Contextual Pattern Recognition with Applications to Biomedical Image Identification

Thesis by

Xubo Song

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

California Institute of Technology

Pasadena, California

1999

(Submitted November 23, 1998)

Acknowledgements

First of all, I am grateful to my advisor Yaser Abu-Mostafa. His insights, guidance, encouragement and support have been the driving force for me through the years. I deeply appreciate his mentorship and tolerance; and the enlightenment that his charisma brought.

It is with affection and appreciation that I mention Amir, Joe, Malik, Zander and Zehra. I thank them not only for their always immediate help on technical issues, but also for their friendship and companionship, and the warm family feeling they brought to the group. I also would like to thank Dr. Harvey Kasdan at IRIS, Inc. The collaboration with him has been very pleasant and productive.

Some other people also have made my life at Caltech rich and rewarding. I thank Shengfa, without whom I wouldn't be here, John, who gave me much valuable advice, and Pat, simply for being my friend. I thank them for the encouragement and emotional support they provided me with whenever I needed it.

And most of all, my thanks to my parents and family, for their unfailing love, support and faith in me.

Abstract

This thesis studies two rather distinct topics: one is the incorporation of contextual information in pattern recognition, with applications to biomedical image identification; and the other is the theoretical modeling of learning and generalization in the regime of machine learning.

In Part I of the thesis, we propose techniques to incorporate contextual information into object classification. In the real world there are cases where the identity of an object is ambiguous due to the noise in the measurements based on which the classification should be made. It is helpful to reduce the ambiguity by utilizing extra information referred to as context, which in our case is the identities of the accompanying objects. We investigate the incorporation of both full and partial context. Their error probabilities, in terms of both set-by-set error and element-by-element error, are established and compared to context-free approach. The computational cost is studied in detail for full context, partial context and context-free cases. The techniques are applied to toy problems as well as real world problems such as white blood cell image classification and microscopic urinalysis. It is demonstrated that superior classification performance is achieved by using context. In our particular application, it reduces overall classification error, as well as false positive and false negative diagnosis rates.

In Part II of the thesis, we propose a novel theoretical framework, called the Bin Model, for learning and generalization. Using the Bin Model, a closed form is derived for generalization that estimates the out-of-sample performance in terms of the in-sample performance. We address the problems of overfitting, and characterize conditions under which it does not appear. The effect of noise on generalization is studied, and the generalization of the Bin Model framework from classification problems to regression problems is discussed.

Contents

List of Figures

List of Tables

Part I

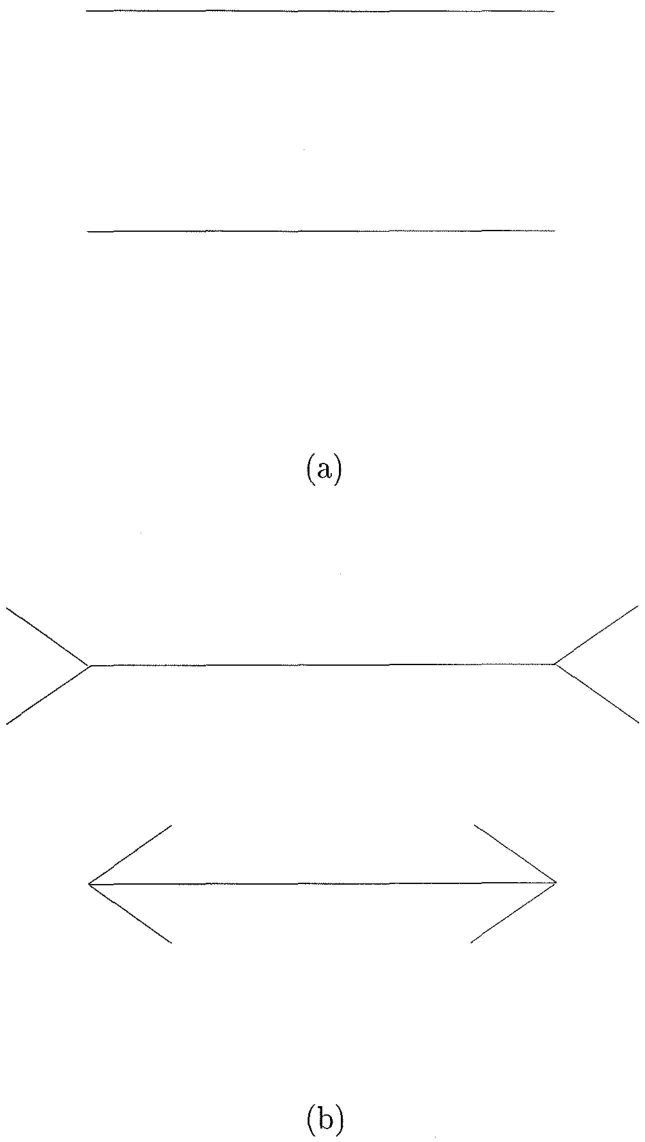
Contextual Pattern Recognition with Applications to Biomedical Image Identification

Chapter 1 Context

The world we live in is interconnected. Things exist in conjunction with one another, instead of in isolation. This interconnected nature of relationship gives rise to *context.* Context describes the dependency among entities, and has been an important notion and tool utilized to better perceive, understand or interpret the world. In this thesis, we focus on the utilization of context in the case of object perception and recognition. The effect of context in this case is that an entity may be perceived differently when viewed in association with other entities, than when viewed in isolation. The information conveyed by the accompanying entities is what we call *contextual information.*

Context occurs at various levels, including perceptual/cognitive, and statistical. For example, consider the two horizontal lines in Figure 1.1 (a). They appear to be of the same length. Now add some context in the forms of arrows at the ends so as to obtain Figure 1.1(b). In Figure 1.1 (b) it appears that the lower line is shorter than the upper one. This is the well-known Muller-Lyer illusion. Another example is Zollner Illusion in Figure 1.3. The diagonal black lines are actually parallel to one another. However, when the short horizontal or vertical lines are added to the picture, the diagonal lines appear to have changed their orientation and they no longer appear parallel. These are examples of the effect of context at a perceptual level. Context can help us perceive or recognize things that are not really there. Another form of context is statistical context, which will be the focus of interest for this thesis. It can be thought of as prior knowledge of the likelihood of the occurrence of any combination of events. For example, look at the two men in Figure 1.2. At first glance they appear to be President Bill Clinton and Vice-President Al Gore, but it is really Clinton and Clinton. The faces on both men are identical, only the hairline and clothing is different. This misclassification is due to the context. When a person's visual system looks at an image it is not so much concerned with the specific details as determining the overall meaning of the image. In this particular case, the viewer is most likely used to such a pose with the Vice-President in the rear. Al Gore's familiar suit and hairline also contributed to the effect. When the viewer doesn't pay attention to the details in the above examples, context can be misleading sometimes. But more often it can help reduce ambiguity and lead to correct classification that would not be achieved without. This is illustrated in Figure 1.4. The words "festival" and "graphics" are written in a noisy way. The "v" in "festival" and the "r" in "graphics" actually have exactly the same appearance. Yet we can still identify them correctly according to the context. In other words, the character is ambiguous and the ambiguity is resolved by context.

Statistical context embodies the dependence or correlation among objects in a statistical sense. It can be knowledge-driven or data-driven. In pattern recognition, the main source of information to identify an object is the set of measurements associated with it, which we call features. However, ambiguity arises when the features are incomplete (missing information) or are contaminated by noise, or simply when class-conditional feature distributions overlap, which leads to misclassification. The use of information-bearing context can help to reduce ambiguity. Human experts apply contextual information in their decision making process. And it makes sense to design techniques and algorithms to make computers mimic human behavior, in the sense that they can aggregate and utilize a more complete set of information in their decision making the way human experts do. The application of context has been investigated in many fields. In remote sensing image classification where each pixel is part of ground cover, certain classes of ground cover are likely to occur in the context of others. For instance, a pixel is more likely to be glacier if it is in a mountainous area, and extremely unlikely if surrounded by residential pixels. In text analysis, one can expect to find certain letters occurring regularly in particular arrangements with other letters (qu, ee, est, tion, etc.). Where ambiguity occurs, we can look at the neighboring letters to gain more information. The same principle applies to speech recognition, where certain arrangements of some phonemes are more likely to occur than others. For example, consonants such as "b", "g", "l" are always followed by



(a)

(b)

Figure 1.1: Muller-Lyer Illusion

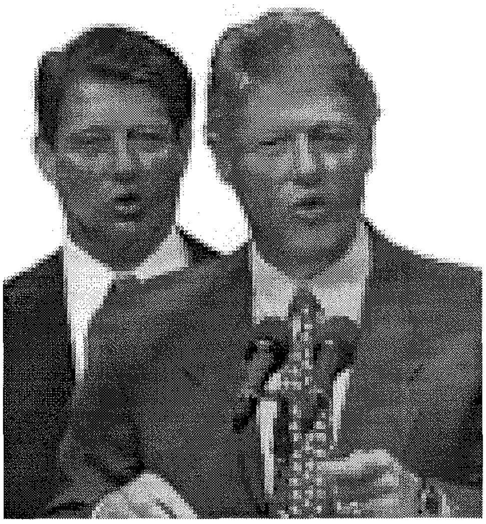


Figure 1.2: An example of the effect of context in face recognition.

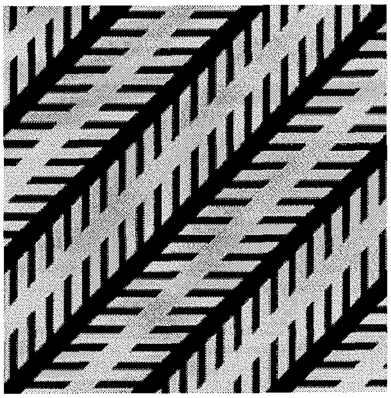


Figure 1.3: Zollner Illusion.

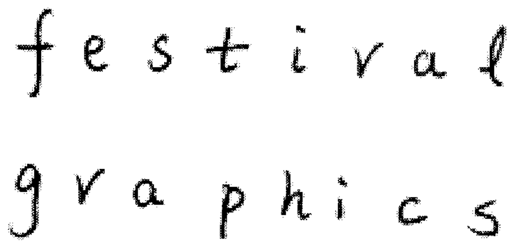


Figure 1.4: An example of the effect of context in character recognition.

vowels. "k" almost never immediately proceeds "d", but it sometimes does "s". In white blood cell identification in the medical field, the composition (the percentage of all blood cell types) of a blood specimen has certain patterns. Abnormal cells are much more likely to appear in groups than in isolation. Specifically, in a sample of several hundred cells, it is more likely to find either no abnormal cells or many abnormal cells than it is to find just a few. Another example of the use of context in medicine is in urinalysis. Bacteria always occur with white blood cells, which is intuitive to us since the white blood cells are there to fight the inflammation often caused by bacteria.

The usefulness of context lies in the fact that it abandons one of the most common assumptions made in the study of machine learning – that the examples are drawn *independently* from some joint input-output distribution, and takes advantage of contextual information that is otherwise ignored if we look at each individual object in isolation.

While the approaches to incorporate context into classification are often closely tied to specific applications, it is possible to see the general frameworks. The approaches broadly fall into three categories. The first is relaxation technique, which has been intensively used for scene labeling/analysis. Relaxation is an iterative technique. The probabilities of neighboring pixels are used to iteratively update the probabilities for a given pixel based on a relation between the pixel labels specified by compatibility coefficients to consider the joint distribution of one neighboring pixel at a time. In [Rosenfeld *et al.*, 1976], a technique is devised that introduces context by means of correlations of the labels between objects and their neighbors. [Zucker and Mohammed, 1978] have suggested schemes that depend instead upon the conditional probability of occurrence of a particular label on an object in view of the labeling on neighbors. Another approach was introduced by [Toussaint, 1978] under the theme of sequential compound decision theory. It attempts to decide the label for one pixel based on the observations of all other pixels in the image. Some of the approximation methods suggested by [Toussaint, 1978] can be found in [Tilton *et al.*, 1982], [Haralick and Joo, 1986], and [Khazenie and Crawford, 1986]. Another technique is to use a fuzzy knowledge representation, adopting the fuzzy-set based quantitative approach. [Binaghi *et al.*, 1997] applied this technique to remote sensing image analysis. While the representations and formulations of context are often different, the implementation of context is often through one of two means. One is the maximum likelihood framework, where the probability of the labeling of an object is maximized in its proper context, as is in [Rosenfeld *et al.*, 1976]. The other is the optimization of some utility function, such as the level of ambiguity, consistency, as in [Faugeras and Berthod, 1981] and [Illingworth and Kittler, 1987].

Some forms of context have distinct structures. For example, in speech recognition and text analysis, the context is *temporal.* The identity of a phoneme or a letter is contingent upon the identities of the ones that immediately proceed or follow it. Elaborate methods, such as Markov Models and State-Space Models, are devised to deal with this kind of context. In scene analysis and remote sensing image classification, the context is *spatial.* The identity of a pixel relies upon those of its neighboring pixels. These kinds of context are *local*, since the farther away the neighbors are, temporally or spatially, the less relevant they become, and thus less contextual information they provide. Most of the research on context has been focused on context with such locality. There are many pattern recognition problems that do not have such a property. This is the case for white blood cell image classification problem and microscopic urinalysis problem, where the task is to identify each element in a set of elements, and the spatial or temporal arrangement of these elements is irrelevant. Locality of context helps to reduce computation cost, because the number of context-bearing elements is small due to locality. Computational difficulty may arise for problems without locality. In white blood cell identification and urinalysis, the amount of context-bearing objects are on the order of hundreds, which poses computational challenges.

The first part of this thesis is organized as follows: Chapter 1 gives an introduction of the background and motivation for using context. The mathematical framework for the formulation of the incorporation of context using a Bayesian approach is provided in Chapter 2. Two forms of context – full context and partial context – are formulated, and their error probabilities and computational complexities are investigated. Chapter 3 and Chapter 4 demonstrate the use of context in the biomedical field. Chapter 3 describes an automated white blood cell image identification system, and the utilization of context in such a system. Chapter 4 focuses on the application of microscopic urinalysis.

Chapter 2 Mathematical Framework for Incorporating Context

2.1 Compound Bayesian Theory for Context

Let us consider a set of N objects T_i, $i = 1, ...N$. We associate each object T_i with a label c_i that is a member of a label set $\Omega = \{\omega_1, \ldots, \omega_D\}$. Each object T_i is characterized by a set of measurements $\mathbf{x}_i \in \mathbf{R}^P$, which we call a feature vector. We consider a situation where the label c_i of object T_i is unknown, but the feature vector $\mathbf{x}_i$ is known. We would like to infer the identity c_i through $\mathbf{x}_i$.

When each object is viewed in isolation, its identity is assumed to depend only on its own feature vector, and is independent of the features and the labels of all other objects. From Bayes' rule, we have

$$p(c_i|\mathbf{x}_i) = \frac{p(\mathbf{x}_i|c_i)p(c_i)}{p(\mathbf{x}_i)} \tag{2.1}$$

This is the element-by-element (context-free) *a posteriori* probability of one single object. (We use the word "element" and "object" interchangeably.) If we classify the objects in a context-free manner, the decision rule selects the class label $\hat{c}_i$ such that

$$\hat{c}_i = \underset{c_i}{\operatorname{argmax}}\, p(c_i|\mathbf{x}_i) \tag{2.2}$$

for $i = 1, \ldots, N$. We call this the context-free maximum likelihood (CFML) decision rule.

Since the accompanying objects $\{T_1, \ldots, T_{i-1}, T_{i+1}, \ldots, T_N\}$ may convey information pertinent to the labeling of T_i, it is logical that we consider not only the feature vector $\mathbf{x_i}$ of this object, but also the feature vector of other objects. Therefore, we are interested in $p(c_i|\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_N)$. A more general form of

$p(c_i|\mathbf{x}_1,\ldots,\mathbf{x}_{i-1},\mathbf{x}_i,\mathbf{x}_{i+1},\ldots,\mathbf{x}_N)$ is the simultaneous labeling of all objects given all the corresponding features $p(c_1,c_2,\ldots,c_N|\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N)$, which is what the calculation of $p(c_i|\mathbf{x}_1,\ldots,\mathbf{x}_{i-1},\mathbf{x}_i,\mathbf{x}_{i+1},\ldots,\mathbf{x}_N)$ will eventually break down to. Since this form incorporates contextual information, we refer to it as the set-by-set context-sensitive *a posteriori* probability.

It follows from Bayes' rule that

$$p(c_1,c_2,\ldots,c_N|\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N)=\frac{p(\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N|c_1,c_2,\ldots,c_N)p(c_1,c_2,\ldots,c_N)}{p(\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N)} \tag{2.3}$$

We make the conditional independence assumption that the feature distribution of an object is dependent only on its own class, not on the features or classes of other objects, *i.e.*,

$$p(\mathbf{x}_i|c_i;c_1,c_2,\ldots,c_J;\mathbf{x}_1,\ldots,\mathbf{x}_{i-1},\mathbf{x}_{i+1},\ldots,\mathbf{x}_K)=p(\mathbf{x}_i|c_i) \tag{2.4}$$

for any $J=0,1,\ldots,N$ and $J\neq i$, and $K=0,1,\ldots,N$. (where $\mathbf{x}_0$ and c_0 are null elements.) This assumption is reasonable since in many cases the feature distribution of a certain class does not change when this class exists in conjunction with some other classes.

Therefore, it follows that

$$p(\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N|c_1,c_2,\ldots,c_N)=p(\mathbf{x}_1|c_1)\ldots p(\mathbf{x}_N|c_N) \tag{2.5}$$

Then Equation 2.3 can be rewritten as

$$p(c_1,c_2,\ldots,c_N|\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N)=\frac{p(\mathbf{x}_1|c_1)\ldots p(\mathbf{x}_N|c_N)p(c_1,c_2,\ldots,c_N)}{p(\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N)}, \tag{2.6}$$

$$=\frac{p(c_1|\mathbf{x}_1)\ldots p(c_N|\mathbf{x}_N)p(\mathbf{x}_1)\ldots p(\mathbf{x}_N)p(c_1,c_2,\ldots,c_N)}{p(c_1)\ldots p(c_N)p(\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N)}$$

where $p(c_i|\mathbf{x}_i)$ is the context-free object-by-object Bayesian *a posteriori* probability, $p(c_i)$ is the *a priori* probability of the classes, $p(\mathbf{x}_i)$ is the marginal probability of the features, and $p(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$ is the joint distribution of all the feature vectors.

Since the features $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$ are given, $p(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$ and $p(\mathbf{x}_i)$ are constant,

$$p(c_1, \ldots, c_N|\mathbf{x}_1, \ldots, \mathbf{x}_N) \propto p(c_1|\mathbf{x}_1) \ldots p(c_N|\mathbf{x}_N) \frac{p(c_1, \ldots, c_N)}{p(c_1) \ldots p(c_N)} \tag{2.7}$$

$$= p(c_1|\mathbf{x}_1) \ldots p(c_N|\mathbf{x}_N) \rho(c_1, c_2, \ldots, c_N) \tag{2.8}$$

where

$$\rho(c_1, c_2, \ldots, c_N) \triangleq \frac{p(c_1, c_2, \ldots, c_N)}{p(c_1) \ldots p(c_N)} \tag{2.9}$$

The quantity $\rho(c_1, c_2, \ldots, c_N)$, which we call the *context ratio* and through which the context plays its role, captures the dependence among the objects. In the case where all the objects are independent, $p(c_1, c_2, \ldots, c_N) = p(c_1) \ldots p(c_N)$, then $\rho(c_1, c_2, \ldots, c_N) = 1$ — there will be no contextual information, and maximizing the context-sensitive *a posteriori* probability in (2.3) is equivalent to maximizing the context-free *a posteriori* probability in (2.2). In the dependent case, $\rho(c_1, c_2, \ldots, c_N) \neq 1$, and the context has an effect on the classifications. In general, the range of ρ is $0 \leq \rho < \infty$.

The context-sensitive decision rule chooses class labels $\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_N$ such that

$$(\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_N) = \underset{(c_1, c_2, \ldots, c_N)}{\text{argmax}} \; p(c_1, c_2, \ldots, c_N|\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N) \tag{2.10}$$

We call this the Context-Sensitive Maximum Likelihood (CSML) decision rule.

2.2 Optimality of Context-Sensitive Maximum Likelihood Decision Rule

2.2.1 CSML Achieves Minimum Set-by-Set Error Probability.

As stated earlier, our task is to classify all elements in a set according to their feature characterizations. Let us use the following notation: for a set of N elements $\{T_1, \ldots, T_N\}$, vector random variable $\underline{c} = (c_1, c_2, \ldots, c_N)$ is the true labeling of the set, $\hat{\underline{c}} = (\hat{c_1}, \hat{c_2}, \ldots, \hat{c_N})$ is the estimated labeling, and $\underline{x} = (\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_N})$ is the feature vector of the set of elements; for any element in the set, scalar random variable c is its true labeling, $\hat{c}$ is its estimated labeling, and $\mathbf{x}$ is its feature vector. The environment from which the examples are generated is characterized by distributions $p(\underline{c})$ and $p(\underline{x}|\underline{c})$, based on which we can derive $p(c)$ and $p(\mathbf{x}|c)$. [1]

We consider an environment in which each object has a unique identity. Errors occur when our inference of the class is different from the true class. We define two types of error: set-by-set error and element-by-element error. The *set-by-set* error probability is defined as

$$P_e^{set} = P(\hat{\underline{c}} \neq \underline{c})$$

$$= \int_{\underline{c}} \int_{\underline{x}} P(\hat{\underline{c}}(\underline{x}) \neq \underline{c}|\underline{x}, \underline{c}) p(\underline{x}, \underline{c}) d\underline{x} d\underline{c}$$

$$= \int_{\underline{c}} \int_{\underline{x}} [1 - \delta(\hat{\underline{c}}(\underline{x}) - \underline{c})] p(\underline{x}, \underline{c}) d\underline{x} d\underline{c}$$

where

$$\delta(\underline{s}) = \begin{cases} 1 & \text{if } \underline{s} = \underline{0} \\ 0 & \text{otherwise} \end{cases}$$

[1] The same conditional independence assumption as in equations 2.4 and 2.5 is made.

is the Kronecker delta.

The *element-by-element* error probability is defined as

$$P_e^{element} = P(\hat{c} \neq c)$$

$$= \int_{\underline{c}} \int_{\underline{x}} [\frac{1}{N} \sum_{n=1}^{N} p(\hat{c}_n(\underline{x}) \neq c_n | \underline{x}, \underline{c})] p(\underline{x}, \underline{c}) d\underline{x} d\underline{c}$$

$$= \int_{\underline{c}} \int_{\underline{x}} \frac{1}{N} \sum_{n=1}^{N} [1 - \delta(\hat{c}_n(\underline{x}) - c_n)] p(\underline{x}, \underline{c}) d\underline{x} d\underline{c}$$

The definition of set-by-set error is such that a set of elements is correctly classified if and only if every single element in the set is correctly classified. If one or more elements in the set are classified incorrectly, then this set, which can be viewed as a big augmented object, is classified incorrectly. The definition of element-by-element error is such that the error is counted on an element-by-element basis. If some elements in a set are classified incorrectly, the rest of the elements in this set are still counted as correct. Errors on different elements are counted separately and weighted equally within a given set for element-by-element error.

We are more concerned with set-by-set error than with element-by-element error. The difference between a set and an element corresponds to the difference between a word and a letter or a syllable, a blood specimen and a blood cell. A word, rather than a syllable, is the basic semantic unit. And it is a blood specimen as a collection of a bunch of cells that conveys information pertaining to the health of a patient.

The benefit of using context lies in the fact that it reduces set-by-set error probability.

Theorem CSML is the decision rule which achieves minimum set-by-set error probability.

Proof:

Since $c_i \in \Omega = \{\omega_1, \ldots, \omega_D\}$ for $i = 1, \ldots, N$, then $\underline{c} = (c_1, c_2, \ldots, c_N) \in \Omega^N = \{\theta_1, \ldots, \theta_{D^N}\}$.

For any given k, $k = 1, \ldots, D^N$, let $\mathcal{X}_k = \{\underline{x} | p(\underline{c} = \theta_k | \underline{x}) \geq p(\underline{c} = \theta_j | \underline{x}), \text{for all } j \neq k \text{ and } 1 \leq j \leq D^N\}$. In other words, $\mathcal{X}_k$ is the region in the input domain $\mathbf{R}^{P \times N}$ where an input $\underline{x}$ is given the label θ_k according to the CSML decision rule.

The set-by-set error probability by using CSML is

$$\begin{aligned}
P_{CSML}^{set}(error) &= 1 - P_{CSML}^{set}(correct) \\
&= 1 - \sum_{k=1}^{D^N} \int_{\underline{x} \in \mathcal{X}_k} p(\underline{x} | \underline{c}_k) p(\underline{c}_k) d\underline{x} \\
&= 1 - \sum_{k=1}^{D^N} \int_{\underline{x} \in \mathcal{X}_k} p(\underline{c}_k | \underline{x}) p(\underline{x}) d\underline{x}
\end{aligned}$$

For any $\underline{x} \in \mathcal{X}_k$, $p(\underline{c}_k | \underline{x})$ is maximized by the definition of CSML decision rule, therefore $\sum_{k=1}^{D^N} \int_{\underline{x} \in \mathcal{X}_k} p(\underline{c}_k | \underline{x}) p(\underline{x}) d\underline{x}$ is maximized, and $P_{CSML}^{set}(error)$ is minimized. **Q.E.D.**

This is essentially the optimality of Bayes Error Rate. Conditioned on the collective feature vector $\underline{x} = (x_1, \ldots, x_N)$, no other decision rule is better in terms of achieving a smaller set-by-set probability of error. The same logic implies that conditioned only on isolated feature x_i, the CFML decision rule that maximizes $p(c_i | x_i)$ for a given x_i achieves minimum element-by-element probability of error. However, it is possible that a decision rule conditioned on $\underline{x} = (x_1, \ldots, x_N)$ has smaller element-by-element probability of error than the one obtained by CFML conditioned only on isolated feature x_i, since more information is being utilized.

2.2.2 Information Theoretic Interpretation of Context

Define N_d as the number of objects in class d, and $\nu_d = \frac{N_d}{N}$ the frequency of class d. Clearly, $\sum_{d=1}^{D} N_d = N$ and $\sum_{d=1}^{D} \nu_d = 1$. Let $\mathbf{P} = (P_1, P_2, ..., P_D)$ be the class prior probability vector, and $\nu = (\nu_1, \nu_2, ..., \nu_D)$ the class frequency vector. Taking logarithms on both sides of Equation 2.7 gives:

$$ln\ p(c_1, ..., c_N|\mathbf{x}_1, ..., \mathbf{x}_N) = \sum_{i=1}^{N} ln\ p(c_i|x_i) + ln\ \frac{p(c_1, ..., c_N)}{p(c_1)...p(c_N)} + constant \tag{2.11}$$

$$= \sum_{i=1}^{N} ln\ p(c_i|x_i) + ln\ p(c_1, ..., c_N) - ln\ P_1^{N_1}...P_D^{N_D} + constant$$

$$= \sum_{i=1}^{N} ln\ p(c_i|x_i) + ln\ p(c_1, ..., c_N) - \sum_{d=1}^{D} N_d ln\ P_d + constant$$

$$= \sum_{i=1}^{N} ln\ p(c_i|x_i) + ln\ p(c_1, ..., c_N) - N\sum_{d=1}^{D} \nu_d ln\ P_d + constant$$

$$= \sum_{i=1}^{N} ln\ p(c_i|x_i) + ln\ p(c_1, ..., c_N) + N\sum_{d=1}^{D} \nu_d ln\ \frac{\nu_d}{P_d} - N\sum_{d=1}^{D} \nu_d ln\ \nu_d + constant$$

$$= \sum_{i=1}^{N} ln\ p(c_i|x_i) + ln\ p(c_1, ..., c_N) + N\mathcal{H}(\nu \parallel \mathbf{P}) + N\mathcal{H}(\nu) + constant \tag{2.12}$$

where $\mathcal{H}(\nu \parallel \mathbf{P}) = \sum_{d=1}^{D} \nu_d ln\ \frac{\nu_d}{P_d}$ is the relative entropy between ν and $\mathbf{P}$, and $\mathcal{H}(\nu) = -\sum_{d=1}^{D} \nu_d ln\ \nu_d$ is the entropy of the class frequency.

The above relation implies that maximizing $ln\ p(c_1, ..., c_N|\mathbf{x}_1, ..., \mathbf{x}_N)$ using context has the effect of trying to achieve a trade-off among several factors: the likelihood of each object given its feature (the first term), the likelihood of the set of objects appearing jointly (the second term), the distance of the class frequency profile of the set of objects from the prior distribution of the classes (the third term), and the entropy of the class frequency profile, the maximization of which implies that the least amount of further information is assumed about the frequency profile. The first term depends on the features, and the other three depend only on the classifications.

2.2.3 Information Gain by Using Context

It follows from the chain rule for conditional entropy that

$$H(c_1, c_2, \ldots, c_N | x_1, x_2, \ldots, x_N)$$

$$= H(c_1|x_1, x_2, \ldots, x_N) + H(c_2|c_1; x_1, x_2, \ldots, x_N) +$$

$$H(c_3|c_1, c_2; x_1, x_2, \ldots, x_N) + \ldots + H(c_N|c_1, c_2, \ldots, c_{N-1}; x_1, x_2, \ldots, x_N)$$

$$\leq H(c_1|x_1) + H(c_2|x_2) + \ldots + H(c_N|x_N)$$

Equality is achieved if and only if the following condition holds,

$$p(c_i|\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N; c_1, \ldots, c_{i-1}) = p(c_i|x_i)$$

for all $i = 1, \ldots, N$. This condition says that c_i is fully determined by x_i, and nothing else, which means that there is no context and therefore no information gain by considering context. When there is contextual information conveyed by other objects, this condition does not hold, in which case $H(c_1, c_2, \ldots, c_N|x_1, x_2, \ldots, x_N)$ is strictly less than $H(c_1|x_1) + H(c_2|x_2) + \ldots + H(c_N|x_N)$, and context provides information gain.

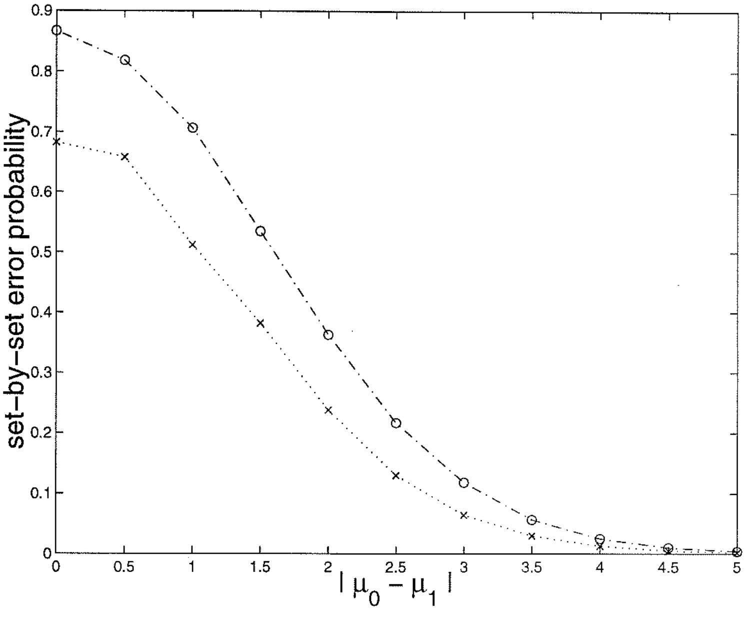
2.3 A Toy Example

We illustrate the effectiveness of using context by a toy example. There are $N = 4$ elements in each set. $\underline{c} = (c_1, c_2, c_3, c_4)$. Each element takes binary values from $\Omega = \{0, 1\}$. The distribution $p(\underline{c})$ is specified in Table 2.1. We choose $p(\underline{x}|\underline{c}) = \prod_{i=1}^{N} p(x_i|c_i)$, as assumed in Equation 2.5. The conditional feature distributions are $p(x|c = 0) = N(\mu_0, \sigma_0)$ and $p(x|c = 1) = N(\mu_1, \sigma_1)$. We choose $\sigma_0^2 = 1.0$ and $\sigma_1^2 = 0.36$. From $p(\underline{c})$ we know that $p(c = 0) = 0.525$ and $p(c = 1) = 0.475$. The distance between μ_0 and μ_1 determines separability of the classes. When $|\mu_0 - \mu_1| = 0$, the two classes are completely overlapping, and there is maximum amount of ambiguity.

Table 2.1: p($\underline{c}$) for the Toy Example

$\underline{c}$	p($\underline{c}$)
0 0 0 0	0.30
0 0 0 1	0.15
0 0 1 0	0.05
0 0 1 1	0
0 1 0 0	0.05
0 1 0 1	0
0 1 1 0	0
0 1 1 1	0.15
1 0 0 0	0.05
1 0 0 1	0
1 0 1 0	0
1 0 1 1	0.05
1 1 0 0	0
1 1 0 1	0.05
1 1 1 0	0.05
1 1 1 1	0.10

When $|\mu_0 - \mu_1| = \infty$, the two classes are completely separable, there is virtually no ambiguity. We compare the performance of context-sensitive maximum likelihood decision rule and the context-free maximum likelihood decision rule for the above setup in terms of both set-by-set and element-by-element error probability as we vary $|\mu_0 - \mu_1|$. Monte Carlo experiments were run and the results are illustrated in Table 2.2 and Figure 2.1. As we can see, by using context, smaller set-by-set and element-by-element error probability are consistently achieved for varying $|\mu_0 - \mu_1|$. Error probabilities, both set-by-set and element-by-element, decrease as $|\mu_0 - \mu_1|$ becomes larger. This is not surprising since the ambiguity is getting smaller. However, the significance of context does not diminish. The ratios of context-sensitive error probabilities to context-free error probabilities actually decrease as $|\mu_0 - \mu_1|$ becomes large, as shown in Figure 2.2, which implies that the effect of context becomes more significant in a relative sense.



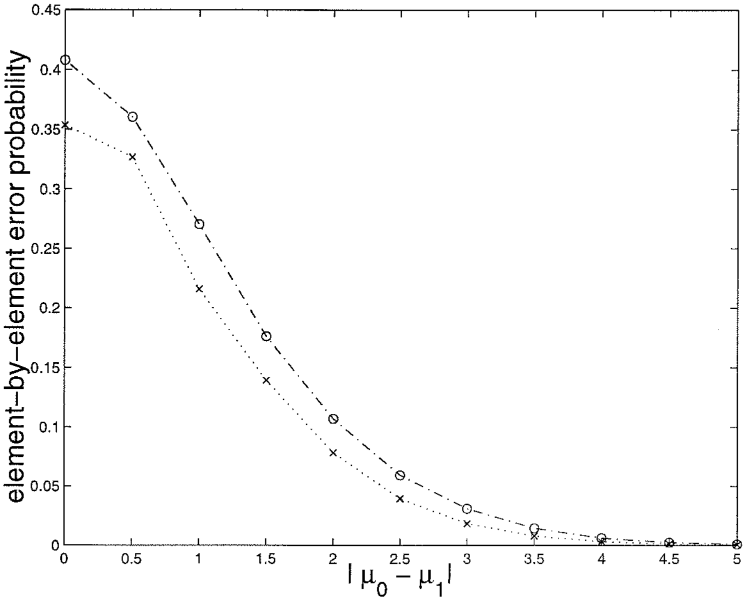


Figure 2.1: Comparison of error probabilities between with context and without context, for both set-by-set error and element-by-element error. In both figures, the dashdot line is without context, and the dotted line is with context.

Table 2.2: Comparison between errors for with and without context

$\mu_0 - \mu_1$	set-by-set w/o	set-by-set w/	ele-by-ele w/o	ele-by-ele w/
0	0.866485	0.682500	0.407871	0.353600
0.5	0.818340	0.658045	0.360449	0.326740
1.0	0.706747	0.512500	0.270263	0.215787
1.5	0.535298	0.382020	0.176094	0.139310
2.0	0.363090	0.237900	0.106931	0.078304
2.5	0.217084	0.130278	0.059470	0.039422
3.0	0.118153	0.064205	0.030936	0.018312
3.5	0.056293	0.029339	0.014382	0.007899
4.0	0.024691	0.012473	0.006231	0.003249
4.5	0.009770	0.004770	0.002454	0.001219
5.0	0.003571	0.001766	0.000894	0.000445

2.4 Special Cases of Context

2.4.1 Only the Counts Count.

We deal with the application of object classification where it is the count in each class, rather than the particular ordering or numbering of the objects, that matters. Such is the case for the application of white blood cell identification and microscopic urinalysis where it is the percentage profile of all classes in a specimen that convey diagnostic information.

Proposition: $p(c_1, c_2, ..., c_N) = p(\pi(c_1, c_2, ..., c_N))$, where $\pi(c_1, c_2, ..., c_N)$ is an arbitrary permutation of $c_1, c_2, ..., c_N$, if and only if $p(c_1, c_2, ..., c_N) = p(N_1, N_2, ..., N_D)$, where N_d is the count of class d, for $d = 1, ..., D$ and $\sum_{d=1}^{D} N_d = N$.

Proof:

A vector $(c_1, c_2, ..., c_N)$ is said to be permuted into *basic arrangement* by π if $\pi(c_1, c_2, ..., c_N) = c_1', c_2', ..., c_N'$ and the arrangement of $c_1', c_2', ..., c_N'$ is such that the first N_1 elements are in class 1, the next N_2 elements are in class 2, *etc.* In other words, $c_1', c_2', ..., c_N'$ has the count profile of $(N_1, N_2, ..., N_D)$.

Since any vectors $(c_1, c_2, ..., c_N)$ that have the count profile of $(N_1, N_2, ..., N_D)$ can be permuted into the same basic arrangement $(c_1', c_2', ..., c_N')$, and since $p(c_1, c_2, ..., c_N) =$

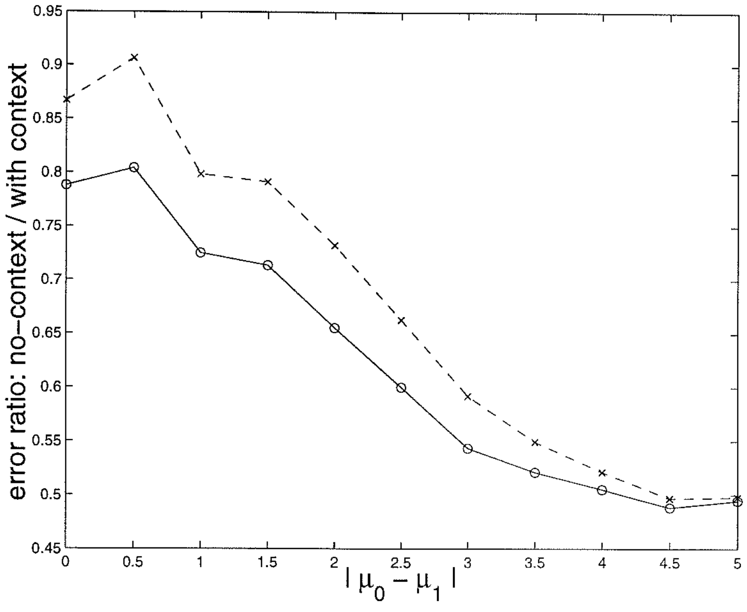


Figure 2.2: The ratio of with-context error probability to without-context error probability. The dashed line is for element-by-element and the solid line is for set-by-set error probability.

$p(\pi(c_1, c_2, ..., c_N)) = p(c_1^{'}, c_2', ..., c_N')$, therefore, $p(c_1, c_2, ..., c_N) = p(N_1, N_2, ..., N_D)$ is a function of the count profile.

The argument also goes the other way: if $p = p(N_1, N_2, ..., N_D)$ is a function only of the count profile $(N_1, N_2, ..., N_D)$, then $p(c_1, c_2, ..., c_N) = p(\pi(c_1, c_2, ..., c_N))$ for any arbitrary permutation $\pi(c_1, c_2, ..., c_N)$ of $(c_1, c_2, ..., c_N)$.

We prove this by contradiction. Assume there exist two vectors $(c_1^1, c_2^1, ..., c_N^1)$ and $(c_1^2, c_2^2, ..., c_N^2)$ that have the same basic arrangement $(c_1, c_2, ..., c_N)$,*i.e.*, $\pi^1(c_1^1, c_2^1, ..., c_N^1) = (c_1, c_2, ..., c_N)$ and $\pi^2(c_1^2, c_2^2, ..., c_N^2) = (c_1, c_2, ..., c_N)$ but $p(c_1^1, c_2^1, ..., c_N^1) \neq p(c_1^2, c_2^2, ..., c_N^2)$. Since $p(c_1^1, c_2^1, ..., c_N^1) = p(\pi^1(c_1^1, c_2^1, ..., c_N^1)) = p(c_1, c_2, ..., c_N)$, and $p(c_1^2, c_2^2, ..., c_N^2) = p(\pi^2(c_1^2, c_2^2, ..., c_N^2)) = p(c_1, c_2, ..., c_N)$, therefore, $p(c_1^1, c_2^1, ..., c_N^1) = p(c_1^2, c_2^2, ..., c_N^2)$, which contradicts the assumption. **Q.E.D.**

As a result, the contextual ratio $\rho(c_1, c_2, ..., c_N)$ is only a function of the count in each class. Since there are $\frac{N!}{N_1!...N_D!}$ ways of arranging N objects that gives rise to a frequency profile of $(\nu_1, \nu_2, ..., \nu_D)$, then

$$p(c_1, c_2, ..., c_N) = \frac{N_1!...N_D!}{N!}p(\nu_1, \nu_2, ..., \nu_D) \tag{2.13}$$

Therefore,

$$\rho(c_1, c_2, ..., c_N) = \frac{p(c_1, c_2, ..., c_N)}{p(c_1)...p(c_N)}$$

$$= \frac{N_1!...N_D!\ p(\nu_1, \nu_2, ..., \nu_D)}{N!\ P_1^{N\nu_1}...P_D^{N\nu_D}} \tag{2.14}$$

$$\doteq \rho(\nu_1, ..., \nu_D)$$

where P_d is the prior probability of class d, for $d = 1, ...D$.

Let

$$\alpha(\nu_1, \nu_2, ..., \nu_D) = \frac{N_1!...N_D!}{N!\ P_1^{N\nu_1}...P_D^{N\nu_D}} \tag{2.15}$$

then,

$$\rho(\nu_1, \nu_2, ..., \nu_D) = \alpha(\nu_1, \nu_2, ..., \nu_D)p(\nu_1, \nu_2, ..., \nu_D) \tag{2.16}$$

Applying Stirling's formula $\sqrt{2\pi}n^{n+\frac{1}{2}}e^{-n}e^{\frac{1}{12n+1}} \leq n! \leq \sqrt{2\pi}n^{n+\frac{1}{2}}e^{-n}e^{\frac{1}{12n}}$ to the factorials in 2.15, we can derive the upper and lower bounds for $\alpha(\nu_1, \nu_2, ..., \nu_D)$,

$$(\sqrt{2\pi})^{D-1}e^{N\mathcal{H}(\nu,\mathbf{P})}\sqrt{\nu_1 \dots \nu_D}e^{\sum_{d=1}^{D} \frac{1}{12N_d+1} - \frac{1}{12N}} \tag{2.17}$$

$$\leq \quad \alpha(\nu_1, \nu_2, ..., \nu_D) \tag{2.18}$$

$$\leq \quad (\sqrt{2\pi})^{D-1}e^{N\mathcal{H}(\nu,\mathbf{P})}\sqrt{\nu_1 \dots \nu_D}e^{\sum_{d=1}^{D} \frac{1}{12N_d} - \frac{1}{12N+1}} \tag{2.19}$$

where $H(\nu, \mathbf{P}) = \sum_{d=1}^{D} \nu_d ln(\frac{\nu_d}{P_d})$ is the relative entropy between the frequency profile and the prior distribution of the classes. The upper and lower bounds of context ratio $\rho(\nu_1, \nu_2, ..., \nu_D)$ follows immediately.

$$
\begin{aligned}
& (\sqrt{2\pi})^{D-1} e^{N\mathcal{H}(\nu,\mathbf{P})} \sqrt{\nu_1 \ldots \nu_D} e^{\sum_{d=1}^{D} \frac{1}{12N_d+1} - \frac{1}{12N}} p(\nu_1, \nu_2, ..., \nu_D) && (2.20) \\
\leq & \quad \rho(\nu_1, \nu_2, ..., \nu_D) && (2.21) \\
\leq & \ (\sqrt{2\pi})^{D-1} e^{N\mathcal{H}(\nu,\mathbf{P})} \sqrt{\nu_1 \ldots \nu_D} e^{\sum_{d=1}^{D} \frac{1}{12N_d} - \frac{1}{12N+1}} p(\nu_1, \nu_2, ..., \nu_D) && (2.22)
\end{aligned}
$$

2.4.2 Only the Presence Matters.

In real world applications, there are cases where the contextual information is in the form of the *presence* of some classes. It is the sheer presence, rather than the amount, that provides context. For example, in microscopic urinalysis, the presence of white blood cells indicates the presence of bacteria. We would like to formulate context in the form of the presence of the classes.

We define random variable A_d in the following way for $d = 1, \ldots, D$.

$$
A_d = \begin{cases} 1 & \text{if class } d \text{ is present} \\ 0 & \text{if class } d \text{ is absent} \end{cases}
$$

A classification vector $(c_1, c_2, \ldots, c_N)$ can be mapped into a presence vector $(A_1, \ldots, A_D)$, where $A_d = 1$ if at least one of the c_i's are d and $A_d = 0$ otherwise. Under the assumption that all arrangements of the elements in a set $(c_1, c_2, \ldots, c_N)$ that give the same $(A_1, \ldots, A_D)$ are equally likely, we can derive the relation between the two,

$$
p(c_1, \ldots, c_N) = \frac{p(A_1, \ldots, A_D)}{T_{N,||A||}}
$$

where $||A||$ is the total number of $1's$ in A, and $A = (A_1, \ldots, A_D)$ is the binary presence vector.

The denominator $T_{N,||A||}$ is the number of arrangements of N different elements that result in designated $||A||$ non-empty classes with at least one element in each of them and $D - ||A||$ empty classes. Some calculation leads to the iterative relation for $T_{N,||A||}$ given by

$$\begin{cases} T_{n,1} = 1 \\ T_{n,2} = 2^n - 2 \\ T_{n,d} = \sum_{k=1}^{n-(d-1)} \binom{n}{k} T_{n-k,d-1} \end{cases}$$

A sketch of the derivation is given in the Appendix.

As a result, the contextual ratio $\rho(c_1, c_2, ..., c_N)$ is only a function of the presence in each class, *i.e.*,

$$\rho(c_1, c_2, ..., c_N) = \frac{p(c_1, c_2, ..., c_N)}{p(c_1)...p(c_N)}$$

$$= \frac{p(A_1, \ldots, A_D)}{T_{N,||A||} P_1^{N\nu_1} ... P_D^{N\nu_D}} \tag{2.23}$$

$$= \rho(A_1, ..., A_D; N_1, \ldots, N_D)$$

2.5 Complexity Problem

When implementing the context-sensitive maximum likelihood decision rule, we want to find a $(c_1, \ldots, c_N)$ that maximizes $p(c_1, \ldots, c_N | \mathbf{x}_1, \ldots, \mathbf{x}_N)$. There are D possible choices for each c_i, and there are N elements, $i = 1, \ldots, N$. Therefore, there are D^N possibilities that $(c_1, \ldots, c_N)$ can take on. We need to compute and compare D^N cases to find the maximum of $p(c_1, \ldots, c_N | \mathbf{x}_1, \ldots, \mathbf{x}_N)$. Suppose the D dimensional probability vector $p(c_i | x_i)$ is given for all $i = 1, \ldots, N$, then for the computation of $p(c_1, \ldots, c_N | \mathbf{x}_1, \ldots, \mathbf{x}_N)$ for each case of $(c_1, \ldots, c_N)$, $2N+1$ multiplications are needed (see 2.7). The total number of multiplication for all D^N cases is $(2N + 1)D^N$. Since finding the maximum of of n numbers has complexity n, then finding the maximum of of D^N numbers has complexity of D^N. For the blood cell recognition problem, $D = 14$ and N is typically around 600, the computation is enormous and virtually impossible to implement. In the context-free case, we deal with each element individually by maximizing $p(c_i|\mathbf{x}_i)$ with respect to c_i for all $i = 1, \ldots, N$. Since the D dimensional probability vector $p(c_i|x_i)$ is already given for all $i = 1, \ldots, N$, we only need to find the maximum of all D-dimensional vectors, whose total complexity is ND. Incorporating context entails computation cost, both in terms of total number of multiplications and in terms of the complexity of finding maximum of D^N numbers. In some cases, additional constraints can be used to reduce computation, as is the case in white blood cell identification, which will be demonstrated in the following section. In some other cases when such simplifications are not feasible, we will have to resort to methods that get around the computation problem, possibly at the cost of accuracy. This is the motivation behind utilizing context in an indirect way, which will be described in the following section.

2.6 Partial Context

2.6.1 Mathematical Formulation

Context-sensitive maximum likelihood leads to computation problems for relatively large D and N. The alternative of using $p(c_i|\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_N)$ also has exponential computation cost. Instead of using the primary "raw" context contained in the feature vector $(\mathbf{x}_1, \ldots, \mathbf{x}_N)$, we use the "intermediate-stage" context. The context is called "intermediate-stage" because it is *derived* from "raw" context or estimated without using context.

Let A be the "intermediate-stage" context. The physical definition of A depends on the problem at hand. For example, A can be the percentage profile of all the classes, or the binary presence vector of the classes, or the presence of one or a few particular classes. A can also represent certain external information sources, such as the chemistry result in urinalysis which is a urine test different from, yet related to, microscopic urinalysis [Boehringer-Mannheim-Corporation, 1991].

By Bayes' Rule,

$$\begin{aligned} p(c_i|x_i, A) &= \frac{p(c_i, x_i; A)}{p(x_i; A)} \\ &= \frac{p(x_i|c_i, A)P(c_i; A)}{p(x_i; A)} \end{aligned}$$

Once again, we make the conditional independence assumption that the feature distribution of an object is dependent only on its own class, not the features or classes of other objects, *i.e.,* $p(x_i|c_i, A) = p(x_i|c_i)$, then

$$\begin{aligned} p(c_i|x_i, A) &= \frac{p(x_i|c_i)p(c_i; A)}{p(x_i; A)} \\ &= \frac{p(x_i|c_i)p(c_i|A)p(A)}{p(x_i; A)} \end{aligned}$$

$$= p(c_i|x_i)\frac{p(c_i|A)}{p(c_i)}\frac{p(A)p(x_i)}{p(x_i; A)} \tag{2.24}$$

$$\propto p(c_i|x_i)\frac{p(c_i|A)}{p(c_i)} \tag{2.25}$$

The context-partial-sensitive *a posteriori* probability $p(c_i|x_i, A)$ is obtained through the context-free *a posteriori* probability $p(c_i|x_i)$ modified by context ratio $\rho = \frac{p(c_i|A)}{p(c_i)}$.

The partial-context maximum likelihood (PCML) decision rule chooses class label $\hat{c}_i$ for element i such that

$$\hat{c}_i = \underset{c_i}{\operatorname{argmax}}\, p(c_i|\mathbf{x}_i, A) \tag{2.26}$$

It is important to point out two aspects of using partial context. One is that the context contained in random variable A needs to be reliable for formula 2.25 to be carried out directly. Otherwise, using wrong context may lead to worse performance. Another point is that A needs to be context-bearing. In other words, the value of A does convey information about c_i. There are many ways to measure the level of relevance of A to c_i, such as the mutual information $I(c_i, A)$, or the relative entropy between the prior distribution $p(c_i)$ and the context conditional distribution $p(c_i|A)$.

2.6.2 A Toy Example

We demonstrate the use of partial context to improve performance by a toy problem. There are $N = 3$ elements in each set. $\underline{c} = (c_1, c_2, c_3)$. Each element takes ternary values from $\Omega = \{0, 1, 2\}$, therefore $D = 3$. The joint distribution $p(\underline{c})$ is specified in Table 2.3. From the joint distribution $p(\underline{c})$, it can be calculated that the prior distribution of the three classes are $p(0) = 0.26$, $p(1) = 0.49$ and $p(2) = 0.25$. From formula 2.25 we know that $p(c_i|A_d)$ is needed to calculate the context ratio $\rho = \frac{p(c_i|A)}{p(c_i)}$. Table 2.4 lists $p(c|A_d = 1)$ and $p(c|A_d = 0)$ for $c = 0, 1, 2$ and $d = 0, 1, 2$.

We choose $p(\underline{x}|\underline{c}) = \prod_{i=1}^{N} p(x_i|c_i)$, as assumed in Equation 2.5. The conditional feature distributions are Gaussians $p(x|c = 0) = N(\mu_0, \sigma_0)$, $p(x|c = 1) = N(\mu_1, \sigma_1)$, and $p(x|c = 2) = N(\mu_2, \sigma_2)$. In this toy problem, the intermediate-stage context is the presence or absence of class 0. As mentioned earlier, the context A that $p(c_i|x_i, A)$ is conditioned on has to be reliable. We choose $\mu_0 = 0$, $\sigma_0 = 0.01$ and both $\mu_1, \mu_2 \gg \mu_0$ so that there is hardly any ambiguity in the classification of class 0, therefore, the context, *i.e.*, the detection of presence of class 0, is accurate. We choose $\sigma_1 = 1.0$ and $\sigma_2 = 0.6$. The distance between μ_1 and μ_2 determines the separability of class 1 and 2. We compare the performance of context-sensitive, context-partial-sensitive, and context-free maximum likelihood decision rules for the above in terms of both set-by-set and element-by-element error probability as we vary $|\mu_1 - \mu_2|$. Monte Carlo experiments were run and the results are illustrated in Table 2.5, Table 2.6 and Figure 2.3. The main moral of this toy problem is, as we can see, that the context-partial-sensitive algorithm consistently outperformed the context-free algorithm, for both set-by-set and element-by-element error probability. In terms of set-by-set error, the context-sensitive algorithm is the best, which is expected due to its optimality. In terms of element-by-element error probability, both the context-sensitive and the context-partial sensitive algorithms are better than the context-free algorithm, but there is no clear winner between the two. Error probability, both set-by-set and element-by-element, decrease as $|\mu_1 - \mu_2|$ becomes larger. This is not surprising since the ambiguity is getting smaller. However, the significance of context does not diminish. The ratios of context-sensitive error probabilities, for both full and partial context, to context-free error probabilities actually decrease as $|\mu_0 - \mu_1|$ becomes large, as shown in Figure 2.4, which implies that the effect of context becomes more significant in a relative sense.

2.6.3 Computational Cost Using PCML

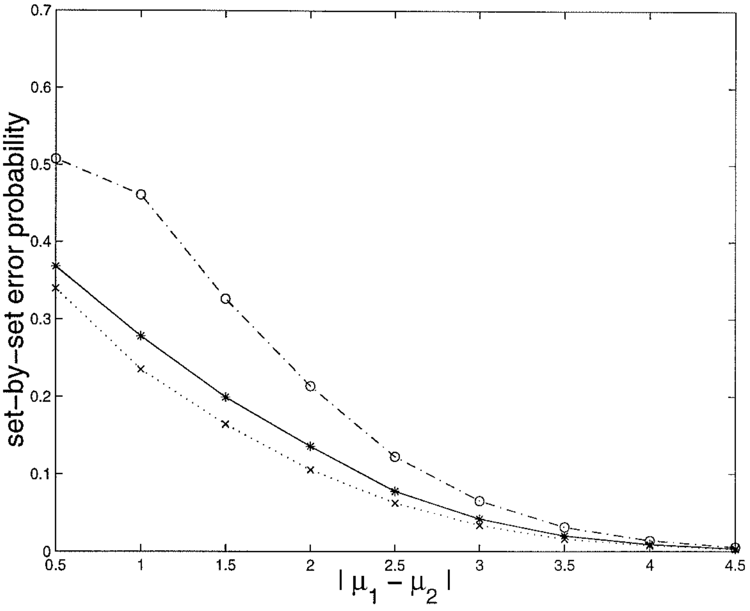
Similar to the context-free approach, the partial-context approach treats each element in a set individually, with additional information from context-bearing factor A. Again the D dimensional probability vector $p(c_i|x_i)$ is already given for all $i = 1, \ldots, N$. Once the context A is obtained, we want to maximize $p(c_i|x_i, A)$ from D possible values that c_i can take on. For each i, we need to do 2 multiplications (see 2.25) and to find the maximum of D numbers. Then the total number of multiplications is $2N$, which is linear in N, compared to the exponential relation $(2N+1)D^N$ in the full context case. The total complexity for finding the maximum is ND, the same as in the context-free approach, compared to the exponential complexity $D^N N$ in the full context case.

2.7 Appendix

Claim: Let $T_{n,d}$ be the number of arrangements of n different elements into d classes with at least one element in each class. It holds true that

$$\begin{cases} T_{n,1} = 1 \\ T_{n,2} = 2^n - 2 \\ T_{n,d} = \sum_{k=1}^{n-(d-1)} \binom{n}{k} T_{n-k,d-1} \end{cases}$$

Proof: $T_{n,1}$ is the number of ways to arrange n elements in 1 class. Obviously, $T_{n,1} = 1$. When we arrange n elements into 2 designated classes, since both 2 classes are non-empty, the possibilities are: there are k elements in the first class, $n-k$ in the second for $k = 1, \ldots, n-1$. Therefore the total number of ways to result in 2 non-



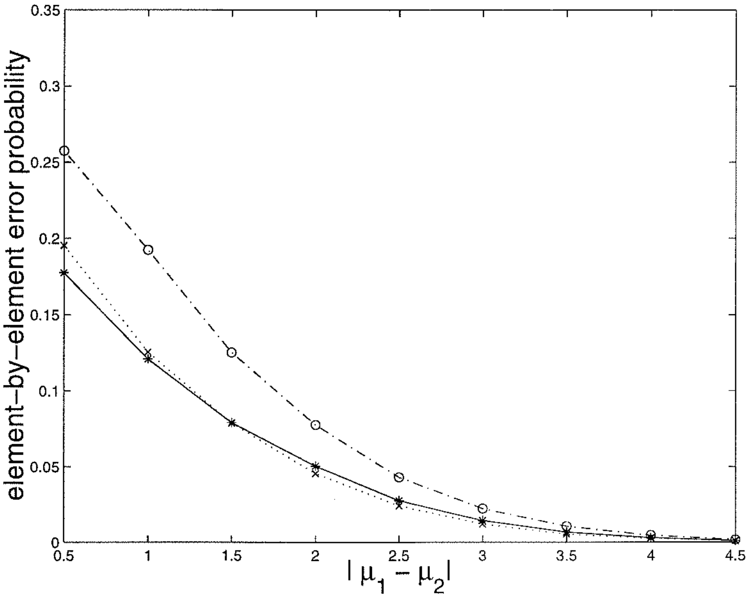
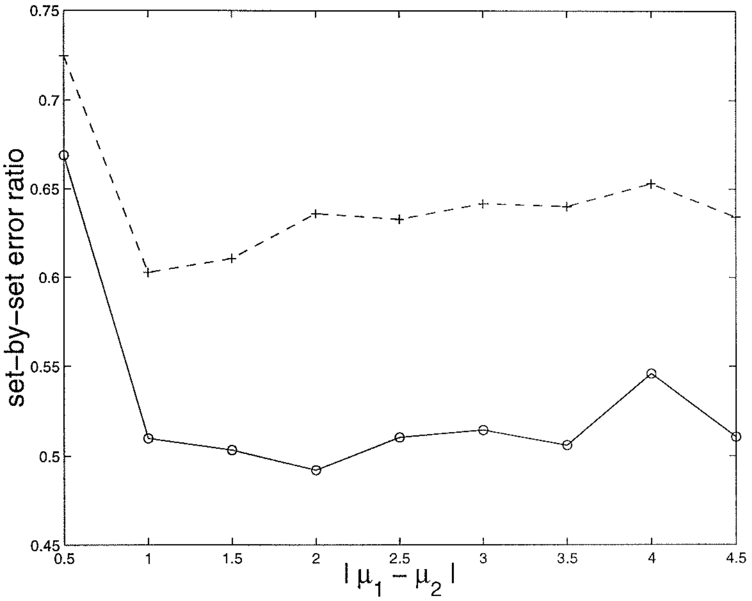


Figure 2.3: Comparison of error probabilities among with full context, partial context and without context, for both set-by-set error and element-by-element error. In both figures, the dashdot line is without context, the dotted line is with full context, and the solid line is with partial context.



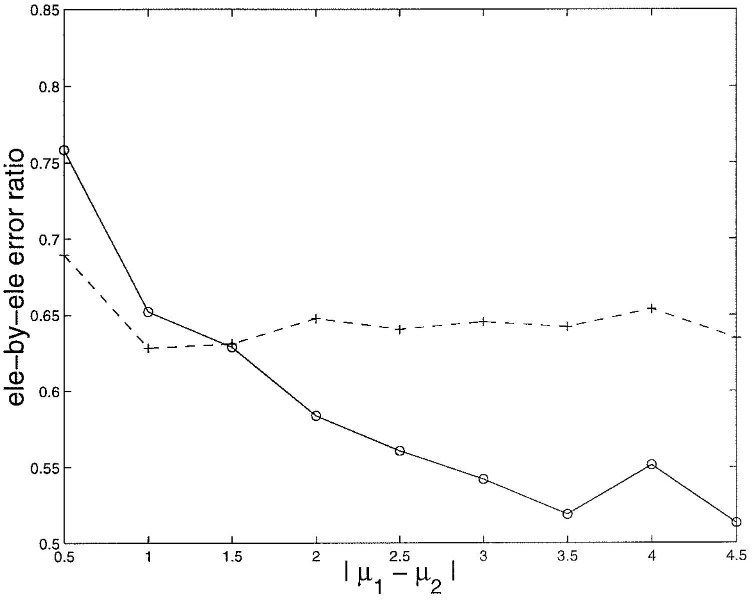


Figure 2.4: The ratio of with-context error probability to without-context error probability, in terms of both set-by-set and element-by-element error probabilities. In both figures, the dashed line is with partial context, and the solid line is with full context.

empty classes is $\binom{n}{1}+\binom{n}{2}+\ldots+\binom{n}{n-1} = 2^n-2$. When there are 3 non-empty classes, the possibilities are: there are k element in the first class with the remaining $N-k$ elements being arranged into the other two classes for $k=1,\ldots,n-2$. Therefore, the total number of ways to result in 3 non-empty classes is $\binom{n}{1}T_{n-1,2}+\binom{n}{2}T_{n-2,2}+\ldots+\binom{n}{n-2}T_{2,2}=T_{n,3}$, which can be calculated since we already know $T_{n,2}$. Following the same logic, to result in d non-empty classes, the possibilities are: there are k elements in the first class with the rest $N-k$ elements being arranged into the remaining $d-1$ classes for $k=1,\ldots,n-(d-1)$. Therefore, the total number of ways to result in d non-empty classes is $\binom{n}{1}T_{n-1,d-1}+\binom{n}{2}T_{n-2,d-1}+\ldots+\binom{n}{n-(d-1)}T_{d-1,d-1}=T_{n,d}$. Once we know $T_{m,d-1}$ for $m=d-1,\ldots,n-1$, $T_{n,d}$ can be calculated. **Q.E.D.**

Note: All the way, though, we assume that $n \geq d$ to make $T_{n,d}$ a legitimate quantity. Using this iterative relation, it follows that $T(n,n)=n!$, which is expected.

Table 2.3: p($\underline{c}$) for the Toy Problem

$\underline{c}$	p($\underline{c}$)
0 0 0	0
0 0 1	0.15
0 0 2	0
0 1 0	0.05
0 1 1	0.02
0 1 2	0
0 2 0	0
0 2 1	0
0 2 2	0
1 0 0	0.08
1 0 1	0.15
1 0 2	0
1 1 0	0.05
1 1 1	0.05
1 1 2	0.10
1 2 0	0
1 2 1	0.15
1 2 2	0.02
2 0 0	0
2 0 1	0
2 0 2	0
2 1 0	0
2 1 1	0.02
2 1 2	0.02
2 2 0	0
2 2 1	0.02
2 2 2	0.12

	$A_0 = 1$	$A_0 = 0$	$A_1 = 1$	$A_1 = 0$	$A_2 = 1$	$A_2 = 0$
$p(c = 0 \mid A_d)$	0.52	0	0.296	0	0	0.47
$p(c = 1 \mid A_d)$	0.48	0.50	0.557	0	0.44	0.53
$p(c = 2 \mid A_d)$	0	0.50	0.148	1.00	0.56	0

Table 2.4: $p(c \mid A_d = 1)$ and $p(c \mid A_d = 0)$

$\mu_1 - \mu_2$	w/o	w/p	w/f
0.5	0.507672	0.368006	0.339566
1.0	0.460798	0.277748	0.234935
1.5	0.326270	0.199237	0.164245
2.0	0.213740	0.135970	0.105204
2.5	0.122439	0.077499	0.062495
3.0	0.065066	0.041755	0.033470
3.5	0.031372	0.020084	0.015873
4.0	0.013825	0.009028	0.007548
4.5	0.005370	0.003406	0.002742

Table 2.5: Comparison between set-by-set error probabilities for with full context, partial context and without context

$\mu_1 - \mu_2$	w/o	w/p	w/f
0.5	0.257425	0.177472	0.195237
1.0	0.192419	0.120881	0.125462
1.5	0.125242	0.079026	0.078751
2.0	0.077659	0.050297	0.045337
2.5	0.042798	0.027409	0.023997
3.0	0.022220	0.014337	0.012044
3.5	0.010575	0.006790	0.005487
4.0	0.004630	0.003027	0.002553
4.5	0.001793	0.001138	0.000920

Table 2.6: Comparison between element-by-element error probabilities for with full context, partial context and without context

Chapter 3 White Blood Cell Identification

3.1 Introduction

White blood cell (WBC) analysis is one of the major routine laboratory examinations. The utility of WBC classification in clinical diagnosis relates to the fact that in various physiological and pathological conditions the relative percentage composition of the WBC changes. An estimate of the percentage of each class present in a blood sample conveys information which is pertinent to the hematological diagnosis. Most WBC differentiation depends almost entirely on manual specimen preparation and human interpretation, and more than ninety percent of the direct costs are labor. The availability of Automated Intelligent Microscopy Flow Imaging technology (see [Kasdan *et al.*, 1994]) makes it possible to have automated differentiation, which will reduce labor and health care costs, and is more efficient. Typical commercial differential WBC counting systems are designed to identify five major mature cell types. But blood samples may also contain immature cells. These cells occur infrequently in a normal specimen, and most commercial systems will simply indicate the presence of these cells because they can't be individually identified by the systems. But it is precisely these cell types that relate to the production rate and maturation of new cells and thus are important indicators of hematological disorders. Our system is designed to differentiate fourteen WBC types which includes the five major mature types: segmented neutrophils, lymphocytes, monocytes, eosinophils, and basophils; *and* the immature types: bands (unsegmented neutrophils), metamyelocytes, myelocytes, promyelocytes, blasts, and variant lymphocytes; as well as nucleated red blood cells and artifacts. Differential counts are made based on the cell classifications, which further leads to diagnosis or prognosis. Table 3.1 gives a range of differential counts of all cell types within which a specimen is considered normal. A specimen is abnormal if the differential counts of one or more cell types fall out of their ranges.

The data was provided by International Remote Imaging Systems (IRIS), Inc. Blood specimens are collected at Harbor UCLA Medical Center from local patients, then dyed with Basic Orange 21 metachromatic dye supravital stain. The specimen is then passed through a flow microscopic imaging and image processing instrument, where the blood cell images are captured. Each image contains a single cell with full color. There are typically 600 images from each specimen. The task of the cell recognition system is to categorize the cells based on the images. Figure 3.1 is an example of cell images of various types.

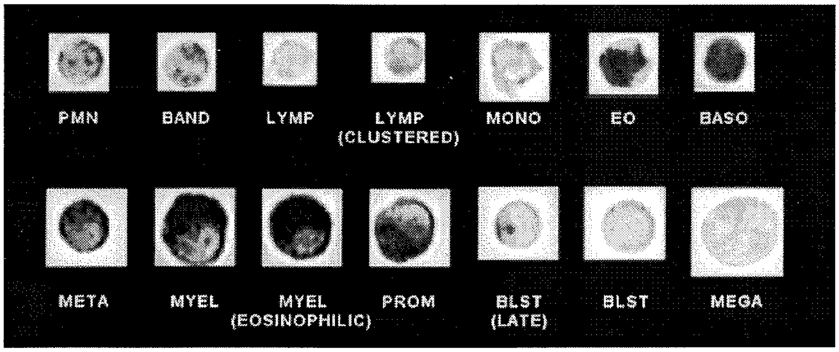


Figure 3.1: Example of some of the cell images.

3.2 Image Processing and Feature Extraction

The size of cell images are automatically tailored according to the size of the cell in the images. Images containing larger cells have bigger sizes than those with small cells. The range varies from 20x20 to 40x40 pixels. The average size is around 25x25 (see Figure 3.1). At the preprocessing stage, the images are segmented to set the cell interior apart from the background. We use adaptive thresholding for image segmentation.

cell type	normal range
pmn	34.4836 - 72.7435 %
lymp	14.5011 - 48.9739 %
mono	4.5973 - 12.2653 %
band	0 - 7.6776 %
eo	0 - 9.7089 %
baso	0 - 1.5436 %
meta	0 - 0.2757 %
vlymp	0 - 1.1691 %
myel	0 - 0.0467 %
blast	0
mega	0
prom	0

Table 3.1: Normal ranges of the percentages of all white blood cell types.

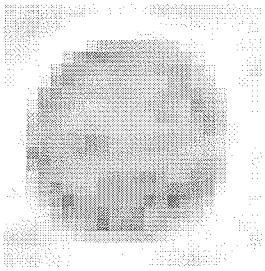


Figure 3.2: Example of a cell.

3.2.1 Adaptive Thresholding for Image Segmentation

Each color cell image is decomposed into red, green and blue frames; each frame itself is a gray level image. In ideal situations, the average intensity of cells of the same type should be more or less the same for the same color frame, and the background should have homogeneous texture and have the same intensity level. However, this is not true due to the instability of exposure time, variation in lens focus and lighting intensity in the microscopic imaging systems and other various sources of noise in the system. Some images appear brighter, some darker. There is also heterogeneous noise in both the background and the foreground, which makes thresholding with constant threshold an inferior approach. This motivates us to adopt an adaptive thresholding

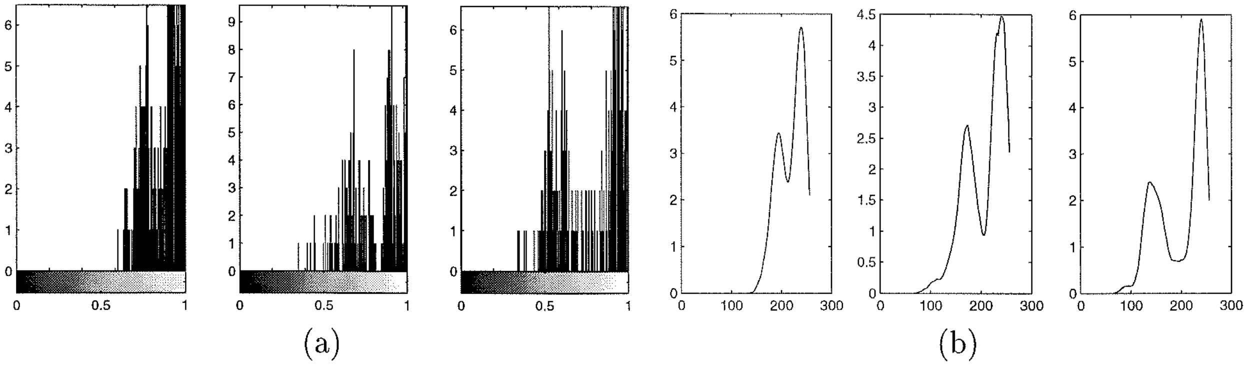


Figure 3.3: (a)Histograms of red, green, blue frames. (b) Smoothed version of (a).

technique, where the thresholds for each color frame of one cell vary according to the statistics of the image.

A gray level image is first filtered with a symmetric Gaussian filter for noise reduction.

$$I_s(i,j) = I(i,j) * G(r,c)$$

where * is convolution operator. $I(i,j)$ is the original gray level image, $I_s(i,j)$ is the smoothed image.

$$G(r,c) = \frac{e^{-\frac{r^2+c^2}{2\sigma^2}}}{\sum\limits_{(r,c)\in W} e^{-\frac{r^2+c^2}{2\sigma^2}}}$$

is the low-pass Gaussian kernel, and W is the size of the kernel, which is chosen to be a 5x5 window in our system $(-2 \leq r, c \leq 2)$ and σ controls the level of smoothness of the filter.

A gray level histogram $H(t)$ is obtained for smoothed gray level image $I_s(i,j)$, where $t = 0, \ldots, 255$. $H(t)$ is a non-smooth discrete series with various peaks and valleys. See Figure 3.3 (a). This series is filtered through a smoothing filter

$$H_s(t) = H(t) * [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]/20$$

The smoothed version $H_s(t)$ of a histogram generally has one significant valley, which corresponds to the highlighted threshold value that sets apart the background and the foreground. See Figure 3.3 (b). An intensity value t is a threshold if

$$H_s^{'}(t) = 0 \tag{3.1}$$

and

$$H_s^{''}(t) > 0 \tag{3.2}$$

A 5^{th} order difference model is used to calculate the derivatives.

$$H_s^{'}(t) = \frac{1}{12}(H(t-2) - 8H(t-1) + 8H(t+1) - H(t+2))$$

.

Instead of using Equation 3.1 and Equation 3.2, we use the following simplified rules to locate the threshold: A value t is the threshold if $H_s^{'}(t-2) < 0$ and $H_s^{'}(t-1) < 0$ and $H_s^{'}(t+1) > 0$ and $H_s^{'}(t+2) > 0$ and $T_l < t < T_u$, where T_l and T_u are the upper and lower bounds of the range that a reasonable threshold should fall in, which is decided *a priori*.

Once the threshold T is decided upon, the original gray level image is segmented according to

$$B(i,j) = \begin{cases} 1 & \text{if } I(i,j) > T \\ 0 & \text{otherwise} \end{cases}$$

where $B(i,j)$ is the binary image after segmentation.

Segmentation is done separately for all three color frames of each color image, resulting in three binary images. A binary NOR operation is taken for the binary images on a pixel-by-pixel basis, i.e.,

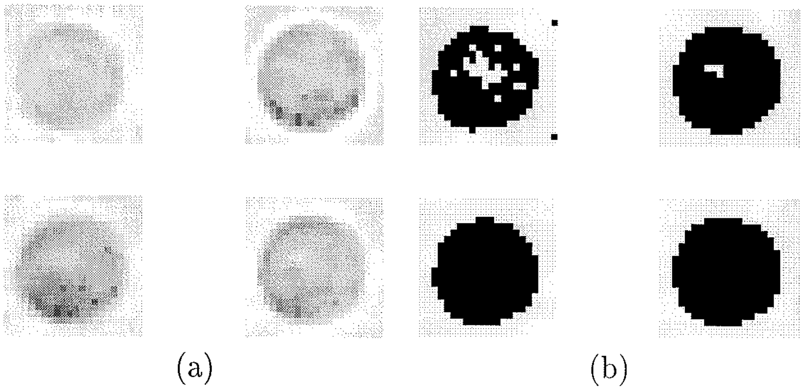


Figure 3.4: (a)Gray level red, green, blue frames and intensity image. (b) Segmentation of red, green, blue frames and the final segmentation.

$$B(i,j) = \begin{cases} 0 & \text{if } B_r(i,j) = 0 \text{ or } B_g(i,j) = 0 \text{ or } B_b(i,j) = 0 \\ 1 & \text{otherwise} \end{cases}$$

This gives rise to a binary image with small holes in the foreground and small islands in the background. A hole-filling and island-removing process is carried out to deal with this problem, obtaining a compact island-free cell segmentation. See Figure 3.4 (b).

3.2.2 Feature Extraction

Features based on the interior of the cells are extracted from the images. The features include size, shape, color and texture. See Table 3.2 for the list of features. [1] These features by design are rotation and translation invariant.

[1] The red-blue distribution is the pixel-by-pixel $ln(red) - ln(blue)$ distribution for pixels in cell interior. The red distribution is the distribution of the red intensity in cell interior.

Table 3.2: Features extracted from cell images

feature number	feature description
1	cell area
2	number of pixels on cell edge
3	the 4th quantile of red-blue distribution
4	the 4th quantile of green-red distribution
5	the median of red-blue distribution
6	the median of green-red distribution
7	the median of blue-green distribution
8	the standard deviation of red-blue distribution
9	the standard deviation of green-red distribution
10	the standard deviation of blue-green distribution
11	the 4th quantile of red distribution
12	the 4th quantile of green distribution
13	the 4th quantile of blue distribution
14	the median of red distribution
15	the median of green distribution
16	the median of blue distribution
17	the standard deviation of red distribution
18	the standard deviation of green distribution
19	the standard deviation of blue distribution
20	the standard deviation of the distance from the edge to the mass center

3.3 Learning and Classification

A feature vector is a numerical representation of a cell image, which is of a particular type. We would like to design a learning machine that can, through a learning algorithm, map a feature vector to its correct classification. This is done through supervised learning, where a set of labeled examples (the training set) is given, and the learning machine tries to infer the mapping between the feature vectors (the input) and the classifications (the output) by observing and extracting relevant information from the set of labeled input-output pairs. We choose the learning model to be parametric. The learning process takes the the training set as input and tries to adjust the parameters of the learning model to optimize some utility function. The optimal parameters hopefully correspond to an input-output relation that is close to the real one. The utility function is an error function measuring the deviation between the desired output and the model output on the training set.

Let P be the input dimension (number of features in a feature vector), and D be the output dimension (number of classes). Let $\mathbf{x}$ be the input vector, $\mathbf{g}(\mathbf{x}, \mathbf{w})$ be the output vector of the learning model, and $\mathbf{w}$ the model parameters. We choose our learning model to be Radial Basis Functions (RBF) [Bishop, 1995], where the functional relation between the input and output is defined by the following:

$$g_d(\mathbf{x}, \mathbf{w}) = \frac{exp(a_d(\mathbf{x}, \mathbf{w}))}{\sum_{d=1}^{D} exp(a_d(\mathbf{x}, \mathbf{w}))} \tag{3.3}$$

where

$$a_d(\mathbf{x}, \mathbf{w}) = \sum_{p=1}^{P} u_{d,p} exp(-\frac{||x_p - \mu_p||^2}{\sigma_p^2}) \tag{3.4}$$

where $d = 1, \ldots, D$ and $\mathbf{w} = (\mathbf{U}, \mu, \sigma^2)$ are the parameters. $\mathbf{U} = \{u_{d,p}\}$ for $d = 1, \ldots, D$ and $p = 1, \ldots, P$. $\mu = \{\mu_d\}$ and $\sigma^2 = \{\sigma_d^2\}$ for $d = 1, \ldots, D$. Figure 3.5 illustrates the architecture of the RBF learning model.

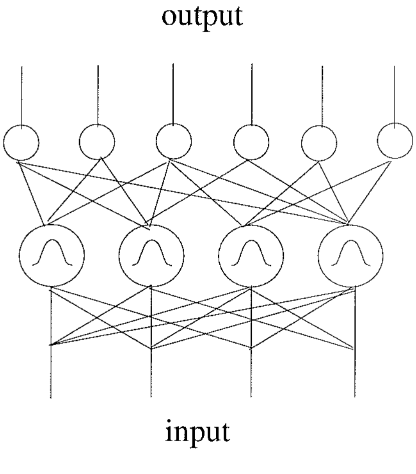


Figure 3.5: Structure of a radial basis function network

The functional mapping that the RBF model implements is $\mathbf{g}_{\mathbf{w}}(\mathbf{x}) : \mathbf{R}^P \to [0, 1]^D$. By design, Equation 3.3, the output vector $\mathbf{g}(\mathbf{w}, \mathbf{x}) = \{g_d(\mathbf{w}, \mathbf{x})\}_{d=1}^{D}$ has the property that $0 \leq g_d(\mathbf{w}, \mathbf{x}) \leq 1$ and $\sum_{d=1}^{D} g_d(\mathbf{w}, \mathbf{x}) = 1$. $g_d(\mathbf{w}, \mathbf{x})$ can be interpreted as the probability that $\mathbf{x}$ is in class d for a fixed set of parameters $\mathbf{w}$. Therefore, $g_d(\mathbf{w}, \mathbf{x})$ for $d = 1, \ldots, D$ is a probability vector.

The outputs of the training set inputs are encoded in such a way that it is a D-dimensional vector, with only one element being 1, the rest being 0, and the position of 1 indicating which class the cell belongs to. Let $\mathbf{y}(\mathbf{x})$ be the desired output for input $\mathbf{x}$. The error function $E(\mathbf{w})$ is defined as the summation of relative entropy between the desired $\mathbf{y}(\mathbf{x})$ and the real outputs probability vector $\mathbf{g}_{\mathbf{w}}(\mathbf{x})$ for any $\mathbf{x}$ in the training set.

$$E(\mathbf{w}) = \sum_{n} \sum_{d=1}^{D} y_d(\mathbf{x_n}) ln \frac{y_d(\mathbf{x_n})}{g_d(\mathbf{x}, \mathbf{w})}$$

The optimal set of weights is the one that minimizes the error function, *i.e.*,

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \, E(\mathbf{w})$$

According to [Richard and Lippmann, 1991], the learning process described above gives a convenient interpretation of the output. The d^{th} element of the output vector of the learning model at the optimal weights $\mathbf{w}^*$ for an input $\mathbf{x}$ corresponds to the *a posteriori* probability that $\mathbf{x}$ is in class d, i.e.

$$g_d(\mathbf{w}^*, \mathbf{x}) = \text{ Prob}(\ c(\mathbf{x}) = d \mid \mathbf{x} \ )$$

3.3.1 No-Context Performance

The maximum likelihood context-free decision rule at this stage is

$$d(\mathbf{x}) = \underset{d}{\operatorname{argmax}} \ p(d|\mathbf{x})$$

The context-free performance is illustrated in the confusion matrix in Table 3.3, where the entry at (i, j) represents number of cells that are really in class i and are classified into class j by the decision rule.

	lym	pmn	art	mono	eo	baso	mega	band	meta	prom	blst	myel	atly	rbc
lym	**348**	2	25	4	0	1	0	1	0	0	7	0	0	0
pmn	6	**691**	1	5	2	0	0	0	1	0	1	0	0	0
art	13	4	**123**	2	2	0	0	0	1	0	1	0	0	0
mono	9	5	0	**145**	0	1	0	8	0	0	0	0	0	0
eo	0	0	1	0	**13**	0	0	1	1	0	0	0	0	0
baso	0	0	1	0	0	**31**	0	0	0	0	0	0	0	0
mega	0	0	0	0	0	0	**0**	0	0	0	0	0	0	0
band	1	0	0	7	5	1	0	**138**	1	0	0	0	0	0
meta	0	6	0	0	1	0	0	5	**27**	0	0	0	0	0
prom	0	0	0	3	0	1	0	0	0	**0**	0	0	0	0
blst	18	1	3	3	0	0	0	0	0	0	**27**	0	0	0
myel	0	0	0	0	0	0	0	1	3	0	0	**0**	0	0
atly	0	0	0	0	0	0	0	0	0	0	1	0	**0**	0
rbc	5	0	3	0	0	0	0	0	0	0	0	0	0	**0**

Table 3.3: Confusion matrix generated by context-free maximum likelihood decision rule.

The average classification rate is 89.71%.

3.4 Incorporating Context into WBC Identification

3.4.1 Introduction

The "context-free" cell-by-cell decision is only based on the features presented by a cell, without looking at any other cells. When human experts make decisions, they always look at the whole specimen, taking into consideration the identities of other cells and adjusting the cell-by-cell decision on a single cell according to the company it keeps. On top of the visual perception of the cell patterns, such as shape, color, size, texture, etc., comparisons and associations, either mental or visual, with other cells in the same specimen are made to infer the final decision. A cell is assigned a certain identity if the company it keeps supports that identity. For instance, the difference between lymphocyte and blast can be very subtle sometimes, especially when the cell is large. A large unusual mononuclear cell with the characteristics of both blast and lymphocyte is more likely to be a blast if accompanied by other abnormal cells or an abnormal distribution of the cells.

This scenario fits in the framework we described in Chapter 1. Context incorporation is treated as the post-processing of the cell-by-cell decisions.

As was mentioned in Chapter 1, the contextual information is incorporated by maximizing $p(c_1, c_2, ..., c_N|\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N)$ given all the cells in a specimen simultaneously instead of maximizing $p(c_i|\mathbf{x}_i)$ for individual cells $i = 1, \ldots, N$ separately.

$$p(c_1, ..., c_N|\mathbf{x}_1, ..., \mathbf{x}_N) \propto p(c_1|\mathbf{x}_1)...p(c_N|\mathbf{x}_N)\frac{p(c_1, ..., c_N)}{p(c_1)...p(c_N)} \tag{3.5}$$

$$= p(c_1|\mathbf{x}_1)...p(c_N|\mathbf{x}_N)\rho(c_1, c_2, ..., c_N) \tag{3.6}$$

where

$$\rho(c_1, c_2, ..., c_N) \equiv \frac{p(c_1, c_2, ..., c_N)}{p(c_1)...p(c_N)} \tag{3.7}$$

In the case of white blood cell identification, only the counts in each class, not the ordering of the cells, that matters. Therefore,

$$\rho(c_1, c_2, ..., c_N) = \frac{p(c_1, c_2, ..., c_N)}{p(c_1)...p(c_N)}$$

$$= \frac{N_1!...N_D!\; p(\nu_1, \nu_2, ..., \nu_D)}{N!\; P_1^{N\nu_1}...P_D^{N\nu_D}} \tag{3.8}$$

$$= \rho(\nu_1, \ldots, \nu_D) \tag{3.9}$$

3.4.2 Estimation of the Context Ratio $\rho(\nu_1, \ldots, \nu_D)$

To avoid encountering astronomical factorial numbers such as $N!$, we approximate Equation 3.8 in the following way.

Since

$$p(\nu_d) = \binom{N}{N_1} P_d^{N_d}(1 - P_d)^{N-N_d}$$

We have

$$p(\nu_1)p(\nu_2)\ldots p(\nu_D) =$$

$$\binom{N}{N_1}\binom{N}{N_2}\cdots\binom{N}{N_D} P_1^{N_1}P_2^{N_2}\dots P_D^{N_D}(1-P^1)^{N-N_1}(1-P^2)^{N-N_2}\dots(1-P^D)^{N-N_D}$$

$$=\binom{N}{N_1}\binom{N}{N_2}\cdots\binom{N}{N_D}(1-P^1)^{N-N_1}(1-P^2)^{N-N_2}\dots(1-P^D)^{N-N_D}p(c_1)p(c_2)\dots p(c_N)$$

since $p(c_1)p(c_2)\dots p(c_N) = P_1^{N_1}P_2^{N_2}\dots P_D^{N_D}$.

Therefore,

$$p(c_1)p(c_2)\dots p(c_N) = \frac{p(\nu_1)p(\nu_2)\dots p(\nu_D)}{\binom{N}{N_1}\binom{N}{N_2}\cdots\binom{N}{N_D}(1-P^1)^{N-N_1}(1-P^2)^{N-N_2}\dots(1-P^D)^{N-N_D}}$$

And since

$$p(c_1,c_2,\dots,c_N) = \frac{p(\nu_1,\nu_2,\dots,\nu_D)N_1!N_2!\dots N_D!}{N!}$$

it follows that

$$\rho(\nu_1,\dots,\nu_D) = \frac{p(c_1,c_2,\dots,c_N)}{p(c_1)p(c_2)\dots p(c_N)}$$

$$=\frac{p(\nu_1,\nu_2,\dots,\nu_D)}{p(\nu_1)p(\nu_2)\dots p(\nu_D)}\frac{N_1!N_2!\dots N_D!}{N!}\binom{N}{N_1}\binom{N}{N_2}\times\dots$$
$$\times\binom{N}{N_D}(1-P^1)^{N-N_1}(1-P^2)^{N-N_2}\dots(1-P^D)^{N-N_D}$$

$$=\frac{p(\nu_1,\nu_2,\dots,\nu_D)}{p(\nu_1)p(\nu_2)\dots p(\nu_D)}\frac{N(N-1)...(N-N_1+1)\dots N(N-1)...(N-N_D+1)}{N!}$$

$$\times(1-P^1)^{N-N_1}(1-P^2)^{N-N_2}\dots(1-P^D)^{N-N_D}$$

Taking logarithms on both sides, and using Stirling's formula

$$N! \approx (2\pi)^{\frac{1}{2}} N^{N+\frac{1}{2}} e^{-N}$$

we get

$$ln\ \rho(\nu_1, \ldots, \nu_D)$$

$$= lnp(\nu_1, \nu_2, \ldots, \nu_D) - \sum_{d=1}^{D} lnP(\nu_d) + \sum_{d=1}^{D} (N - N_D) ln(1 - P_d) +$$

$$\sum_{d=1}^{D} \sum_{k_d=1} N_d(N - N_d + 1) + (N + \frac{1}{2}) lnN - N + \frac{1}{2} ln\pi$$

.

Therefore, instead of maximizing $p(c_1, ..., c_N | \mathbf{x}_1, ..., \mathbf{x}_N)$ we maximize

$$ln\ p(c_1, ..., c_N | \mathbf{x}_1, ..., \mathbf{x}_N)$$

$$= \sum_{i=1}^{N} ln\ p(c_i | \mathbf{x}_i) + ln\ p(\nu_1, \nu_2, \ldots, \nu_D) - \sum_{d=1}^{D} lnP(\nu_d) + \sum_{d=1}^{D} (N - N_D) ln(1 - P_d) +$$

$$\sum_{d=1}^{D} \sum_{k_d=1} N_d(N - N_d + 1)$$

In our case, $D = 14$ and N is typically around 600, and $p(\nu_1, \nu_2, \ldots, \nu_D)$ is a discrete distribution with D^N possible values. The number of examples in the data set we have to estimate the distribution is around 4000. Distribution estimation is an ill-posed problem, and the problem is worsened with limited amount of data. We try to get around this problem by grouping classes into 5 groups and grouping the cell number into 5 ranges, which effectively reduce both D and N to 5 from 14 and 600 respectively. See Table 3.4.

$p(\nu_1, \nu_2, \ldots, \nu_D)$ and $p(\nu_1), p(\nu_2), \ldots, p(\nu_D)$ are estimated by grouping.

class subgroups	range 1	range 2	range 3	range 4	range 5
Neutrophil	$0-20\%$	$20-40\%$	$40-60\%$	$60-80\%$	$\geq 80\%$
Lymphocyte	$0-20\%$	$20-40\%$	$40-60\%$	$60-80\%$	$\geq 80\%$
Mono+eo+baso	$0-10\%$	$10-20\%$	$20-30\%$	$30-40\%$	$\geq 40\%$
Immature non-blast	0	$1-4$	$5-9$	$10-14$	≥ 15
blast	0	$1-3$	$4-7$	$8-11$	≥ 12

Table 3.4: Grouping of classes and counts

3.4.3 Observations and Simplifications

Direct implementation of the proposed algorithm is difficult due to the computational complexity. In the application of WBC identification, simplification is possible. We observed the following: First, we are primarily concerned with one class: blast, the presence of which has clinical significance. Secondly, we only confuse blast with another class lymphocyte. In other words, for a potential blast, $p(\text{blast}|\mathbf{x}) \gg 0$, $p(\text{lymphocyte}|\mathbf{x}) \gg 0$, $p(\text{any other class}|\mathbf{x}) \approx 0$. Finally, we are fairly certain about the classification of all other classes, *i.e.*, $p(\text{a certain class}|\mathbf{x}) \approx 1$, $p(\text{any other class}|\mathbf{x}) \approx 0$. Based on the above observations, we can simplify the algorithm, instead of doing an exhaustive search.

Let $p_i^d = p(c_i = d|\mathbf{x}_i), i = 1, ..., N$. More specifically, let $p_i^B = p(\text{blast}|\mathbf{x}_i)$, $p_i^L = p(\text{lymphocyte}|\mathbf{x}_i)$ and $p_i^* = p(\text{class} * |\mathbf{x}_i)$ where * is neither a blast nor a lymphocyte.

Suppose there are K potential blasts. Order the $p_1^B, p_2^B, ..., p_K^B$'s in a descending manner over i, such that

$$p_1^B \geq p_2^B \geq \cdots \geq p_K^B$$

then the probability that there are k blasts is

$$P_B(k) = p_1^B ... p_k^B p_{k+1}^L ... p_K^L \; p_{K+1}^* ... p_N^* \; \rho(\nu_B = \tfrac{k}{N}, \nu_L = \nu_L^{'} + \tfrac{K-k}{N}, \nu_3, ..., \nu_D)$$

where $\nu_L^{'}$ is the proportion of unambiguous lymphocytes and $\nu_3, ..., \nu_D$ are the proportions of the other cell types.

We can compute the $P_B(k)$'s recursively.

$$P_B(0) = p_1^L ... p_K^L \; p_{K+1}^* ... p_N^* \; \rho(\nu_B = 0, \nu_L = \nu_L^{'} + \frac{K}{N}, \nu_3, ..., \nu_D)$$

$$P_B(k+1) = P_B(k) \frac{p_{k+1}^B \; \rho(\nu_B = \frac{k+1}{N}, \nu_L = \nu_L^{'} + \frac{K-k-1}{N}, \nu_3, ..., \nu_D)}{p_{k+1}^L \; \rho(\nu_B = \frac{k}{N}, \nu_L = \nu_L^{'} + \frac{K-k}{N}, \nu_3, ..., \nu_D)}$$

for k = 1, ..., K-1, and

$$P_B(K) = p_1^B ... p_K^B \; p_{K+1}^* ... p_N^* \; \rho(\nu_B = \frac{K}{N}, \nu_L = \nu_L^{'}, \nu_3, ..., \nu_D)$$

This way we only need to compute K terms to get $P_B(k)$'s . We pick the optimal number of blasts k^* that maximizes $P_B(k), k = 1, ..., K$.

3.4.4 The Algorithm and Complexity

Step 1 Estimate $\rho(\nu_1, ..., \nu_D)$ from the database, for $d = 1, ..., D$.

Step 2 Compute the object-by-object "no context" *a posteriori* probability $p(c_i|\mathbf{x}_i), i = 1, ..., N$, and $c_i \in \{1, ..., D\}$.

Step 3 Compute $P_B(k)$ and find k^* for $k = 1, ..., K$, and relabel the cells accordingly.

We would like to point out that the number of terms to compute and compare drops from D^N to 2^N after simplification, and further to N after ordering.

3.4.5 Results

The algorithm has been intensively tested at IRIS, Inc. on the specimens obtained at Harbor UCLA Medical Center. We compared the performances with and without using contextual information on blood samples from 220 specimens (consisting of 13,200 cells). In about 50% of the cases, a false alarm would have occurred had context not been used. Most cells are correctly classified, but a few are incorrectly labeled as immature cells, which raises a flag for the doctors. Change of the classification of the specimen to abnormal requires expert intervention before the false alarm is eliminated, and it may cause unnecessary expenses and worry. When context is applied, the false alarms for most of the specimens were eliminated, and no false negative was introduced. See Table 3.5. Table 3.6 illustrates how context changes the labeling for a few specimens.

methods	cell classification	normality identification	false positive	false negative
no context	88%	$\sim$ 50%	$\sim$50%	0%
with context	89%	$\sim$ 90%	$\sim$10%	0%

Table 3.5: Comparison of with and without using contextual information

pmn %	lymp %	mono +eo +baso %	num of immature non-blast	num of blast	prob of blast	num of proposed changes	ρ	prod times ρ	num changed from blast to lymp
60.37	29.93	9.35	1	1	0.72	0	0.00	0.00	1
						1	1.80	0.51	
68.07	12.28	18.77	3	2	0.57	0	2.18	0.43	1
					0.34	1	2.18	0.81	
						2	1.33	0.37	
70.58	7.61	13.67	45	2	0.76	0	83.38	35.38	0
					0.55	1	83.38	28.71	
						2	2.57	0.27	
42.96	46.28	8.54	0	8	0.94	0	0.	0.	8
					0.91	1	0.	0.	
					0.89	2	0.	0.	
					0.82	3	0.	0.	
					0.79	4	0.	0.	
					0.76	5	0.	0.	
					0.53	6	0.	0.	
					0.52	7	0.	0.	
					0.51	8	1.83	0.0002	

Table 3.6: Illustration of how context changes the labeling for a few specimens.

Chapter 4 Incorporating Context into Urinalysis

4.1 Introduction

Urine is one of the most complex body fluid specimens: it potentially contains about 60 meaningful elements. Urinalysis is probably the physician's oldest laboratory procedure. Examination of the urine sediment plays a critical role in urinalysis. It detects the presence of elements that often provide early diagnostic information concerning dysfunction, infection, or inflammation of the kidneys and urinary tract. Thus this non-invasive technique can be of great value in clinical case management. Traditional manual microscopic sediment examination is time-consuming, labor-intensive and difficult to standardize. Automated microscopy of all specimens is more practical than manual microscopy because it eliminates variation among different technologists and the variation that becomes more pronounced when the same technologist examines increasing numbers of specimens. Also, it is less labor-intensive and thus less costly than manual microscopy. It also provides more consistent and accurate results.

An automated urinalysis system work station (The Yellow Iris, International Remote Imaging Systems Inc.) has been introduced in numerous clinical laboratories for automated microscopy. Urine samples are processed and examined at 100x (low power field) and 400x magnifications (high power field) with bright-field illumination. Manual microscopic urinalysis systems rely on human operators who read the samples visually and identify them. The Yellow IRIS automated system collects video images of formed elements in a stream of uncentrifuged urine passing an optical assembly. These images are given to a computer algorithm for automatic identification.

The elements found in microscopic urinalysis are casts (including hyaline casts, granular casts and cellular casts), epithelial cells (including renal epithelial cells, transitional epithelial cells and squamous epithelial cells), blood cells (including both white and red blood cells), crystals (include amorphous crystals, uric acid crystal, calcium oxalate crystals and triple phosphate crystals), as well as other elements including bacteria, yeast (including both budding yeast and hyphae yeast), mucus, fat body and spermatozoa. Some of these analytes are pathological. Specimens are considered abnormal when any of the following is found: red blood cell count exceeds 3; white blood cell count exceeds 5; presence of yeast; presence of non-squamous epithelial cells (i.e., presence of renal and transitional epithelial cells); large amount of mucus; any type of casts other than hyaline casts.

Context is rich in urinalysis and plays a crucial role in analyte classification. Some combinations of reasonable analytes are more likely than others. Some analytes go together, some don't. For instance, the presence of bacteria indicates the presence of white blood cells, since bacteria tend to cause infection and thus trigger the production of more white blood cells. Some analytes, such as renal epithelial cells, transitional epithelial cells and white blood cells, can be captured in both low and high power fields. Thus, if they are detected in one power field, they probably exist in another power field as well. Existence of cellular casts (collection of cells) is strongly correlated with those of white and red blood cells. And if amorphous crystals show up, they tend to show up in bunches and in all sizes. Therefore, if there are amorphous crystal look-alikes in various sizes, it is quite possible that they are amorphous crystals. Squamous epithelial cells can appear both flat or rolled up. If squamous epithelial cells in one form are detected, then it is likely that there are squamous epithelial cells in the other form. White blood cell clusters in the low power field usually indicate white blood cells in high power field. Utilizing such context will hopefully improve classification accuracy.

The task of automated microscopic urinalysis is, given a urine specimen that consists of up to a few hundred images of analytes, to classify each analyte into one of the classes. Figure 4.5 to Figure 4.8 are examples of analyte images of various types. Similar to the white blood cell identification task discussed in the previous chapter, the automated urinalysis consists of the following steps: image processing and feature extraction, learning and pattern recognition, and context incorporation. The first two steps are very similar to that of the white blood cell identification, therefore these details will not be discussed. Table 4.8 gives a list of features extracted from analyte images.[1] The classes we are looking at are artifacts, bacteria, calcium oxalate crystals, red blood cells, white blood cells, budding yeast, amorphous crystals, and uric acid crystals. All these analytes are in the high power field.

4.2 Incorporating Partial Context into Urinalysis

The form of context in urinalysis, especially the fact that context is contained in the presence of some types of analytes, makes it well suited for using the partial-context framework discussed in Chapter 2. The partial-context maximum likelihood decision rule uses intermediate-stage context and has the advantage of being computationally efficient. In urinalysis, the intermediate-stage context A is the presence of several relevant classes. The criteria for relevance will be discussed in next section.

Assume $p(x_i|c_i, A) = p(x_i|c_i)$, according to 2.24,

$$p(c_i|x_i, A) = p(c_i|x_i)\frac{p(c_i|A)}{p(c_i)}\frac{p(A)p(x_i)}{p(x_i; A)}$$

$$\propto p(c_i|x_i)\frac{p(c_i|A)}{p(c_i)} \tag{4.1}$$

where $\rho = \frac{p(c_i|A)}{p(c_i)}$ is the context ratio.

The partial-context maximum likelihood (PCML) decision rule chooses class label $\hat{c}_i$ for element i such that

$$\hat{c}_i = \underset{c_i}{\text{argmax}}\, p(c_i|\mathbf{x}_i, A) \tag{4.2}$$

Since the context ratio $\rho = \frac{p(c_i|A)}{p(c_i)}$ is where context is contained, Table 4.1 and

[1] λ_1 and λ_2 are respectively the bigger and the smaller eigenvalues of the second moment matrix of an image.

	art	bact	caox	rbc	wbc	byst	amor	uric
art	1.0378	1.0286	1.0103	1.0069	1.0400	1.0392	0.4047	1.0364
bact	0.6699	1.5816	0.9588	1.1276	1.1800	1.0784	0.3458	1.0545
caox	0.8564	1.2463	1.9691	1.1468	0.9900	1.0000	0.5049	0.7636
rbc	0.7387	1.2913	0.8247	1.9081	1.2150	0.8824	0.4440	0.8182
wbc	0.9626	1.1327	0.7629	0.8368	1.9800	1.3137	0.3694	1.2182
byst	0.8699	1.2374	1.1443	1.1509	1.0750	2.2353	0.4479	0.5455
amor	0.8166	1.1786	1.1031	1.0988	0.9700	0.9804	1.6090	0.5091
uric	0.8614	1.2389	1.1443	1.1632	1.0250	1.0196	0.5108	1.7818

Table 4.1: Context ratio $\frac{p(ci=d|A_j=1)}{P(c_i=d)}$

	art	bact	caox	rbc	wbc	byst	amor	uric
art	0	0.0777	0.0052	0.0609	0.0006	0	0.8541	0.0014
bact	0.7253	0	0.0135	0.1160	0.0046	0.0054	0.1285	0.0067
caox	0.5017	0.3197	0	0.0940	0.0213	0.0055	0.0522	0.0056
rbc	0.5742	0.2936	0.0150	0	0.0071	0.0073	0.0941	0.0088
wbc	0.5306	0.2806	0.0120	0.0851	0	0.0035	0.0838	0.0044
byst	0.5052	0.3253	0.0097	0.0794	0.0196	0	0.0546	0.0061
amor	0.5362	0.3391	0.0103	0.0807	0.0220	0.0057	0	0.0062
uric	0.5104	0.3269	0.0097	0.0744	0.0204	0.0053	0.0529	0

Table 4.2: Context ratio $\frac{p(ci=d|A_j=0)}{P(c_i=d)}$

Table 4.2 give the context ratios for present and absent classes respectively. Figure 4.1 is a figurative plot of the ratios for both cases. In both tables, the $(i,j)^{th}$ entry is $\frac{p(c=j|A_i)}{p(c=j)}$. If $\frac{p(c=j|A_i)}{p(c=j)} > 1$, it implies that A_i, the presence or absence of class i, enhances the likelihood of class j. If $\frac{p(c=j|A_i)}{p(c=j)} < 1$, it implies that A_i inhibits the likelihood of class j.

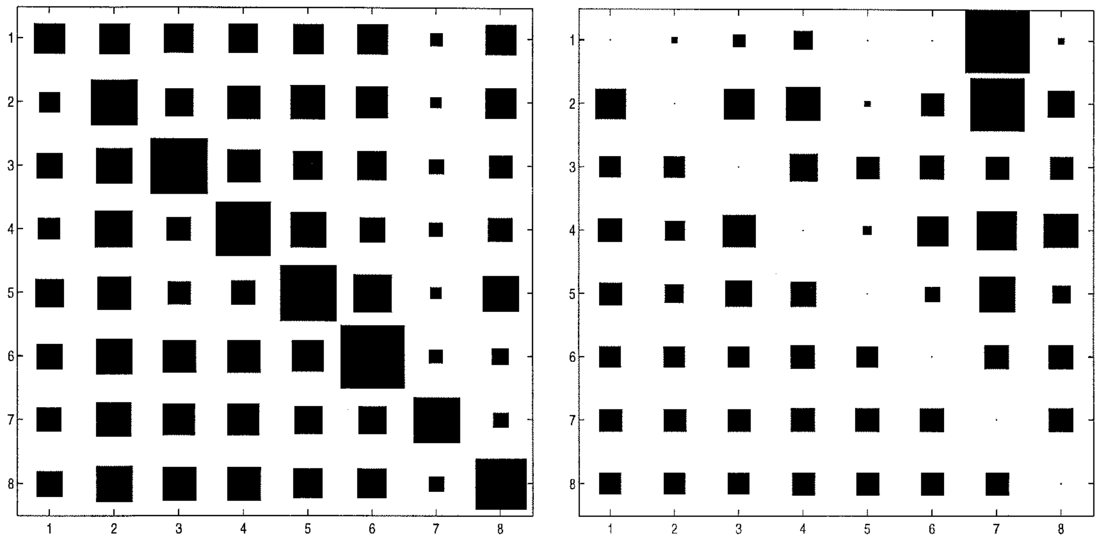


Figure 4.1: Context ratios when the classes are present and absent.

4.3 Identification of Relevant Classes

Not all classes are relevant in terms of carrying contextual information. We propose three criteria based on which we can systematically investigate the relevance of the class presence. To use these criteria, we need to know the following distributions: the joint distribution of the presence of all classes $p(A_1, A_2, \ldots, A_D)$; the class prior distribution $p(c)$ for $c = 1, \ldots, D$; the conditional class distribution $p(c|A_d = 0)$ and $p(c|A_d = 1)$ for $c = 1, \ldots, D$ and $d = 1, \ldots, D$; and the class presence prior distribution $p(A_d)$ for $A_d \in 0, 1$ and $d = 1, \ldots, D$. Table 4.4 gives the conditional class distribution given the presence of all the classes. The $(i, j)^{th}$ entry in the matrix is $p(c = j|\text{class i is present})$. Table 4.5 gives the conditional class distribution given the absence of all the classes. The $(i, j)^{th}$ entry in the matrix is $p(c = j|\text{class } i \text{ is absent})$. Table 4.6 lists the class presence prior distribution $P(A_d = 1)$ and the class prior distribution $P(c)$.

The first criterion is the correlation coefficient between the presence of any two classes; the second one is the classical mutual information $I(c; A_d)$ between the presence of a class A_d and the class probability $p(c)$, where $I(c; A_d)$ is defined as

$$I(c; A_d) = H(c) - H(c|A_d)$$

where $H(c) = \sum_{i=1}^{D} p(c = i)ln(p(c = i))$ is the entropy of the class priors and $H(c|A_d) = P(A_d = 1)H(c|A_d = 1) + P(A_d = 0)H(c|A_d = 0)$ is the conditional entropy of c conditioned on A_D, and the third one is what we call the *expected relative entropy* $D(c||A_d)$ between the presence of a class A_d and the labeling probability $p(c)$, which we define as

$$D(c||A_d) = P(A_d = 1)D(p(c)||p(c|A_d = 1)) + P(A_d = 0)D(p(c)||p(c|A_d = 0))$$

where

$$D(p(c)||p(c|A_d = 1)) = \sum_{i=1}^{D} p(c = i|A_d = 1)ln(\frac{p(c = i|A_d = 1)}{p(c = i)})$$

and

$$D(p(c)||p(c|A_d = 0)) = \sum_{i=1}^{D} p(c = i|A_d = 0)ln(\frac{p(c = i|A_d = 0)}{p(c = i)})$$

Table 4.3 is the correlation coefficient matrix of the class presence. According to the first criterion, one type of analyte is considered relevant to another if the absolute value of their correlation coefficient is beyond a certain threshold. The graph in Figure 4.2 illustrates the relevance between any two analyte types according to various thresholds. In this figure, two types are related or relevant to each other only if their nodes are connected by a line. The solid lines correspond to threshold 0.25 and the added dotted lines to 0.10. Not surprisingly, lowering the threshold leads to more relevant classes. It shows that uric acid crystals, budding yeast and calcium oxalate crystals are not relevant to any other types even by a generous threshold of 0.10. Table 4.7 lists mutual information $I(c; A_d)$ and expected relative entropy $D(c||A_d)$, both in unit of nats. The bigger the mutual information between the presence of a class and the class distribution, the more relevant this class is. Ranking the analyte types in terms of $I(c; A_d)$ in a descending manner gives rise to the following list: bacteria, amorphous crystals, artifact, red blood cells, white blood cells, uric acid crystals,

	art	bact	caox	rbc	wbc	byst	amor	uric
art	1.0000	-0.0128	0.0706	0.0721	0.1244	0.0674	-0.5121	0.0385
bact	-0.0128	1.0000	0.0952	0.1588	0.2664	0.1139	-0.2708	-0.0847
caox	0.0706	0.0952	1.0000	0.0529	0.1010	-0.0218	0.0470	-0.0567
rbc	0.0721	0.1588	0.0529	1.0000	0.2786	-0.1195	-0.1773	-0.0310
wbc	0.1244	0.2664	0.1010	0.2786	1.0000	0.1090	-0.1866	-0.0469
byst	0.0674	0.1139	-0.0218	-0.1195	0.1090	1.0000	-0.0789	-0.0207
amor	-0.5121	-0.2708	0.0470	-0.1773	-0.1866	-0.0789	1.0000	-0.0450
uric	0.0385	-0.0847	-0.0567	-0.0310	-0.0469	-0.0207	-0.0450	1.0000

Table 4.3: Correlation coefficient matrix of presence of eight classes.

	art	bact	caox	rbc	wbc	byst	amor	uric
art	0.5303	0.3341	0.0098	0.0734	0.0208	0.0053	0.0206	0.0057
bact	0.3423	0.5137	0.0093	0.0822	0.0236	0.0055	0.0176	0.0058
caox	0.4376	0.4048	0.0191	0.0836	0.0198	0.0051	0.0257	0.0042
rbc	0.3775	0.4194	0.0080	0.1391	0.0243	0.0045	0.0226	0.0045
wbc	0.4919	0.3679	0.0074	0.0610	0.0396	0.0067	0.0188	0.0067
byst	0.4445	0.4019	0.0111	0.0839	0.0215	0.0114	0.0228	0.0030
amor	0.4173	0.3828	0.0107	0.0801	0.0194	0.0050	0.0819	0.0028
uric	0.4402	0.4024	0.0111	0.0848	0.0205	0.0052	0.0260	0.0098

Table 4.4: $P(c|A_d = 1)$

budding yeast and calcium oxalate crystals. The relevance level decreases in the list. Similarly, ranking the analyte types in terms of $D(c||A_d)$ in a descending manner gives rise to the following list: bacteria, artifact, red blood cells, amorphous crystals, white blood cells, calcium oxalate crystals, budding yeast and uric acid crystals. Thresholding correlation coefficient explores the pairwise relevance of classes, whereas mutual information and expected relative entropy indicate the general relevance of a class to all other classes in an expectation sense. All three criteria lead to similar conclusions regarding the relevance of all classes.

	art	bact	caox	rbc	wbc	byst	amor	uric
art	0	0.0777	0.0052	0.0609	0.0006	0	0.8541	0.0014
bact	0.7253	0	0.0135	0.1160	0.0046	0.0054	0.1285	0.0067
caox	0.5017	0.3197	0	0.0940	0.0213	0.0055	0.0522	0.0056
rbc	0.5742	0.2936	0.0150	0	0.0071	0.0073	0.0941	0.0088
wbc	0.5306	0.2806	0.0120	0.0851	0	0.0035	0.0838	0.0044
byst	0.5052	0.3253	0.0097	0.0794	0.0196	0	0.0546	0.0061
amor	0.5362	0.3391	0.0103	0.0807	0.0220	0.0057	0	0.0062
uric	0.5104	0.3269	0.0097	0.0744	0.0204	0.0053	0.0529	0

Table 4.5: $P(c|A_d = 0)$

class	$p(A_d = 0)$	$p(c)$
art	0.9646	0.5110
bact	0.5788	0.3248
caox	0.2295	0.0097
rbc	0.5690	0.0729
wbc	0.4115	0.0200
byst	0.0696	0.0051
amor	0.0488	0.0509
uric	0.0305	0.0055

Table 4.6: The prior distribution of the presence of all classes, and the class prior distribution of all classes.

A_d	$I(c; A_d)$	$D(c\|\|A_D)$
art	0.0938	0.0925
bact	0.1735	0.2338
caox	0.0016	0.0160
rbc	0.0431	0.0812
wbc	0.0281	0.0307
byst	0.0063	0.0071
amor	0.1231	0.0513
uric	0.0191	0.0062

Table 4.7: Mutual information $I(c; A_d)$ and the expected relative entropy between the presence of a class A_d and the labeling probability $p(c)$.

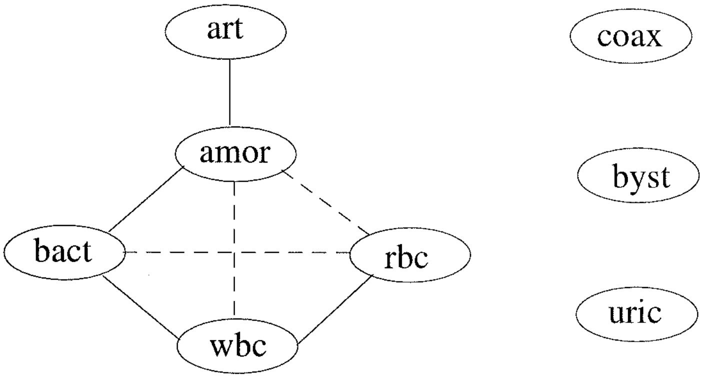


Figure 4.2: Relevant classes

4.4 The Algorithm

Once we identify the M relevant classes, we use the following algorithm to incorporate partial context.

Step 0 Estimate the priors $p(c|A_d)$ and $p(c)$, for $c \in 1, 2, \ldots, D$ and $d \in 1, 2, \ldots, D$.

Step 1 For a given x_i, compute $p(c_i|x_i)$ for $c_i = 1, 2, \ldots, D$.

Step 2 Let the M relevant classes be $R_1, \ldots, R_M$. According to the no-context $p(c_i|x_i)$ and certain criteria for detecting the presence or absence of all the relevant classes, get $A_{R_1}, \ldots, A_{R_M}$.

Step 3 Let $p(c_i|x_i, A_0) = p(c_i|x_i)$, where A_0 is the null element. Then, for $m = 1$ to M, iteratively compute

$$\begin{aligned} & p(c_i|x_i; \underline{A_0, \ldots, A_{R_{m-1}}, A_{R_m}}) \\ &= p(c_i|x_i, \underline{A_0, \ldots, A_{R_{m-1}}}) \frac{p(c_i|A_{R_m})p(A_{R_m})}{p(c)} \end{aligned}$$

Step 4 Repeat step 3 until the algorithm converges.

Step 5 Label the objects according to the final context-containing $p(c_i|\mathbf{x}_i, \underline{A_{R_1}, \ldots, A_{R_M}})$, *i.e.*,

$$\hat{c}_i = \underset{c_i}{\operatorname{argmax}}\, p(c_i|\mathbf{x}_i, \underline{A_{R_1}, \ldots, A_{R_M}})$$

for $i = 1, \ldots, N$.

This algorithm is invariant with respect to the ordering of the M relevant classes in $(A_1, \ldots, A_M)$. A sketch of the proof follows:

Proof:

Denote the final *a posteriori* probability of class label given the feature vector after taking into account all context contained in M relevant classes as $p(c_i|x_i; \underline{A_1, \ldots, A_M})$.

According to the above algorithm (step 3), what the algorithm effectively does is

$$\begin{aligned}
&p(c_i|x_i; \underline{A_{R_1}, \ldots, A_{R_M}}) \\
&= p(c_i|x_i, A_0) \prod_{m=1}^{M} \frac{p(c_i|A_{R_m})p(A_{R_m})}{p(c_i)p(x_i, A_{R_m})} \\
&= p(c_i|x_i) \prod_{m=1}^{M} \frac{p(c_i|A_{R_m})}{p(c_i)} \frac{\prod_{m=1}^{M} p(A_{R_m})}{\prod_{m=1}^{M} p(x_i, A_{R_m})} \\
&\propto p(c_i|x_i) \prod_{m=1}^{M} \frac{p(c_i|A_{R_m})}{p(c_i)}
\end{aligned}$$

Since $p(c_i|A_{R_m})$ are multiplied, therefore, the order does not matter.

Q.E.D.

4.5 Results

The algorithm using partial context was tested on a database with 83 urine specimens that contains 20276 analytes total. Four classes are considered relevant according to the criteria described in section 4.3. The four classes are: bacteria, red blood cells, white blood cells and amorphous crystals. The criteria for the presence of these classes in each specimen are: bacteria exceeds 15% of the total amount of analytes in a specimen, the number of red blood cell exceeds 5, the number of white blood cell exceeds 5, and the number of amorphous crystals exceeds 100. We measure two types

Table 4.8: Features extracted from urine anylates images

feature number	feature description
1	area
2	length of edge
3	$\frac{\text{square root of area}}{\text{length of edge}}$
4	$\frac{\text{standard deviation}}{\text{mean}}$ of distance from center to edge
5	$\frac{\lambda_1}{\lambda_2}$
6	$\frac{\text{sum of length of two longest straight edges}}{\text{total length of edge}}$
7	$\frac{\text{sum of length of four longest straight edges}}{\text{total length of edge}}$
8	$\frac{\text{sum of length of two longest semi-straight edges}}{\text{total length of edge}}$
9	$\frac{\text{sum of length of four longest semi-straight edges}}{\text{total length of edge}}$
10	the mean of red distribution
11	the mean of blue distribution
12	the mean of green distribution
13	15^{th} percentile of gray level histogram
14	85^{th} percentile of gray level histogram
15	the standard deviation of gray level intensity
16	energy of the Laplacian transformation of grey level image

of error: element-by-element error, and specimen diagnosis error. The element-by-element confusion matrices are list in Table 4.9 and Table 4.10 for without context and with context respectively. The average element-by-element error is 44.48% without context, and is 36.66% with context (see Table 4.11.) The diagnosis for a specimen is either normal or abnormal. Table 4.12 and Table 4.13 compare the diagnosis performance of with and without context. We can see that context helps to increase correct diagnosis for both normal and abnormal specimens, and to reduce both false positive and false negative.

	art	bact	caox	rbc	wbc	byst	amor	uric
art	6304	368	10	122	149	0	705	1
bact	6053	1116	0	4	4	0	391	0
caox	30	4	9	34	2	0	95	1
rbc	58	11	3	301	64	0	220	0
wbc	34	0	2	7	158	0	35	0
byst	45	0	0	5	0	0	86	0
amor	1959	173	3	23	12	0	1653	2
uric	1	0	10	5	0	0	4	0

Table 4.9: Confusion matrix generated by context-free maximum likelihood decision rule

	art	bact	caox	rbc	wbc	byst	amor	uric
art	5148	1671	0	194	272	0	372	2
bact	2965	4409	0	7	13	0	174	0
caox	24	33	5	63	15	0	33	2
rbc	34	64	1	392	128	0	38	0
wbc	18	7	0	5	199	0	7	0
byst	11	84	0	20	0	0	21	0
amor	1583	469	5	56	16	0	1694	2
uric	0	0	8	9	0	0	3	0

Table 4.10: Confusion matrix generated by partial-context maximum likelihood decision rule.

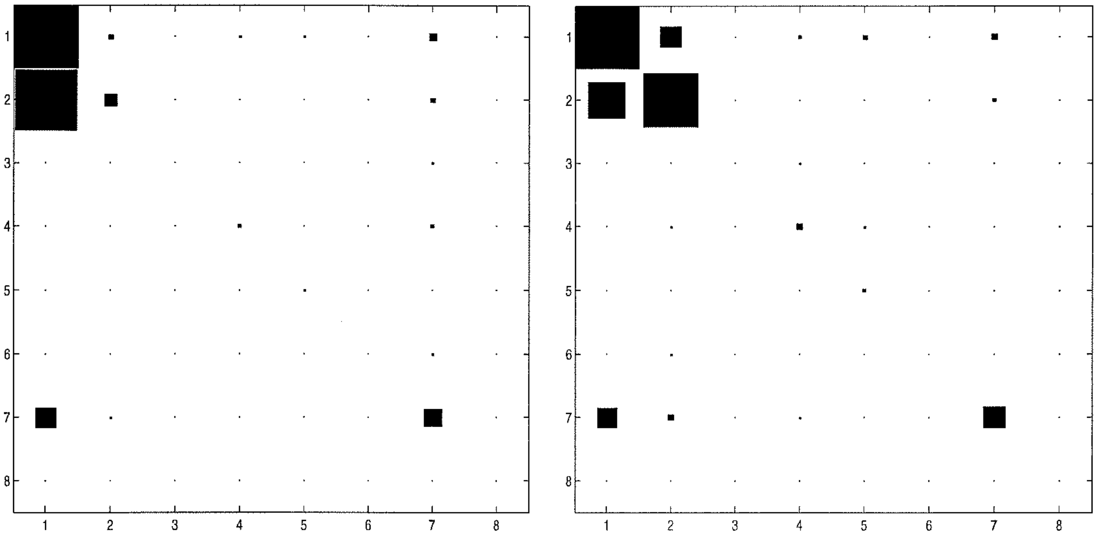


Figure 4.3: Confusion matrix without and with context in terms of total numbers

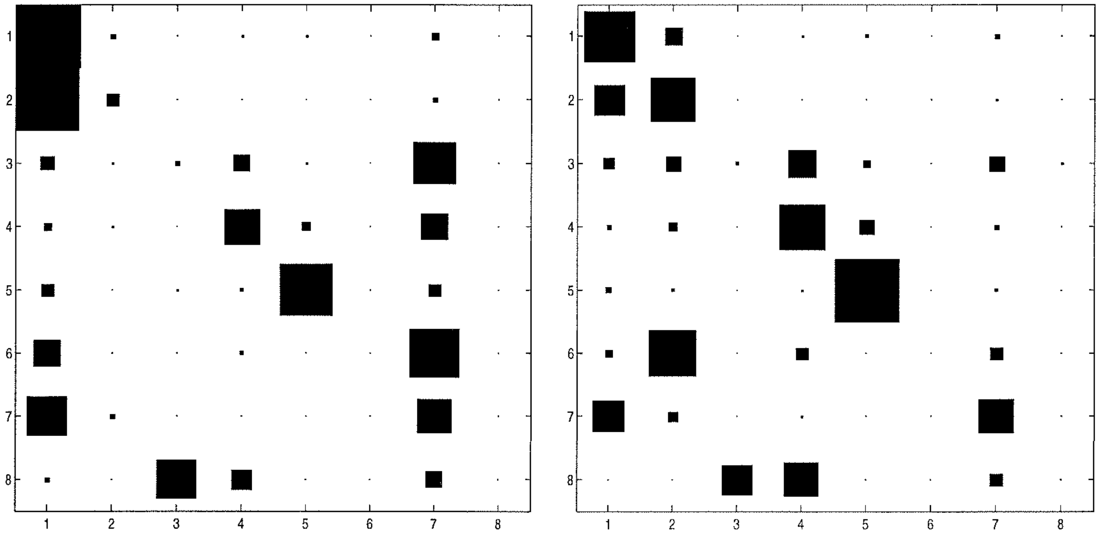


Figure 4.4: Confusion matrix without and with context in terms of percentages

	without context	with context
average element-by-element error	44.48 %	36.66 %

Table 4.11: Comparison of with and without using contextual information for element-by-element error.

	estimated normal	estimated abnormal
truly normal	40.96 %	7.23 %
truly abnormal	19.28 %	32.53 %

Table 4.12: Diagnosis confusion matrix without context

	estimated normal	estimated abnormal
truly normal	42.17 %	6.02 %
truly abnormal	16.87 %	34.94 %

Table 4.13: Diagnosis confusion matrix with context

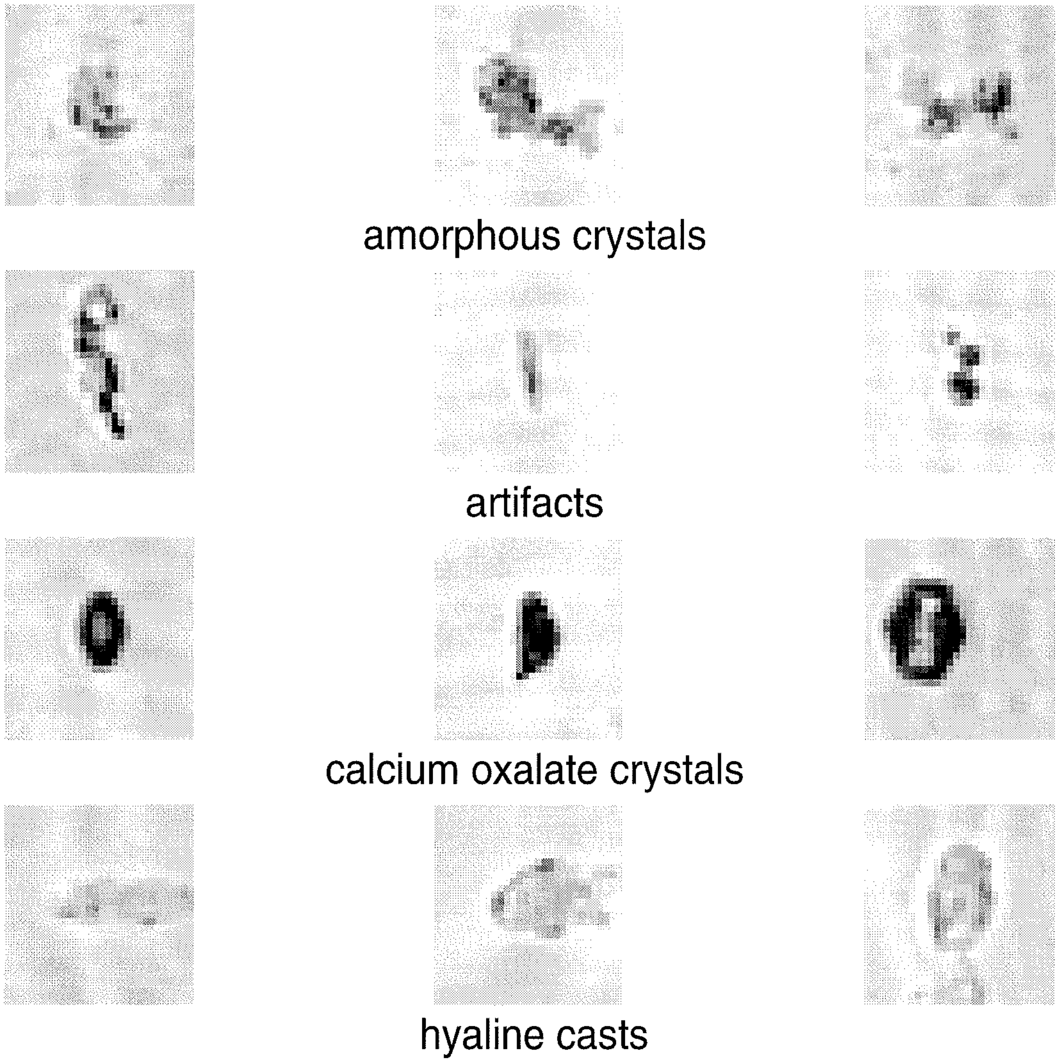
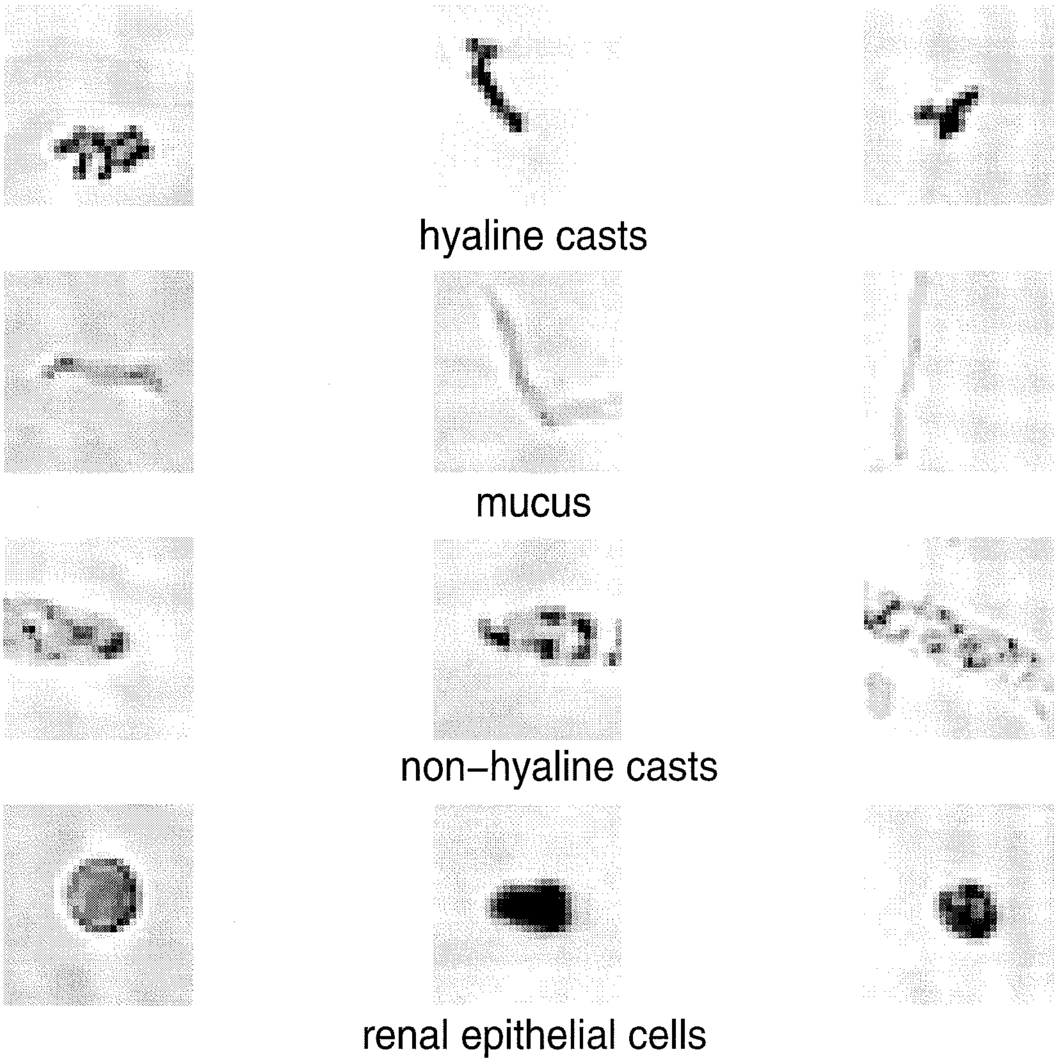


Figure 4.5: Analyte images



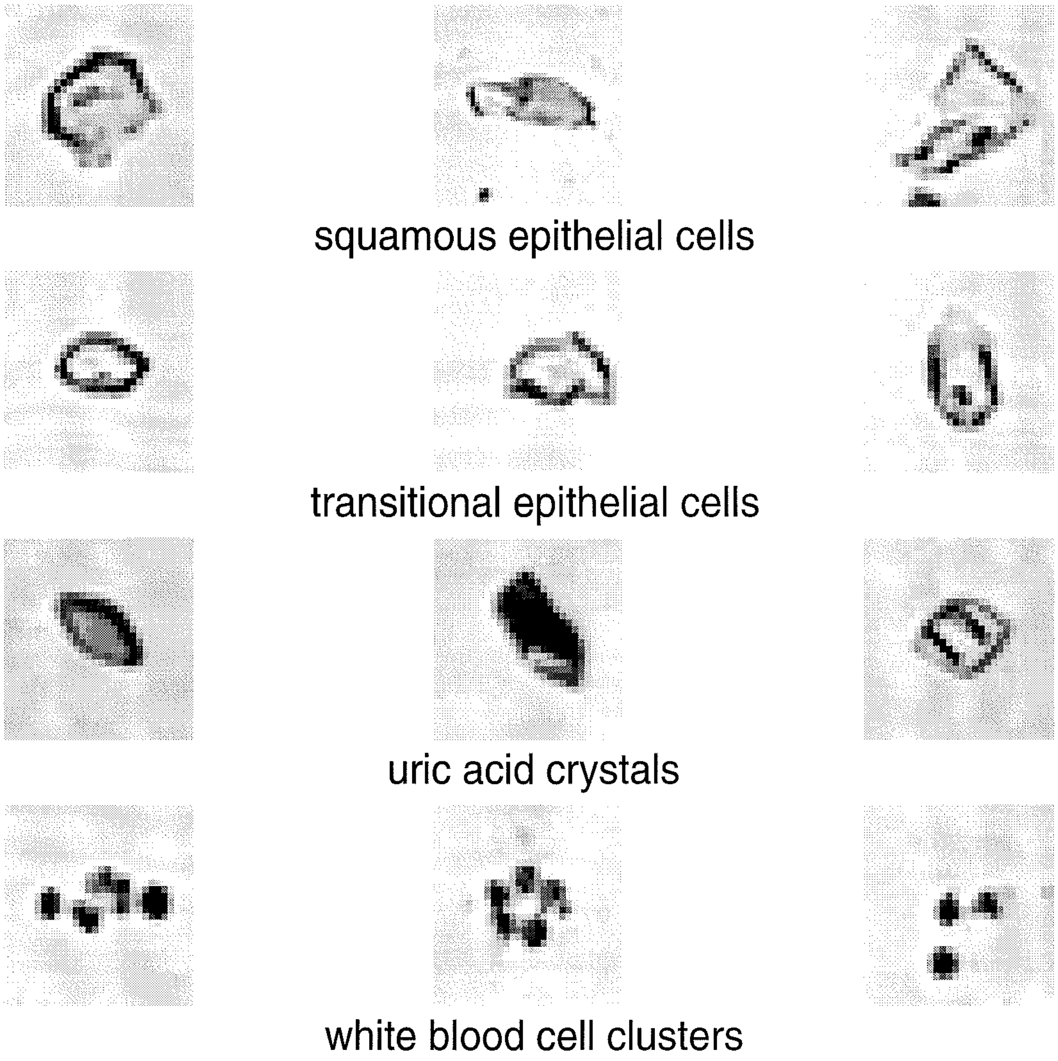
hyaline casts

mucus

non−hyaline casts

renal epithelial cells

Figure 4.6: Analyte images



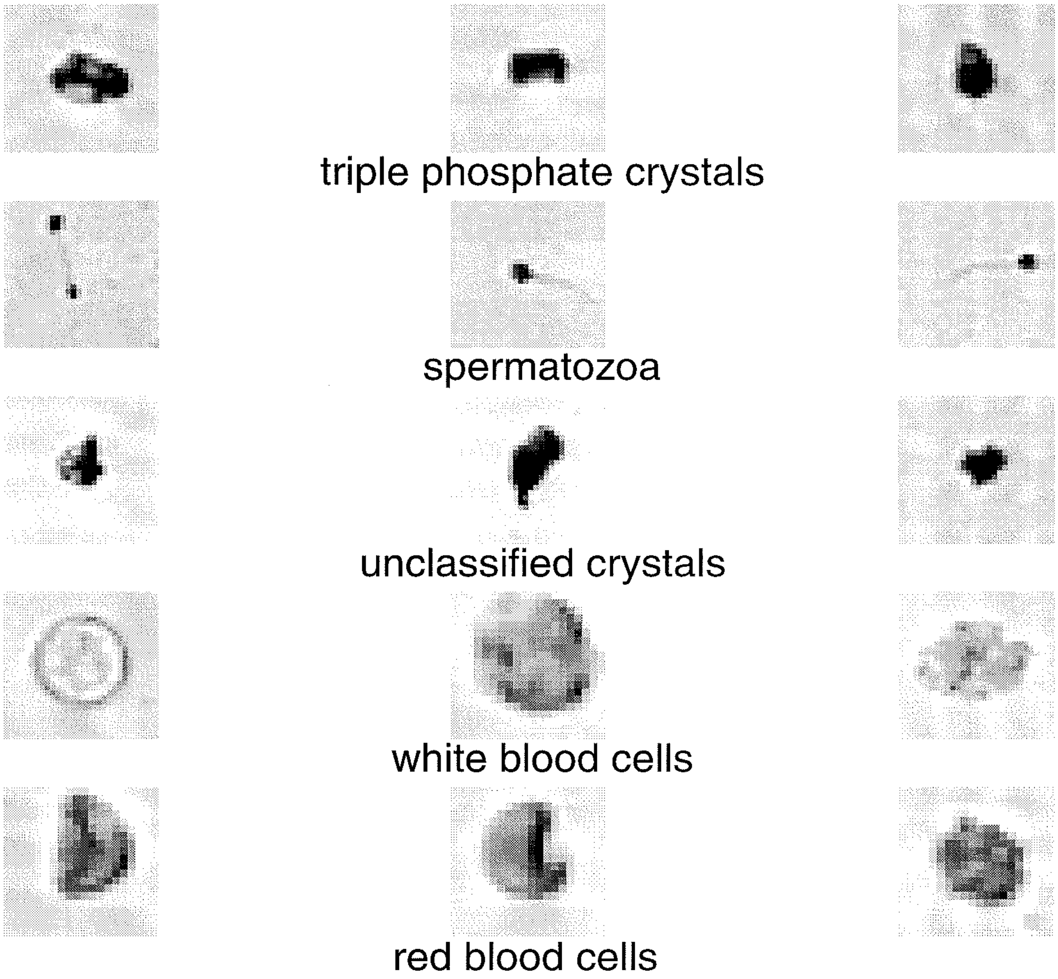
squamous epithelial cells

transitional epithelial cells

uric acid crystals

white blood cell clusters

Figure 4.7: Analyte images



triple phosphate crystals

spermatozoa

unclassified crystals

white blood cells

red blood cells

Figure 4.8: Analyte images

Part II

The Bin Model for Learning and Generalization

Chapter 5 The Bin Model for Learning and Generalization

5.1 Introduction

Part II of this thesis investigates the issue of generalization in the paradigm of *learning from examples.* Learning is a general concept that describes the process of formulating a hypothesis from a finite set of data. Data is presented as a set of instances drawn from an underlying process, and one attempts to determine the underlying process from the data. We hereby describe the set up for a learning process, which is illustrated in Figure 5.1.

We are presented with a data set $\mathcal{D}_N$, the *training set*, which consist of N input-output pairs $\{x_i, y_i\}_{i=1}^{N}$. The input-output pairs are generated according to an *unknown* underlying function $f : \mathbf{R}^d \rightarrow \{0, 1\}$, which we call the *target function.* Each $x_i \in \mathbf{R}^d$ is drawn according to a governing distribution $\Omega(X)$. The training set is our sole knowledge pertaining to the target function, and our goal is to infer the target function based on the training set. Learning entails choosing a hypothesis function $g : \mathbf{R}^d \rightarrow \{0, 1\}$ from a collection of candidate functions $\mathcal{H}$ as our inference of the target function. The set $\mathcal{H}$ is called the *learning model* because it reflects how we choose to model the target function. The hypothesis function is chosen by a *learning algorithm*, $\mathcal{A}$. The learning algorithm takes as inputs the training set $\mathcal{D}_N$ and the learning model $\mathcal{H}$, outputs a hypothesis $g \in \mathcal{H}$, usually based on some performance criterion on the training set. For example, a typical learning algorithm might be to choose from the learning model the hypothesis which minimizes an error measure on the training set.

The "goodness of fit" of a hypothesis is measured by how close it is to the target function averaged over the input space. Error occurs when a hypothesis deviates

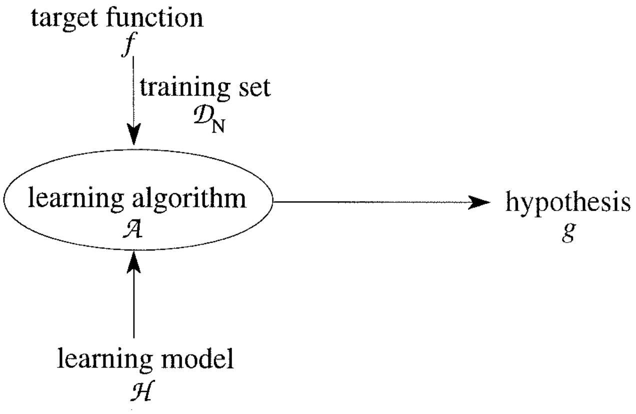


Figure 5.1: Illustration of a learning system

from the target function on any point in the input space. The error a hypothesis function commits on the training set is usually referred to as "in-sample error," or the "training error," and the error on points out of the training set "out-of-sample error," or "test error." A good hypothesis should give low out-of-sample error. A "clever" learning algorithm might be able to find a hypothesis that fits the training set well and achieve small in-sample error. However, in-sample error often has no bearing on the out-of-sample error. Typically, two possible scenarios occur in a learning process as illustrated in Figure 5.2. In the first scenario, the out-of-sample error decreases as the in-sample error decreases, all the way until the in-sample error reaches its minimum. In this case, it is good for a learning algorithm to find a hypothesis that achieves the minimum in-sample error, since the corresponding out-of-sample error is also minimal. In the second scenario, the out-of-sample error decreases at the beginning together with the in-sample error as the learning algorithm is getting a grasp of the general properties of the target function contained in the training set. Then at a certain point, the out-of-sample error starts to rise as the in-sample error is further reduced. The learning algorithm is finding hypotheses that fit the training data better, but that also fit any idiosyncrasies in the training set. As a result, the hypothesis approximates the training data set well, but fails to generalize to out-of-sample data. This phenomenon is usually referred to as "over-fitting." Over-fitting occurs when the learning model is overly complex. It is especially prominent when the training set is noisy.

A question naturally arises: When can we be confident that our model has truly learned and not simply memorized the examples it is given? How much does the in-sample error tell us about the out-of-sample error? Does the in-sample performance generalize to unseen data points? This is the issue of *generalization.* It is an important issue when learning from examples because the training error is the only quantity we have access to, and the whole merit of learning lies in the hope that training error, to some extent, provides some information about the test error.

In general, both the target function and the learning model are nonlinear, and the training set is noisy, thus making technical analysis difficult. It would be nice if we can find a general framework that, for arbitrary target functions, arbitrary learning models, arbitrary input distributions and noisy data sets, can estimate the out-of-sample performance in terms of the in-sample performance, characterize the conditions for over-fitting, quantify the effect of noise on generalization, and eventually, lead to rules for selecting learning models and learning algorithms that can achieve better generalization performance. This is the motivation for our proposition and design of the Bin Model. The Bin Model is a general yet manageable mathematical framework from which we extract the essential property $p(\pi)$ of a learning process that combines the properties of learning model, input distribution and target function. Given some knowledge of $p(\pi)$, a closed form relationship between in- and out-of-sample performance can be derived. In this framework, we can handle the general nonlinear and noisy cases of learning. Important issues in learning, such as generalization, over-fitting, the role of model complexity and the impact of noise, can be addressed using the Bin Model.

We describe the set up of the Bin Model framework in section 5.2. In section 5.3, we derive a closed form relationship between in- and out-of-sample error using the Bin Model. Section 5.4 provides the condition that characterizes over-fitting. The Bin Model result is compared with other paradigms addressing generalization issues in section 5.5. The Bin Model captures fundamental properties of a learning process

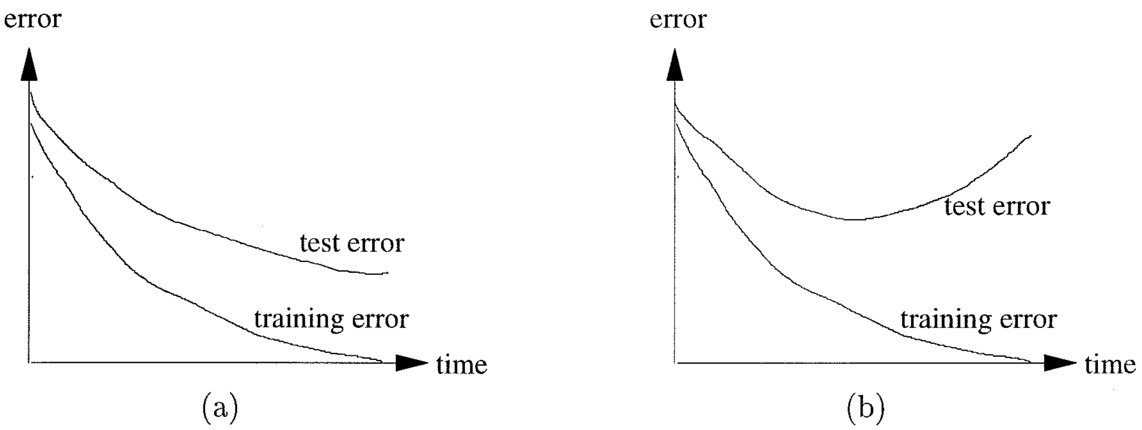


Figure 5.2: Two learning scenarios: over-fitting occurs in the second one.

and is not sensitive to certain details of the target function and the learning model. Section 5.6 describes the relevant features of a learning process that characterize its Bin Model. Section 5.8 studies the impact of noise on generalization, and section 5.9 generalizes the bin model to regression problems, and provides the condition that characterizes over-fitting for regression problems.

5.2 The Bin Model

We consider a target function of the form $f : X \rightarrow \{0, 1\}$. A hypothesis $g : X \rightarrow \{0, 1\}$ is characterized by a parameter π, which is the probability that $g(x)$ disagrees with $f(x)$ on a point x randomly chosen from the input space X according to the possibly unknown input distribution $\Omega(X)$.

$$\pi \triangleq Pr[g \neq f] = \int [g(x) - f(x)]^2 \Omega(x) dx \tag{5.1}$$

The "goodness of fit" for a hypothesis g is measured by its π. The smaller π is, the closer g is to f. If $\pi = 0$, then g is exactly f. The worst case scenario is that in which $\pi = 1$ which implies $g(x) \neq f(x)$ for all x. In general, $0 \leq \pi \leq 1$.

A more visual way to look at a single hypothesis function is to see it as a "bin" that contains red and green marbles (hence the name "Bin Model"), with red marbles representing input points on which the hypothesis disagrees with the target, and green marbles representing those on which they agree. Then the fraction of red marbles in the bin is π. When we draw an example x from the input space and compare $f(x)$ with $g(x)$, we are virtually drawing a marble from the bin and checking its color. A sample can be treated as a Bernoulli trial with probability π of error (being a red marble) and probability $1 - \pi$ of success (being a green marble). When we have a training set with N examples, we have N i.i.d. Bernoulli trials. We denote by ν the fraction of red marbles (frequency of error) in the sample. Therefore, ν is the in-sample error of hypothesis g for this training set, and π is its out-of-sample error. The Law of Large Numbers tells us that for a given bin, as $N \rightarrow \infty$, the random variable ν approaches its mean π. That is to say, the frequency of error will be a good estimate of the probability of error for large sample size N.

So far we have established that one hypothesis is mapped to a bin parameterized by π. When we have a learning model with a *set* of hypothesis functions, we have an *array* of bins (Figure 5.3). Each hypothesis has its own error probability π. The array of $\pi_1, \pi_2, ..., \pi_M$, where $M = |\mathcal{H}|$ is the number of functions in the learning model $\mathcal{H}$, leads to a π-distribution $p(\pi)$. (Note that two different hypotheses can have the same π.) The π-distribution indicates the suitability of the learning model for approximating the target function. The smaller the probability of error π and the more the functions that have small π, the better suited this learning model is for the approximation of f.

Figure 5.4 (a) is an example of π-distribution in which we have some "best" functions with error probability around 0.05, some "worst" ones around 0.95, and a lot of mediocre ones that lie in between. (The π's are ordered in an increasing way for illustrative purposes.)

It is worth noticing that the π-distribution and M are all the information that we need to specify a bin model. We do not need the detailed information about

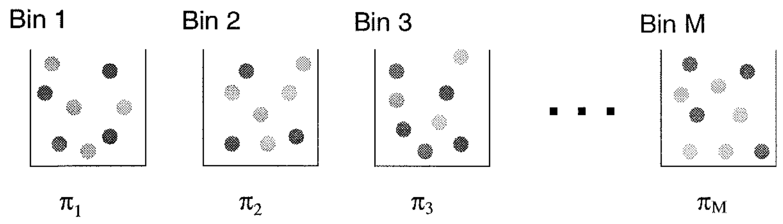
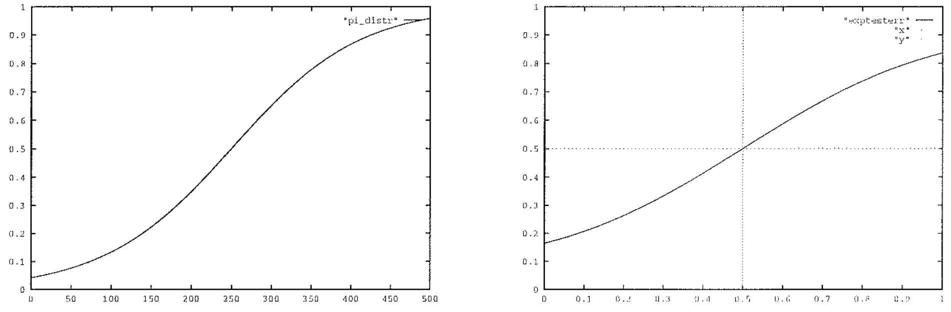


Figure 5.3: The Bin Model: We have a set of M bins, each containing red and green marbles. The frequency of red marbles in bin i is denoted by π_i.



(a)

(b)

Figure 5.4: (a)An example π-distribution for $M = 500$ (note that the x-axis indicates bin index). In this case the hypothesis that best approximates the target function makes approximately 5% error. (b) The expected test error as a function of training error ν for the π-distribution in (a).

the specific forms the target and the hypothesis functions assume. Nor do we need to know anything about the inputs and input distribution. The bin model abstracts relevant quantities from the target function and the learning model. It can be applied to arbitrary, nonlinear target function and arbitrary, nonlinear learning model. This makes it possible to model learning processes in a very simple way without loss of generality.

5.3 Generalization Using Bin Model

We define the expected out-of-sample error given the in-sample error as the measure of generalization, whose analytic expression can be derived from the bin model.

The learning game of interest is the following. We have a learning model and a training set of size N. One hypothesis is randomly picked from the learning model. Its in-sample error is measured, which is equivalent to drawing a sample of N marbles from the bin this hypothesis corresponds to. An error frequency is obtained. If it matches some predetermined level of training error ν, we keep this function as our approximation to the target function f, otherwise it is put back. This reminds us of what we do in learning – we stop at a certain training error, and keep the function that achieves this training error as our hypothesis. We repeat this experiment, and keep all the hypotheses that have in-sample error ν. We refer to the set of hypotheses with error frequency ν on the training set as $\mathcal{H}_\nu$. From $\mathcal{H}_\nu$ one hypothesis is randomly chosen as the final hypothesis. We would like to know the expected out-of-sample error of the final hypothesis, that is $E[\pi|\nu]$.[1]

The set $\{\pi_i\}$ induces a probability distribution p on π. Randomly choosing a hypothesis function from the learning model $\mathcal{H}$ gives us a hypothesis with probability of error which can be assumed to be randomly chosen from $p(\pi)$. For a given hypothesis with error probability π, the probability for an in-sample error ν is the binomial distribution,

[1] We use $E[\cdot]$ to denote expectations.

$$p(\nu|\pi) = \binom{N}{N\nu} \pi^{N\nu}(1-\pi)^{N(1-\nu)}$$

Therefore, it follows that the expected out-of-sample error given in-sample error ν is

$$\begin{aligned}
\pi(\nu) \triangleq E[\pi \mid \nu] &= \int_0^1 \pi p(\pi \mid \nu) d\pi && (5.2)\\
&= \frac{\int_0^1 \pi p(\pi, \nu) d\pi}{\int_0^1 p(\pi, \nu) d\pi} && (5.3)\\
&= \frac{\int_0^1 \pi p(\pi) \pi^{N\nu}(1-\pi)^{N(1-\nu)} d\pi}{\int_0^1 p(\pi) \pi^{N\nu}(1-\pi)^{N(1-\nu)} d\pi} && (5.4)
\end{aligned}$$

where

$$p(\pi, \nu) = p(\pi) \binom{N}{N\nu} \pi^{N\nu}(1-\pi)^{N(1-\nu)}$$

We refer to the relationship between $\pi(\nu)$ and ν in 5.4 as the *generalization curve.* A perfect generalization curve should be $\pi(\nu) = \nu$, which implies that in-sample error ν is a perfect indication of out-of-sample error in expectation. It is not the absolute value of ν and $\pi(\nu)$, but the deviation between them that characterizes the generalization behavior of a learning process. The further away a generalization curve is away from $\pi(\nu) = \nu$ at any point, the worse the generalization is at that point. It is important to point out that no assumption is made about the dependence among the hypotheses. The statistical dependence among the hypotheses in the learning model does not enter the π-distribution.

Figure 5.4 (b) is the generalization curve based on the π-distribution given in 5.4 (a). According to this model, when the training error ν is zero, the expected out-of-sample error is actually 0.17, indicating this sample error is an optimistic estimation of the corresponding out-of-sample error.

Based on the Bin Model, we can also derive the variance of out-of-sample error given in-sample error.

$$\begin{aligned} var(\pi|\nu) &\triangleq E[\pi - E[\pi|\nu]]^2 \\ &= \int_0^1 (\pi - E(\pi|\nu))^2 p(\pi|\nu) d\pi \\ &= \int_0^1 \pi^2 p(\pi|\nu) d\pi - (\int_0^1 \pi p(\pi|\nu) d\pi)^2 \\ &= E[\pi^2|\nu] - (E[\pi|\nu])^2 \end{aligned}$$

Since

$$p(\pi|\nu) = \frac{p(\nu|\pi)p(\pi)}{p(\nu)}$$

$$p(\nu) = \int_0^1 p(\nu|\pi)p(\pi)d\pi$$

and

$$p(\nu|\pi) = \binom{N}{N\nu} \pi^{N\nu}(1-\pi)^{N(1-\nu)}$$

Therefore,

$$var(\pi|\nu) = \frac{\int_0^1 \pi^{N\nu+2}(1-\pi)^{N(1-\nu)}p(\pi)d\pi}{\int_0^1 \pi^{N\nu}(1-\pi)^{N(1-\nu)}p(\pi)d\pi} - (\frac{\int_0^1 \pi^{N\nu+1}(1-\pi)^{N(1-\nu)}p(\pi)d\pi}{\int_0^1 \pi^{N\nu}(1-\pi)^{N(1-\nu)}p(\pi)d\pi})^2.$$

The variance measures the consistency of error probability of hypotheses whose in-sample error is ν.

For some $p(\pi)$ we can get both $\pi(\nu)$ and $var(\pi|\nu)$ in explicit form.

Special Case: For the π-distribution $p(\pi) = (d+1)\pi^d$, for any $d \in \mathbf{Z}^+$, the generalization curve $\pi(\nu)$ is

$$\pi(\nu) = \frac{N\nu + d + 1}{N + d + 2}, \tag{5.5}$$

and its variance is

$$var(\pi|\nu) = \frac{(N\nu + d + 1)(N - N\nu + 1)}{(N + d + 2)^2(N + d + 3)}. \tag{5.6}$$

The proof is provided in appendix A. When $d = 0$, $p(\pi) = 1$ is a uniform distribution, which implies that we have no bias or preference over the distribution of π. If there is only one training example, *i.e.*, $N = 1$, then $\pi(\nu) = \frac{1}{3}\nu + \frac{1}{3}$, and the variance is $var(\pi|\nu) = \frac{(\nu+1)(2-\nu)}{36}$. If there are infinitely many training examples, *i.e.*, $N = \infty$, then $\pi(\nu) = \nu$ and $var(\pi|\nu) = 0$, which means the expected test error is consistently the same as the training error and thus the generalization is perfect. This is expected when we have a large training set.

5.4 Unbiased Optimization

As mentioned earlier, when attempting to learn a function from a finite data set by minimizing the in-sample error, it is not uncommon to observe over-fitting. We argue that over-fitting is an artifact induced by the learning algorithm. During the "learning period" (*i.e.*, when out-of-sample error is decreasing together with the in-sample error) in Figure 5.2 (b), the learning algorithm is looking for hypotheses that better fit the general properties of the target function . During the "over-fitting period" (*i.e.*, when out-of-sample error increases as in-sample error continues to decrease), the learning algorithm is looking for hypotheses that are so complex and powerful that they can fit the data almost perfectly yet are far away from the target function.

In the Bin Model framework, however, the over-fitting phenomenon does not appear. It can be shown that $\pi(\nu)$ is monotonically increasing with respect to ν, *i.e.*, $\frac{d\pi(\nu)}{d\nu} > 0$ for all ν. This implies that smaller in-sample error always corresponds to smaller out-of-sample error in expectation. Thus there will be no over-fitting and we can always benefit from getting smaller in-sample error. However, this does not contradict our observation of over-fitting. Instead, it clarifies the condition under which over-fitting does not occur – the condition of *unbiased optimization.* Unbiased optimization is a learning algorithm that chooses all hypotheses with the same in-sample error with equal probability. In other words, it has no preference for some hypotheses over others as long as they achieve the same in-sample error. Fundamental to the Bin Model analysis is the random selection of a hypothesis from the resulting $\mathcal{H}_\nu$ (we assumed that all hypotheses with the same in-sample error ν are chosen with equal probability from $\mathcal{H}$.) The choice is not subject to a controlled tour of parameter space. The learning game we set up in section 5.3 and the resulting derivation of $\pi(\nu)$ utilize an unbiased optimization learning algorithm.

Theorem : Under unbiased optimization, $\frac{d\pi(\nu)}{d\nu} > 0$ for any ν.
Proof: See Appendix B.

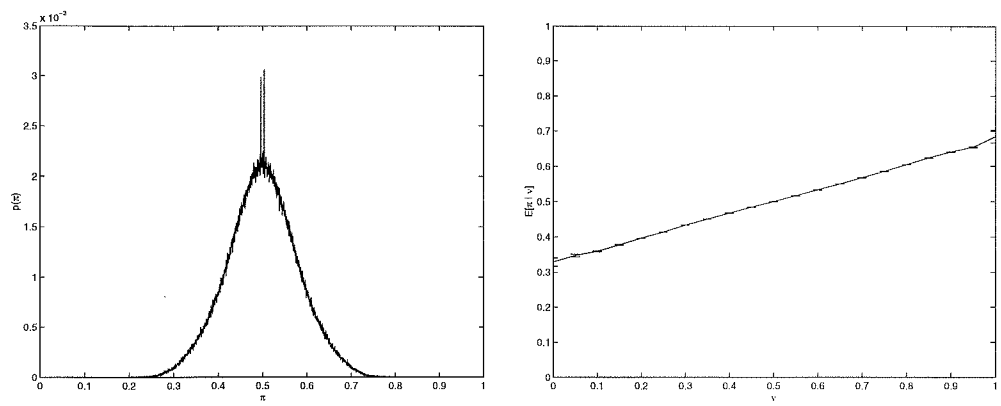
Notice that this theorem holds true for arbitrary π-distribution $p(\pi)$.

Figure 5.5 is an example of the generalization behavior under unbiased optimization. A histogram approximating $p(\pi)$ is shown, along with the empirical generalization function $\pi(\nu)$. $p(\pi)$ is estimated by randomly sampling hypotheses from the learning model and estimating π by the test error on a large data set. We can see that the expected generalization is poor for such a small data set (even for $\nu = 0$ we can only expect $\pi \sim 0.3$), and that $\pi(\nu)$ is indeed monotonic – we obtain the best out-of-sample error by finding the minimum training error (over-fitting is not observed).

With the idea that over-fitting is usually accompanied by using overly complex models, it is important to point out that no explicit measure of the model complexity appears in the calculation of $\pi(\nu)$. The generalization ability is completely captured by $p(\pi)$, which may be independent of traditional complexity measures.

More importantly, it points out that model complexity in and of itself is not the cause for over-fitting. Under unbiased optimization, model complexity does not lead to over-fitting. Here we demonstrate it with an example illustrated in Figure 5.6 and 5.7. Figure 5.6 illustrates this for a learning model of two-layer neural networks with sigmoidal hidden units.[2] Target functions were chosen randomly from the model and

[2] In this case the output function of the network was used as a decision boundary for a binary



(a)

(b)

Figure 5.5: (a)The π distribution for a noisy learning problem. The target function is $f(\mathbf{x}) = sign(sin(\frac{x_1}{2})sin(\frac{x_2}{4}))$ and the learning model is 2-10-1 neural network with hyperbolic tangent transfer function and linear output. The training set is of size 20. The noise in the training set is realized by randomly flipping the sign of the output with probability 0.1. The input distribution is uniform in [-2 π, 2 π]. (b) The expected test error as a function of training error ν for the π distribution in (a) when using unbiased optimization.

trained on a data set with $N = 50$ examples. In comparison, the number of hidden units in the model was varied from less than 10 to 1234 (the number of parameters in the model was varied from less than 50 to more than 5000). When the model was trained using gradient descent to a fixed level of training error, the out-of-sample error remained relatively unchanged as the number of hidden units increased.

The independence of (5.4) on model complexity implies that, given a target function f, all models with π-distributions of the same shape will exhibit the same generalization behavior. If we can vary the model size without changing $p(\pi)$, then we can separate the phenomenon of over-fitting from conventional complexity measures.

Figure 5.7 shows an estimate of the π-distribution [3] for three of the models used in the experiments of of Figure 5.6. The distribution changes only slightly as the model complexity increases. Given the preceding discussion, then, the results in Figure 5.6 are not surprising.

One discrepancy between the theory and experiment lies in the method of selecting

classification problem.

[3] Based on 100000 random samples of ν on 100 data points.

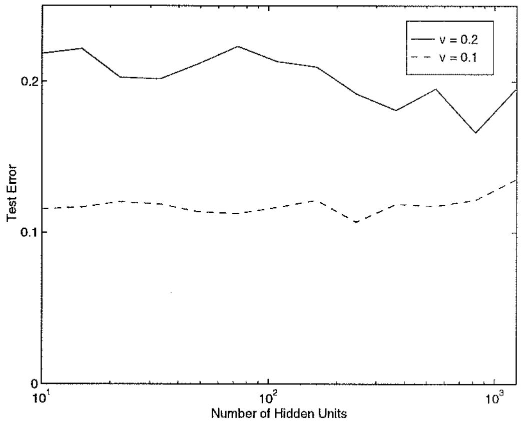


Figure 5.6: Generalization error for a 2-layer neural network as the number of hidden units varies.

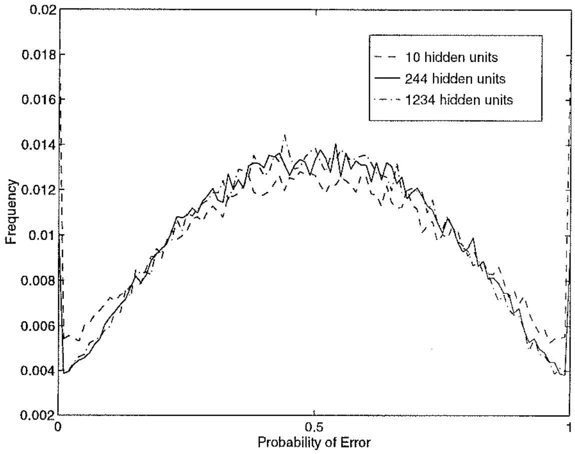


Figure 5.7: Histogram of the empirical π distributions for experiments of Figure 5.6

candidate bins. For the result of (5.4) we have assumed that bins are chosen randomly. Training with gradient descent restricts our choice of bins to a small subset at each step. In the experiments presented here, averaging over random targets and starting points seems to have compensated for any differences that might have arisen. In general, however, fitting an arbitrary learning algorithm into the framework of the Bin Model is an interesting and open problem.

5.5 Other Paradigms

Many attempts have been made to assess the generalization performance of a learning process. These include the Prediction Error ([Akaike, 1970], [Moody, 1992]), VC analysis ([Vapnik and Chervonenkis, 1971], [Abu-Mostafa, 1989]), and the Exhaustive Learning paradigm ([Solla, 1992], [Schwartz *et al.*, 1990]).

The Prediction Error approach takes a general form of out-of-sample error that consists of the sum of two terms:

$$\text{out-of-sample error} \approx \text{in-sample error} + \frac{2C}{N}\sigma^2 \tag{5.7}$$

where C embeds model complexity and therefore the second term can be viewed as a penalty for model complexity. σ^2 is the variance of noise in the data. This criterion has the nice property of formulating the impact of model complexity and data noise on the gap between in- and out-of-sample error. The prediction error criterion is an asymptotic result, and it requires knowing σ^2 which is almost never known. For the nonlinear case [Moody, 1992], it is assumed that the input density is discrete with support only at input points in the training set which is a crude estimation and thus limits the accuracy of this criterion.

Some insights into generalization are gained by bounding the deviation between training and test error in the worst-case scenario. This is formulated by VC-theory, which gives a bound of the form

$$Prob\ [\ \sup_{g\in\mathcal{H}} |\pi_g - \nu_g| < \epsilon] < \delta$$

where ϵ is a tolerance level for the deviation between in-sample error ν and the out-of-sample error π for a hypothesis. This criterion highlights the impact of number of training examples N and model complexity (both appearing in δ) on generalization. This approach provides a bound, not a definite relationship. For many cases the bound turns out to be a trivial bound, *i.e.*, δ is larger than 1. Also, the confidence level δ is a function of the "growth function" [Abu-Mostafa, 1989] the calculation of which is formidable, if not impossible, for most learning models. This seriously limits the practical use of VC analysis.

[Solla, 1992] and [Schwartz *et al.*, 1990] suggested a framework with a somewhat similar formulation as our Bin Model, which leads to the following results:

$$G_N = \int_0^1 g\rho_N(g)dg \tag{5.8}$$

$$= \frac{\int_0^1 g^{N+1}\rho_0(g)dg}{\int_0^1 g^N\rho_0(g)dg}$$

where G_N is the 'mean generalization ability', effectively our $E(\pi)$; g is equivalent to our definition of π, and $\rho_0(g)$ is $p(\pi)$ in the Bin Model. $\rho_N(g)$ corresponds to $p(\pi|\nu = 0)$ in the Bin Model setting In other words, Equation 5.8 addresses only the expected test error given zero training error, namely $E(\pi|\nu = 0)$, instead of establishing a mapping between $E[\pi|\nu]$ for any possible ν.

5.6 Characteristics of the π-distribution

It is very interesting to notice that the generalization behavior of a learning process, characterized by $\pi(\nu)$, is fully determined by its π-distribution $p(\pi)$. Two learning processes may have different learning models (and hence different model complexities), target functions and input distributions, but as long as they have the same π-

distribution, they will have exactly the same generalization in terms of expected test error given training error. This is striking because it implies that the π-distribution of a learning process abstracts and fully contains what is important as for as generalization is concerned.

Figure 5.8 gives some examples of $p(\pi)$ and their corresponding generalization curves $\pi(\nu)$. It seems that $\pi(\nu)$ is not sensitive to the detailed shape of $p(\pi)$, as long as there exists a "decent" amount of functions with π of all possible values. This raises the question of what features of $p(\pi)$ are most relevant to generalization. There are two ways to characterize relevant features of a π-distribution. One is the moments of $p(\pi)$, and the other is the geometry of $p(\pi)$ in terms of its derivatives.

Theorem: The generalization curve $\pi(\nu)$ of a learning process is a function of the first N moments of its π-distribution $p(\pi)$.

Proof:

Since,

$$(1-\pi)^{N-N\nu} = \sum_{k=0}^{N-N\nu} (-1)^{N-N\nu-k} \pi^{N-N\nu-k}$$

Then,

$$\pi(\nu) = \frac{\int_0^1 \pi \pi^{N\nu}(1-\pi)^{N(1-\nu)} p(\pi) d\pi}{\int_0^1 \pi^{N\nu}(1-\pi)^{N(1-\nu)} p(\pi) d\pi}$$

$$= \frac{\sum_{k=0}^{N-N\nu}(-1)^{N-N\nu-k} \int_0^1 \pi^{N-k+1} p(\pi) d\pi}{\sum_{k=0}^{N-N\nu}(-1)^{N-N\nu-k} \int_0^1 \pi^{N-k} p(\pi) d\pi}$$

$$= \frac{\sum_{k=0}^{N-N\nu}(-1)^{N-N\nu-k} \mu_{N-k+1}}{\sum_{k=0}^{N-N\nu}(-1)^{N-N\nu-k} \mu_{N-k}}$$

where

$\mu_k = \int_0^1 \pi^k p(\pi) d\pi$ is the k^{th}-order moments of $p(\pi)$ distribution. **Q.E.D.**

So, as far as generalization is concerned, we don't need to know the full details of $p(\pi)$, only its first $N+1$ moments, which fully determine the generalization curve. Two

π-distributions with the same first $N+1$ moments and different higher order moments will have exactly the same generalization. However, for two learning processes to have the same variance $var(\pi|\nu)$, they need to have the same $N+2$ moments.

5.7 Positive Learning

According to 5.4, for any given in-sample error ν and π-distribution $p(\pi)$, the generalization curve $\pi(\nu)$ is a function of the number of training examples N. We denote $\pi(\nu)$ as $\pi(\nu, N)$. We would like to study the effect the number of training examples has on generalization. It is plausible to think that the bigger N, the better a learning system generalizes, since we have more information provided by the training set about the underlying target function. To formalize this, we introduce what we call *positive learning* curve, which is defined as follows:

$$PL(N,\nu) \triangleq \frac{|\pi(N+1,\nu)-\nu|}{|\pi(N,\nu)-\nu|} \tag{5.9}$$

Definition: A learning process is *positive* if more training examples brings the expected out-of-sample error closer to in-sample error, formally speaking, if $\frac{|\pi(N+1,\nu)-\nu|}{|\pi(N,\nu)-\nu|} \leq 1$.

Theorem: A learning process characterized by π-distribution $p(\pi) = (d+1)\pi^d$ is positive.

Proof:

For a learning process characterized by $p(\pi) = (d+1)\pi^d$, according to the derivation provided in Appendix A, its generalization curve $\pi(\nu, N)$ is

$$\pi(\nu,N) = \frac{N\nu+d+1}{N+d+2} \tag{5.10}$$

Therefore,

$$PL(N,\nu) = \frac{|\pi(N+1,\nu)-\nu|}{|\pi(N,\nu)-\nu|}$$

$$= \frac{N+d+2}{N+d+3}$$

$$\leq 1$$

Q.E.D.

5.8 Noise

When we draw a sample from a bin and obtain error frequency ν, it generally deviates from the mean π. We define the discrepancy between ν and π as *noise*. At a closer look, we find that noise comes from two sources. One is *Bernoulli noise*, which is the discrepancy between Bernoulli frequency ν and the mean π due to finite sample size. The other is *data noise* which corresponds to the physical noise in real world. We formalize the latter source of noise by flipping the samples with certain probability ε. It is equivalent to putting the sample through a binary symmetric channel(BSC) with cross probability ε (Figure 5.9). We are not able to tell the noise source when we draw a sample and observe discrepancy. However, they are intrinsically different. Bernoulli noise can be overcome by taking large sample size, whereas data noise can not. Let $\nu_{Bernoulli}$ denote the error frequency effected only by Bernoulli noise, and ν_{BSC} be that effected also by data noise. It can be shown that for Bernoulli noise,

$$E(\nu_{Bernoulli}) = \pi \tag{5.11}$$

$$Var(\nu_{Bernoulli}) = \frac{\pi(1-\pi)}{N} \tag{5.12}$$

whereas for data noise,

$$E(\nu_{BSC}) = (1-2\varepsilon)\pi + \varepsilon \tag{5.13}$$

$$Var(\nu_{BSC}) = \frac{(1-2\varepsilon)^2\pi(1-\pi) + \varepsilon(1-\varepsilon)}{N} \tag{5.14}$$

It can also be shown that introducing data noise with noise level ε is equivalent to replacing the original error probability π by a diluted version $(1-2\varepsilon)\pi+\varepsilon$ (replacing π in (5.11) and (5.12) with $(1-2\varepsilon)\pi+\varepsilon$ leads to (5.13) and (5.14).) Introducing data noise has the effect of pushing the original π-distribution closer to a random function with $\pi = 0.5$ which has equal chance of agreeing and disagreeing with the target function, as we can see from Figure 5.8. Taking large sample size reduces $Var(\nu_{BSC})$ to zero, and makes ν close to $E(\nu_{BSC}) = (1-2\varepsilon)\pi+\varepsilon$. However, there is still a deviation of $\varepsilon - 2\varepsilon\pi$ from $E(\nu_{BSC})$ to the real mean π. This leads us to the important conclusion that data noise gives rise to intrinsic generalization error (the deviation of test error from training error), which can not be overcome by taking large sample size.

5.9 Analog Error

So far we have considered classification problem where the output of a target function takes binary values. We would like to generalize the Bin Model framework to functions with analog outputs. We consider target functions of the form $f : X \rightarrow \mathbf{R}$, and hypothesis functions $g : X \rightarrow \mathbf{R}$.

The deviation of hypothesis $g(x)$ from the target $f(x)$ gives rise to the error function $e(x)$. Squared error $e(x) = (f(x) - g(x))^2$ is often used, though in principle $e(x)$ can be any measure of error. For a given target $f(x)$ and a given hypothesis $g(x)$, the error function $e(x)$ and the distribution $p(x)$ induces a distribution $p(e)$ whose mean and variance are as follows:

$$\pi = \int_{-\infty}^{\infty} e(x)p(x)dx \tag{5.15}$$

$$= \int_{0}^{\infty} ep(e)de \tag{5.16}$$

and

$$\sigma^2 = \int_{-\infty}^{\infty} (e(x) - \pi)^2 p(x) dx \tag{5.17}$$

$$= \int_{0}^{\infty} (e - \pi)^2 p(e) de \tag{5.18}$$

Note that $0 \leq \pi \leq \infty$.

When a single example is drawn from input space X, its error has the distribution of $p(e)$. When we have a sample of size N, the in-sample error is

$$\nu = \frac{1}{N} \sum_{i=1}^{N} e_i$$

where the e_i are i.i.d. from $p(e)$. We would like to know the distribution of ν for a hypothesis g. In general, for a finite N, we can not get $p(\nu)$ unless we know the exact form of $p(e)$. However, for large N, the central limit theorem can be used to estimate $p(\nu)$ for a given hypothesis. Since $p(e)$ has mean π and variance σ^2, then according to the central limit theorem, the limit distribution of $p(\nu)$ for a given hypothesis is Gaussian with mean π and variance $\frac{\sigma^2}{N}$.

$$p(\nu|g) \sim \mathcal{N}(\pi, \frac{\sigma^2}{N})$$
$$= \frac{1}{\sqrt{\frac{2\pi\sigma^2}{N}}} exp(-\frac{(\nu - \pi)^2}{\frac{2\sigma^2}{N}})$$

Analogous to the binary case, let's assume a hypothesis is parameterized by its expected error π. Therefore,

$$p(\nu|\pi) = p(\nu|g)$$
$$= \frac{1}{\sqrt{\frac{2\pi\sigma_2}{N}}} exp(-\frac{(\nu - \pi)^2}{\frac{2\sigma_2}{N}})$$

A set of hypotheses induces a π-distribution, similar to the binary case. To get the closed form of generalization, the learning game of interest is the same as the binary case. We have a learning model and a training set with N examples. One hypothesis is randomly chosen from the learning model, and its in-sample error is measured. An error frequency is obtained. If it falls into some predetermined interval of training error $[\nu - \delta, \nu + \delta]$, we keep this function as our approximation to the target function f, otherwise it is put back. Among the hypotheses that achieve the same interval of training error, a hypothesis is randomly chosen as the final hypothesis. We would like to know the expected out-of-sample error of the final hypothesis given in-sample error ν, formalized by $\pi(\nu) \triangleq E[\pi|\nu]$ as follows:

$$\begin{aligned}
\pi(\nu) \triangleq E[\pi \mid \nu] &= \int_0^\infty \pi p(\pi \mid \nu) d\pi && (5.19)\\
&= \frac{\int_0^\infty \pi p(\pi, \nu) d\pi}{\int_0^\infty p(\pi, \nu) d\pi} && (5.20)\\
&= \frac{\int_0^\infty \pi p(\nu \mid \pi) p(\pi) d\pi}{\int_0^\infty p(\nu \mid \pi) p(\pi) d\pi} && (5.21)\\
&= \frac{\int_0^\infty \frac{\pi}{\sigma} exp(-\frac{N}{2\sigma^2}(\nu - \pi)^2) p(\pi) d\pi}{\int_0^\infty \frac{1}{\sigma} exp(-\frac{N}{2\sigma^2}(\nu - \pi)^2) p(\pi) d\pi} && (5.22)\\
& && (5.23)
\end{aligned}$$

In the binary case the $p(e)$ for any hypothesis is a Bernoulli distribution with mean π. Under unbiased optimization, $\pi(\nu)$ is monotonically increasing in terms of ν for arbitrary ν and arbitrary $p(\pi)$, which always favors small training error for better average out-of-sample error. However, the monotonicity property does not in general hold for analog error.

Assume for any hypothesis the error variance is a function of the mean, *i.e.*, $\sigma = \sigma(\pi)$. This is true for a wide variety of distributions. Also assume that the error function $e(x)$ is bounded for any x and thus can be normalized to $[0, 1]$. This normalization leads to $0 \leq \nu, \pi \leq 1$. We then have the following theorem.

Theorem A sufficient condition for $\frac{d\pi(\nu)}{d\nu} > 0$ for arbitrary ν is:

$$\begin{cases} \pi_2 > \pi_1 \quad and \quad \sigma_2 < \sigma_1 \implies \frac{\sigma_1^2}{\sigma_2^2} < \frac{1-\pi_1}{1-\pi_2} \\ \pi_2 > \pi_1 \quad and \quad \sigma_2 > \sigma_1 \implies \frac{\sigma_1^2}{\sigma_2^2} > \frac{\pi_1}{\pi_2} \end{cases} \tag{5.24}$$

Proof:

Let

$$\pi(\nu) = \frac{\int_0^\infty \frac{\pi}{\sigma} exp(-\frac{N}{2\sigma^2}(\nu-\pi)^2)p(\pi)d\pi}{\int_0^\infty \frac{1}{\sigma} exp(-\frac{N}{2\sigma^2}(\nu-\pi)^2)p(\pi)d\pi} \tag{5.25}$$

$$= \frac{A(\nu)}{B(\nu)}$$

where

$$A(\nu) = \int_0^\infty \frac{\pi}{\sigma} exp(-\frac{N}{2\sigma^2}(\nu-\pi)^2)p(\pi)d\pi$$

and

$$B(\nu) = \int_0^\infty \frac{1}{\sigma} exp(-\frac{N}{2\sigma^2}(\nu-\pi)^2)p(\pi)d\pi$$

Then

$$\frac{d\pi(\nu)}{d\nu} =$$

$$\frac{N}{2B^2(\nu)} \int_0^\infty \int_0^\infty \frac{(\pi_2-\pi_1)}{\sigma_1\sigma_2}[(\frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2})\nu + \frac{\pi_2}{\sigma_2^2} - \frac{\pi_1}{\sigma_1^2}]exp[-\frac{N}{2\sigma_1^2}(\nu-\pi_1)^2 - \frac{N}{2\sigma_2^2}(\nu-\pi_2)^2]p(\pi_1)p(\pi_2)d\pi_1 d\pi_2$$

All the terms in the product in the integral are positive except

$$(\pi_2 - \pi_1)[(\frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2})\nu + \frac{\pi_2}{\sigma_2^2} - \frac{\pi_1}{\sigma_1^2} \tag{5.26}$$

It is sufficient for $\frac{d\pi(\nu)}{d\nu} \geq 0$ for any ν that the quantity in 5.26 is positive for any ν. We discuss all the possible scenarios.

Suppose $\pi_2 - \pi_1 > 0$.

Case 1: $\frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2} < 0$. If we want $(\pi_2 - \pi_1)[(\frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2})\nu + \frac{\pi_2}{\sigma_2^2} - \frac{\pi_1}{\sigma_1^2}] \geq 0$ for all ν, then the worst case is when $\nu = 1$. Consider the worst case, we need to have

$$[(\frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2}) + \frac{\pi_2}{\sigma_2^2} - \frac{\pi_1}{\sigma_1^2}] \geq 0$$

which implies

$$\frac{\sigma_1^2}{\sigma_2^2} < \frac{1-\pi_1}{1-\pi_2}$$

Case 2: $\frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2} > 0$. If we want $(\pi_2 - \pi_1)[(\frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2})\nu + \frac{\pi_2}{\sigma_2^2} - \frac{\pi_1}{\sigma_1^2}] \geq 0$ for all ν, then the worst case is when $\nu = 0$. Consider the worst case, we need to have

$$\frac{\pi_2}{\sigma_2^2} - \frac{\pi_1}{\sigma_1^2} > 0$$

which implies

$$\frac{\sigma_1^2}{\sigma_2^2} < \frac{\pi_1}{\pi_2}$$

Since π_1 and π_2 are interchangeable in the integral, there is no need to discuss the case $\pi_1 - \pi_2 \geq 0$.

Q.E.D.

This condition can be written in a different way,

$$\text{If } \ \pi_2 > \pi_1, \text{ then either } \sigma_1\sqrt{\frac{1-\pi_2}{1-\pi_1}} < \sigma_2 < \sigma_1 \quad \text{or} \quad \sigma_1 < \sigma_2 < \sigma_1\sqrt{\frac{\pi_2}{\pi_1}} \qquad (5.27)$$

The sufficient condition given in 5.24 for monotonicity to hold true is an asymptotic result. Whether or not this condition is satisfied is a property of the learning process. In other words, this condition puts constrains on the learning process (modulo the target function) for monotonicity to hold. There are some common $p(e)$ distributions that satisfy this condition. For example, for the Bernoulli $p(e)$ where $\sigma(\pi) = \pi(1-\pi)$, the above condition is satisfied, therefore the monotonicity holds for binary bins, which is confirmed by the discussions in earlier sections. There are some other distributions that don't satisfy the condition. For uniform error distribution, where $0 < e < C$ and $p(e) = \frac{1}{C}$, $\pi = \frac{C}{2}$ and $\sigma^2 = \frac{\pi^2}{4}$, the condition is violated.

5.10 Conclusion

In the second part of the thesis, we proposed a novel theoretical framework, called the Bin Model, which provided a framework for analyzing generalization in a general classification learning process. The Bin Model abstracts the relevant quantities from a learning process by parameterizing each hypothesis function with a single value, π. Using the Bin Model, we derived a closed form for generalization that estimates the out-of-sample performance in terms of the in-sample performance. The expected generalization error was shown to depend only on the π-distribution for the learning process. Within the Bin Model framework, we can derive properties that hold for any $p(\pi)$ (for example, the lack of over-fitting with unbiased optimization), and therefore that hold for any learning model, target function and input distribution. Thus the Bin Model can provide a tool for further study of distribution independent properties. The Bin Model is a very powerful model which is simple enough for mathematical manipulation and general enough to address and give insights into many important issues in learning. While in general the π-distribution is unknown, we can still gain insights from invariant properties of the Bin Model analysis. Work in progress includes the extension of the model to noisy problems, regression problems and the identification of relative features of $p(\pi)$ that characterize commonly encountered learning problems.

5.11 Appendix A

Special Case: If $p(\pi) = (d+1)\pi^d$ for any $d \in \mathbf{Z}^+$. Then,$\pi(\nu) = \frac{N\nu+d+1}{N+d+2}$ and $var(\pi|\nu) = \frac{(N\nu+d+1)(N-N\nu+1)}{(N+d+2)^2(N+d+3)}$.

Proof:

$$\pi(\nu) = \frac{\int_0^1 \pi\pi^{N\nu}(1-\pi)^{N(1-\nu)}p(\pi)d\pi}{\int_0^1 \pi^{N\nu}(1-\pi)^{N(1-\nu)}p(\pi)d\pi}$$

$$= \frac{\int_0^1 \pi^{N\nu+1+d}(1-\pi)^{N-N\nu}p(\pi)d\pi}{\int_0^1 \pi^{N\nu+d}(1-\pi)^{N-N\nu}p(\pi)d\pi}$$

$$= \frac{B(N\nu + d + 2, N - N\nu + 1)}{B(N\nu + d + 1, N - N\nu + 1)}$$

where $B(x, y) = \int_0^1 t^{x-1}(1-t)^{y-1}dt$ is the Bessel function. Since $B(x, y)$ can be expressed in terms of Gamma function $B(x, y) = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}$, and $\Gamma(x) = (x-1)!$ for $x \in \mathbf{Z}^+$, then

$$\pi(\nu) = \frac{\Gamma(N\nu + d + 2)\Gamma(N + d + 2)}{\Gamma(N\nu + d + 1)\Gamma(N + d + 3)}$$

$$= \frac{(N + d + 1)\ !\ (N\nu + d + 1)\ !}{(N + d + 2)\ !\ (N\nu + d)\ !}$$

$$= \frac{N\nu + d + 1}{N + d + 2} \tag{5.28}$$

For the variance,

$$var(\pi|\nu) = E[\pi^2|\nu] - (E[\pi|\nu])^2$$

$$= \frac{\int_0^1 \pi^2 \pi^{N\nu}(1-\pi)^{N(1-\nu)}p(\pi)d\pi}{\int_0^1 \pi^{N\nu}(1-\pi)^{N(1-\nu)}p(\pi)d\pi} - \pi^2(\nu)$$

since $p(\pi) = (d+1)\pi^d$, then the first term is

$$\frac{\int_0^1 \pi^2 \pi^{N\nu}(1-\pi)^{N(1-\nu)}p(\pi)d\pi}{\int_0^1 \pi^{N\nu}(1-\pi)^{N(1-\nu)}p(\pi)d\pi}$$

$$= \frac{\int_0^1 \pi^{N\nu+2+d}(1-\pi)^{N-N\nu}p(\pi)d\pi}{\int_0^1 \pi^{N\nu+d}(1-\pi)^{N-N\nu}p(\pi)d\pi}$$

$$= \frac{B(N\nu + d + 3, N - N\nu + 1)}{B(N\nu + d + 1, N - N\nu + 1)}$$

$$= \frac{\Gamma(N\nu + d + 3)\Gamma(N + d + 2)}{\Gamma(N\nu + d + 1)\Gamma(N + d + 4)}$$

$$= \frac{(N+d+1)\,!\,(N\nu+d+2)\,!}{(N+d+3)\,!\,(N\nu+d)\,!}$$

$$= \frac{(N\nu+d+1)(N\nu+d+2)}{(N+d+2)(N+d+3)} \qquad (5.29)$$

Therefore,

$$var(\pi|\nu)$$

$$= \frac{(N\nu+d+1)(N\nu+d+2)}{(N+d+2)(N+d+3)} - (\frac{N\nu+d+1}{N+d+2})^2$$

$$= \frac{(N\nu+d+1)(N-N\nu+1)}{(N+d+2)^2(N+d+3)}$$

Q.E.D.

5.12 Appendix B

Theorem: Under unbiased optimization, $\frac{d\pi(\nu)}{d\nu} > 0$ for any ν.

Proof:

$$\pi(\nu) = \frac{\int_0^1 \pi p(\pi)\pi^{N\nu}(1-\pi)^{N(1-\nu)}d\pi}{\int_0^1 p(\pi)\pi^{N\nu}(1-\pi)^{N(1-\nu)}d\pi}$$

Let

$$A(\nu) = \int_0^1 \pi p(\pi)\pi^{N\nu}(1-\pi)^{N(1-\nu)}d\pi \qquad (5.30)$$

$$B(\nu) = \int_0^1 p(\pi)\pi^{N\nu}(1-\pi)^{N(1-\nu)}d\pi \qquad (5.31)$$

Then

$$\pi(\nu) = \frac{A(\nu)}{B(\nu)} \qquad (5.32)$$

$$\frac{dA(\nu)}{d\nu} = N \int_0^1 \pi^{N\nu+1}(1-\pi)^{N(1-\nu)} ln\ \frac{\pi}{1-\pi} p(\pi) d\pi$$

$$\frac{dB(\nu)}{d\nu} = N \int_0^1 \pi^{N\nu}(1-\pi)^{N(1-\nu)} ln\ \frac{\pi}{1-\pi} p(\pi) d\pi$$

Then,

$$\frac{d\pi(\nu)}{d\nu} = \{\frac{dA(\nu)}{d\nu} B(\nu) - A(\nu)\frac{dB(\nu)}{d\nu}\}\frac{1}{B^2(\nu)}$$

$$= \frac{1}{B^2(\nu)} \int_0^1 \int_0^1 (\pi\rho)^{N\nu}(1-\pi)^{N(1-\nu)}(1-\rho)^{N(1-\nu)}\ \pi ln\ \frac{\pi(1-\rho)}{\rho(1-\pi)} p(\pi)p(\rho) d\pi d\rho$$

Let

$$S(\pi,\rho) = (\pi\rho)^{N\nu}(1-\pi)^{N(1-\nu)}(1-\rho)^{N(1-\nu)}$$

and

$$T(\pi,\rho) = \pi ln\ \frac{\pi(1-\rho)}{\rho(1-\pi)}$$

.

Then,

$$\frac{d\pi(\nu)}{d\nu} = \int_0^1 \int_0^1 S(\pi,\rho)T(\pi,\rho)p(\pi)p(\rho) d\pi d\rho$$

.

We can see that $S(\pi,\rho) = S(\rho,\pi)$, and $T(\pi,\rho) + T(\rho,\pi) = (\pi - \rho) ln \frac{\pi(1-\rho)}{\rho(1-\pi)}$. Since π and ρ are interchangeable inside the integral, *i.e.*,

$$\frac{d\pi(\nu)}{d\nu} = \int_0^1 \int_0^1 S(\rho,\pi)T(\rho,\pi)p(\pi)p(\rho) d\pi d\rho$$

.

Therefore,

$$\frac{d\pi(\nu)}{d\nu} = \frac{1}{2}(\int_0^1 \int_0^1 S(\pi,\rho)T(\pi,\rho)p(\pi)p(\rho)d\pi d\rho + \int_0^1 \int_0^1 S(\rho,\pi)T(\rho,\pi)p(\pi)p(\rho)d\pi d\rho)$$

$$= \frac{1}{2}\int_0^1 \int_0^1 S(\pi,\rho)(T(\pi,\rho)+T(\rho,\pi))p(\pi)p(\rho)d\pi d\rho$$

$$= \frac{1}{2}\int_0^1 \int_0^1 (\pi\rho)^{N\nu}(1-\pi)^{N(1-\nu)}(1-\rho)^{N(1-\nu)}(\pi-\rho)ln\ \frac{\pi(1-\rho)}{\rho(1-\pi)}p(\pi)p(\rho)d\pi d\rho$$

All the terms of the product inside the integral are positive with the possible exception of $(\pi-\rho)ln\frac{\pi(1-\rho)}{\rho(1-\pi)}$.

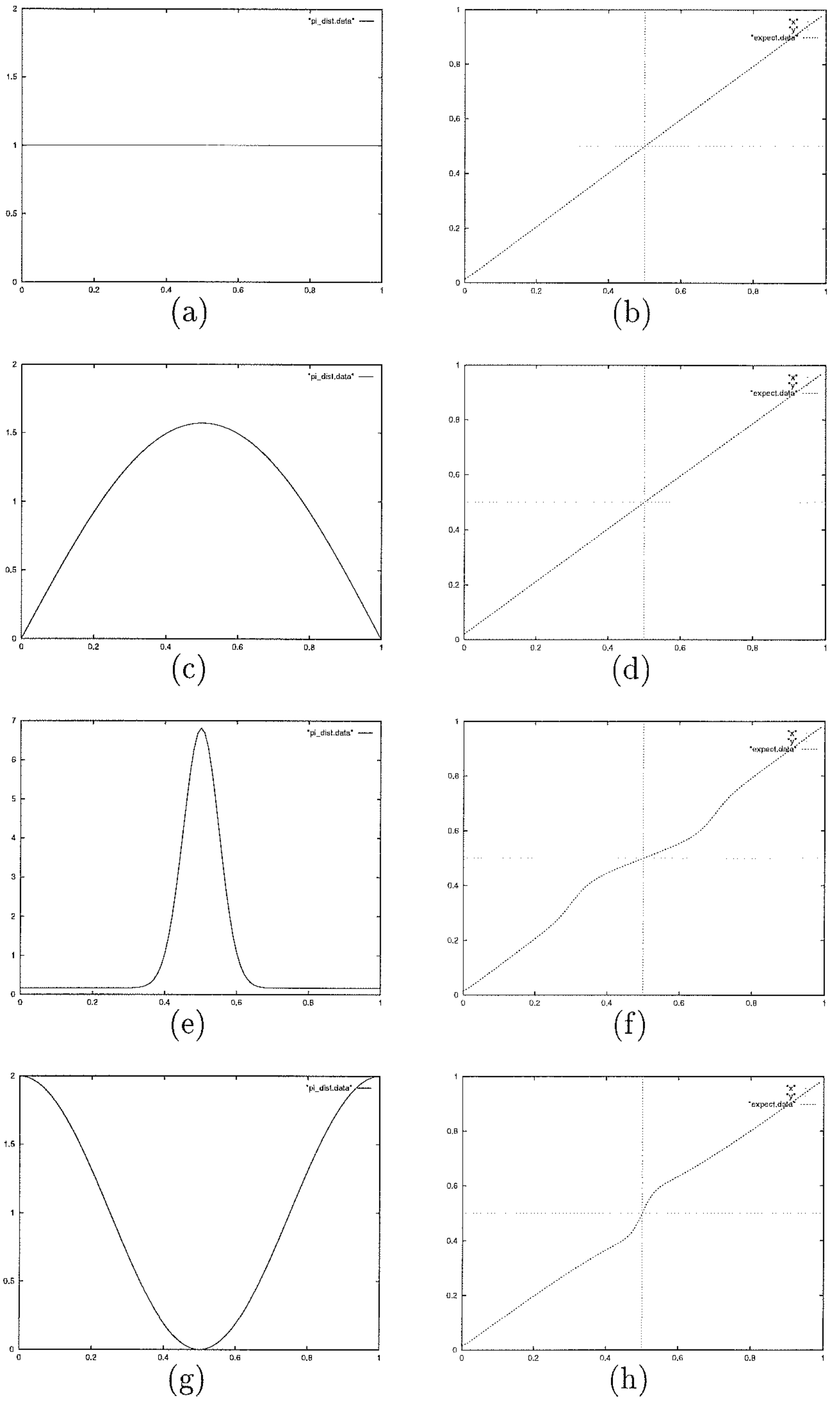
If $\pi \geq \rho$, then $\pi - \rho \geq 0$ and $\frac{\pi(1-\rho)}{\rho(1-\pi)} \geq 1$, which implies $(\pi-\rho)ln\frac{\pi(1-\rho)}{\rho(1-\pi)} \geq 0$.

If $\pi \leq \rho$, then $\pi - \rho \leq 0$ and $\frac{\pi(1-\rho)}{\rho(1-\pi)} \leq 1$, which again implies $(\pi-\rho)ln\frac{\pi(1-\rho)}{\rho(1-\pi)} \geq 0$.

Therefore,

$$\frac{d\pi(\nu)}{d\nu} \geq 0$$

Q.E.D.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 5.8: π-distributions (the left column) and their corresponding generalization curves (the right column).

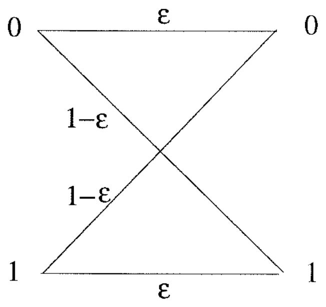


Figure 5.9: Binary symmetric channel

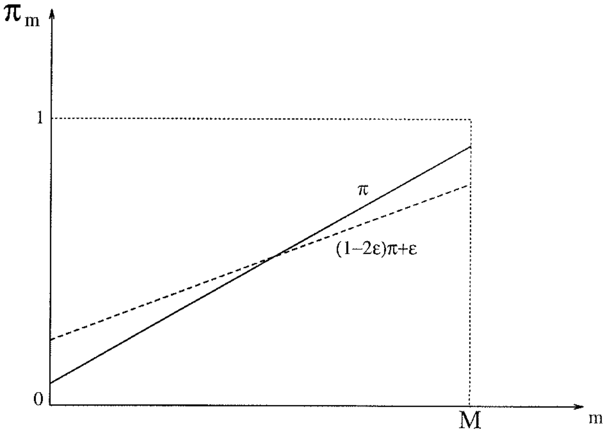


Figure 5.10: Effect of Noise – the original π is pushed closer to random function.

Bibliography

[Abu-Mostafa and Song, 1996] Y. Abu-Mostafa and X. Song. Bin model for neural networks. In S. Amari, L. Xu, L. Chan, I. King, and K. Leung, editors, *Proceedings of the International Conference on Neural Information Processing*, volume 1, pages 169–173. Springer, 1996.

[Abu-Mostafa, 1989] Y. Abu-Mostafa. The vapnik-chervonenkis dimension: Information versus complexity in learning. *Neural Computation*, 1:312–317, 1989.

[Akaike, 1970] H. Akaike. Statistical predictor identification. *Annals of the Institute of Statistical Mathematics*, 22:203–217, 1970.

[Aus *et al.*, 1987] H. A. Aus, H. Harms, V. ter Meulen, and U. Gunzer. Statistical evaluation of computer extracted blood cell features for screening population to detect leukemias. In Pierre A. Devijver and Josef Kittler, editors, *Pattern Recognition Theory and Applications*, pages 509–518. Springer-Verlag, 1987.

[Bacus and Gose, 1972] J. W. Bacus and E. E. Gose. Leukocyte pattern recognition. *IEEE Transactions on Systems, Man and Cybernetics*, 4:513–526, 1972.

[Binaghi *et al.*, 1997] E. Binaghi, P. Madella, M. Montesano, and A. Rampini. Fuzzy contextual classification of multisource remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 35:326–339, 1997.

[Bishop, 1995] C. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, New York, 1995.

[Boehringer-Mannheim-Corporation, 1991] Boehringer-Mannheim-Corporation. *Urinalysis Today.* Boehringer-Mannheim-Corporation, 1991.

[Cover and Thomas, 1991] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* Wiley Series in Telecommunications. Wiley, 1991.

[Duda and Hart, 1973] R. Duda and P. Hart. *Pattern Classification and Scene Analysis.* J. Wiley & Sons, 1973.

[Faugeras and Berthod, 1981] O. D. Faugeras and M. Berthod. Improving consistency and reducing ambiguity in stochastic labeling: An optimization approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3:412–424, 1981.

[Feller, 1950] W. Feller. *An Introduction to Probability Theory and Its Applications.* John Wiley & Sons, New York, 1950.

[Haralick and Joo, 1986] R. M. Haralick and H Joo. A context classifier. *IEEE Transactions on Geoscience and Remote Sensing*, 24:997–1007, 1986.

[Haralick and Shapiro, 1992] R. M. Haralick and L.G. Shapiro. *Computer and Robot Vision.* Addison-Welsley, 1992.

[Hertz *et al.*, 1991] J. Hertz, A. Krogh, and R. Palmer. *Introduction to the Theory of Neural Computation.* Addison-Wesley, Redwood City, 1991.

[Illingworth and Kittler, 1987] J. Illingworth and J. Kittler. Optimization algorithms in probability relaxation labeling. In Pierre A. Devijver and Josef Kittler, editors, *Pattern Recognition Theory and Applications*, pages 109–117. Springer-Verlag, 1987.

[Kasdan *et al.*, 1994] H.K. Kasdan, J.P. Pelmulder, L. Spolter, G.B. Levitt, M.R. Lincir, G.N. Coward, S. I. Haiby, J. Lives, N.C.J. Sun, and F.H. Deindoerfer. The $WhiteIRIS^{TM}$ leukocyte differential analyzer for rapid high-precision differentials based on images of cytoprobe-reacted cells. *Clinical Chemistry*, 40:1850–1861, 1994.

[Khazenie and Crawford, 1986] N. Khazenie and M. M. Crawford. Spatial-temporal autocorrelated model for contextual classification. *IEEE Transactions on Geoscience and Remote Sensing*, 24:997–1007, 1986.

[Kittler and Illingworth, 1985] J. Kittler and J. Illingworth. Relaxation labelling algorithms - a review. *Image and Vision Computing*, 3:206–216, 1985.

[Kittler, 1987] J. Kittler. Relaxation labelling. In Pierre A. Devijver and Josef Kittler, editors, *Pattern Recognition Theory and Applications*, pages 99–108. Springer-Verlag, 1987.

[Moody, 1992] J. E. Moody. The effective number of parameters: An analysis of generalization and regularization in nonlinear learning systems. In J. E. Moody, S. J. Hanson, and R. P. Lippmann, editors, *Advances in Neural Information Processing Systems 4*, volume 4, pages 847–854. Morgan Kaufmann, 1992.

[Richard and Lippmann, 1991] M.D. Richard and R.P. Lippmann. Neural network classifiers estimate bayesian *a posteriori* probabilities. *Neural Computation*, 3:461–483, 1991.

[Rosenfeld *et al.*, 1976] A. Rosenfeld, Hummel R., and Zucker S. Scene labeling by relaxation operations. *IEEE Transactions on Systems, Man and Cybernetics*, 6:420–433, 1976.

[Schwartz *et al.*, 1990] D. B. Schwartz, V.K. Samalam, S. Solla, and J. S. Denker. Exaustive learning. *Neural Computation*, 2:374–385, 1990.

[Silverman, 1993] B. Silverman. *Density Estimation for Statistics and Data Analysis.* Chapman and Hall, London, UK, 1993.

[Solla, 1992] Sara Solla. Supervised learning: A theoretical framework. In M. Casdagli and S. Eubank, editors, *Nonlinear Modeling and Forecasting*, pages 25–38. Addison-Wesley, 1992.

[Tilton *et al.*, 1982] J. C. Tilton, S. B. Vardman, and P.H. Swain. Estimation of context for statistical classification of multispectral image data. *IEEE Transactions on Geoscience and Remote Sensing*, 20:445–452, 1982.

[Toussaint, 1978] G. Toussaint. The use of context in pattern recognition. *Pattern Recognition*, 10:189–204, 1978.

[Vapnik and Chervonenkis, 1971] V. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and Its Applications*, 16:264–280, 1971.

[Vapnik, 1995] V. N. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag, New York, 1995.

[Zucker and Mohammed, 1978] S. W. Zucker and J. L. Mohammed. Analysis of probabilistic relaxation labeling processes. In *IEEE Conf. on Image Processing and Pattern Recognition*, pages 307–312, 1978.