron is determined as follows:

$$u_i{}' = l \ \text{ iff } \sum_{j=1}^{n} w_{ij}\ u_j \in J_l \ .$$

As before, operation can be in either synchronous or asynchronous modes.

For each neuron, then, the weights $(w_{i1},...,w_{in})$, together with the thresholds $t_{-\kappa},...,t_{\kappa-1}$, determine a set of 2κ parallel hyperplanes which effectively partition the pattern space K^n into $2\kappa+1$ decision regions. The effectiveness of this structure as an associative memory clearly depends upon an appropriate choice of weights w_{ij} and thresholds t_l.

In fig. 9.1 we illustrate the partitioning of a pattern space by multiple, parallel hyperplanes. Here we are considering a two-dimensional structure, $n=2$. The set of values that the neurons can take is $K = \{-1,0,1\}$. Fig 9.1 (a) illustrates the decision regions pertaining to neuron 1, while fig. 9.1 (b) illustrates the decision regions pertaining to neuron 2.

B. Outer Products Revisited

We again have recourse to the outer product algorithm. Let $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(m)} \in K^n$ be an m-set of fundamental memories. We assume that the components $u_i{}^{(\alpha)}$ of the fundamental memories are i.i.d. random variables with

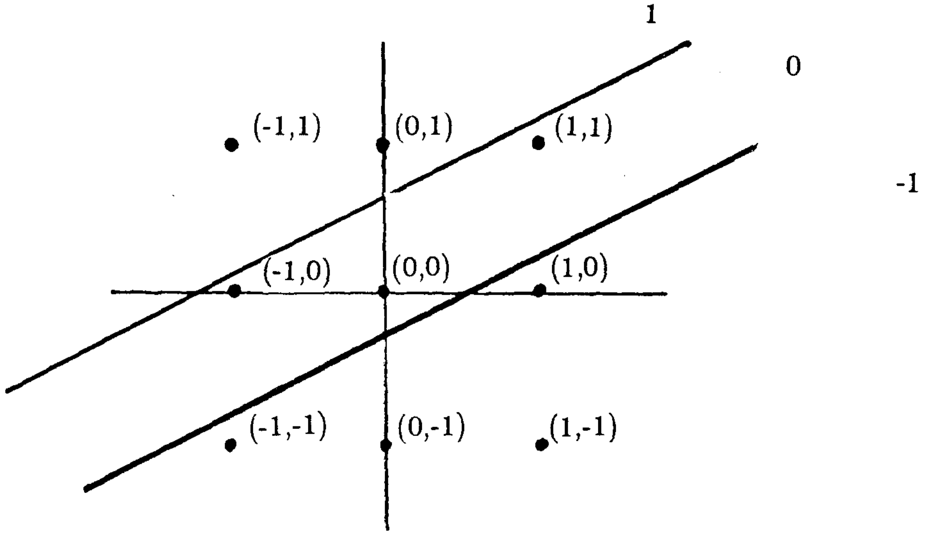

Fig. 10.1 (a). Three-state neurons: decision regions for neuron 1.

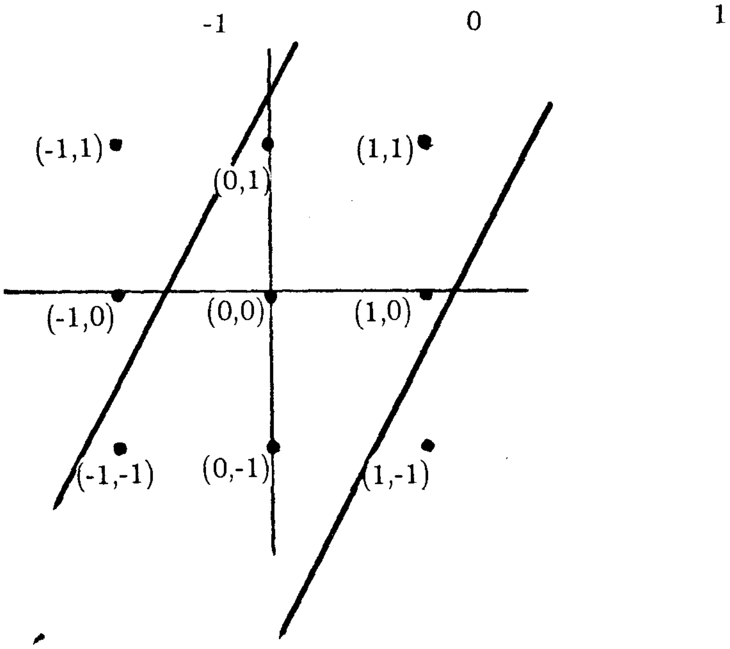

Fig. 10.1 (b). Three-state neurons: decision regions for neuron 2.

$$\mathbf{P}\left\{u_i^{(\alpha)} = l\right\} = \begin{cases} \dfrac{1}{2\kappa+1} & -\kappa \leq l \leq \kappa \\ 0 & \text{otherwise} \end{cases} .$$

The interconnection weights, w_{ij}, are again formed as the sum of Kronecker outer products of the fundamental memories, with

$$w_{ij} = \begin{cases} \sum_{\beta=1}^{m} u_i^{(\beta)} u_j^{(\beta)} & \text{if } j \neq i \\ 0 & \text{if } j = i \end{cases} .$$

Again, the zero-diagonal symmetry constraint effects a sensible improvement in system dynamics.

The thresholds t_l, $l = -\kappa,\ldots,\kappa - 1$ are specified by

$$t_l = \left(l + 1/2\right) \frac{\kappa(\kappa+1)(n-1)}{3} ; \qquad (9.3.1)$$

(for reasons that will soon be clear).

Let us consider, for simplicity, capacity under a fixed-point constraint. Are the fundamental memories fixed points? Define

$$\begin{aligned} X_i^{(\alpha)} &= \sum_{j=1}^{n} w_{ij} u_j^{(\alpha)} \\ &= u_i^{(\alpha)} \sum_{j=1}^{n} (u_j^{(\alpha)})^2 + \sum_{j \neq i} \sum_{\beta \neq \alpha} u_i^{(\beta)} u_j^{(\beta)} u_j^{(\alpha)} \\ &= Y_i^{(\alpha)} + Z_i^{(\alpha)} , \end{aligned}$$

where $Y_i^{(\alpha)}$ corresponds to the single sum, and $Z_i^{(\alpha)}$ corresponds to the second sum.

We have

$$\mathbf{E}\left\{Y_i^{(\alpha)} \mid u_i^{(\alpha)} = l\right\} = (n-1)l \ \mathbf{E}\left\{u_j^{(\alpha)2}\right\}$$

$$= (n-1)l \sum_{j=-\kappa}^{\kappa} \frac{j^2}{2\kappa+1}$$

$$= \frac{l\,\kappa(\kappa+1)(n-1)}{3} .$$

Also

$$\text{Var}\left\{Y_i^{(\alpha)} \mid u_i^{(\alpha)} = l\right\} = l^2 \sum_{j \neq i} \mathbf{E}\left\{u_j^{(\alpha)4}\right\}$$

$$= \frac{l^2\kappa(\kappa+1)(3\kappa^2+3\kappa-1)(n-1)}{15} .$$

Similarly,

$$\mathbf{E}\left\{Z_i^{(\alpha)} \mid u_i^{(\alpha)}\right\} = 0 ,$$

and

$$\text{Var}\left\{Z_i^{(\alpha)} \mid u_i^{(\alpha)}\right\} = \frac{\kappa^3(\kappa+1)^3(m-1)(n-1)}{27} .$$

The rationale for choosing the thresholds as in equation (9.3.1) is now clear; the mean values of $X_i^{(\alpha)}$ given the various values that $u_i^{(\alpha)}$ can take are spread at equal distances of $\kappa(\kappa+1)(n-1)/3$, while the variances are asymptotically compatible. Modifying the definition of the signal-to-noise ratio (SNR) for the present case of multiple thresholds, we get

$$SNR = \frac{\left(\mathrm{E}\left\{Y_i^{(\alpha)} \mid u_i^{(\alpha)} = l\right\} - \dfrac{(l+\frac{1}{2})\kappa(\kappa+1)(n-1)}{3}\right)^2}{\text{Var}\left\{Y_i^{(\alpha)} \mid u_i^{(\alpha)} = l\right\} + \text{Var}\left\{Z_i^{(\alpha)} \mid u_i^{(\alpha)} = l\right\}}$$

$$= \frac{3(n-1)}{\kappa(\kappa+1)(m-1)[1+o(1)]} .$$

Going along the lines of the previous proof, we have with the Poisson conjecture for the distribution of errors that the storage capacity, $C(n)$, is given by

$$C(n) = \frac{3n}{4\kappa(\kappa+1)\log n} .$$

Thus, the capacity for the outer product scheme remains the order of $\frac{n}{\log n}$. Some deterioration is present in the constant, as expected, however, because the thresholds effect a finer partition of the real line.

C. Spectral Approaches

As in chapter VII, we can utilise a spectral approach to improve on the performance of the outer product scheme. Again, let $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(m)} \in K^n$ be a randomly chosen m-set of fundamental memories. (For $m \leq n$ they will be linearly independent with probability approaching one as $n \rightarrow \infty$.) Let $\mathbf{U} = [\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(m)}]$ be the $n \times m$ matrix of fundamental memories. Let $\lambda > 0$ be fixed. Analogously with the pseudo-inverse scheme of strategy 1 in chapter VII, we define the matrix of weights $\mathbf{W} = [wij]$ by $\mathbf{W} = \mathbf{U}\,\mathbf{\Lambda}\,(\mathbf{U}^T\ \mathbf{U})^{-1}\mathbf{U}^T$, where $\mathbf{\Lambda} = \mathbf{dg}(\lambda, \ldots, \lambda) = \lambda\mathbf{I}$ is a constant diagonal matrix of eigenvalues. The eigenvalues of $\mathbf{W}$ are hence m-fold degenerate. (We could equally well have used any other spectral scheme, of course, but we will restrict ourselves to the pseudo-inverse strategy for simplicity.) It is now simple to verify that the fundamental memories are eigenvectors of $\mathbf{W}$ with positive eigenvalue λ. (This will be true modulo the assumption of linear independence of the m-set of fundamental memories; this, however, is true with probability approaching 1 as $n \rightarrow \infty$, as noted earlier.) Specifically,

$$(\mathbf{W}\,\mathbf{u}^{(\alpha)})_i = \sum_{j=1}^{n} w_{ij}\ u_j^{(\alpha)} = \lambda\ u_i^{(\alpha)} \quad i=1,...,n\ ,\ \ \alpha=1,...,m.$$

The thresholds t_l are specified by

$$t_l = (\ l + 1/2)\ \lambda \quad ,\ \ l = -\kappa, \ .\ .\ .\ , \kappa\text{-}1\ .$$

The fundamental memories are fixed points with probability approaching 1 as $n \rightarrow \infty$ for a choice $m \leq n$, as $(u_i^{(\alpha)} - 1/2)\ \lambda \leq \lambda\ u_i^{(\alpha)} < (u_i^{(\alpha)} + 1/2)\ \lambda$. Thus, with the spectral approach, we can again realise a storage capacity of the order of n fundamental memories. Notice, however, that this is a considerable improvement over the earlier instance where we stored binary n-tuples as fundamental memories. The capacity of n memories derived earlier, corresponded to the storage of n^2 *bits*. In the present example, the storage capacity is actually of the order of $[\log_2(2\kappa + 1)]\ n^2$ *bits*. The multiplicative factor of $\log_2 (2\kappa + 1)$ comes about because, now, each component of a memory can take on $(2\kappa + 1)$ values (instead of two, as before), and hence requires $\log_2 (2\kappa + 1)$ bits to specify it. The usage of multiple threshold hence gives us an improvement of a factor of $\log_2 (2\kappa + 1)$ in information storage.

Ultimate capacities are somewhat harder to specify for this case. Generalising arguments by Olafsson and Abu-Mostafa [3] which count the number of disjoint regions created in the pattern space by multiple thresholds, it appears that the ultimate storage capacity lies between $2\,(n + \kappa)$ and $2\,(2n + \kappa)$ memories. Again, there is an incremental multiplicative factor of $\log_2 (2\kappa + 1)$ in information storage in bits because of the multiple states attainable by each neuron.

4. DISTORTION INVARIANCE

A. Generalisation: Outer Product Algorithm

As we have seen thus far, neural associative nets can efficiently perform nearest neighbour searches (or error correction). Specifically, for the instance of a binary pattern space, neural networks can be constructed to map points in Hamming spheres surrounding the fundamental memories to the memories themselves, which act as fixed-points, or absorbing states of a Markov chain. The volume of the Hamming sphere, or the number of points in it, is a measure of the error-correction capability available to the network. Thus, random errors that occur in the specification of individual components of a stored memory can be effectively compensated for by an associative neural net. Such error occurrence may be considered to be at a *local* or *microscopic* level.

There are certain commonly occurring *macroscopic* errors, however, which affect a large number of memory components, and which cannot be compensated for by a neural network of the structure we have described thus far. Instances of such macroscopic distortions are translational shifts of pattern vectors, rotations, and scales of images, and in general, any (fixed) non-singular transformation acting on the pattern vectors. Such non-singular transformations create macroscopic changes in the memories as all components are altered, albeit in some invertible fashion. Thus, in terms of Hamming distances, such transformations create patterns which are far removed from the fundamental memories, and hence are not "recognised" by the neural network.

For known macroscopic distortions of patterns of this nature, an approach toward solving the problem would be to utilise a pre-processing stage which maps distorted patterns into Hamming balls surrounding the memories. This approach, however, has some inherent problems. First, the pre-processing stage itself could be extremely complex. Second, and perhaps more important, in the process of correcting for a deterministic (macroscopic) distortion, we lose part or all of the Hamming space originally reserved for random (microscopic) bit error.

An alternative approach which does not sacrifice error correction capability to achieve compensation for fixed distortions is suggested by a closer examination of the outer product algorithm of chapter VI. Assume $\mathbf{u}^{(\alpha)}$ is the initial state of the system. The potential $X_i{}^{(\alpha)}$ seen at each neuron can be rewritten as

$$X_i{}^{(\alpha)} = \sum_{\beta=1}^{m} \left(\sum_{j=1}^{n} u_j{}^{(\beta)} u_j{}^{(\alpha)} \right) u_i{}^{(\beta)} .$$

Thus, the potential at each neuron can be arrived at by taking individual inner-products (or correlations) of the input state with each of the fundamental memories in turn, and using these correlations as weights for corresponding memory conponents. Thus, a fully equivalent processing scenario is obtained by considering m individual filtering stages, with the outputs of the filters being combined to form the neural post-synaptic potentials. The results of the first section on correlators (specifically, chapter III) indicate that if we generalise the filter construct by adding a point rule after every filter, we might be able to achieve correction of prescribed distortions.

Fig. 9.2 illustrates the sort of structure we envisage. We have m channels corresponding to each memory. Each channel comprises two filtration stages with the parameters of each filter determined solely by the particular memory corresponding to the channel. The filtration stages are separated by a point rule which acts pointwise on the output of the first filter to produce the input for the second filter. (If the point rule is the identity, we end up with the original outer product formulation.) Thus we have two point rules in operation: an intermediate point rule **D** introduced to compensate for specified distortions, and a threshold point rule which, as before, compensates for random bit errors.

B. Translational Shift Invariance

Let $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(m)}$ be independently chosen fundamental memories as before. To allow of translational shifts, we assume that each fundamental memory is a vector comprised of n independently chosen ± 1's in sequence, and padded with 0's on either

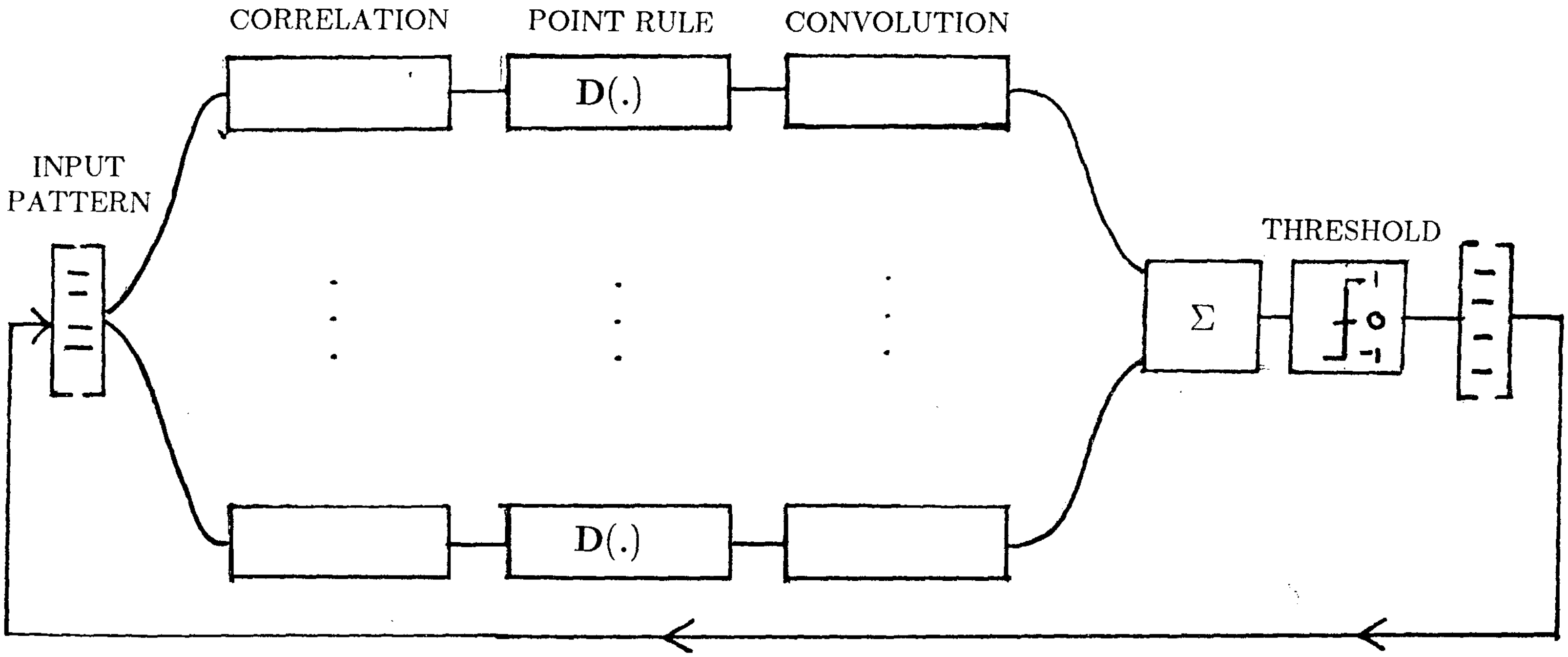

Fig. 10.2. System architecture for distortion invariant memory recall.

side to allow of up to a shift of n on either side. For simplicity of representation, we allow the non-zero components of $\mathbf{u}^{(\alpha)}$ to run from 0 to $n-1$. The padding with zeroes is an artifice which allows the creation of a "dead zone" in which shifts are allowed. Thus we can think of a shifted memory as the occurrence of a string of n ± 1's somewhere in an otherwise blank one-dimensional register of infinite extent.

Our approach is predicated upon the fact that the autocorrelation of a shifted pattern with itself results in an autocorrelation peak at the shifted position. If the autocorrelation peak is enhanced, then a succeeding convolution with the pattern itself will essentially result in the specified pattern dominating a bunch of weak cross-correlations. We elaborate on this theme, using an approach described in [4]. For each channel we choose a filter impulse response $\mathbf{u}^{(\beta)^-}$, with components $u_j^{(\beta)^-} = u_{-j}^{(\beta)}$ for the first filtration stage (a correlation), and a filter impulse response $\mathbf{u}^{(\beta)}$ for the second filtration stage (a convolution) and interpose a square-law point rule between the two filtration stages. The square-law point rule acts pointwise on each point of the correlation resulting from the first filter, and produces a vector with support over $2n - 1$ points as input to the convolutional stage to follow. The square-law point rule acts as a peak enhancer, and the convolutional stage that follows essentially picks out the particular (shifted) memory that was presented to the system. Clearly, other schemes can be chosen which function as well. We consider just the above algorithm as an archetypal example illustrating the possibilities in this area.

The algorithm can easily be seen to be insensitive to translational shifts. Shifted inputs just cause the correlations in each channel to shift by equivalent amounts. As the square-law rule acts pointwise on the vector components, the amount of shift is unaltered, so that the vector after thresholding is just shifted by the amount of the input shift. It only remains to be shown that the addition of the square-law point rule does not significantly increase the "noise" content, which in turn would reduce storage capacity.

Let fundamental memory $\mathbf{u}^{(\alpha)}$ be the initial state (probe) of the system. The potential seen by the i-th neuron can be written as

$$X_i^{(\alpha)} = \sum_{\beta-1}^{m} \sum_{j=-(n-1)}^{n-1} c_j^{(\alpha,\beta)^2} u_{i-j}^{(\beta)} ,$$

where

$$c_j^{(\alpha,\beta)} = \sum_{k=0}^{n-1} u_k^{(\alpha)} u_{j+k}^{(\beta)} = \begin{cases} \sum\limits_{k=0}^{n-1-j} u_k^{(\alpha)} u_{j+k}^{(\beta)} & \text{if } j \geq 0 \\ \sum\limits_{k=-j}^{n-1} u_k^{(\alpha)} u_{j+k}^{(\beta)} & \text{if } j < 0 \end{cases} .$$

Expanding terms, we obtain

$$X_i^{(\alpha)} = n^2 u_i^{(\alpha)} + Y_i^{(\alpha)} + Z_i^{(\alpha)} ,$$

where

$$Y_i^{(\alpha)} = 2 \sum_{j=1}^{n-1} \sum_{k=0}^{n-1-j} \sum_{l=0}^{n-1-j} u_k^{(\alpha)} u_l^{(\alpha)} u_{j+k}^{(\alpha)} u_{j+l}^{(\alpha)} u_{i-j}^{(\alpha)} ,$$

and

$$Z_i^{(\alpha)} = \sum_{\beta \neq \alpha} \left[\sum_{j=0}^{n-1} \sum_{k=0}^{n-1-j} \sum_{l=0}^{n-1-j} u_k^{(\alpha)} u_l^{(\alpha)} u_{j+k}^{(\beta)} u_{j+l}^{(\beta)} u_{i-j}^{(\beta)} \right.$$

$$\left. + \sum_{j=-(n-1)}^{-1} \sum_{k=-j}^{n-1} \sum_{l=-j}^{n-1} u_k^{(\alpha)} u_l^{(\alpha)} u_{j+k}^{(\beta)} u_{j+l}^{(\beta)} u_{i-j}^{(\beta)} \right] .$$

We have

$$\left| E(X_i^{(\alpha)} \mid u_i^{(\alpha)}) \right| = n^2 ,$$

as $Y_i^{(\alpha)}$ and $Z_i^{(\alpha)}$ are zero-mean. To obtain the signal-to-noise ratio, we now compute the variance of $Y_i^{(\alpha)}$ and $Z_i^{(\alpha)}$. We can write $Zi^{(\alpha)^2}$ as a sum over 8 variables $(\beta_1, \beta_2, j_1, j_2, k_1, k_2, l_1, l_2)$ of a product of 10 terms, each ±1. Hence, Var $Z_i^{(\alpha)}$ is an 8-sum of the expectation of the product of ten terms. A careful examination of the product yields that the only non-zero contribution to the 8-sum results under the following circumstances:

$$\beta_1 = \beta_2\,,\ \ j_1 = j_2\,,\ \ l_1 = l_2\,,\ \text{ and } k_1 = k_2;$$

$$\beta_1 = \beta_2\,,\ \ j_1 = j_2\,,\ \ l_1 = k_2\,,\ \text{ and } l_2 = k_1\,;$$

$$\beta_1 = \beta_2\,,\ \ j_1 = j_2\,,\ \ l_1 = k_1\,,\ \text{ and } l_2 = k_2\,.$$

For each of the above cases the expectation of the 10-product is one. Hence

$$\text{Var }(Z_i^{(\alpha)}) = 3\left[(m-1)\,2\sum_{j=0}^{n-1}(n-1-j)^2\right]$$

$$= 6\,(m-1)\,\frac{(n-1)\,n\,(2n-1)}{6}$$

$$\sim 2mn^3 \qquad \text{as } n \to \infty\,.$$

Similarly

$$\text{Var }(Y_i^{(\alpha)}) = O(n^3).$$

Choosing m so that $m \to \infty$, as $n \to \infty$, we have the signal-to noise-ratio given by

$$SNR = \frac{[\,\mathrm{E}\,(\,X_i^{(\alpha)} \mid u_i^{(\alpha)}\,)\,]^2}{\text{Var }(\,Z_i^{(\alpha)}\,) + \text{Var }(\,Y_i^{(\alpha)}\,)}$$

$$\sim \frac{n^4}{2mn^3(1 + o(1))}$$

$$\sim \frac{n}{2m}\,.$$

The signal-to-noise (power) ratio for the original outer product algorithm was $\frac{n}{m}$, so that this result augurs a loss of about a factor of 1/2 in capacity. This loss in the storage capacity for fundamental memories is, however, offset by achieving full invariance to translational shifts of the memories, while retaining a Hamming sphere of correction of (random) bit errors around each memory. Experimental results are available in Ref. [4].

(An argument could be made that we actually store $(2n-1)m$ memories, by viewing each shifted version of a memory as a separate entity. This would, however, not be strictly accurate from our definition of capacity. In the definition we require that it be possible to store almost all choices of memories within capacity. In the present case, each shifted version of a memory is clearly related to the memory; it would not be possible to store an *arbitrary* choice of $(2n-1)m$ memories within the present schema.)

Thus, the addition of a second point rule (in addition to the threshold decision rule) results in considerably more associative processing power. A hidden price paid here, however, is that the system loses *robustness*. The original outer product scheme is very resilient from system damage [2], as evinced in section 2 in this chapter; binarising the interconnection weights does not significantly affect system performance. The shift invariant system, however, is much more susceptible if system parameters are incorrectly specified. For instance, losing a single correlator would result in that particular memory being lost.

C. Rotation and Shift Invariance

The same architecture could be applied to correct other forms of input distortion with suitable choices of system parameters. Consider a pattern space of bipolar images $f(x,y) \in \mathbb{B}$. Let images $f^{(1)}, \ldots, f^{(m)}$ represent m fundamental memories. We can utilise the results of chapter III now to obtain an associative memory which corrects for rotations, and shifts as well. The correlator impulse responses in this case are chosen to be rotation insensitive filters $h^{(\alpha)}(x,y)$, (as in theorem (3.2.1)) with the filter in each channel so chosen that it produces a correlation peak if the corresponding memory is the input. The remaining system parameters are kept as before. We can argue, as before, that the resultant system is invariant now to both rotations and translational shifts of image. The potential now seen by neuron indexed by coordinates (x, y) when $f^{(\alpha)}$ is the initial state of the system is

$$X_{(x,y)}^{(\alpha)} = \sum_{\beta=1}^{m} \int \int \ | C^{(\alpha,\beta)}(u, v) |^2 f^{(\beta)}(x-u, y-v) du \ dv \ ,$$

where

$$C^{(\alpha,\beta)}(u, v) = \int \int f^{(\alpha)}(s, t) \overline{h^{(\beta)}}(s+u, t+v) \ ds \ dt \ .$$

As before, we can show that we are left with a signal term corresponding to the correlation peak $C^{(\alpha,\alpha)}(0,0)$, and a noise term. If the rotation insensitive filters are suitably chosen so that good correlation peaks develop, then the signal term will dominate the noise term, and we have the requisite associative behaviour. Note, however, that there is some loss in performance compared to the pure shift invariance case. This corresponds to the discussion in chapter III on the loss in performance that accrues when rotation insensitive filters are compared to matched filters.

5. HIGHER ORDER NETWORKS

A. Polynomial Maps

In the last two sections we have seen that improving the elemental computational capabilities of the system can result in significant improvements in overall computational capability. Thus, by utilising multiple thresholds instead of single thresholds, we improved the information capacity of the system whereas by introducing an additional layer of square-law point rules we could obtain distortion invariant, associative recognition. An alternative approach to increasing capacity is to improve inter-neural "communication." Specifically, while retaining a simple threshold point rule, we replace the linear map **W** by a more complex (non-linear) map.

In this section, we illustrate the gains that may be had by this approach by utilising polynomial maps, **W**, to disseminate information about neural states throughout the network. Polynomial maps are, in a sense, the natural generalisation of the linear interconnections that we have been considering so far.

We again consider an n-neuron system with the instantaneous system state being a binary n-tuple $\mathbf{u} \in \mathbb{B}^n$. We define polynomial map of degree d on the pattern space $\mathbf{W} : \mathbb{B}^n \rightarrow \mathbb{R}^n$ by

$$(\mathbf{W}\,\mathbf{u})_{i_0} = \sum_{k=1}^{d} \sum_{1 \le i_1 < \ldots < i_n \le n} w_{i_0 i_1 \cdots i_k} \, u_{i_1} \ldots u_{i_k} \quad , \; i_0 = 1, \ldots, n \; . \qquad (9.5.1)$$

Where the $w_{i_0 i_1 \cdots i_k}$ are real coefficients (weights) for the polynomial. The potential seen by neuron i_0 is now given by equation (9.5.1). (Note that for $d=1$, we have the case of a linear mapping.) State changes at each neuron are determined by the same threshold rule as before, with neuron i_0 taking on state +1 if the potential (9.5.1) is non-negative, and taking on state -1 if the potential is negative. Operation could be either synchronous of asynchronous mode, as before. Again, with the nature of the decision rule being fixed, the demonstration of memory encoding and associative recall within the structure rests purely on the choice of the coefficients $w_{i_0 i_1 \cdots i_k}$ and the degree d of the polynomial.

B. Outer Products Again

Let us once again consider a generalisation of what is fast becoming an old friend–the outer product algorithm. Let $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(m)} \in \mathbb{B}^n$ be randomly specified fundamental memories, as before. Let d be the degree of the polynomial map. We form the coefficients $w_{i_0 i_1 \cdots i_k}$ of the polynomial as a generalisation of the outer product algorithm as follows:

$$w_{i_0 i_1 \cdots i_k} = \begin{cases} \displaystyle\sum_{\beta=1}^{m} u_{i_0}^{(\beta)} u_{i_1}^{(\beta)} \cdots u_{i_k}^{(\beta)} & \text{if } i_0, \ldots, i_k \text{ are all distinct} \\ 0 & \text{otherwise} \end{cases} \qquad (9.5.2)$$

To demonstrate the working of the algorithm, consider the storage of fixed points. Assume $\mathbf{u}^{(\alpha)}$ is the initial state of the system. Because of the symmetricity of the terms involved in equation (9.5.2) (all permutations of $i_0, i_1, \ldots, i_k$ yield the same coefficient value), we consider only terms without repetition in the sum (9.5.1). We have the potential $X_i^{(\alpha)}$ at the i-th neuron when $\mathbf{u}^{(\alpha)}$ is the initial state, given by

$$X_i^{(\alpha)} = \sum_{k=1}^{d} \sum_{\substack{1 \le i_1 < \ldots < i_k \le n \\ i_1, \ldots, i_k \ne i_0}} \sum_{\beta=1}^{m} u_{i_0}^{(\beta)} u_{i_1}^{(\beta)} \cdots u_{i_k}^{(\beta)} u_{i_1}^{(\alpha)} \cdots u_{i_k}^{(\alpha)}$$

$$= u_{i_0}^{(\alpha)} \sum_{k=1}^{d} \binom{n-1}{k} + Z_{i_0}^{(\alpha)} ,$$

where

$$Z_{i_0}^{(\alpha)} = \sum_{\beta \ne \alpha} \sum_{k=1}^{d} \sum_{\substack{1 \le i_2 < \ldots < i_k \le n \\ i_2, \ldots, i_k \ne i_1}} u_{i_0}^{(\beta)} \; u_{i_1}^{(\beta)} \; \ldots u_{i_k}^{(\beta)} \; u_{i_1}^{(\alpha)} \cdots u_{i_k}^{(\alpha)} . \qquad (9.5.3)$$

We again utilise a signal-to-noise criterion. We have

$$\left| \mathrm{E} \, (X_{i_0}^{(\alpha)} \;\mid\; u_{i_0}^{(\alpha)} \;) \right| = \sum_{k=1}^{d} \binom{n-1}{k} ,$$

as $Z_{i_0}^{(\alpha)}$ is zero mean. Furthermore, the variance of $Z_{i_0}^{(\alpha)}$ can be easily found by exploiting the independence of the terms $u_i^{(\alpha)}$. Specifically,

$$\mathrm{Var} \, (Z_{i_0}^{(\alpha)} = (m-1) \sum_{k=1}^{d} \binom{n-1}{k} .$$

The signal-to-noise (power) ratio is hence given by

$$SNR = \frac{\sum_{k=1}^{d} \binom{n-1}{k}}{(m-1)} . \qquad (9.5.4)$$

Consider the case of fixed polynomial degree d. Allowing $m \to \infty$ as $n \to \infty$, we get that asymptotically with n ,

$$SNR \sim \frac{n^d}{d\,!} m \ .$$

Recalling that for the unvarnished outer product scheme we had an $SNR \sim \frac{n}{m}$, we get, in analogy with results obtained in chapter VI (or using the Poisson conjecture for error distribution), that the storage capacity $C(n\,;d\,)$ of the generalised outer product scheme for a polynomial of degree d is given by

$$C(n\,;d\,) = \frac{n^d}{2\,(d+1)!\,\log n} \ .$$

For $d=1$ (the linear case) this reduces to the form $\frac{n}{4\log n}$ obtained earlier. For $d=2$ we get $\frac{n^2}{12\log n}$, already a considerable improvement in capacity. Optical implementations of quadratic associative nets with $d=2$ have been proposed by Psaltis and Park [5]. The interesting feature of the proposed implementation is that the quadratic form (9.5.1) with $d=2$ is formed using solely linear maps and square-law point rules, and in fact this generalises to arbitrary d. This can be seen specifically from equation (9.5.3) . The innermost sum can be replaced by the term

$$\left[\sum_{\substack{1 \le i_1 < .. < i_k \le n \\ i_1, \ldots, i_k \neq i_0}} u_{i_0}^{(\beta)}\,,\ u_{i_1}^{(\beta)} \ldots u_{i_k}^{(\beta)}\ \ u_{i_1}^{(\alpha)} \ldots u_{i_k}^{(\alpha)} \right]$$

$$= u_{i_0}^{(\beta)} \ \left[\left(\sum_{j \neq i_0} u_j^{(\beta)} u_j^{(\alpha)} \right)^k - \text{terms with index duplication} \ \right] .$$

It is easily verified that for fixed d, the terms with index duplication do not contribute significantly to the sum. Hence, the outer product polynomial map can be implemented by a linear map in conjunction with k-th law point devices, $k = 1,\ldots,d$.

This may have considerable practical import. As has been long known, the dominant problem in VLSI systems is that of communication [6], and in a general polynomial map of the form (9.5.1), we require truly massive communication. Reducing the communication demands of the outer product polynomial map to that of a linear map is hence of some importance.

C. Generalised Spectral Approaches

The spectral algorithm also generalises to the polynomial case to yield a considerable increase in capacity. Let $\mathbf{U} = [\,\mathbf{u}^{(1)} \cdots \mathbf{u}^{(m)}]$ be an $n \times m$ matrix formed by the m column vectors, $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(m)}$, corresponding to the m fundamental memories. Now corresponding to each $\mathbf{u} \in \mathbb{B}^n$, we form a column vector $\mathbf{u}^*$ with $\sum_{k=1}^{d} \binom{n}{k}$ components according to the following prescription:

$$\mathbf{u}^* = \begin{bmatrix} u_1 \\ \vdots \\ u_n \\ u_1\, u_2 \\ u_{n-1}\, u_n \\ \vdots \\ u_1 \cdots u_d \\ \vdots \\ u_{n-d+1} \cdots u_n \end{bmatrix} . \tag{9.5.5}$$

Set

$$N_d = \sum_{k=1}^{d} \binom{n}{k} . \tag{9.5.6}$$

Form the $N_d \times m$ matrix

$$\mathbf{U}^* = [\,\mathbf{u}^{(1)^*} \cdots \mathbf{u}^{(m)^*}] .$$

Let $\mathbf{W}$ denote the $n \times N_d$ matrix of coefficients for the polynomial in (9.5.1) arranged lexicographically; i.e.,

$$\mathbf{W} = \begin{bmatrix} w_{11} & \cdots & w_{112} & \cdots & w_{1,n-d+1,\ldots,n} \\ w_{21} & \cdots & w_{212} & \cdots & w_{2,n-d+1,\ldots,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ w_{n1} & \cdots & w_{n12} & \cdots & w_{n,n-d+1,\ldots,n} \end{bmatrix} . \qquad (9.5.7)$$

The right hand side of equation (9.5.1) can now be identified with the matrix product $\mathbf{WU}^*$.

Let $\lambda^{(1)}\ldots,\lambda^{(m)}$ be m positive real number. Let $\mathbf{\Lambda} = \mathbf{dg}\ [\lambda^{(1)}, \ldots, \lambda^{(m)}]$. For each of the fundamental memories to be a fixed point it suffices that

$$\mathbf{W}\ \mathbf{u}^{\alpha^*} = \lambda^{(\alpha)}\mathbf{u}^{(\alpha)} \quad , \alpha{=}1,\ldots,\ m\ ;$$

The fundamental memories $\mathbf{u}^{(\alpha)}$ are hence *generalised eigenvectors* with *generalised eigenvalues* $\lambda^{(\alpha)}$ of the polynomial map $\mathbf{W}$. We hence set out to solve for $\mathbf{W}$ such that

$$\mathbf{W}\,\mathbf{U}^* = \mathbf{U}\,\mathbf{\Lambda}\ . \qquad (9.5.8)$$

A solution of (9.5.8) yielding a generalisation of the pseudo-inverse strategy of chapter VII is

$$\mathbf{W} = \mathbf{U}\,\mathbf{\Lambda}\,(\mathbf{U}^{*T}\ \mathbf{U}^*)^{-1}\,\mathbf{U}^{*T}\ . \qquad (9.5.9)$$

Of course, other solutions are possible, as pointed out in chapter VII, but (9.5.9) is often the easiest to compute. Note that the unequivocal evaluation of equation (9.5.8) depends on the $N_d \times m$ matrix $\mathbf{U}^*$ being full rank. This will be guaranteed as $n \to \infty$ by Komlös theorem for $m \leq N_d$.

Therefore, *the storage capacity of the generalised spectral scheme is of the order of* $N_d = \sum_{k=1}^{d} \binom{n}{k}$ *memories. For* d *fixed, this evaluates to the order of* $\frac{n^d}{d\,!}$ *memories.*

There is again an improvement by a factor $\log n$ over the outer product approach.

D. Maximal Capacity of Polynomial Maps

The map $\mathbf{W}$ defined by equation (9.5.1) defines a polynomial map of degree d on the space of binary n-tuples $\mathbb{B}^n$. For N_d defined by equation (9.5.6), define $\mathbb{B}^{*n} \subseteq \mathbb{B}^{N_d}$ to be the subset of binary N_d-tuples comprised of binary N_d-tuples $\mathbf{u}^*$ of the form (9.5.5) derived from binary n-tuples $\mathbf{u}$. Clearly, $| \mathbb{B}^{*n} | = 2^n$ as equation (9.5.5) describes a 1–1 map. From the discussion just concluded, it is clear that we could equivalently regard $\mathbf{W}$ as a *linear mapping* on the space $\mathbb{B}^{*n}$ whose matrix of coefficients is the $n \times N_d$ matrix of equation (9.5.7) .

The results of chapter VIII now hold by treating $\mathbf{W}$ as a linear map on an N_d-dimensional space. Specifically, we have the following assertion.

Theorem 9.5.1. The epsilon capacity of a polynomial neural network of degree d is $\frac{2\,N_d}{1-2\epsilon}$, where $0 \leq \epsilon < 1/2$ is the prescribed error tolerance.

Note that for fixed d, the epsilon capacity is $\frac{2\,n^d}{(1-2\,\epsilon)\,d\,!}$. If the degree d is allowed to grow with n, we could potentially store an exponential number of memories.

References

[1] M. Minsky and S. Papert, *Perceptrons: An Introduction to Computational Geometry*. Cambridge, Massachusetts: MIT Press, 1969.

[2] J.J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Natl. Acad. Sci. USA*, vol. 79, pp. 2554–2558, 1982.

[3] S. Olafsson and Y.S. Abu-Mostafa, "The capacity of multi-level threshold functions," in preparation.

[4] D. Psaltis, J. Hong, and S.S. Venkatesh, "Shift invariance in optical associative memories," *SPIE Proc. Conf. Optical Computing* pp. 635–627, Los Angeles, California, Jan. 1986.

[5] D.Psaltis and C.H. Park "Nonlinear descriminant functions and associative memories," *Conf. on Neural Networks for Computing*, Snowbird, Utah, April 1986.

[6] I.E. Sutherland and C.A. Mead, "Microelectronics and computer science," *Scientific American*, vol. 237, pp. 210–228, 1977.

CHAPTER X

CONCLUSIONS

Generalising linear discriminant functions to include point rules results in systems of moderate complexity which have the potential of expanding considerably the repertoire of problems that can be tackled by linear classification machines. The point rules themselves need not be very complex, and even very simple point rules can be efficacious for specific problems as we illustrated in our considerations of the threshold, and square-law point rules. The general utility of the approach to various classes of pattern recognition problems is yet under investigation. It appears, however, that (linearly non-separable) classification problems of specific structure, such as specified distortions of reference patterns, may well be amenable to solution by usage of general point rules in conjunction with appropriate linear maps. The choice of linear map and point rule, of course, has to be tailored to the specific problem at hand. For such a general classification problem, it may be appropriate to think of the point rule as generating a multitude of partial (generalised) decisions, with the final classification (linear discriminant function) stage producing a decision based upon these partial decisions. If point rules, and linear maps, can be so chosen (for a particular, cooperative problem structure) that each partial decision is independently biased towards correct classification, then the overall decision is liable to be accurate. This was the case, as we saw, for multiple channel machines using rotation insensitive filters in each channel, as well as for the binary filters.

The usage of point rules in reduced dimensionality situations cannot, however, increase the essential capacity of the classification system in terms of increasing the *number* of states of nature that the system can identify. As we saw, however, in situations where the problem structure is cooperative, point rules can be used fruitfully to separate states of nature which are not linearly separable, while working within the capacity of the system. In such cases then, the major advantage conferred by utilising generalised linear discriminant functions is in increased *flexibility* in classification, while leaving system capacity unchanged.

The usage of cascades of these structures and feedback results in considerable increases in computational capability. Non-trivial problems in association, for instance, can be readily handled by relatively simple constructs. However, as we saw in our analysis of neural networks, systems incorporating simple linear maps and threshold point rules are intrinsically limited in capacity. In the hunt for systems of wider applicability, and greater computational capability, it is essential to quantify the exact tradeoffs between communication and elemental, or raw computing, power. In this regard, the extensions of neural network structure that we considered, yield encouraging results. Several open questions remain, however.

While polynomial maps for communicating between processing nodes in such a feedback system yield reasonable capacities, it may well pay to investigate other general forms of communication between nodes (neurons). Replacing single and multiple threshold point rules by more general Boolean point rules is another approach to consider. In all such generalisations, of course, difficulties in practical implementation and system cost are important factors. Other questions we have not delved into in depth include the universality of applicability, and programmability of these structures.

While much of the effort in this area has gone into the characterisation of the fixed point structure of these networks, much less is known about the dynamics of the network in even very simple structures. Charting the dynamics of state flow can be of importance in utilising these networks for more general computation. Another problem of interest (of which not much is known) is the distribution of extraneous fixed points, and their basins of attraction. While some early work is being done in this area, much needs to be done. Gains in the analyses of these problems can be put to use, for instance, in sculpting basins of attraction, or establishing desired paths in the state space.

Stochastic systems such as the so-called Boltzman machines, and algorithms such as simulated annealing may be of use in particular computational problems in these networks. Work remains to be done in the role of the stochastic structure in imposing order on chaotic systems.

Allied problems of interest include the characterisation of locally interconnected systems, such as cellular automata. Multi-layer machines utilising many so called "hidden units" may be very effective for particular computations. These and other similar problems can be mapped onto large systems with dense local interconnectivity, and sparse global interconnectivity. The analysis of these systems promises to lead to characterisations of a wide class of problems which fall within the province of such networks.

An issue we have not touched upon at all is that of *learning*, wherein the system parameters change in time, converging towards more optimal values. The characterisation of learning rules, and adaptation (both supervised and unsupervised) remains far from complete.

In fine, of particular interest is the role of "analog" neurons which take on a continuum of values, with some suitable non-discrete point rule. While early efforts indicate that these can be used fruitfully for computation, very little is known of their capabilities. They exhibit behaviour considerably different from their discrete counterparts, particularly in the appearance of chaotic behaviour in systems incorporating such analog structures. The analysis of these structures could well prove to be critical in evaluating the ultimate use of such densely interconnected networks for computation.