Chapter 1

Introduction

> "I'm disappointed in *Wired*, and perhaps in the good professor as well, for not displaying the self-control to decline publication of what reads as a classic red herring. Where are discussion and references to support this claim of getting something for nothing?"
>
> "Can you believe that someone would have such a fundamental misunderstanding of basic mathemetics [sic] and information theory that they would base a medical diagnosis on features produced by data interpolation? I hope it's not my doctor doing it. Prettying up pictures is great. Looking for tumors, etc. is insane. By definition, you're looking for an aberration, which, by definition, this algorithm would not produce."
>
> "Would you be willing to gamble your life on interpolated data where the shortcoming would be missing clinical pathology? A MRI image contains many subtle shades of gray in abstract shapes. For an artistic image clear, sharp edges and vivid colors may enhance, but to render a line on a diagnostic image that appears to abruptly end and restart may draw a complete artery where there is really an occlusion or to smooth out faint variations representing a brain tumor can be deadly."
>
> "Sounds great, but extraordinary claims demand extraordinary proof to distinguish them from hype."
>
> "You can't make something from nothing. You can create something by inference, but there's no certainty what you're making is right."
>
> "You won't be able, anyway, to save the data nobody ever entered... ."

The above quotes are from readers' comments on the online version of a 2010 *Wired Magazine* article [Ell10] about compressed sensing (CS). To sheltered academics in certain fields, compressed sensing is just another buzzword; these quotes serve to remind that there really has been great progress. This thesis can be viewed as an answer to these skeptical readers. The power of compressed

sensing techniques is highlighted, while maintaining that the theory really is intuitive (with the benefit of hindsight), since the basic tenet is that it is necessary to *sample at the information rate.*

To the several thousand academics just mentioned, the results of this thesis can be read and digested without causing much commotion. The same content, but in the year 2000, would have been met with astonishment; in the year 1985, incredulity. To be sure, beginning in the late 1980s, a small group of researchers had premonitions of what was possible, but the main notions behind compressed sensing were not widespread. What has changed since the 80s? Not just one major advance, but advances in many fields of science, math, and engineering. Namely:

- Signal processing and statistical advances, mainly in the ability to exploit sparsity. The results of compressed sensing in 2004 were fundamental [CRT06, Don06], although maybe not a complete surprise to a handful of statisticians since around 1988. The real significance of compressed sensing was a change in the very manner of thinking of many engineers and mathematicians. Instead of viewing ℓ_1 minimization as a post-processing technique to achieve better signals, CS has inspired devices, such as the RMPI system described in this thesis, that acquire signals in a fundamentally novel fashion, regardless of whether ℓ_1 minimization is involved.

- Computing power advances. For believers in Moore's law, this is no surprise, but nonetheless it is still impressive. The experiments in this thesis all require massive computation; so much, in fact, that before the development of the algorithms presented in Chapters 3 and 4, the simulations were performed on the massive Caltech SHC cluster.

- Circuits advances (also relating to the computing power). The design of the integrated circuit (IC) discussed in this thesis began in 2008, and the IC has now been fabricated in 90 nm CMOS technology; in 2011, there are even 32 nm technologies. Beyond just fabrication improvements, the design knowledge and design toolkits (such as powerful SPICE simulators) have increased. This allowed precise design of the IC. We note that we are still on the border of the feasible, since a full SPICE calibration of the system would take roughly a month of computational time on a multi-core workstation.

- Algorithmic advances. By 1997, with the publications of [Wri97,NN94], interior point methods (IPM) were a mature technology, and the powerful models of semi-definite programming [VB96] were being increasingly used. However, this recent decade has seen tremendous growth in data collection; as just one example, the Compact Muon Solenoid detector at the LHC at CERN produces 40 terabytes *per second* [The08]. Thus problems that were once "solved" have now been re-opened because these modern scientific problems are of such large scale. On the extreme side are tools like MapReduce and algorithms like Google's PageRank. For large—but not gigantic—datasets, a new theme is first-order methods. And, as always, parallel methods

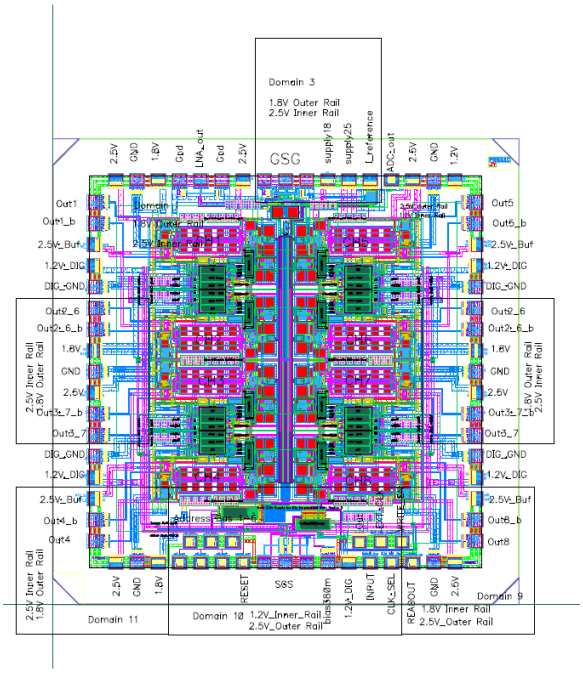


Figure 1.1: Diagram of the version 1 RMPI chip

are still crucial, with increasing attention on the GPGPU and on minimal-communication distributed computing.

This thesis brings together all of these advances and presents a state-of-the art system that bypasses a current barrier in ADC technology. The actual physical system is a receiver/ADC called the "Random Modulator Pre-Integrator" (RMPI); Figure 1.1 shows a layout of the actual integrated circuit. The general concept of the RMPI is not original to this thesis (see [SOS+05, KLW+06, LKD+07, TLD+10]), but it is one of the world's first working hardware devices to be built based on the compressed sensing paradigm. In designing the system, numerous engineering and mathematical problems were overcome, and we hope that the results presented in this thesis will be of use to others in the field.

The second component of the system is the signal-processing back-end, which involves many steps, but the heart of the recovery is solving an optimization program. This key computation is "just" a linear program (LP), or a second-order cone program (SOCP) for fancier variants, but we will show why existing algorithms are inadequate to deal with the large quantities of data, and how our proposed algorithms make the computation tractable.

Chapter 2 discusses the design of the RMPI receiver which was part of a larger "analog-to-information" (A2I) project. We take a basic CS idea, the RMPI, and from a high-level block diagram, turn it into a working radar receiver integrated circuit that has now been fabricated in 90 nm CMOS. The chapter explores both hardware decisions as well as signal processing tricks, but

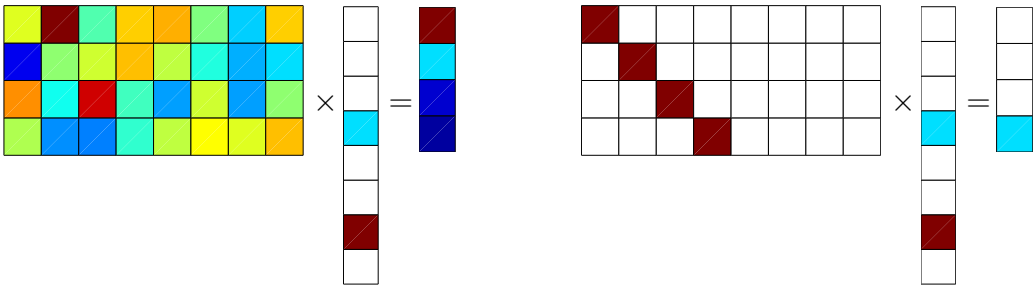


Figure 1.2: Democracy/incoherence in measurements. Pictoral representation of linear measurements $Ax = b$. Historically, good operators A were square and assumed to be easily invertible, such as the identity I or Fourier matrix $\mathcal{F}$. For *undersampling*, choosing A to be a partial identity matrix is bad, and recovery depends on luck. For sparse signals, it turns out that we can recover all of them if measurements are sufficiently global and incoherent. We informally refer to this idea as "democracy": each measurement provides the same amount of information as every other measurement, so there is no one special measurement without which reconstruction is impossible.

without treating the optimization problems in detail.

Chapter 3 presents an optimization algorithm "NESTA" that efficiently solves the signal-processing optimization problem for the RMPI. Chapter 4 presents an alternative optimization algorithm "TFOCS" that has some additional attractive features and is applicable to diverse problems.

The emphasis of this thesis is on the RMPI section, since this has not yet been published. The solvers (NESTA and TFOCS) have been previously described in publications [BBC11, BCG10a]. Chapters 3 and 4 are based on these publications, but contain some new material (see the chapter introductions for an overview of what is new). These chapters were written with co-authors, named in the chapter introductions.

The rest of this introduction presents some background material. The theory of compressed sensing is not discussed in much detail since there are already several excellent review articles and monographs [Can06, CW08, Bar07, FR11]; see also the resources at [RU]. Instead, we exploit the unconstrained nature of this thesis to elaborate a little on the history of some ideas that contributed to our current understanding of ℓ_1 processing, since it is rare that current literature has the luxury of going into background.

What compressed sensing is. A brief overview of the theoretical results is reserved for §1.2.4; we first mention the main ideas in an intuitive style. Compressed sensing (CS) addresses the situation in which there is an unknown mathematical object of interest that lies in a large ambient space, but because of prior information, the information content of the signal is less than the dimension of the space. The prior information need not be interpreted in the Bayesian sense; a more appropriate interpretation is that the signal is "compressible".

An implication of the premise that the signal is compressible is that sampling can be performed at the "information rate", or only marginally faster. This generally means that under-sampling is possible. As always, more samples improve the performance in the presence of noise.

The other key aspect of compressed sensing is the idea of incoherent measurements. Each mea-

surement of a signal should give some global information; see Figure 1.2 for a depiction. This comes at a price: when all the measurements are combined, the reconstruction algorithm is non-linear. In fact, CS theory says that the most efficient measurements look like a random sum, regardless of the reconstruction algorithm or specific signal. Think of the Voyager 1 spacecraft taking compressed measurements and sending them back to Earth; such measurements would allow modern techniques to extract as much information as possible, even using techniques such as wavelet analysis which did not exist at the time of Voyager's launch.

The *Wired Magazine* article failed to convey to its readers the importance of incoherence, hence the skepticism. To the readers, it seems as though subsampling drops information. But with incoherent measurements, every measurement gives a little bit of information, so the aberration tumor *is* measured, just not directly. However, compressed sensing does not give "something for nothing". In the case of noisy measurements, it is always helpful to take more measurements since this reduces the effect of the noise, and so under-sampling will always perform worse than exact- or over-sampling (assuming the same post-processing techniques).

What compressed sensing is not. Because compressed sensing deals with sparsity and compressibility, it is related to many other fields, such as sparse approximation, classic problems in image and signal processing such as denoising and deconvolution, dictionary learning, computational harmonic analysis, etc. In brief, these fields are related to the first ingredients of compressed sensing: sparsity and compressibility. But since they do not fundamentally involve incoherent measurements, they are distinct fields.

The architecture proposed in Chapter 2 is a pure compressed sensing architecture, because fundamental to its operation is the fact that measurements are incoherent. The optimization algorithms proposed in Chapters 3 and 4 are not limited to compressed sensing; they solve a wide class of problems which include CS problems, but also include problems from sparse approximation, machine learning, etc. In particular, they solve so-called ℓ_1 minimization problems, which have a broad applicability.

Perhaps one of the biggest impacts of CS is that it has spurred research in related fields, with the idea of exploiting prior knowledge. Yet the impact on hardware devices is much more limited: even though compressed sensing theory is about 7 years old and is quite well understood, there are very few pure compressed sensing applications. A few novel schemes are discussed in §1.1.

1.1 Applications

Before describing compressed sensing in detail, we motivate it with a few applications. Of course this thesis presents one such application, but the techniques of this thesis extend beyond this specific

case. As mentioned earlier, only true CS devices are mentioned, and thus we consider only hardware that takes incoherent measurements.

- **Optical and near-optical imaging.** The iconic "single pixel camera" developed at Rice University [WLD+06] is the best known compressed sensing device; it was singled out in 2007 as one of the top 10 emerging technologies by MIT Technology Review [Mag07]. The idea behind the camera is to trade spatial resolution for temporal resolution; for frequencies such as infrared, where each pixel is extremely expensive, this is a smart trade off. The camera uses a single pixel, but it takes many measurements over time. Each measurement needs to be different and also encode the entire scene; this is achieved by placing a micro-mirror array in front of a conventional optical lens system. The array is a grid of pixel-like mirrors called a digital micro-mirror device (DMD); each micro-mirror can be oriented so as to reflect light toward the single receiving pixel, or in some other direction. The effect is that of a binary mask.

  This is similar to the ideas used in coded aperture imaging in astronomy since the 1960s, except that due to sub-sampling, direct inversion is no longer possible. Coded aperture imaging uses a mask that is spatially moved in order to acquire a full set of data. Similar to radio interferometry, coded aperture is used as a means to increase spatial resolution, since radio telescopes' antenna have limited spatial resolution.

  Overall, these CS-based optical devices seem promising, and InViewCorp has recently mobilized to commercialize such a system. Similar ideas can also be applied to microscopy [WCW+09], where again DMDs are used to reduce the number of raster scans needed in confocal imaging.

  However, the subject is not yet closed, for there remain a few key difficulties. The first challenge is calibration. If the system can be modeled sufficiently accurately, then calibration is not an issue, but for high dynamic range acquisition, it is likely that the system will need to be characterized. Calibration at optical frequencies is challenging, since the phase is not as easy to control as in radio frequency (RF) systems.

  Another difficulty is related to the binary sensing matrix and the type of noise. The receiver sees a superposition of light from many sources for every measurement, and the photon count is greater than it would be in a multi-pixel setting. The DMD acts as a mask, and is equivalent to a matrix of zeros and ones. Because there are no negative numbers, the average of each row is necessarily greater than zero. It is this effect that causes the problem. Photons arrive according to a Poisson process, so there is "shot" noise. The distinguishing feature of this Poissonian shot noise is that the variance is equal to the mean. Because the single-pixel sensor sees many pixels worth of light, it consequently sees greater fluctuations about its mean. This appears to be a fundamental problem with no clever workaround. Bright point-like sources

can be resolved, but sources that have an even intensity are completely lost in the shot-noise and impossible to recover. This is currently the limiting factor in these systems.

- Medical Resonance Imaging (MRI). One of the "hot-topic" CS applications is MRI, which has seen much research, and it is also one of the first, with conference proceedings of the authors in [LDP07] going back to 2005. The work of Lustig [LDP07] has even been incorporated into the Stanford research hospital [Ell10], and current MRI manufactures are paying close attention to the field; the Siemens Corporate Research office has stayed active in the field of compressed sensing. MRI works by acquiring points in 2D or 3D k-space (i.e., Fourier space), and conventional MRI acquires specific grids of points so that the image may be reconstructed by the inverse Radon transform. Using sparse-approximation ideas, it is possible to sub-sample this grid and solve a linear inverse problem to recover the signal. CS applies to MRI by allowing k-space samples that are not on standard grids, and fewer samples than conventionally needed. Fewer samples leads to faster scans, which is significant since scans are performed on living, moving, people. However, the true potential of CS for MRI is that it might allow for non-standard types of measurements or certain types of systematic errors, such as non-linearities in the magnetic field, or weaker magnetic fields. To our knowledge, these types of breakthroughs have not yet happened.

- Microarray sequencing. Microarrays are used commonly in biology to identify specimens, using fluorescent tags to identify where samples bind; most samples have only a few active parts, thus using ideas from CS and group testing suggest that it is possible to take many fewer measurements and still accurately infer which specimens are present. Recent work in industry has suggested that this approach is useful in practice [MSS$^+$10].

- Seismic imaging. Because the Earth is made of discrete layers and so separated by sparse boundaries, seismic imaging has benefited from sparse recovery techniques since the 1970s. However, a true compressed-sensing-based seismic imaging system goes further and acquires images in a different fashion; for example, by changing the type of excitation signal sent by a ship or controlled explosion. Another method to apply compressed sensing is by controlling the rate and location of samples, much like the non-uniform sampler (NUS) introduced in §1.4.1. Undersampling is a fact of life for many seismic imaging systems; the system in [HH08] proposes that undersampling should be nearly at random, like the NUS, in order to turn coherent aliases into incoherent noise. They also suggest that the undersampling should not be completely at random; instead, the maximum separation between measurements should be controlled.

- Hyperspectral imaging. Hyperspectral imaging is the practice of acquiring images at many wavelengths (though usually not a continuum of wavelengths). It can be useful because it may

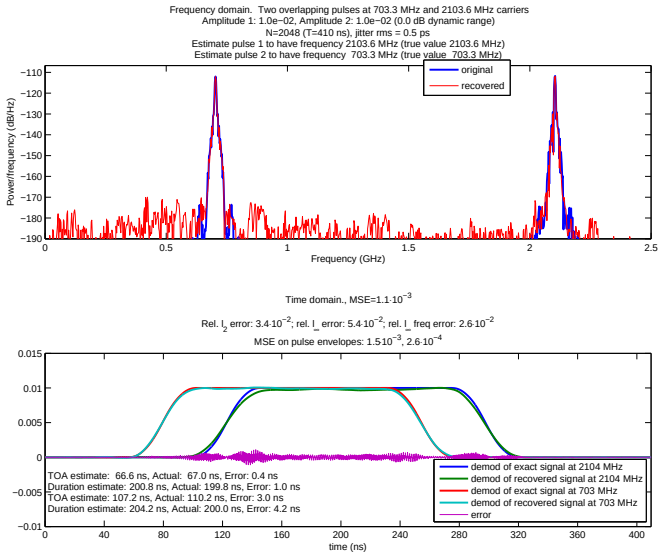


Figure 1.3: Preview: beating Nyquist with the RMPI. Reconstructing two pulses from 2.5 GHz of bandwidth using only 400 MHz sampling rate in a realistic simulation with noise and non-idealities.

ease identification; for example, it may aid a satellite imaging the ground, trying to determine the type of foliage on the surface. Images are typically compressible, and hyperspectral images even more so. See the work in [WGB07] for a proposed hyperspectral imaging system that uses a binary aperture mask (similar to the single pixel camera) with a multi-pixel array.

- Radar. Much like seismic imaging, it is possible to design special radar pulses to exploit CS; for example, [HS09] proposes sending a type of chirp called an Alltop sequence to probe targets, and shows that when there are only a few targets, this allows greater resolution than traditional radar. This is distinct from the radar problems discussed in this thesis. Typical CS radar applications design an emitter and a receiver; in this thesis, we are concerned with learning pulse characteristics from a foreign object's radar emitter, so not only is there no control over the emitted pulses, but the pulse parameters are unknown *a priori*.

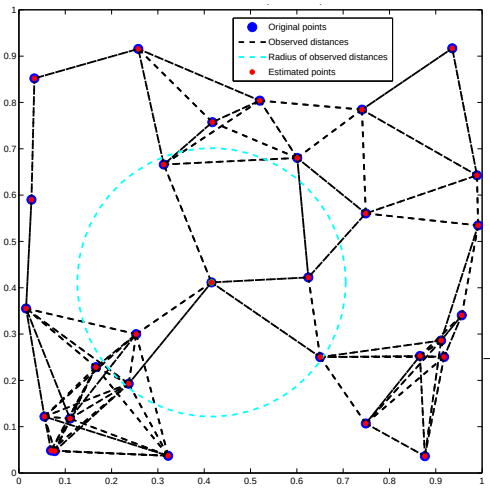
- CS has much potential for astronomy by extending what is possible with various types of coded aperture and, recently, DMD arrays. The benefits and problems are similar to those already discussed in the optical imaging section. It does not appear that a true CS-based astronomy device exists, but a CS-based coding scheme has been applied in astronomy [BSO08]. The Herschel satellite, launched on May 14, 2009, takes images with a photometer and needs to compress the images by a factor of 16 (in real-time) because of the limited rate of the downlink to Earth. Using lossless entropy coding, the data can be compressed by a factor of 2.5, but this is still insufficient. Because of the unique situation (low-powered sensor, but unlimited time and power for the receiver), [BSO08] proposed using a particular rectangular pseudo-

random matrix (a noiselet transform [CGM01]) to multiply groups of 6 images; importantly, this operation is easy to carry out in the satellite. The compressed data is sent to Earth and decompressed using ℓ_1 techniques or similar. The Herschel satellite adopted this scheme, so it is one of the first compressed sensing hardware devices, even if the image acquisition is still conventional. For more details of CS in astronomy, see [SB10].

- Quantum physics. In quantum computing and related fields, quantum states are prepared for specific purposes, and it is important to verify that an experiment has produced the expected state. The process of verifying a quantum state is known as quantum tomography. In an n qubit system, there are n particles but the spin of the system cannot be represented by a n-dimensional state vector due to entanglement. Instead, the system can be represented by a $n \times n$ Hermitian matrix. In a generic high entropy system, this matrix is rank n, but in specially prepared states, the matrix has low rank, or even rank 1. In a collaboration with quantum information physicists, this author worked on the problem of estimating the quantum state via subsampled measurements [GLF$^+$10]. Measurements are taken in an incoherent fashion; unfortunately, the best type of measurements are difficult to enact experimentally. Feasible measurements that are slightly less optimal are also proposed. The situation is interesting for several reasons. The first reason is that there is much experimental control over which measurements are taken, and no calibration is needed. It even allows the possibility of adaptive measurements. Secondly, measurements are inherently noisy because they are samples of a quantum wave function. Each measurement is really a Bernoulli random variable, so repeated measurements are binomial random variables. Thus the noise can be made arbitrarily small at the expense of requiring more time. Furthermore, in the experimental setup, there is a time penalty for switching to a new measurement, so it is advantageous to take repeated measurements of the same quantity. These tradeoffs make for interesting future study.

- ADC (Analog-to-Digital Converters). A compressed sensing ADC device is presented in Chapter 2 and also in §1.4, where history and background will be presented in detail, so we defer discussion. Figure 1.3 shows the recovery of two radar pulses from a realistic simulation that included non-idealities of the circuit.

Applications of sparse recovery. As pointed out, the above applications are true compressed sensing devices. The algorithms developed in Chapters 3 and 4 are designed for compressed sensing, but also for generic sparse recovery problems. Sparse recovery problems include topics mentioned in the CS applications section, as well as image and signal processing applications (interpolation, deconvolution, denoising, etc.; see Figures 1.5 and 1.6), and various applications in dictionary learning, blind source separation and morphological separation, and blind deconvolution. They also have ap-



	Denver	Chicago	Atlanta	Dallas	Seattle
Denver	0	?	591	398	?
Chicago	?	0	218	?	982
Atlanta	591	218	0	634	?
Dallas	398	?	634	0	521
Seattle	?	982	?	521	0

Figure 1.4: Preview: distance completion. In general, this is an NP-hard problem, but sometimes the convex relaxation is exact. The algorithm proposed in Chapter 4 can solve these minimization problems that involve matrix variables. This problem is like filling in the unknown "?" entries in the sample mileage chart on the right. The plot on the left shows a set of points for which about 29% of the pairwise distances are known (line segments represent known distances); from this, their true positions are recovered.

plications in many communications and network problems, and can be used to solve general statistics estimation problems. Thus our algorithms apply to almost all fields in the sciences. The algorithms can also solve certain matrix-variable problems in low-rank (and/or sparse) recovery. Such problems include matrix completion [CR09, Gro11] and various variants of robust PCA [CLMW09, MT10] (Figure 1.4). TFOCS can solve general semi-definite programs (SDP) [VB96]. Over the past 15 years, an increasing number of problems from engineering and control theory, as well as the fields mentioned above, have been cast in this framework. SDP relaxations of difficult problems such as MAX-CUT and matrix completion are also of high interest.

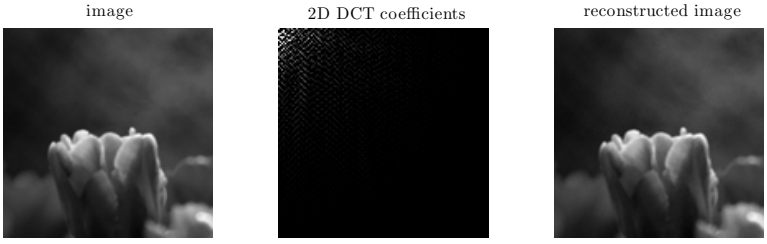


Figure 1.5: Compression. The image on the right is reconstructed from less than half the DCT coefficients of the original image on the left. It is an empirical fact that real images are usually compressible in the 2D frequency domain. JPEG uses an 8×8 2D DCT (in HSV color space) and quantizes coefficients, and has revolutionized the internet. The algorithms in Chapters 3 and 4 can be used for much more advanced image processing problems; see Figures 1.6 and 4.8.

1.1.1 Problems of interest

This section covers the formulation of some linear inverse problems. The meaning of the variables may change slightly in subsequent sections, but we aim for consistency of notation. In a linear

Figure 1.6: Preview: image denoising with TFOCS. For full details, see Figure 4.8. The image on the right is a denoised version of the noisy image on the left.

inverse problem, we seek an estimate $\hat{x}$ of a true signal x_0 given (possibly noisy) measurements of the formulation

$$b = Ax_0 + z, \quad x \in \mathbb{R}^n, b \in \mathbb{R}^m, A \in \mathbb{R}^{m \times n} \tag{1.1.1}$$

where z is a noise vector (usually stochastic). On occasion we allow complex numbers, and this will be explicitly stated. The model $b = A(x + \tilde{z})$ also arises, which is equivalent to (1.1.1) if we set $z = A\tilde{z}$. The *linearity* of the problem means that A is a linear operator, and since this thesis only considers finite dimensional problems, A is a matrix. Estimates for $\hat{x}$ are not necessarily linear functions of A and b.

This thesis is concerned with the under-determined case, $m < n$, but the algorithms are also useful in the ill-conditioned case where $m = n$ but A is either singular or has large condition number. The focus is not on the $m = n$ case, and it is assumed that A has full column rank unless otherwise stated. Recovering an arbitrary vector x from a given A and b is then ill-posed because there are infinitely many solutions to this under-determined equation.

Consider temporarily a problem in which $m > n$, so the situation is over-determined. A standard approach (if z is Gaussian, this is the maximum likelihood estimator) is to set $\hat{x} = \operatorname{argmin}_x \|Ax - b\|_2^2$, which is the famous "least-squares" estimator made popular by Gauss. This can be written in closed form as $\hat{x} = A^\dagger b$ where $A^\dagger = (A^T A)^{-1} A$ is the Moore-Penrose pseudo-inverse of A. In the case that A is ill-conditioned, a classical technique is to regularize the problem (known as Tikhonov regularization or ridge regression):

$$\hat{x}_\gamma = \operatorname*{argmin}_x \|Ax + b\|_2^2 + \gamma \|x\|_2^2 = (A^T A + \gamma^2 I)^{-1} A^T b. \tag{1.1.2}$$

The second equality holds when γ is large enough that $A^T A + \gamma^2 I$ is invertible. The answer depends

on the choice of γ. If it is assumed that the signal x_0 comes from a normal distribution with mean 0 and variance $\sigma^2 I$ (which we will write as $x_0 \sim \mathcal{N}(0, \sigma^2 I)$) and $z \sim \mathcal{N}(0, I)$, then this is the maximum *a posteriori* (MAP) estimator if we choose $\gamma = \sigma$. In some contexts it may be possible to choose γ via cross-validation.

The ℓ_2 norm used in (1.1.2) is part of the class of ℓ_p norms, defined on a vector x as $\|x\|_p = (\sum_i |x_i|^p)^{1/p}$; so $\|x\|_1 = \sum_i |x|$ is just the sum of the absolute values. The ℓ_0 quasi-norm is not a norm: it just measures the number of non-zero entries of a vector.

We use Tikhonov regularization to motivate our first problem of interest, called the LASSO [Tib96] (Least Absolute Shrinkage and Selection Operator):

$$\hat{x} = \underset{x}{\operatorname{argmin}} \ \frac{1}{2}\|Ax - b\|_2^2 + \lambda \|x\|_1. \tag{1.1.3}$$

This is quite similar in form to Tikhonov regularization. From a Bayesian perspective, (1.1.3) is the MAP estimator if x has a Laplacian prior and z is i.i.d. Gaussian. At the solution $\hat{x}$, the first-order stationary condition [BV04] gives

$$A^T(b - A\hat{x}) \in \lambda \partial \|\hat{x}\|_1$$

where ∂f is the sub-differential of f [Roc70] (that is, the set of all sub-gradients). In particular, the absolute value of every entry of $\partial \|x\|_1$ is bounded by 1. If the original signal x_0 were to be a solution, then this gives the necessary condition that $\|A^T z\|_\infty \leq \lambda$. When z is a general stochastic vector, it is not expected that $\hat{x}$ is exactly x_0, but it turns out that $\lambda \approx \|A^T z\|_\infty$ is a reasonable value for the parameter. If $z \sim \mathcal{N}(0, \sigma^2 I)$, then with high probability $\|A^T z\|_\infty \leq 2\sqrt{\log(n)}$.

The key difference between the LASSO and Tikhonov regularization is that the ℓ_1 norm has a sharp discontinuity at 0; see Figure 1.7.

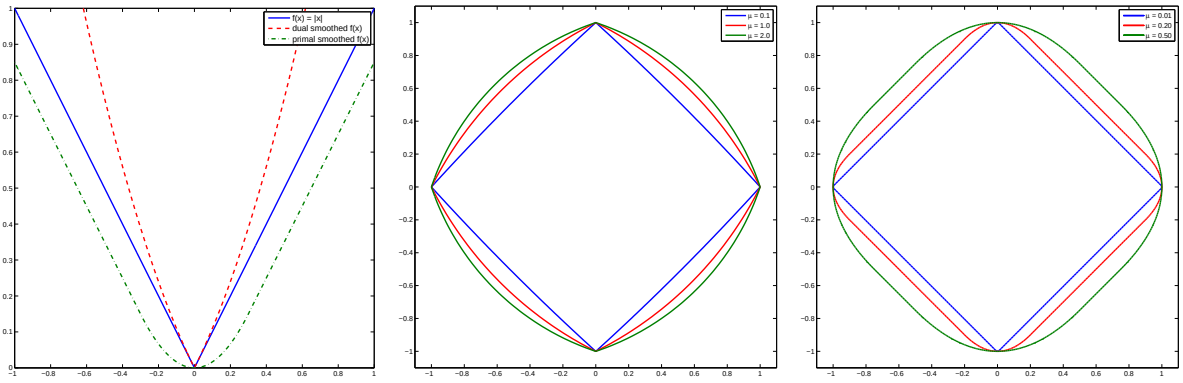
A closely related problem is the following:

$$\underset{x}{\text{minimize}} \ \|Ax - b\|_2^2 \quad \text{such that} \quad \|x\|_1 \leq \tau. \tag{1.1.4}$$

Confusingly, this is the version suggested by Tibshirani in his original LASSO paper [Tib96] so it also goes by the name "LASSO". To keep the variants clear, we will stick with the λ or τ parameters, and refer to the first problem as the "unconstrained LASSO". The unconstrained LASSO has seen much more attention in the literature than the ℓ_1 constrained LASSO.

Another variant is known as basis pursuit (BP) [CDS98], which is the linear program (LP):

$$\underset{x}{\text{minimize}} \ \|x\|_1 \quad \text{such that} \quad Ax = b \tag{1.1.5}$$



(a) The absolute value function (blue), a primal smoothed version known as the Huber function which will be used in Chapter 3, and a dual smoothed version which will be used in Chapter 4

(b) Contours of the dual smoothed absolute value function, for various levels of smoothing

(c) Contours of the Huber function for various levels of smoothing

Figure 1.7: The ℓ_1 norm promotes sparsity because of the kink at 0. Any function which is quadratic near zero (e.g., least-squares norm, or the Huber function [Hub64]) does not promote sparsity because of the "diminishing returns" in pushing a coefficient to zero. The dual-smoothed version keeps this feature. Unfortunately, the non-smoothness of ℓ_1 also makes it difficult to work with; only the Huber function has a continuous derivative. See also Figure 4.1

which always has the same value as its dual problem

$$\underset{\nu}{\text{maximize}}\ b^T\nu \quad \text{such that} \quad \|A^T\nu\|_\infty \leq 1.$$

This is mainly used when $z = 0$. For noisy cases, we turn to basis pursuit denoising (denoted BPDN or BP_ε), which is a second-order cone program (SOCP) [BV04]:

$$\underset{x}{\text{minimize}}\ \|x\|_1 \quad \text{such that} \quad \|Ax - b\|_2 \leq \varepsilon \tag{1.1.6}$$

and its dual

$$\underset{\nu}{\text{minimize}}\ \|\nu\|_2 - b^T\nu \text{ such that } \|A^T\nu\|_\infty \leq \varepsilon.$$

This dual problem has the same value as the primal problem when A has full row-rank and $\varepsilon > 0$ (see the Slater conditions [BV04]). BPDN is sometimes referred to as the LASSO as well.

BPDN will be the focus of the optimization algorithms introduced in later chapters. A basic result of convex optimization [Roc70, BNO03] is that unconstrained LASSO, the ℓ_1 constrained LASSO, and BPDN are all equivalent for some values of ε, τ, and λ (as long as these are all non-zero and finite). That is, given ε, there is some $\lambda(\varepsilon)$ such that the unconstrained LASSO estimator is the same as the BPDN estimator. In general, calculating $\lambda(\varepsilon)$ is not easy, and requires at least as much computation as solving the estimation problem; for example, to calculate $\varepsilon(\lambda)$, we set $\varepsilon = \|A\hat{x}_\lambda - b\|_2$ which requires knowing $\hat{x}_\lambda$.

The value of ε is typically chosen by $\varepsilon \simeq \mathbb{E}\,\|z\|_2$. If the entries of z are independent and have a finite second moment (and zero first-moment), this is just the square root of the sum of the variances. Methods for estimating variances, using the sample variance or a robust statistic like the median absolute deviation, are well-understood, so it is quite reasonable to assume that a reasonable value of ε is known. In practice, some researchers may prefer to estimate ε rather than λ; this was the motivation behind NESTA's development.

The final canonical problem we discuss is the Dantzig selector [CT07a], which motivated the development of the TFOCS algorithm. The Dantzig selector is the solution to

$$\underset{x}{\text{minimize}} \ \|x\|_1 \quad \text{such that} \quad \|A^T(Ax-b)\|_\infty \leq \delta. \tag{1.1.7}$$

The parameter δ should be about the same as the parameter λ from the unconstrained LASSO. The Dantzig selector is used for the same types of problems as BPDN, and enjoys similar theoretical properties. Indeed, the differences between the two was the subject of much discussion in the Annals of Statistics (see [EHT07, CT07b] and related papers in the same issue).

Lastly, we present two variants of BPDN that are useful when a signal x_0 is compressible in some basis or dictionary Ψ; these variants can also be applied to the LASSO and Dantzig formulations. Suppose $x_0 = \Psi\alpha_0$ (this is useful when α_0 is sparse or has decaying coefficients), and take measurements $b = \Phi x_0 + z = \Phi\Psi\alpha_0 + z$. There are two ways to use BPDN. The first is known as "synthesis":

$$\underset{\alpha}{\text{minimize}} \ \|\alpha\|_1 \quad \text{such that} \quad \|A\alpha - b\|_2 \leq \varepsilon, \quad A \triangleq \Phi\Psi \tag{1.1.8}$$

and the second is "analysis":

$$\underset{x}{\text{minimize}} \ \|\Psi^T x\|_1 \quad \text{such that} \quad \|\Phi x - b\|_2 \leq \varepsilon. \tag{1.1.9}$$

Since we are often interested in solving the analysis problem, it is convenient to allow BPDN (and the other variants) to use weighted ℓ_1 norms; we often write W for the weighting matrix, e.g., $W = \Psi^T$. The differences between synthesis and analysis are discussed in §2.7.2.1, §3.6.1, and §4.7.5.

1.2 Historical development

Here, we present a selective history of some key developments leading to compressed sensing, as well as some classic signal processing results.

1.2.1 Classical signal processing

We start with the following fundamental theorem.

Theorem 1.2.1 (Shannon-Nyquist-Whittaker, this form modified from [Mal08]). *Let $x(t)$ be a band-limited signal with frequencies inside the band $[-B, B]$ (in Hz). Then*

$$x(t) = \sum_{n=-\infty}^{\infty} x(n\Delta T)\phi_s(t - n\Delta T)$$

where $f_s \triangleq 2B$ is the "Nyquist rate" or "Nyquist frequency", $\Delta T \triangleq 1/f_s$, and $\phi_s = \frac{\sin(2\pi f_s t)}{2\pi f_s}$ is a scaled sinc *function.*

The theorem is obvious using the machinery of the inverse Fourier transform and convolutions, and noting that the Fourier transform of a *sinc* is a boxcar function; this may explain why it was discovered independently by several people in the first half of the 20th century. In [ME09a], it is referred to as the WKS theorem, where the K stands for Kotelńikov.

In plain language, Theorem 1.2.1 states that if a signal has bandwidth B, then it is completely specified by sampling at rate $f_s = 2B$. For an arbitrary bandlimited signal, this sampling rate is also necessary, otherwise aliasing occurs and the original signal cannot be reconstructed from the samples. Sampling theorems such as these are important since most algorithms manipulate digitized signals, and digitizing an analog signal requires sampling. The bandlimited assumption is quite natural since most receivers and emitters have finite bandwidths (or more accurately, nearly finite bandwidths: at very high frequencies, signals are attenuated so much that they can be neglected).

Suppose now that a bandlimited signal only has small spectral occupancy. If these are in a few continuous regions, and are known, then down-conversion tricks (see §1.3) show that the sampling rate only needs to be at twice the amount of occupied bandwidth. A more precise, general and robust result is due to Landau [Lan67] from the 1960s. In the following, $\mathcal{F}$ denotes the Fourier transform.

Theorem 1.2.2 (Landau sampling theorem; this form modified from [ME09a]). *Define*

$$\mathcal{B}_\Omega = \{x(t) \in L^2(\mathbb{R}) \mid \operatorname{supp}(\mathcal{F}x(f)) \subset \Omega\}.$$

A set of time samples $R = \{r_n\}$ is a sampling set *for $\mathcal{B}_\Omega$ if the sequence of samples $x_R[n] = x(r_n)$ is stable; stability means $\exists \alpha > 0, \beta < \infty$ such that*

$$\alpha\|x - y\|^2 \leq \|x_R - y_R\|^2 \leq \beta\|x - y\|^2 \quad \forall x, y \in \mathcal{B}_\Omega.$$

Then a necessary condition for R to be a sampling set for $\mathcal{B}_\Omega$ is

$$D^-(R) \triangleq \lim_{t\to\infty} \inf_{y\in\mathbb{R}} \frac{|R \cap [y, y+t]|}{t} > |\Omega|$$

where $|\cdot|$ is the Lebesgue measure. The term D^- is known as the lower Beurling density.

For example, if R takes uniform samples with spacing ΔT, then $D^- = 1/\Delta T$ which is the average sampling density. If in addition $\Omega = [-B, B]$, then the theorem requires $1/\Delta T > 2B$, which is just the Shannon-Nyquist theorem.

Note an interesting consequence of the theorem: if $x(t) = \sin(2\pi f t)$ is a pure tone, then $|\Omega| = 0$ so the theorem does not provide a minimal necessary sampling rate to recover this signal. In fact, any periodic signal has a zero measure frequency support, and hence the theorem provides no bound.

For practical recovery from such a model, and sampling at the minimal rate, it is necessary to know Ω. An example of a related compressed sensing result is due to [ME09a], which, to paraphrase, says that if the frequency support Ω of X is *unknown* but $|\Omega|$ is bounded and it is known that $\Omega \subset [-B, B]$, then a necessary condition for a sampling set that stably recovers all signals in this model is that its lower Beurling density must be twice the value it would have been had Ω been a fixed, known set.

1.2.2 Estimation and the rise of alternatives to least-squares

Least-squares is the quintessential linear estimation technique; the ordinary least-squares (OLS) solution to an overdetermined system $\min_x \|Ax - b\|_2$ is $\hat{x}_{OLS} = A^\dagger b$ which is linear in b. Starting with at least Gauss, the nice properties of least-squares estimation (smoothness, convexity, closed-form expressions) have been much touted.

Particularly because of the rise of computers in the 20th century, there have been an increasing number of alternatives to linear methods. These alternatives have accumulated gradually but steadily: Dantzig's simplex algorithm in the '40s, message passing and belief propagation used for decoding (Viterbi's algorithm in the '60s), and graphical models used for inference in the past two decades, to name just a few. Linear estimation is certainly still a cornerstone technique and has produced successes like the Kalman filter. Below is a brief and partial history of some ideas that form a foundation for modern compressed sensing. Most of these are related to processing sparse signals and using ℓ_1 techniques; the idea of using incoherent measurements has much less precedent, with perhaps the exception of coded aperture receivers.

In 1965, Logan's dissertation [Log65] proved a result now known as "Logan's phenomenon": if a continuous-time band-limited signal $x(t)$ is corrupted by noise z and the noise has bounded L_1 norm and has a very sparse time support (but the support is unknown), then finding the minimal L_1 norm projection of the corrupted data can *perfectly* recover the original signal x. This is remarkable,

because it is independent of the magnitude of the error.

The discrete version of this result, using the ℓ_1 norm, was empirically demonstrated by Santosa and Symes [SS86] in 1986, and proved in [DS89]. The work of [SS86] proved that recovery of a k-sparse vector from just m Fourier measurements was possible if $m \geq (1 - 1/(2k))n$, using the unconstrained LASSO formulation (they also note the pessimistic nature of this bound). Even by 1986, the fact that "it is possible to construct a sparse spike train from part of its spectrum using the minimum ℓ_1 criterion" was "well-known (and believed)" [SS86].

The 1989 article by Donoho and Stark [DS89] extended Santosa and Symes' results by proving a more general new type of uncertainty principle (see also [DL92]). The classical time-frequency uncertainty principle says that a signal cannot have small support in both time and frequency. The [DS89] result extended this result by relaxing the restriction that a signal's support be on intervals. To be concrete, let x be a discrete signal of length n with discrete Fourier transform X. Donoho and Stark prove

$$\|x\|_0\|X\|_0 \geq n \quad \Big(\text{and hence,} \quad \|x\|_0 + \|X\|_0 \geq 2\sqrt{n}\Big). \tag{1.2.1}$$

Intuitively, the result says that there are not many discrete signals that are sparse in both time and frequency. Thus, if a signal happens to be sparse in both time and frequency, it has no "neighboring" signals, and for this reason it is easy to recover. The multiplicative inequality is sharp, because if $n = d^2$, then a Dirac comb of spacing d has $\|x\|_0 = \|X\|_0 = d$; this example is also sharp for the additive identity (the additive identity follows from the multiplicative one by the arithmetic-geometric mean inequality). Another example is when x is a single Fourier element, in which case $\|x\|_0 = n$ and $\|X\|_0 = 1$, so the multiplicative inequality is sharp but the additive inequality is not. It turns out the additive inequality is only sharp in cases such as $n = d^2$; the situation when n is prime is quite different [Tao05].

[DS89] also reports numerical experiments suggesting that much stronger results are possible if the sparsity is "scattered" in a random way, so it is clear that the authors had some insight into what was achievable; their official bound on measurements was the same as Santosa and Symes. It is worth mentioning that their numerical experiments went only as large as $n = 256$, and it is not easy to distinguish $\mathcal{O}(n)$, $\mathcal{O}(\sqrt{n})$, and $\mathcal{O}(\log n)$ growth from experiments with just $n = \{64, 128, 256\}$.

This theory in the late 1980s built on empirical work from the previous 15 years, as different scientific communities were using ℓ_1 minimization and/or sparsity. The work in [CM73] in 1973 argues that the geophysics community should consider ℓ_1 regularized problems for the sake of robustness to outliers, using the mean and median estimators as examples. They also argue that ℓ_1 generates sparsity, but only to the extent that there is always a vertex solution to a linear program (so sparsity of a solution can be chosen less than m). In addition to sparsity in space, they give

examples of using sparsity in first- and second-differences; this was further explored in the highly influential total-variation (TV) minimization paper in 1992 [ROF92].

Similar works in geophysics, such as [TBM79], cite ℓ_1 studies going back to 1964. Incremental results continued in geophysics, exemplified by [OSK94] in 1994 which cites numerous ℓ_1 results from the 1980s (but appears to be unaware of [SS86, DS89]). Many of these works proposed special algorithms to avoid recasting the problem as a linear program (LP).

In radio astronomy, the CLEAN algorithm [Hög74], introduced around 1971, exploits sparsity of radio sources and is similar to a matching pursuit algorithm [MZ93] as shown by [WJP+09], though it is not equivalent to ℓ_1 minimization (indeed, the first analysis of CLEAN [Sch78] compares it to ℓ_2 minimization). Its idea of exploiting sparsity and post-processing has been extremely useful in astronomy; to quote from [Cor09], "The impact of CLEAN on radio astronomy has been immense. First, there is the accumulated science from the telescopes that have used CLEAN—GBI, MERLIN, WSRT, VLA, VLBI, etc. ... Second, by showing what could be achieved with some postprocessing, CLEAN has encouraged a wave of innovation in synthesis processing that continues to this day."

The article [PC79] introduces the authors' algorithms from 1975 to the application of estimating the transfer function of an unknown medium using ultrasound pulses, and exploits sparsity in the number of layers in most mediums. The algorithm they propose is similar to alternating projections between the time domain (which is sparse; this set is not convex) and the frequency domain (where it is assumed that reliable data exists in some band). The 1982 article [ME82] discusses an application of ℓ_1 minimization for improving diffraction-limited images, and also suggests its use in video compression.

A noticeable similarity in all these works is that they almost exclusively work with deconvolution, and hence operate in the Fourier and time/spatial domains.

1.2.3 Leading up to compressed sensing

Another key milestone in the work preceding compressed sensing was a paper by Donoho and Huo in 1999 [DH01], and was followed by several papers in the next five years, including [DE03, EB02, GN03, Tro04, DET06, Tro06]. These results concern sparse approximation: suppose a signal $x \in \mathbb{C}^n$ can be represented as $x = \Psi\alpha$ where $\Psi = (I, \mathcal{F})$ contains "spikes and sines" ($\mathcal{F}$ is the DFT). Since Ψ is rectangular, there are infinitely many α which satisfy this. The result of [DH01] is that if there is some $\|\alpha_0\|_0 < \frac{1}{2}\sqrt{n}$ and $x = \Phi\alpha_0$, then ℓ_1 minimization

$$\min_{\alpha} \|\alpha\|_1 \quad \text{such that} \quad x = \Phi\alpha$$

will return α_0. Similar results hold for solving by Orthogonal Matching Pursuit (OMP; see §1.5) [Tro04]. These results were generalized to other bases using a convenient tool called the coherence (1.2.2).

The result means that a signal $\alpha \in \mathbb{C}^{2n}$ of length $N = 2n$ which is k sparse, with $k < \frac{1}{2}\sqrt{n}$, can be recovered from $m = n$ entries. In terms of the dimension N and measurements m, the sparsity must obey $k < \frac{1}{\sqrt{2}}\frac{m}{\sqrt{N}}$. This $\sqrt{N}$ sparsity may not seem very sparse to readers with knowledge of recent results, who might expect that sparsity can grow with rate $k \simeq \frac{m}{\log N}$, which only weakly depends on N. But as proved in [DH01], these results are sharp. Other than special cases (such as n prime; see [Tao05]), there are counter examples showing that better bounds cannot be obtained, such as with the Dirac comb. So how does compressed sensing improve these bounds? The key difference is that compressed sensing bounds typically do not hold for *all* measurement matrices, but rather hold with overwhelming probability[1] or with high probability[2].

1.2.4 Compressed Sensing

This section gives a brief review of basic CS theory; for more details, the short review article [CW08] is recommended.

Let a signal $x(t) = \sum_{i=1}^{n} \alpha_i \psi_i(t)$ where Ψ is the matrix with columns ψ_i. We deal mainly with discrete, finite-length samples, so $x \in \mathbb{R}^n$, and Ψ is a $n \times n$ matrix, typically chosen to be orthogonal. For example, the signal x might just be a sparse vector, so then Ψ can be the identity matrix I.

Let Φ be another orthogonal $n \times n$ matrix; we will eventually subsample this to create the $m \times n$ "sampling matrix"; outside of the introduction, it is always assumed that Φ is $m \times n$ with $m < n$, and except for the NESTA algorithm in Chapter 3, the rows of Φ will not need to be orthogonal. Since Φ and Ψ are both orthogonal, and therefore invertible (and well-conditioned), it is possible to recover x (or equivalently, α) from samples

$$b = \Phi x = \Phi\Psi\alpha, \quad \text{or} \quad b = A\alpha, A \triangleq \Phi\Psi.$$

So far, this is not remarkable, and there is no benefit to doing so (even in the noisy case, since Gaussian white noise is unchanged under orthogonal transformations).

The remarkable results from compressed sensing say that if α is sparse (we typically use k to denote the sparsity, and say that α is "k-sparse") and Φ and Ψ are incoherent, then it suffices to only take a little more than k rows of the Φ matrix in order to stably reconstruct x.

Define the *coherence* of Ψ and Φ to be

$$1 \leq \mu(\Phi, \Psi) \triangleq \sqrt{n} \max_{1 \leq k,j \leq n} |\langle \psi_k, \phi_j \rangle| \leq \sqrt{n}. \tag{1.2.2}$$

The two ortho-bases are "incoherent" when $\mu \simeq 1$ or has a weak-dependence on the dimension n.

[1] Used here, this is a technical term (see [Tao11]); if a statement or event E_n depends on n, then E_n holds *with overwhelming probability* if $\forall \alpha > 0$ we have $\forall n, \mathbb{P}(E_n) \geq 1 - c_\alpha n^{-\alpha}$ where c_α is a constant independent of n; in compressed sensing, n is the dimension of the signal.

[2] Similarly, we say E_n holds *with high probability* if $\exists \alpha > 0$ such that $\mathbb{P}(E_n) \geq 1 - cn^{-\alpha}$.

Complete coherence, $\mu = \sqrt{n}$, is obtained if, for example, $\Psi = \Phi$. Maximal incoherence, $\mu = 1$, is obtained for $\Psi = F$, the Fourier basis, and $\Phi = I$, the identity. One remarkable fact is that for *any* fixed basis Ψ, if Φ is chosen from the general orthogonal ensemble (GOE), the coherence of Φ and Ψ is $\sqrt{2\log n}$ with high probability. Thus random matrices Φ are universally good measurement matrices. Intuitively, we want low coherence so that each row of Φ takes an equal amount of information about the signal. If Φ and Ψ are very coherent, then each row of Φ only tells us how much of one particular basis element ψ_i is present in the signal, and thus we would need all n measurements, since leaving out the ith measurements is taking the very risky gamble that $\alpha_i = 0$. See Figure 1.2 on page 4 for a graphical depiction.

An early result [CR07a] is that if

$$m \geq C\mu^2(\Phi, \Psi)k \log n$$

then if α is k-sparse and Φ_m is Φ restricted to m rows chosen uniformly at random, then with overwhelming probability, the measurements $b = \Phi_m x$ are sufficient to *tractably* recover x. The recovery is tractable because it can be done by solving basis pursuit (1.1.5), which can be solved in polynomial time. Here, C is a constant that is independent of k and n.

Very recent results [CP10a] have extended this, which we briefly paraphrase. For simplicity, let A incorporate both Φ and Ψ, $A = \Phi\Psi$. Write the rows of A as a_i^T, and assume each row is drawn iid from some probability distribution P, and that each row has zero mean and identity covariance matrix (so in particular, all the entries $a_{i,j}$ of A are uncorrelated, and furthermore entries from different rows are independent). Define the self-coherence to be the smallest number μ_s such that for all rows a_i,[3]

$$\max_{1 \leq j \leq n} |a_i(j)|^2 \leq \mu_s. \tag{1.2.3}$$

If P is a sub-Gaussian distribution, then $\mu = c \log n$ for some c independent of n, and if P is sub-exponential, $\mu = c \log^2 n$.

From now on we'll assume x is the sparse object, e.g., $x = \alpha$. The following theorem differs from other compressed sensing results in that we assume x is a *fixed* k-sparse vector. This is a stronger assumption than before because for any given realization of a sampling matrix A, we can no longer pick an adversarial signal (and one might argue that since adversarial signals rarely arise in practice, this stronger assumption is palatable).

Theorem 1.2.3 (due to [CP10a])**.** *Let x be a fixed k-sparse vector, and take m measurements $b = Ax$. If $m \geq (1+\beta)C\mu_s k \log n$ then with probability $1 - 5/n - e^{-\beta}$, basis pursuit* (1.1.5) *will recover x.*

[3] This need not hold deterministically; see [CP10a] for details.

This can be extended to cover compressible signals and noisy measurements. Let x_k denote the best k-term approximation to x; in loose terms, we say x is compressible if there is a small k such that $\|x - x_k\|_2$ is small. In particular, if x is k-sparse, then $x = x_k$. We present a simplified theorem below:

Theorem 1.2.4 (due to [CP10a]). *Let $x \in \mathbb{R}^n$ be arbitrary; in particular, it need not be sparse. Let $y = Ax + z$ where $z \sim \mathcal{N}(0, \sigma^2)$, and define $\lambda = 10\sigma\sqrt{\log n}$. Then if*

$$m \geq (1 + \beta)C\mu_s k \log n$$

then with probability at least $1 - 6/n - 6e^{-\beta}$, the LASSO (1.1.3) will recover an estimate $\hat{x}$ satisfying

$$\|\hat{x} - x\|_2 \leq \tilde{C}(1 + \log^2 n)\left(\frac{\|x - x_k\|_1}{\sqrt{k}} + \sigma\sqrt{\frac{k \log n}{m}}\right).$$

Identical results hold for the Dantzig selector (1.1.7).

To make these results meaningful, we compare to an "oracle". Similarly to how an unbiased estimator's performance can be compared to the optimal performance given by the Cramér–Rao bound, comparing with an oracle gives an idea of how close this performance is to the optimal performance. Consider recovery of a k-sparse vector x given noisy observations $y = x + \sigma z$, where z is any noise vector with independent entries and unit variances. Suppose an oracle whispers in your ear that the nonzero entries of x are precisely the first k. Then a quite reasonable strategy (though not optimal in terms of mean-square error, as Willard James and Charles Stein famously showed) is to take $\hat{x}_i = y_i$ for $i \leq k$ and $\hat{x}_i = 0$ otherwise; in fact, with this information, we only need to take k measurements. The expected error is $\mathbb{E}\,\|\hat{x} - x\|^2 = k\sigma^2$, and this is referred to as the oracle error.

The oracle takes k measurements and the error is $k\sigma^2$. The above theorem says that without the benefit of knowing *a priori* which entries are nonzero, then if we take slightly more measurements (by a factor of about $\log n$), we can do almost as well, with an error that is worse by a factor of about $\log^5 n$. Theorem 1.2.6 will improve this to a constant factor, and in Chapter 2 we show empirically that this factor is about 3.

For completeness, we cover what is now the *de facto* "standard model" of compressed sensing, which relies on a tool called the RIP:

Definition 1.2.5 (RIP (Restricted Isometry Property)). *The restricted isometry constant δ_k of order k for a matrix A is the smallest number such that*

$$(1 - \delta_k)\|x\|_2^2 \leq \|Ax\|_2^2 \leq (1 + \delta_k)\|x\|_2^2$$

holds for all k-sparse vectors x.

For example, if A is an orthogonal matrix, then A is an isometry, and $\delta_k = 0$ for any k, which is the best possible constant. If A is $m \times n$, then it has a non-trivial nullspace, so $\delta_k \geq 1$ for $k \geq m$, but it is still possible for $\delta_k < 1$ for $k < m$. In general, adding rows to a matrix A will improve (i.e., decrease) its RIP constant. The literature often refers informally to a "matrix having the RIP" whenever that class of matrices satisfies $\delta_{\gamma k} \ll 1$ with high probability whenever $m \simeq k \log(n)$, where γ is typically 2 or 3. There are many variations of results using the RIP, but we show one of the most compact results.

Theorem 1.2.6 (due to [Can08]). *Let $\delta_{2k} < \sqrt{2}-1$ and $b = Ax + z$ for any (possibly deterministic) z, with $\|z\|_2 \leq \varepsilon$. Then basis pursuit denoising* (1.1.6) *with this ε gives an estimator $\hat{x}$ satisfying*

$$\|\hat{x} - x\| \leq C_0 \frac{\|x - x_k\|_1}{\sqrt{k}} + C_1 \varepsilon$$

for some constants C_0 and C_1 (which are not unreasonably large).

The theorem is deterministic, but unfortunately, there are few deterministic matrices which have small RIP constants. For a fixed number of rows m, random matrices have the best known RIP constants (although these bounds are not deterministic, and hence there is a nonzero chance of failure). The bound $\delta_{2k} < \sqrt{2}-1$ can be slightly improved, but this has little qualitative significance to our study here.

There are different bounds for random matrices, but we present here a paraphrase of an elegant result that sums up the general idea:

Theorem 1.2.7 (due to [BDDW08]). *Let the entries of the $m \times n$ matrix A be drawn i.i.d. according to either a zero-mean Gaussian or $\{+1, -1\}$ Bernoulli distribution, and scaled so the expected value of each column is 1. For any k and δ, with overwhelming probability (which depends on δ) the matrix A will satisfy $\delta_k(A) \leq \delta$ as long as*

$$m \geq Ck \log(n/k)$$

for some constant C independent of n (but dependent on δ).

Note that there is a qualitative difference between the $\{+1, -1\}$ Bernoulli matrices (also known as signed Bernoulli, or Rademacher) and the $\{0, 1\}$ Bernoulli matrices in their compressed sensing performance; the simple shift makes a big difference. In fact, $\{0, 1\}$ matrices do not obey the RIP unless m scales with either k^2 or n (as opposed to $k \log n$) [Cha08]. This does not mean that the matrices are useless for CS, rather just that RIP-based techniques fail; for example, types of expander graphs [JXHC09] fail the RIP but still have probable recovery results. However, in some applications, non-negative measurement matrices induce problems; for example, the micro-mirror optical systems that work in either on- or off-states are affected by this non-negativity due to an increase in noise; see the discussion in §1.1.

Figure 1.8: Group testing. Suppose N soldiers are tested for a disease using a sensitive but expensive blood test; if blood from different soldiers is grouped together, the test is sensitive enough to declare a "positive" if any of the contributed blood samples were positive. If k soldiers are infected, how many measurements M are necessary? If k is large, then $M = N$ is the best we can do. If $k = 1$ (and k is known *a priori*), then $M = \log_2(N)$ is possible (with or without adaptive measurements). The key idea is to take global measurements and then use reasoning. In this pictorial example, soldier # 45 tests positive, and 45 is 101101 in binary, so using the binary tests depicted by the gray shading, and assuming $k = 1$, we can deduce that this is the infected soldier. Each row represents one test, which involves the columns that are gray. See [GS06a] for more info. This was first proposed in the 1940s by two economists helping the military screen draftees for syphilis; their proposal was to group together the blood of five men and apply the test. Since few men were likely to have syphilis, this saves tests on average. The test was never put into practice, in part because the test was barely sensitive enough to detect a positive hit from diluted blood. The subject was revived again in the late 1950s.

Another useful result of CS is that random rows of a discrete Fourier matrix also work well for sampling when the signal is sparse. This is significant since the fast Fourier transform allows one to compute measurements faster than simply taking a matrix-vector product. The number of required measurements is similar to that of the Gaussian ensemble case, except the $\log(n)$ factor is $\log^5(n)$; this bound may be lowered in the future, so it is not necessarily indicative that Fourier matrices are inferior to Gaussian matrices in practice.

To summarize the intuition behind compressed sensing: if a signal has a sparse representation, then typically only $\mathcal{O}(k \log n)$ measurements are necessary (this is an intuitive number: for each of the k nonzeros, we need some constant number of measurements to encode the amplitude, and $\log n$ number of measurements to encode the location of the entry).

The other intuition is that for a noisy signal, the undersampling hurts us, but only by an amount proportional to m/n. This is a fundamental limitation: to accurately estimate a quantity given noisy measurements (say, estimate the population mean given the sample mean), it is always beneficial to take more measurements.

Our final remark is that because random sensing matrices have good properties, CS tells us they are universal encoders. Furthermore, the measurements are not adaptive. Clearly, well-chosen adaptive measurements are not detrimental, but perhaps they are not as beneficial as one might expect. For example, consider the case of group testing [GS06a, DH93] depicted in Figure 1.8, with n samples and k samples that test "positive". We wish to find the k positives. If $k = 1$ and k is known, then one obvious solution is an adaptive binary tree scheme (aka bisection method) which will take $\log_2 n$ measurements. But a non-adaptive scheme that groups the measurements according to their binary representation also takes $\log_2 n$ measurements, so there is no advantage to the adaptive scheme. If k is larger and unknown, then the adaptive scheme is better, but not by too much.

Alternatives to RIP. The RIP is a convenient tool, but so far it has not proven to be the perfect tool to analyze deterministic matrices. Most undesirably, it is generally NP-hard to check if the RIP holds for a specific matrix. Furthermore, the RIP is not a necessary condition. It is clearly a useful tool, but because of the importance of CS, there has been work on alternatives. We mention just a few of these alternatives here, in addition to the coherence property which was already discussed.

The work in [BGI+08] introduces a simple extension of the RIP to the so-called RIP-p, and in particular the RIP-1. The RIP-p is stated as the RIP is, except using ℓ_p norms instead of the ℓ_2 norm; thus RIP-2 is the standard RIP. The RIP-1 is particularly useful for expander graphs [JXHC09] and other sparse encoding matrices. Another extension of the RIP is model-based RIP [BCDH10] which concerns all x restricted to a specific model; if the model is the set of k-sparse signals, then this is just the usual RIP. A variant on this is the restricted-amplification property [BCDH10].

Another approach is that taken by Donoho in his original paper [Don06]; see [DT10] for an overview. Results are proven using combinatorial geometry, and using results on Gel'fand n-widths. The approach in [XH11] is a simple nullspace condition that is both necessary and sufficient, and uses Grassmannian angles to prove that matrices satisfy the condition. A complicated hierarchy of conditions, some of them implying others, is collected and discussed at Terence Tao's website [Tao] for those seeking further information.

1.3 The need for the RMPI

The random modulation pre-integrator (RMPI) is an analog-to-digital converter (ADC). Technically, the RMPI also combines a RF front-end so it is also a receiver, but its novelty resides in its approach to ADC. The RMPI is part of a new class of so-called "analog-to-information converters" (which is abbreviated variously as A2I or AIC). The purpose of an ADC is to digitize analog information so that it may analyzed and/or stored on a computer. For example, when recording music, an ADC is used to sample the input of a microphone and convert to a digital bit-stream. This is actually an example of an application that could benefit from CS, since CD-quality recordings require sampling at 44.1 kHz in order to capture frequencies up to about 20 kHz. The bit rate is 1411.2 kbit/s (e.g., two stereo 16-bit channels), yet on a computer, this is typically encoded by MP3 (or improvements, such as AAC) to 128 kbit/s, throwing away data.

Sampling audio-frequency sounds in the kHz range is not difficult, but sampling wide-band radio in the GHz range is. As a rule, ADC performance degrades as a function of the sampling speed. The most fundamental measure of an ADC performance is the number of bits $\tilde{B}$ of its output, meaning that the digitized output takes on $2^{\tilde{B}}$ possible values.

To describe both the bits and distortion, the effective number of bits (ENOB) term is used. This comes from the error (peak SNR) that an *ideal* $\tilde{B}$-bit ADC incurs due to quantization. Let the

desired sampling rate be f_s (i.e., the Nyquist rate), and suppose the ADC samples at γf_s, where $\gamma \geq 1$. The peak SNR of the ideal ADC is $\text{SNR}_p = \frac{3}{2}2^{2\tilde{B}}\gamma$ for a sinusoidal signal. Converting this to dB gives $1.76 + 6.02\tilde{B} + 10\log_{10}(\gamma)$ [GSS02].

For a non-ideal ADC, the SNR (signal-to-noise-ratio) or SINAD (signal-to-noise-and-distortion) may be different. The ENOB is thus an expression of the equivalent number of bits in an ideal ADC that would achieve the same SNR. In the following, we use SNDR (signal-to-noise-and-distortion-ratio) that accounts for all noise and distortion. Hence for $\gamma = 1$

$$B \triangleq \text{ENOB} = \frac{\text{SNDR}_{dB} - 1.76}{6.02}. \tag{1.3.1}$$

This is convenient because it incorporates distortion and the number of bits.

In the 2005 review [LRRB05], which supersedes the widely cited but now outdated review by Walden [Wal99], two useful figures-of-merit are discussed:

$$P = 2^B f_s \quad \text{and} \quad F = \frac{2^B f_s}{P_{\text{diss}}} \tag{1.3.2}$$

where P_{diss} is the power dissipation of the ADC. Walden suggested that the F improves over time due to more efficient power management, but that P is rather flat. Reviewing newer technology, [LRRB05] finds that P is not flat, but has slightly improved when f_s is slow.

There are many types of ADC: sigma-delta, folding, half-flashed, pipelined, SAR, and flash. The fastest ADC are of flash type, and reach about 1 GHz sampling rate, with about 7 ENOB; they are limited to a maximum of about 8 ENOB due to non-linearities. The ENOB for an ADC of 1 MHz is much better, going up to about 15 or 16. Fast ADC with high ENOB and accuracy are greatly desired. The cell phone revolution has spurred receivers to routinely deal with 100 dB dynamic range, and hence they need high accuracy, while the growing trend toward spread-spectrum RF systems also requires systems with high bandwidth.

Among current applications of high-rate ADC is the sensing of wide-band signals, such as several GHz of radio frequency bandwidth. This is of particular interest to the military, who seek to monitor these frequencies for signals of interest; this is referred to as SIGINT (signal intelligence) in military lingo. One application is detection of foreign radar pulses. This is quite different than usual radar, since it involves detecting radar pulses from *other* emitters, so the characteristics of the signal are not known. To prevent jamming, radar pulses may "frequency hop", meaning that the carrier frequency is changed over time. The pattern of carrier frequencies is known to the foreign radar emitter, but unknown to our third-party receiver.

Basics of heterodyne receivers. One of the most fundamental tools for designing a receiver is the principle of heterodyning. In heterodyning, a low-bandwidth signal (e.g., human voice, or

a radar pulse envelope) is mixed (i.e., multiplied in the time domain) with a carrier sinusoid for the purposes of transmission. The receiver then mixes with the same carrier and filters in order to recover the low-bandwidth signal.

To see how this works mathematically, let $a(t)$ be the low-bandwidth signal, and ω_c be the frequency of the high-rate carrier. Then the transmitted signal is

$$x(t) = a(t)\sin(\omega_c t).$$

For simplicity, let $a(t) = \sin(\omega_s t)$ where $\omega_s \ll \omega_c$ is a slow frequency. Using trigonometric identities, we express $x(t)$ as

$$x(t) = \sin(\omega_s t)\sin(\omega_c t) = \frac{1}{2}(\cos((\omega_s - \omega_c)t - \cos((\omega_s + \omega_c)t)).$$

To recover a, one multiplies the received signal $x(t)$ by another carrier of $\sin(\omega_c t)$. Then using the same trigonometric identities,

$$\begin{aligned}\sin(\omega_c t)x(t) = \frac{1}{4}(&\sin((\omega_c - (\omega_s - \omega_c))t) + \sin((\omega_c + (\omega_s - \omega_c))t) + \\ &- \sin((\omega_c - (\omega_s + \omega_c))t) - \sin((\omega_c + (\omega_s + \omega_c))t) \\ = -\frac{1}{2}&\sin(\omega_s t) + \frac{1}{2}\sin(\omega_s t)\cos(\omega_c t).\end{aligned}$$

When this new signal is low-pass filtered to frequencies below $\omega_c - \omega_s$, all that remains is $\frac{-1}{2}a(t)$ because $\omega_c \gg \omega_s$ and so only $a(t)$ is in the passband of the filter.

This means that if a signal has a known carrier frequency $f_c = 2\pi\omega_c$, then it can be "down-coverted" by mixing with $\sin(\omega_c t)$ at the receiver. Thus the bandwidth constraint depends only on $a(t)$.

For SIGINT applications, f_c is not known in advance, which complicates the system. The receiver needs to cover the entire range of possible carrier frequencies. As mentioned earlier, ADCs can only go up to one GHz, and do so at considerable expense of ENOB, but a frequency hopping signal might easily range over two or three GHz. The conventional approach is to channelize the receiver: a received signal is split into different paths or channels, then each channel k mixes the signal with a carrier frequency ω_k and then low-pass filters the result. The effect is that each channel views a small section of the spectrum. The clear downside of this approach is that it requires many channels, which requires more power and complexity. Consider a system to monitor about 2.5 GHz of bandwidth. Typical hardware uses 50 MHz channels, so it requires 50 channels! Since each channel is 50 MHz, the ADCs must sample at 100 MHz; a typical high ENOB 100 MHz ADC optimistically requires about 1 Watt of power, so the whole receiver requires about 50 Watts. In contrast, the design we

propose in Chapter 2 consumes about 1 Watt and covers the same bandwidth.

Because of the huge power requirements of current high-bandwidth receivers, in 2007 the DARPA called for proposals of innovative ADC designs [Hea07]. A joint Caltech and Northrop Grumman Corporation (NG, or NGC) team answered this proposal and in 2008 began the "A2I" project after completing a preliminary "Phase 0" proof-of-concept study. The on-site Caltech team consists of Professors Emmanuel Candès and Azita Emami, and graduate student Juhwan Yoo and this author. Further signal processing help is provided by the extended Caltech team, consisting of Dr. Michael Grant, and Professors Michael Wakin and Justin Romberg and their graduate students. The Nothrop Grumman team is based at the Northrop Grumman Space Technologies division, and led by Dr. Emilio Sovero, with chief designer Dr. Eric Nakamura.

The A2I project consists of three devices:

1. Caltech Random Modulator Pre-Integrator (RMPI). This is an eight channel RMPI device built in complementary metal–oxide–semiconductor (CMOS) by the Caltech team. It is designed to capture radar pulses over a 2.5 GHz bandwidth.

2. NG RMPI. This is a four channel RMPI device built in indium-phosphide (InP) by the Northrop Grumman team, and designed as a backup to the Caltech design. The design goals are the same as the Caltech design.

3. NG Non-Uniform Sampler (NUS). This is built by Northrop Grumman, and is a specialized ADC that samples at irregular time intervals. It is designed to capture signals that are longer in duration than radar pulses, and specifically, the GSM cell phone band. It has a 1.2 GHz bandwidth.

Signal processing techniques are shared between the Caltech and NG teams, as are some hardware plans.

At various stages, the team met with DARPA and the program manager (first, Professor Dennis Healy, and later, Dr. Daniel Purdy) to review progress. Several other teams, such as a joint Rice University and Applied Signal Technology team headed by Richard Baraniuk, answered the same proposal and are working on similar projects. Other groups include a team at HRL led by Peter Petre, a team at L-3 lead by Jerry Fudge, and a team at Texas A&M led by Sebastian Hoyos. Most of these projects focus on frequency sparsity, whereas the RMPI attempts to exploit time-frequency sparsity, which is more complicated.

The review meetings allowed teams to interact, and consequently some ideas have spread throughout all the teams. For example, the Hoyos team has a planned design that is quite similar to the RMPI presented here: 8 channels, seeking to cover 1.5 GHz of bandwidth.

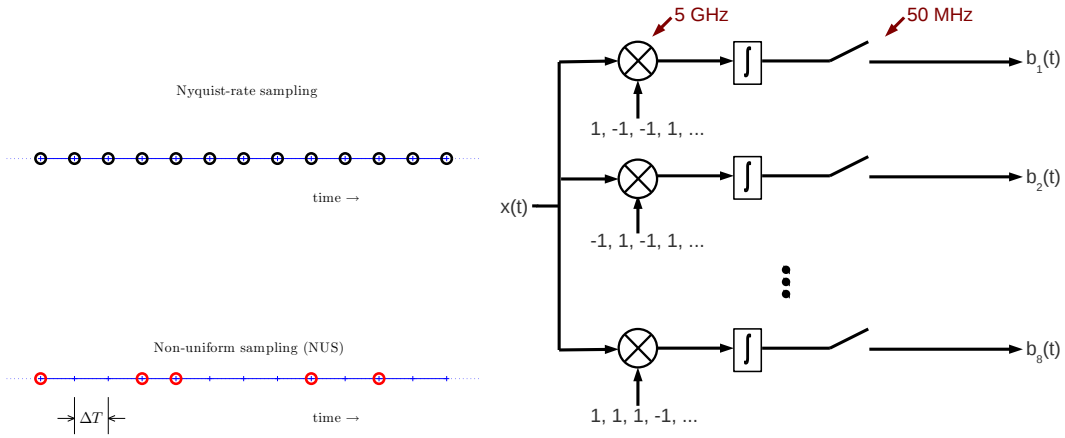


Figure 1.9: Pictoral representations of NUS (left) and RMPI (right). In the NUS, the large circles represent time-domain samples. The NUS samples at irregularly spaced intervals, though always in multiples of ΔT. In the RMPI, each channel mixes with a unique pseudo-random bit sequence (PRBS), and then integrates and samples.

1.4 Principles of the RMPI

In this section, the high-level operating principles of the RMPI are discussed. First, we introduce the NUS, which is conceptually simpler, and also simpler to build.

1.4.1 The NUS

The non-uniform sampler (NUS) acts as an ADC that samples at irregularly spaced time intervals. If the Nyquist rate is f_s, then a traditional ADC samples a signal $x(t)$ at regularly spaced time intervals $t_1, t_2, \ldots$ where $t_{i+1} - t_i = \Delta T = 1/f_s$. Limiting our discussion to a finite time window T, there are then $N = T/\Delta T$ Nyquist samples. Let x be the vector of samples $x[k] = x(k\Delta T)$, so the vector of conventional measurements can be written

$$b_{\text{conventional}} = Ix + z$$

where z is a noise vector and I is the $N \times N$ identity matrix. See the upper left of Figure 1.9 for a cartoon representation.

The NUS takes only a subset of these samples. Let Ω be a subset of $\{1, \ldots, N\}$, and let the cardinality of Ω be written as M; sometimes we may abuse notation and also refer to Ω when we really mean the time samples indexed by Ω. Write S for the matrix formed by keeping only the rows of I that have an index in Ω, so S is a $M \times N$ matrix. Then

$$b_{\text{NUS}} = Sx + z.$$

For example, if $\Omega = \{2, 4, 6, 8, \ldots\}$, then the NUS would be sampling at $2\Delta T$. This is *not* a

desirable choice for Ω, because there is no way to distinguish between a tone at frequency f and a tone at frequency $f + f_s/4$: both tones look identical if sampled on Ω. This can be remedied by replacing the 2 entry in Ω with either a 1 or 3. However, this change does not guarantee that arbitrary signals can be reconstructed, it merely addresses this particular ambiguity.

One of the first compressed sensing results [CRT06] is that the Fourier and time domains are usually extremely incoherent. Even though there are some bad sampling schemes (such as sampling at $2\Delta T$), *random* sampling in one domain means that coherence is likely small. If the signal x is sparse in frequency, then the time samples Ω should be chosen randomly. Hence the NUS chooses Ω via a pseudo-random number generator (note that Ω obviously must be known when trying to reconstruct the signal). The authors in [HH08] nicely state the intuition: "random undersampling renders coherent aliases into harmless incoherent random noise, effectively turning the interpolation problem into a much simpler denoising problem."

This is almost the entire story: the final complication is that hardware limitations impose some restrictions on Ω so it cannot be chosen completely randomly. Note also that the samples are not chosen uniformly over the time period $[0, T]$, but chosen uniformly from the discrete time grid $t_1, t_2, \ldots, t_N$. This is because it is more practical for hardware implementation and calibration, and because the theory is well understood (in terms of coherence between the Fourier and impulse discrete bases). Continuous-time compressed sensing theory has been developed to a limited extent but is certainly not as mature as discrete-space (finite dimensional) CS. See Figure 1.9 (left) for a depiction of the NUS.

What is the advantage of a design like NUS over a conventional ADC? The NUS might take samples spaced only ΔT apart, so at its fastest, it still needs to sample at f_s. However, the *average sampling rate* is M/N. For applications of interest, the spectrum is typically sparse, so it is possible to severely undersample with $M < N$. By using hardware tricks (beyond the scope of this thesis) it is possible to exploit this lower average sampling rate. This has the net effect of obtaining higher resolution samples than would be possible using an ADC sampling directly at f_s.

1.4.2 The RMPI

The RMPI system is designed to capture radar pulses; see Figure 1.9 (right) for a block diagram. A radar pulse consists of a pulse envelope, which is nearly trapezoidal except smoothed at the corners, which is modulated by a high-frequency carrier signal. The relevant features of a radar pulse is that it is sparse in frequency, since the bandwidth of the pulse envelope is typically not great (e.g., 10 MHz), and it is also sparse in time, since the pulse window is of finite duration. Radar signals typically repeat (on the order of 10 kHz), but the gaps between the pulses is greater (typically much greater) than the length of the pulses themselves, so the sparsity assumption is still valid.

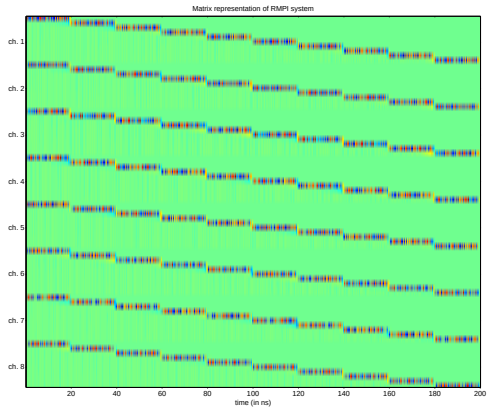


Figure 1.10: Sample measurement matrix. The green entries are zero or nearly zero. This particular matrix has rows ordered such that the measurements from each channel are grouped together.

Why not use NUS? The NUS operates on the principle that sparsity in the time domain and sparsity in the frequency domain are mutually exclusive. It takes non-uniform time samples, which are almost always incoherent with pure tones. On the other hand, the RMPI is designed to detect pulses that are also sparse in the time domain. If a short-duration pulse is sampled non-uniformly in time, it will only appear in some of the measurements, which is clearly inefficient. Viewed another way, the pulses are sparse in a Gabor time-frequency dictionary (see §2.7.2.2) which is not maximally incoherent with the impulse domain.

Operating principles of the RMPI. What *is* incoherent with the Gabor time-frequency dictionary? Another early result of CS [CT06] is that an instance of a *random* sensing matrix with iid entries from a Gaussian or signed Bernoulli distribution is overwhelmingly likely to be incoherent with a sparse signal in *any* fixed basis. The Gabor dictionary is not a basis, so these results do not directly apply; for example, the theory in [TLD+10] assumes sparsity in the Fourier domain, though recent work [CENR11] may help extend this to work with the Gabor dictionary. However, the intuition is that a Gaussian or Bernoulli matrix is a good sensing matrix. We note that these are not the only random matrices with good properties, but as they are particularly simple we focus on them.

The goal of the RMPI design is to approximate a random signed Bernoulli matrix as closely as possible. Due to practical considerations, the RMPI ends up approximating a block Bernoulli matrix. We motivate the RMPI with this approach for now, though in Chapter 2 it will also be motivated in a frequency-domain viewpoint.

Consider a Bernoulli matrix Φ that has r_{ch} rows and N_{int} columns. For example, if $r_{\text{ch}} = 3$ and

$N_{\text{int}} = 8$, it might look like

$$\Phi = \begin{pmatrix} 1 & -1 & -1 & -1 & 1 & -1 & -1 & -1 \\ -1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 \\ -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 \end{pmatrix}.$$

Suppose $x(t), t \in [0, T_{\text{int}}]$, is represented by a vector $x \in \mathbb{R}^{N_{\text{int}}}$. For now, let this representation be obtained by averaging:

$$x[n] = \frac{1}{\Delta T} \int_{(n-1)\Delta T}^{n\Delta T} x(t)dt.$$

In future sections, digital representations are usually point samples $x[n] = x(n\Delta T)$. For low-frequency samples, there is not much difference in the two representations.

With this representation, the action of the first row of Φ applied to x is just

$$b_1 = \sum_{n=1}^{N_{\text{int}}} p_n x[n] = \sum_{n=1}^{N_{\text{int}}} p_n \frac{1}{\Delta T} \int_{(n-1)\Delta T}^{n\Delta T} x(t)dt = T_{\text{int}} \int_0^{T_{\text{int}}} x_p(t)$$

where p_n is the n^{th} column of the first row of Φ, and x_p is the signal x after mixing with p:

$$t \in [(n-1)\Delta T, n\Delta T) \implies x_p(t) = p_n x(t).$$

The two key components of the RMPI are thus mixing with a sequence p_n, followed by integration. The different rows of the Φ matrix correspond to different channels. Each channel is identical except that it uses a different sequence p_n so that it collects unique information.

The integrator must be sampled at some point in time T_{int}. Each channel sends its sample to an ADC which operates at frequency $f_{\text{ADC}} = 1/T_{\text{int}}$; the benefit of the RMPI is that $f_{\text{ADC}} \ll f_s$ so the ADCs do not need to be high-rate, and therefore they can be very accurate (e.g., 15 ENOB). To describe the system behavior after a long time T, multiple Φ matrices are concatenated. For example, if $T = 2T_{\text{int}}$,

$$\Phi = \begin{pmatrix} \Phi_1 & 0 \\ 0 & \Phi_2 \end{pmatrix} = \begin{pmatrix} 1 & -1 & -1 & -1 & 1 & 1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & -1 & 1 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & -1 & 1 & -1 & 1 \end{pmatrix}.$$

Note that Φ_1 and Φ_2 do not need to be the same.

For a given time period T, assuming T_{int} divides T evenly, the number of rows of Φ is $M = r_{\text{ch}} \frac{T}{T_{\text{int}}}$. Since each row is a measurement, the number of measurements can be increased by either sampling more often (decreasing T_{int}, which means increasing sampling rate f_{ADC}), or increasing the number of channels r_{ch}. For a given number of measurements per time M/T (i.e., the information rate), a decrease in channels can be compensated for by an increase in sampling rate.

For example, the Caltech design uses 8 channels and a sampling rate of $f_{\text{ADC}} = 50$ MHz. The Northrop Grumman design uses 4 channels and a sampling rate of $f_{\text{ADC}} = 100$ MHz.

For long periods of time $T \gg T_{\text{int}}$, the Φ matrix is not Bernoulli, but rather block Bernoulli. For a fixed information rate, the number of channels can be increased and T_{int} increased, so that the matrix is exactly a Bernoulli. This is expected to have better performance, but it is impractical to have arbitrarily many channels, so in practice we deal with a finite value of r_{ch}. This trade off is explored in Chapter 2. The other extreme is a single channel operating with a short T_{int}; this case is dangerously close to a sub-sampled identity matrix, which it is coherent with signals that are sparse in the time domain, so it is not a useful CS matrix.

In a real system, the matrix is only an approximation of a block diagonal Bernoulli matrix. For reasonable perturbations, this does not affect the recovery, as long as the recovery algorithm knows Φ. Estimating the true value of Φ is the subject of the calibration section in Chapter 2. The effect of perturbations will also be discussed.

Finally, we note that because the RMPI is designed to mimic a Bernoulli matrix, is is nearly optimal for any kind of compressible signal, not just time-frequency sparse radar signals! In this sense, it is much more general than NUS. The tradeoff is extra complexity in the design and recovery.

1.5 Optimization background

The main post-processing step for recovering a signal from compressed measurements is solving one of the optimization problems described in §1.1.1. The optimization chapters of this thesis will cover state-of-the-art optimization algorithms, but do not discuss in depth the alternatives to optimization-based approaches. Below is a brief survey of the various methods researchers use to attack linear inverse problems; for a readable recent review, see [TW10]. We eschew specifics here; some state-of-the-art ℓ_1 solvers, as of 2009, are compared in Chapter 3. See also the masters thesis [Pop09] and the paper [Lor09] which both review ℓ_1 solvers as of 2009.

Below is a list of categories of algorithms that are used for recovering signals from undersampled or ill-posed data; these categories are not necessarily mutually exclusive. For example, soft-thresholding algorithms fit into several of these categories, and some greedy algorithms can be viewed as convex optimization algorithms on dual problems.

- Convex relaxation methods. These are true ℓ_1 solvers.
    - — Interior-point methods (IPM). These are second-order methods that require solving a linear system at every iteration; in return for this expense, they typically need few iterations (20 to 50) to converge. The theory behind them is well understood [NN94]. General purpose solvers are SeDuMi, SDPT3, and CVXOPT. Specialized solvers are l1ls [KKB07]

and l1Magic [CR07b]. The downside of IPM is the poor scaling with dimension, and recent years has seen trends away from IPM when solving very large problems. For medium sized problems, IPM are the gold standard. We also note the useful modeling framework CVX [GB10], which uses IPM solvers.

— First-order methods. These rely only on gradient calculations. Each iteration is cheaper than those in an IPM method, but typically more iterations are required to reach convergence. However, this trade off may be worth it. The algorithms presented in Chapters 3 and 4 are of this type.

— Splitting methods. Some splitting methods, such as the forward-backward method (used in [HYZ07]), can be motivated also as gradient methods. Others, such as the Douglas-Rachford, are distinct from gradient-based approaches, and may work quite well in practice. For an example, see Figure 4.15. These methods were originally inspired by PDE splitting methods. Convergence of the Douglas-Rachford splitting is proved in [Com04].

— Homotopy/pivoting methods. For a problem like the unconstrained LASSO with parameter λ_0, these methods find all solutions for values of λ between $[\lambda_0, \infty]$. This is useful for model selection when λ is unknown. Unfortunately, these methods are typically too slow for signal processing applications. Examples of homotopy methods are LARS [EHJT04] and HOMOTOPY [OPT00] which solve the LASSO (see also [DT08]), and DASSO [JRL09] which solves the Dantzig selector.

— Coordinate descent. If done properly, this can exploit prior computation and be quite efficient. By itself, it is an extremely old optimization method; see [FHT10] for recent versions. It is related to greedy methods.

— Variants. Noiseless basis pursuit (BP) is a linear program, and so can be solved via the simplex method. The other variants, such as BPDN, are not linear programs, but can be solved by various standard optimization techniques such as sequential quadratic programming (SQP), augmented Lagrangian methods, subgradient methods, etc. These are really classes of techniques, not specific algorithms.

  In terms of compressed sensing, two algorithms deserve special mention. The first is non-linear conjugate gradient; whenever this can be used, it is likely to be fast. It is used to solve the non-convex rank minimization problem in [BM03] and recently in [MT10]. The other algorithm is the alternating direction method of multipliers (ADMM), aka alternating direction augmented Lagrangian. This method "splits" a variable into two distinct copies of itself, and then imposes a penalty function that encourages the two copies to be equal. The two copies of the variable are updated in an alternating fashion (much like the Jacobi method). Very recent theoretical work has extended the ADMM

method to updates in a Gauss-Seidel-like fashion, which crucially allows for the first ever convergence rate estimates [GM10] and proves convergence for the case when a variable has been split into more than two copies (this had been used in practice but not analyzed). An extension by the same authors proposes an accelerated variant, using Nesterov-style ideas, which converges more rapidly [GMS10]. One reason for the recent interest in ADMM is that the subproblems are tractable for nuclear-norm minimization problems, and especially one type of robust PCA problem [CLMW09]. Another interest in ADMM is that multiple splittings potentially allow for very low-communication parallel algorithms [BPC+10].

- Thresholding. Soft-thresholding can be cast in other frameworks (e.g., gradient-based, splitting). Hard thresholding is sometimes possible to cast as a solution to a convex functional using a Landweber iteration. The iterative hard thresholding algorithm itself can be analyzed on its own, and recent work in [BD09] has established noiseless recovery guarantees similar to ℓ_1 minimization under the assumption of the RIP; the main issue seems to be dealing with noisy signals.

- Iterative ℓ_1 algorithms. The canonical example is reweighted ℓ_1 [CWB08], which is a more advanced version of the adaptive LASSO [Zou06]. Other variants include iterative-support detection (ISD) [WY10] and active-set methods, such as FPC-AS [WYGZ10]. The ISD method is quite similar in principle to CoSaMP, a greedy algorithm, but can also be viewed as a reweighting algorithm that allows weights with zero value. These methods solve convex sub-problems, but the overall method itself need not be convex. Reweighting is one of the techniques used extensively in Chapter 2. So far, there is no theoretical result showing that reweighted ℓ_1 converges to a *global* minimum. Other algorithms, such as ISD, are shown to converge (locally) in the noiseless case. The algorithms considered in [Zou06, KXAH10] are both two-step methods, so convergence is not an issue; both also offer quantitative theoretical performance improvements over the LASSO. Another two-step method in [Chr09] appears to work as well. See further discussion in the reweighting §2.7.2.3 and debiasing §2.7.2.4 sections in Chapter 2.

- Greedy pursuits. These methods seek to update the signal with the best possible update, without taking into account global structure. Early methods (OMP) update one entry at a time, while more recent methods update many entries at once (CoSaMP). In statistics, this is a type of forward stepwise regression; in signal processing these are often called "pursuits" (such as matching pursuit [MZ93]); and in approximation theory and optimization they are called "greedy methods". These do not exist independently of other optimization techniques, and are closely related to variants of coordinate descent.

  The prototypical greedy pursuit is orthogonal matching pursuit (OMP) [TG07] which im-

proves on matching pursuit. OMP has recovery guarantees almost as good as ℓ_1 recovery [TG07, DW10] for the case of Gaussian matrices, and was one of the first methods to be analyzed [Tro04] after the seminal work [DH01] in the early 2000s. OMP is important and simple enough that we describe it briefly. If b are the observations and Φ is the measurement matrix, start with $x = 0$ and the residual $r(x) = \Phi x - b$. Every step consists of two parts. First, select the atom ϕ_i that is most correlated with the residual $r(x)$; second, update x by solving the least-squares problem using only the atoms that have been previously selected.

Improvements on OMP are stagewise OMP (StOMP) [DTDS06] and regularized OMP (ROMP) [NV10, NV09]. So far, the best results are from the CoSaMP method, which adds up to k entries at every step, solves the least-squares problem, and then prunes the support set; it is not unlike the iterative support detection algorithm in [WY10]. In [NT09], it is proved that the RIP implies CoSaMP has the same recovery guarantees as ℓ_1 minimization. Furthermore, the running time is fast: it has linear convergence. One downside is that you must specify the desired sparsity level (though this is not to say that the algorithm fails on compressible signals; in that case, it returns an estimate close to the best k-term approximation, just as ℓ_1 recovery does).

- Bayesian framework. This puts a prior on the signal x and attempts to solve for the maximum *a posteriori* (MAP) estimate [WR04]. The functional may not be known in closed form, which makes rigorous results difficult. The recent work on correcting for jitter (discussed in Chapter 2) [WG11] is of this type.

- Nonconvex optimization. This set of problems seeks to minimize a ℓ_p norm for $0 < p < 1$, which is non-smooth and non-convex. Other than special non-convex problems such as rank-minimization for low-rank problems [BM05], there are no known results on efficiently finding global minima (of course, non-convex techniques such as simulated annealing can be used, but these are much slower). Work by Chartrand has shown that the global solution of ℓ_p problems does enjoy recovery guarantees [Cha07].

  We also note the very interesting work [WXT10]. It shows that for strong recovery (which is what we have discussed in earlier theorems), ℓ_p recovery outperforms ℓ_1 recovery. Yet for a type of fixed-design case, called "weak recovery," which considers recovery for a fixed sign pattern of the signal, it is shown that ℓ_1 minimization is actually superior to ℓ_p minimization. They also demonstrate that ℓ_p solutions can be denser than ℓ_1 solutions.

- Brute force. The field of subset selection [Mil02] has long been concerned with minimizing the ℓ_0 quasi-norm, since common information criteria such as AIC and BIC are based in terms of ℓ_0. For small problems, brute force combinatorial search is possible, and dynamic programming

techniques may be of some use. In general, this approach is completely infeasible for large signal recovery problems.

- Belief-propagation and message-passing. These methods are well-known in coding theory and graphical modeling, and can be remarkably quick, but often depend upon special properties of the measurement matrix. The work of [BSB10] proposes a belief propagation algorithm for a class of low-density parity check (LDPC) matrices specialized for compressed sensing; similarly, the work in [SBB06] presents a fast algorithm, but relies on specialized measurement matrices. In [DMM09], an approximate message-passing algorithm is explored, but it lacks the theory of ℓ_1 minimization.

  Other ideas related to coding theory are discussed in [XH07, JXHC09] where types of expander graphs are used to encode signals. Decoding is similar to belief propagation of LDPC codes, and the running time of the algorithm is $\mathcal{O}(n \log(n/k))$ which is extremely fast. Futhermore, [JXHC09] has shown that only $k \log(n/k)$ measurements are necessary to recover a k-sparse signal, which is the same order as found for ℓ_1 minimization. The main disadvantage is that the encoding matrix must be chosen to have special properties. It is possible to deterministically construct such matrices, but, for the case of RMPI, it is not yet apparent how to construct the physical realization of an arbitrary matrix. This topic is worth pursuing, and is especially tantalizing because it may allow on-chip real-time recovery without the need for a computer backend.

- Combinatorial algorithms. Similar to the coding-theory based ideas, these use special decoding algorithms that are similar to group testing, and rely on highly structured encoding matrices. So far, they have been shown to recover sparse signals but with more measurements than for ℓ_1 minimization. Examples include heavy hitters on steroids (HHS) [GSTV07], which improves on the related earlier algorithm [GSTV06] by the same authors. A similar algorithm is the Fourier sampling algorithm [GST08].

Discussion. Of the categories discussed, the two most commonly used are the convex optimization methods (perhaps with tweaks), and the greedy methods. Most other methods are too specialized, or not yet thoroughly understood.

One of the benefits of greedy methods is that they can be modified to work with models other than sparsity; for example, [BCDH10] has modified CoSaMP to enforce tree-based structure. See the discussion in §2.7.2.3. However, new works show that optimization-based methods may be able to solve model-based problems, using variants of reweighting techniques (as suggested in this thesis, and by [DWB08]), and also since the proximity operators may be computable [JMOB10].

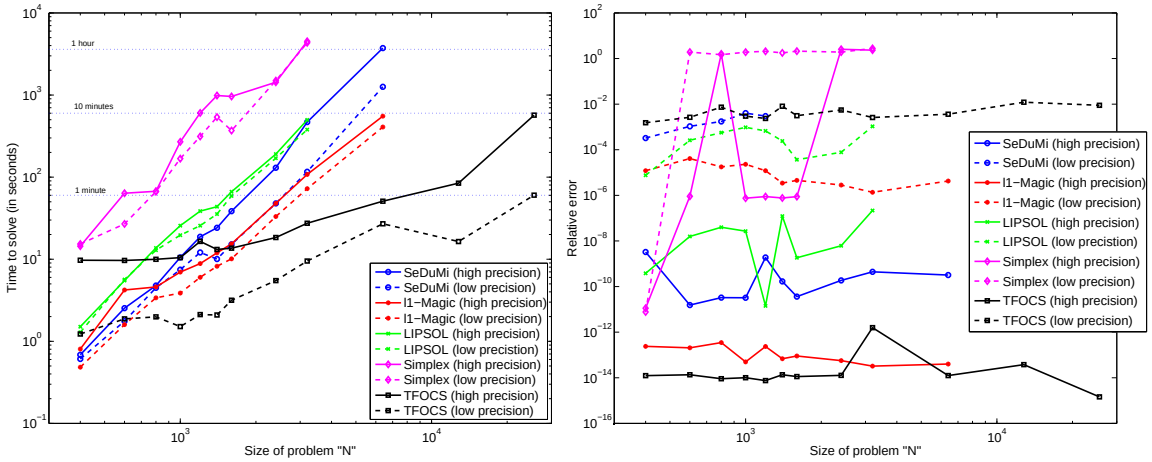


Figure 1.11: Using noiseless basis pursuit (1.1.5) with $m = n/2$ measurements and a sparse solution with $k = m/5$ nonzeros. The TFOCS solver used $\mu = .1$ (see equation (4.1.14)) which was selected on the first try and not "tweaked". In order to be fair to IPM, this example used a dense measurement matrix A. If A is a partial DCT or FFT, then TFOCS can solve problems with 2^{20} variables in about a minute; see Figure 1.12.

The conventional wisdom is that ℓ_1 minimization methods are the best, but at the expense of being slower. Like most conventional wisdom, there is truth behind this yet it is not the whole story. We focus on ℓ_1 minimization methods in this thesis, but remain open to other methods in the future. As research progresses, the disadvantages of each method will be either exposed or fixed, so it is possible that variants of greedy methods will come to be considered as robust as ℓ_1 minimization, and that ℓ_1 minimization will be as fast as greedy methods. In short, the disparate methods may start looking more and more alike.

Further discussion of IPM. The initial approach we took to solve reconstructions for the RMPI was via interior-point methods. A version of l1Magic [CR07b] was modified to allow for analysis (1.1.9); another variant using a null-space formulation was also used, but only covered the noiseless case. The l1Magic software is specially designed to solve the Newton step via iterative methods. In the noiseless synthesis formulation (1.1.8), with $A = \Phi\Psi$, the update step requires solving

$$\begin{pmatrix} D_x & A^t \\ A & 0 \end{pmatrix} \cdot \begin{pmatrix} \Delta x \\ \Delta \nu \end{pmatrix} = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \tag{1.5.1}$$

where D_x is a positive diagonal matrix that changes every iteration. From here, we could solve this "saddle-point" system (e.g., with MINRES, since the system is symmetric and non-singular, but not positive definite unless A is), especially if we have a preconditioner. Or, the standard technique is to reduce further, and get

$$AD_x^{-1}A^t \cdot (\Delta\nu) = \tilde{w} = AD_x^{-1}w_1 - w_2. \tag{1.5.2}$$

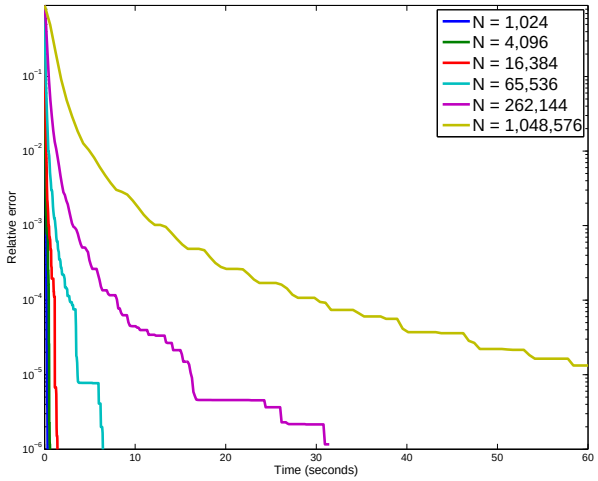


Figure 1.12: Using noiseless basis pursuit (1.1.5) as in Figure 1.11, but now with a DCT measurement matrix. Using $m = .7n$ measurements and a sparse solution with $k = n/16$ nonzeros. Solved on a modern laptop with a quadcore Intel i7 processor.

In this case, since D_x is a positive-definite diagonal matrix, the matrix $AD_x^{-1}A^t$ is positive definite. Recall $A = \Phi\Psi$. The matrix Φ is of reasonable size, but Ψ is an overcomplete Gabor dictionary which is quite large. The key benefit of the Gabor dictionary is that it exploits the FFT to make matrix-vectors multiplies in $\mathcal{O}(n \log n)$ time. Thus l1Magic solves (1.5.2) using the Conjugate Gradient (CG) method; any direct factorization method is impractical. Unfortunately, a well known issue with IPM is that D_x splits into widely different scales near the solution, and hence AD_xA^T has widely separated eigenvalues. This causes CG to take many iterations.

To overcome this, the author and Emmanuel Candès tried preconditioning the saddle-point system. One possible preconditioner for the saddle-point system, motivated by [RG07, GS06b], is

$$P = \begin{pmatrix} D_x^{-1} + \gamma^{-1}A^tA & 0 \\ 0 & \gamma I \end{pmatrix}.$$

The saddle-point system (1.5.1), which is not positive definite, is solved via MINRES, and the preconditioner P is inverted via CG. Hence the overall algorithm requires three levels of iteration: the primal-dual iterations, the MINRES iterations, and CG iterations. This was implemented using both the l1Magic [CR07b] and CVXOPT [DV10] packages. Unfortunately, the three levels of iterations proved to be too slow. After trying other first-order methods, the authors developed the algorithms presented in Chapters 3 and 4. These significantly improve over IPM in terms of running time, and do not suffer in accuracy. See Figure 1.11.

One of the issues with IPM is that the most effective methods are predictor-corrector methods that take two steps every iteration. The equation (1.5.2) is solved via a Cholesky factorization, and this factorization can be re-used for the second step, so the "correction" step is very cheap. When

the Newton step is solved with an iterative method like CG, this is no longer as attractive since the two steps are both full cost. The ill-conditioning issue is another problem. But a basic fundamental problem is that for extremely large numbers of variables, any type of method that solves a linear system of equations at every step is going to be too slow.

There is work on improving the use of warm-starts for IPM [GG03, BS07, YW02], which may lead to better performance. Special versions of conjugate gradient that exploit information from previous solves could also make IPM more attractive. Finally, we mention randomized linear algebra (see [HMT11] and the references therein) which may lead to efficient methods to approximately solve the Newton step.

1.6 Reading guide

Different readers may find certain parts of the thesis more interesting than others. The PDF version of the thesis is equipped with internal links to facilitate quick jumps from section to section.

RMPI design. The reader interested in the RMPI will find Chapter 2 contains the details, and Chapters 3 and 4 may be skipped. The conclusion in Chapter 5 is also relevant to the RMPI.

Signal processing and optimization. Chapters 3 and 4 discuss the NESTA and TFOCS optimization algorithm. From the RMPI chapter, the section on calibration §2.3.4, phase-blind calibration §2.3.5, and recovery §2.7 discuss signal processing techniques. The concluding chapter also contains some discussion.

Novelty. All of Chapter 2 is new and has not appeared in print before. Most of Chapters 3 and 4 are verbatim from published or submitted work, and so may be skipped if the reader is already familiar with the work. However, these chapters contain a few new sections which we mention here. For NESTA, the new section §3.7 discusses an extension to deal with the case when $\Phi\Phi^* \neq I$.

In TFOCS, the new section §4.6 recasts the original derivation of TFOCS in a dual function framework, as opposed to the dual conic framework. A section on convergence §4.6.4 has been added, and the appendix §4.12 on test problems has been rewritten slightly and comments on new literature that appeared in March 2011. The section §4.8 discusses applying TFOCS to conic programs in standard form (e.g., LP, SOCP, SDP) and specialized algorithms for this case; it also discusses some results regarding the special problem of matrix completion and using splitting algorithms such as Douglas-Rachford, as well as a novel automatic restart scheme. Future directions are discussed in the concluding chapter.