TABLE OF CONTENTS

LIST OF TABLES

TABLE **Page**

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Introduction

In order to meet exponentially growing consumer demand for wireless data, radio spectrum regulators considering opportunistic spectrum access policies. Historic spectrum regulation focussed on exclusive frequency assignments (licensing), with spatial and frequency separation to mitigate interference between users. However, this approach leads to considerable underutilisation in both space and time. Thus, faced with a need to provide 1000 times the bandwidth in 10 years, regulators are considering agile access technologies on a licence exempt basis.

Before opportunistic access is a reality, speedy, robust, and accurate estimation of frequency spectra must be made. This is a challenging statistical and engineering problem, limited by characteristics of wireless channels such as multipath fading, and shadowing. The proposed bands have a large bandwidth, containing sub-channels which are not contiguous but statistically correlated, and radio-wave fading environment which can mask high-powered transmissions. Traditional methods are un-viable in these conditions; either requiring expensive hardware to meet the data rate required to perform the sampling, or a large number of RF components to turn a single wideband channel into many narrowband ones.

This project addresses the issue of estimating available frequencies for opportunistic transmission, from a set of underdetermined measurements. The compete set of measurements may be available to a particular sensor. Or, the measurements may be distributed over a network of sensors, improving estimation accuracy, and

The growing number of wireless devices is placing increasing demand on radio spectrum. Consumers are demanding faster speeds and better quality connections in more places. However, there is a limited amount of frequencies to transmit on. Consequently, demand for frequencies which provide sufficient bandwidth, good range and in-building penetration is high.

Not all frequencies are used at all times and in all places. Judicious spectrum management could alleviate the issue of too few frequencies being available to devices by developing approaches to interleaving opportunistic transmissions within established bands.

There are benefits to spectrum sharing beyond simply satisfying consumer demand. Innovative wireless applications such as wireless rural broadband, remote monitoring, and machine-to-

machine applications will be made viable without the need to purchase exclusive access to a specific frequency.

The historic allocation process has placed unused spectrum between adjacent radio channels, in an attempt to avoid interference between consecutive users. However, technical changes in transmission standards are making some bands available as modern coding and modulation techniques are more spectrally efficient. In particular, in the UK, the switchover from analogue to digital terrestrial broadcast has freed many previously used bands.

Recent regulatory focus has been on frequencies in the TV broadcast bands 470-790MHz. In the UK, TV channels are broadcast using up to six multiplexes, each requiring an 8MHz channel. A total of 32 of these channels are allocated for TV broadcast, whilst only 6 of these channels are required to receive the 6 multiplexes at any given location. This is because TV broadcast is high powered and needs spatial separation between coverage areas to avoid interference. As such, the majority of TV frequencies reserved for TV broadcast are unused in any given place. These white spaces are spectrum which has been left over to prevent interference between primary users (such as TV broadcast). These frequencies can be used on an opportunistic basis by relatively low-powered devices. Whilst current focus is on the TVWS bands, the work presented in this thesis can be used in other, as yet unrealised frequencies.

Currently access to spectrum is managed in two ways: licensed and license exempt access. Licensing authorises a particular user/users to exclusive use of a specific frequency band. License exempt access allows any user to access a band, provided they meet technical requirements intended to limit interference in on other users. This is a particularly pressing issue for the co-existence of licensed users and licence exempt users in the same band.

Devices seeking to access white spaces need a robust mechanism for learning which frequencies can be used at a particular time and location. The approach currently being taken by OFCOM and the FCC is to maintain a set of databases, which map the location of white spaces based on knowledge of existing spectrum users. An alternative approach is for devices to monitor the use of spectrum individually.

One approach to white-space access is to maintain a database of currently available frequencies. The database would contain up to date information about incumbents, including television transmitters and wireless microphones. It would also need to maintain information about currently operative secondary users. Devices would register with the database, based on their geo-location and the service would determine availability for the device, either based on in situ measurements, or from a propagation model.

Location based measurements are a costly approach, as they need to be redone every time primary user transmission characteristics change, and they cannot be done economically in difficult to reach places. Propagation models can achieve good accuracy, but at the cost of being complex and computationally intensive. Simplistic models do not agree with ground truth measurements.

Devices must also supply geographic information to a database service. How this is done has yet to be codified, but current methodologies all have drawbacks. GPS positioning works well outdoors, but is inaccurate indoors. Cellular location can yield errors of up to a mile. Using the base-station location as a proxy for the device location results in an unacceptably high loss of whitespaces, as decisions would have to be needlessly conservative.

An alternative to maintaining a database of available frequencies, is for secondary users to independently sense spectrum. This method is subject to a number of technical limitations. The proposed bands have an ultra-wide bandwidth. Traditional sensing approaches to this have been either to divide the band into a number of contiguous narrow bands, or to simply use a high rate processing device to capture the necessary samples. Both methods are prohibitively expensive, and

require dedicated and energy hungry hardware. Further, traditional statistical spectral detection and estimation techniques make assumptions about the primary users signal. Such assumptions include cyclo-stationarity, the presence of specific waveform patters, the use of known pilot sequences, and sufficiently high powered transmissions that energy detection is viable. In practice, some (or all) of these assumptions are violated; either because assumed patterns are not present in the transmission, or because the radio environment has a deep fade between the primary and secondary user masking high powered transmissions.

The biggest issues for single node sensing are multipath fading, shadowing, and PU receiver uncertainty. Cooperation between a network of nodes can improve sensing performance through spatial diversity. In cooperative sensing information from geographically diverse nodes is aggregated in the decision making process. Combining observations can overcome the deficiencies of observations unique to individual cognitive radios. Spatial diversity makes it unlikely that all radios will experience the same fading and receiver uncertainty. Approaches to cooperative sensing based on Nyquist sensing typically involve energy detection at each node, along with a gain combining procedure (which is either performed at a fusion node, or via a decentralised algorithm). Cooperative sensing can use such simple methods at each node as receiver sensitivity can be mitigated by using multiple statistically independent observations. This reduces the complexity and cost of the individual cognitive radios.

Cooperative sensing has drawbacks however. Spatial diversity does not necessarily mean statistically independent observations. A subset of nodes all blocked by the same object will make measurements which are correlated and corrupted by the same shadowing for example. Cooperative sensing also involves an overhead with the being part of a network. This is any extra time, energy and processing when compared to sensing individually.

An alternative sensing strategy is compressive sampling (CS). This is a new paradigm in signal processing which has emerged over the last decade, and has had significant success in imaging problems. In particular compressive sampling strategies have reduced the time needed for an MRI scan from 2 minutes to 30 seconds. Patients previously had to hold their breath for the entire duration. The central innovation in compressive sensing is that randomness is an effective strategy for sensing sparse signals. In practice much of the TVWS spectrum is unoccupied, and CS can leverage this sparsity to reduced the sampling rate of the CRS.

Group Testing is a model sparse inference problem, first introduced by Dorfmann in the context of testing for rare disease. Given a population of items, some small fraction of which has an interesting property (labelled defective), Group Testing proposes algorithms to find those items efficiently. What allows these items to be discovered, in fewer tests than the total number of items, is testing pools of items. That is, items aren't tested individually, but several together. In testing for rare diseases the blood samples are mixed and the mixture is tested. This allows fast elimination of non-defective items. Popular pooling designs are randomised - items are included in a test with independent probabilities. We consider probabilistic algorithms with non uniform probabilities of each item's defectivity. These non-uniform priors could come from a TVWS database of possibly occupied spectrum, or from summarising risk and family history in disease testing.

The structure of this thesis is as follows. Chapter 2 covers the relevant material on Nyquist sensing theory and Nyquist approaches to the spectrum sensing problem. It covers energy detection, feature detection, matched filtering, and cooperative approaches to the problem.

Chapter 3 covers the theory of compressive sensing and

Chapter 4 covers the theory of ADMM and optimisation on graphs.

Chapter 5 covers our approach to solving the optimisation problem.

Chapter 6 Covers the model we used.

Chapter 7 covers
Chapter 8 covers group testing.

CHAPTER 2

CLASSICAL SENSING

2.1 Introduction

Despite the ubiquity, capacity and apparent efficacy, modern communication systems are wasteful, inefficient and in need of reform. Most of the bits of data collected by our sensing systems are unessential, and only serve to necessitate data compression wasting computation time before transmission. For example, people regularly use a camera with a resolution of several megapixels only to upload a file of a few kilobytes to Facebook. Devices are unable to make dynamic decisions about how to transmit this data, leading to both spectral exhaustion on some frequencies whilst much of the available radio spectrum lies fallow.

This project addresses these issues, by reviewing a novel acquisition and decompression framework for data: a way in which we need only sense the most informative bits of data. This framework is then applied to the problem of sensing spectral opportunities dynamically, to make better use of available spectrum.

The key uniting both these applications is that data and spectra are *sparse*: that is they have a representations which are 'smaller' than their respective dimension. For example, images and audio can be compressed into file formats much smaller than when initially recorded (compare the relative sizes of bitmap and JPEG images).

The sole focus of this research is to use the sparsity of the spectrum to uncover transmission opportunities, allowing better use of spectrum more generally.

We are motivated by the need to send more data over wireless networks, whilst at the same time having a constrained frequency set over which to transmit this information. This issue could be alleviated by users dynamically allocating spectrum on a per-transmission basis: without the ability to gain knowledge of spectral conditions this can never become a reality however.

The requirement for increasing bandwidth isn't just a pressing issue for today: in the next decade it is forecast that network operators will need to provide for three-orders of magnitude (1000 times) more capacity. Demand is continually outstripping supply - motivated by the ubiquity of smart-phones, and the consumers appetites for media.

At the same time as this demand for ever more data, there is an increasing scarcity of radio spectrum over which to transmit. New frequencies are rarely cleared for commercial purposes,

and when they are they go for high prices. A decade ago the UK auction for 3G spectrum licenses raised an estimated 22.4 billion pounds for the UK treasury, indicating the seriousness of the market players requirements for new spectrum. The recent 4G spectrum auction raised 2.3 billion pounds with initial networks being rolled out by the end of 2013.

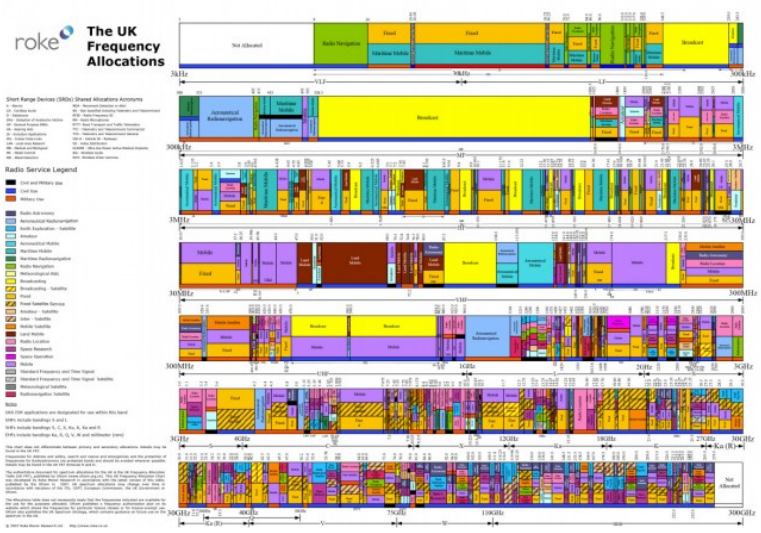

Figure 2.1: A digram of current Spectral allocation [**Strategy2013**]

However, a closer inspection of the frequency allocation suggests this scarcity is artificial, it's more a product of regulatory oversight over time. As the constraints on spectrum requirement became more complex, so did the solutions to that problem - at the cost of leaving much of the spectrum idle for most of the time.

For example: much of the spectrum is allocated to TV broadcast, radio broadcast and mobile. However, if we look closer, the allocations aren't even for specific companies - they're simply categories. Within these, OFCOM may have many licensees within each category.

Also interesting to note is how much frequency the Government allocates to itself (the red bar underneath the blocks indicates Government use). Compare this to the actual utilisation of spectrum: much of it is not used at all. Figure 2.2 shows a snapshot of frequency utilisation in three diverse locations in the UK over te radio specturm, note that many frequencies are not utilised (coloured blue) whilst others have significant activity (coloured yellow). Note that the plot for Southwark (central London) is barely different from Braddock - a rural area.

How do we then go about solving this issue - how can we obtain the most significant bits of information from our sensing mechanism, whilst obviating the need to compress the data once we are done? How do we dynamically assign spectrum? The work of Candes, Tao [**Candes2006**] and Donoho [**donoho2**], has shown that instead of measuring the information we require directly (and then compressing it), we can measure 'holographic' and non-linear random projections between our measurement space and the space where our data is sparse. This requires only the knowledge that the signal is compressible via some transform - both the acquisition protocol and the reconstruction algorithm are agnostic to the type of signal. What is surprising is that the sampling kernels are fixed independently of the signal, are non-adaptive and these projections are sufficient to reconstruct the signal - as if we had an Oracle to tell us where the non-zero components of our signal are.

This work has had a large impact in medical imaging since it's inception: for example, it's now

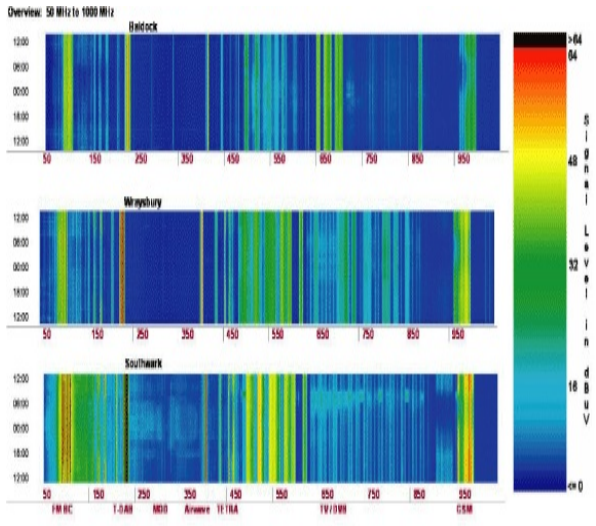

Figure 2.2: A snapshot of frequency utilisation in various areas: many frequencies are not used at all, whilst there is significant activity on others [**Burbidge2007**]

possible to take an image of a patient's heart within a single breath, as well as dynamic imaging of the heart ([**Donoho**] figures 7 and 9).

Modern digital signal processing techniques (such as modulation techniques) are far more spectrally efficient than their historic analogue counterparts, which has in part contributed to the spectrum crisis. All this is changing though: from the beginning of 2013 all TV in the UK will transmitted digitally. Historically, television in the UK was broadcast using analogue signals requiring 32 multiplexes. Digital TV requires 6 multiplexes, on the other hand.

This freeing up of TV frequencies represents an opportunity: these frequencies have good propagation characteristics (they suffer less with free space path loss relative to higher frequencies), whilst sill providing good bandwidth for data transmission. These TV frequencies are being opened up to civilian and commercial users: spectral holes will be able to be exploited opportunistically by devices, so long as they don't interfere with the reception of TV. Historically, this is the single largest gift of new spectrum, and because there is no requirement for licensing this spectrum is free.

As with all technological innovations, this will not only improve existing infrastructure but also new classes of devices to transmit, for instance; applications such as passive sensor networks) which only need spectrum intermittently to transmit monitoring results), inter-vehicle communication for real time traffic monitoring and wireless internet at broadband data rates have all been proposed.

Despite all of this hype, dynamic spectrum access won't become a reality unless spectral holes can be robustly detected. The requirement that secondary users exploit the new spectrum politely, without interference to primary user makes spectrum sensing essential to TV white-space (TVWS) technologies. The realisation of any Cognitive Radio standard (such as IEEE 802.22), requires the co-existence of primary (TV users) and secondary (everybody else who wants to use TVWS spectrum) users of the frequency spectrum to ensure proper interference mitigation and appropriate network behaviour.

Users of TVWS (Cognitive Radios) must sense whether spectrum is available, and must be able to detect very weak primary user signals. Furthermore they must sense over a wide bandwidth (due to the amount of TVWS spectrum proposed), which challenges traditional Nyquist sampling

techniques, because the sampling rates required are not technically feasible with current RF or Analogue-to-Digital conversion technology.

Sensing should enable devices to detect the presence of TV signals in a band and provide smart and adaptive (and possibly distributed) solutions to band identification.

Spectrum sensing should involve:

1. Sensing to detect white spaces.
2. Co-existence with similar devices.
3. Frequency monitoring of other devices.
4. Interference management.
5. Spectrum mobility and transmission power control when needed.

As described earlier, the available spectrum is highly underutilised, and can be thought of as a collection of narrowband transmissions over a wideband channel. As such, the spectrum we're sensing is sparse. This makes it an ideal candidate for sparse recovery techniques such as Compressive Sensing.

The report is divided into three chapters: the remainder of this chapter describes methods for sensing narrowband signals (i.e. channels where the frequency response is approximately flat, and where the bandwidth is smaller than the coherence bandwidth of the channel), and the limitations of these are highlighted for the problem of sensing spectrum for Cognitive Radios.

Classical and Compressive Sensing are then contrasted, including the main ideas such as incoherence and the Restricted Isometry Property, and illustrating the number of samples required for full reconstruction. Somme approaches to solving the optimisation problems posed by the new framework are also discussed.

Chapter 2 introduces Group Testing, and covers new work which has been accepted for publication at Allerton 2014. After some preliminary remarks the Capacity of a Group Testing problem is defined. Then, previous work on variations of an algorithm by Hwang are discussed, including upper and lower bounds on Capacity. Finally, a new algorithm is presented along with an analysis of the average number of tests the algorithm will execute.

The focus of Chapter 3 is again compressive sensing, this time in a distributed setting - given a connected network of nodes, how can we organise sensing to reconstruct a wideband signal? The chapter begins by discussing various signal models in the literature, and justifies the use of a single model. Then the sensing model is presented as a multinode extension of the Modulated Wideband Converter ([**mishali2010theory**]). There follows an extended discussion of constrained convex optimisation, and an introduction to the Alternating Direction Method of Multipliers. Finally, how to sense the requisite signals and use a distributed setup to solve the system is explained. Some tentative results of simulations are also presented.

2.2 Classical Sensing

Classically, for perfect signal reconstruction, we must sample a signal such that the sampling rate must be at least twice the maximum frequency in the bandlimited signal. The continuous time signal can then be recovered using an appropriate reconstruction filter (e.g. a sinc filter). For example, we

can represent a sampled continuous signal as a multiplication of the signal with a train of Dirac delta functions at multiples of the sampling period T.

where

Working the frequency domain, this multiplication becomes convolution (which is equivalent to shifting):

$$\hat{X}_s(f) = \sum_{k=-\infty}^{\infty} x(t - kT) \tag{2.2.0.1}$$

Thus if the spectrum of the frequency is supported on the interval $(-B, B)$ then sampling at intervals $\frac{1}{2B}$ will contain enough information to reconstruct the signal $x((t)$. Multiplying the spectrum by a rectangle function (low-pass filtering), to remove any images caused by the periodicity of the function, and the signal $x((t)$ can be reconstructed from its samples:

$$x(t) = \sum_{n=-\infty}^{\infty} x(nT)\, sinc\left(\frac{t_n T}{T}\right) \tag{2.2.0.2}$$

2.3 Narrowband Spectrum Sensing

The problem of spectrum sensing [**yucek2009survey**] is to decide whether a particular band is available, or not. That is, we wish to discriminate between the following two hypotheses:

$$H_0 : y[n] = w[n], \text{n} = 1 \ldots N \tag{2.3.0.3}$$

$$H_1 : y[n] = x[n] + w[n], \text{n} = 1 \ldots N \tag{2.3.0.4}$$

Where x is the (deterministic) primary users signal, having a specific structure which stems from modern coding and modulation techniques, w is additive white Gaussian noise and y is the received signal.

Any detection strategy is a function, $f : \mathbb{R}^n \rightarrow \{0, 1\}$, mapping the output of sensing to $\{0, 1\}$.) means the received signal is noise, whilst 1 means that a Primary User signal is present.

To decide whether the observations $\mathbf{y}$ were generated under H_0 or H_1 is accomplished by forming a test statistic $\Gamma(y)$ and then comparing this statistic with a predefined threshold λ. Both classical methods, where the hypotheses are assumed to be deterministically true and the goal is to minimise the false detection probability, and Bayesian methods, where it is assumed that the source selects the true hypothesis at random according to some prior probabilities, agree that the test statistic should be likelihood ratio:

$$\Gamma(\mathbf{y}) = \frac{p(\mathbf{y} \mid H_0)}{p(\mathbf{y} \mid H_1)} \tag{2.3.0.5}$$

The performance of a detector is quantified in terms of the probability of detection

(2.3.0.6) $$P_D = Pr\left(\Gamma\left(\mathbf{y}\right) > \lambda \mid H_1\right)$$

and the probability of false alarm

(2.3.0.7) $$P_{FA} = Pr\left(\Gamma\left(\mathbf{y}\right) > \lambda \mid H_0\right)$$

By varying λ the operating point of a detector can be chosen anywhere along its receiver operating characteristics curve.

There are several proposed spectrum sensing methods that enable cognitive radios identify bands and perform dynamic frequency selection. Some of the common (narrowband) spectrum sensing techniques are described below.

Energy Detection

This is a common method for the detection of unknown signals in noise, due to low computational and implementation complexity. This method is quite generic as receivers need no knowledge of the primary users signal.

A typical method would be a bandpass filter with a centre frequency f_s and a bandwidth W. This is followed by a squaring device to measure the received energy and an integrator to determine the observation interval. Finally the output of the integrator is compared with a threshold to determine the presence of a signal. This threshold is determined based upon the noise variance of the channel. I.e. we have a decision metric of the following form:

(2.3.0.8) $$M = \sum_{n=0}^{N} |y[n]|^2$$

Assuming that the signal is a zero mean AWGN variable as well, we can derive expressions for the metric, the detection probability and the false alarm probability:

(2.3.0.9) $$M = \begin{cases} \frac{\sigma_w^2}{2} & H_0 \\ \frac{\sigma_w^2+\sigma_s^2}{2} & H_1 \end{cases}$$

(2.3.0.10) $$P_D = 1 - \Gamma\left(1, \frac{\lambda}{1+\frac{\sigma_2^2}{\sigma_w^2}}\right)$$

(2.3.0.11) $$P_F A = 1 - \Gamma\left(1, \frac{\lambda}{\sigma_w^2}\right)$$

Where $\Gamma(1,x)$ is the incomplete gamma function. From these equations it's clear to see that the performance of energy detection based sensing faces challenges at low SNR values. See (REF) figure 3 for curves quantifying the performance. Also energy detectors perform poorly under extreme

fading conditions as they are unable to distinguish primary users and noise. Further this type of detector is not efficient at detecting spread spectrum signals.

For energy detection we wish to maximise P_D subject to a constraint on P_{FA}. This is done via a threshold λ, which trades off these two probabilities.

Choosing λ requires knowledge of the Primary User transmissions, as well as estimates of the noise power. Given these, calculating the optimal λ is straightforward [**xie2009optimal**]. Estimating the PU power is dependent on the radio environment between the PU transmitter and the CR. Noise power estimation isn't flawless and a small noise power estimation error can cause significant performance loss [**hamdi2010impact**], [**sahai2004some**].

Noise power uncertainty can be mitigated by using an adaptive algorithm [**zhang2011adaptive**], or an a variant of the MUSIC algorithm which separates signal and noise subspaces [**olivieri2005scalable**]. These significantly increase the complexity of the energy detector, making it less attractive relative to other methods.

A more serious concern for energy detection is the SNR wall [**tandra2008snr**]: an SNR below which an energy detector will fail to detect the presence of a PU signal no matter how long the detector observes the channel. This is because at low SNRs the PU signal is no longer well separated from the noise. There has been some work in overcoming this wall using cross correlation between multiple antennas [**oude2011lowering**].

Cyclostationary Feature Detection

Because the signals used in practical communication systems contain distinctive features that can be exploited for detection, it is possible to achieve a detection performance which substantially surpasses the energy detector [**ye2007spectrum**], [**kim2007cyclostationary**]. This is in contrast to the predictions of information theory where maximum entropy signals will be statistically white and Gaussian (if this were the case, then we could do no better than the energy detector). More importantly, known signal features can be exploited to estimate unknown parameters such as noise power.

Examples of well known patterns include pilot signals and spreading sequences. Other examples include preambles and midambles: known sequences transmitted before and in the middle of each slot, respectively. Others include redundancy added by coding, modulation and burst formatting used by the transmitter.

This method exploits cyclostationary features of received signals: man made periodicity in the signal (for example symbol rate, chip rate, cyclic prefix etc) or its statistics - mean, autocorrelation. A cyclic correlation function is used instead of PSD (or autocorrelation sequence) for detecting signals present in a given spectrum. This is able to differentiate noise from primary users signals since noise is wide-sense stationary with no correlation but modulated signals are cyclostationary due to the redundancy of signal correlations.

For clarity, the random processes encountered by a cognitive radio will have a period in both expectation and autocorrelation:

$$\mathbb{E}(t) = \mathbb{E}(t + mT) = \mathbb{E}[x(t)] \tag{2.3.0.12}$$

$$\mathbb{R}(t,\tau) = \mathbb{R}(t + mT,\tau) = \mathbb{E}\left[x(t)\, x(\bar{t_+}\tau)\right] \tag{2.3.0.13}$$

where t is time, τ is the autocorrelation lag, $x(t)$ is the random process we are considering and m is an integer.

Due to the periodicity of the autocorrelation, it can be expressed as a Fourier series over integer multiples of the fundamental frequency in the signal as well as integer multiples of sums and differences of this frequency:

$$\mathbb{R}(t,\tau) = \sum_{\alpha} r(\alpha,\tau)\, e^{2\pi j\alpha t} \tag{2.3.0.14}$$

with Fourier coefficients:

$$r(\alpha,\tau) = \frac{1}{T}\int_{T} x\left(t+\frac{\tau}{2}\right) x\left(\bar{t+\frac{\tau}{2}}\right) e^{-2\pi j\alpha t}\, dt \tag{2.3.0.15}$$

where α is the cyclic frequency

From this we can define the Cyclic Power Spectrum of the signal:

$$S(f) = \int_{-\infty}^{\infty} r(\alpha,\tau)\, e^{-2\pi jf\tau}\, d\tau \tag{2.3.0.16}$$

For a fixed lag τ, 2.3 can be re-written as:

$$R_{xx}(t,\tau) = R_{xx}(\tau) + \sum_{\alpha} r(\alpha,\tau)\, e^{2\pi j\alpha t} \tag{2.3.0.17}$$

i.e. a part dependent on the lag only (the cyclic frequency is zero), and a part which is a periodic function of time.

Under both hypotheses, (2.3, 2.3), the continuous portion of the signal exists, but the cyclo-stationary portion only exists under 2.3 when $\alpha \neq 0$. Thus we only need to test for the presence of a cyclo-statrionary component.

To this end re-write the hypotheses as:

$$H_0 : y[n] = S_w^{\alpha}[n],\ \mathrm{n} = 1 \ldots N \tag{2.3.0.18}$$

$$H_1 : y[n] = S_x^{\alpha}[n] + S_w^{\alpha}[n],\ \mathrm{n} = 1 \ldots N \tag{2.3.0.19}$$

where S_x^{α} is the CPS of white noise which is zero for $\alpha \neq 0$. Using the test statistic:

$$\chi = \sum_{\alpha\neq 0}\sum_{n} S_x^{\alpha}\bar{S_x^{\alpha}} \tag{2.3.0.20}$$

we can formulate the cyclo-stationary detector as:

$$d = \begin{cases} 0 & \chi < \lambda \\ 1 & \chi \geq \lambda \end{cases} \tag{2.3.0.21}$$

where λ is some pre-determined threshold [**Ghozzi2006**].

The advantages of this type of sensing over energy detection, are that its possible to distinguish primary user transmissions (as well as distinguish between different PU signals) [**lunden2007spectrum**]. It is also possible to distinguish noise from PU signals as the noise spectrum has no cyclic correlation [**cabric2004implementation**], [**vcabric2005physical**]. However, cyclic frequencies have to be assumed to be known [**Ghozzi2006**].

Matched Filtering

If all the probability distributions and parameters - noise variance, signal variance, channel coefficients etc - are known under both hypotheses, and the signal to be detected is perfectly known then the optimal test statistic is a matched filter [**cabric2004implementation**], [**yucek2009survey**].

A matched filter is the convolution of a test signal with a template signal (or window) and detects the presence of the template in the unknown signal (as the convolution measures the overlap of two signals).

For example: for a given TV signal, $r(t)$ defined over $0 \leq t \leq T$ the corresponding matched filter is $h(t) = r(T-t)$.

A test statistic can be formed by sampling the output of the filter every nT seconds and choosing 2.3 if the statistic is below some threshold and 2.3 otherwise.

When compared to other methods, matched filtering takes a shorter time to achieve a threshold probability of false alarm. However, matched filtering requires that radios demodulate received signals, and so requires perfect knowledge of primary users signalling features. Matched filtering also requires a prohibitively large power consumption, as various algorithms need to be executed for detection.

The paper [**bhargavi2010performance**], compares the performance of Energy Detection, Matched Filtering and Cyclostationary detection. It concludes that cyclostationarity based detection has the best performance (based on a lower P_{FA} for a given P_D), as this for of detection is naturally insensitive to noise uncertainty as the test statistic for cyclic detection doesn't require knowledge of the noise variance.

Limitations

The methods described above, are appropriate for sensing whether a single channel is available for transmission, based upon the result of measurements of that channel. However, Cognitive Radios aim to exploit spectral holes in a wide band spectrum (i.e. a channel whose frequency response is not flat over the bandwidth) and will usually have to make a decision regarding transmission from measurements from this type of channel.

There are two proposed approaches to this: Multiband sensing and Compressive Sensing. Multiband sensing splits the wideband spectrum into a number of independent (not necessarily contiguous) sub-channels (whose frequency response is flat), and performs the hypothesis test for each sub-channel. However, in practice, there are correlations across sub-channels that this method fails to address. For example, digital TV signals are transmitted as spread spectrum signals so that primary user occupancy is correlated across channels. A related issue is that noise variance could be unknown but correlated across bands. Binary hypothesis testing then fails in this case, needing to be replaced by composite hypothesis tests which grow exponentially with the number of sub-channels. Such problems are typically non-convex and require prohibitively complex detectors.

2.4 Wideband Spectrum Sensing

This section presents a new method of sensing sparse signals, and its application to the problem of sensing over wideband spectra in Cognitive Radios. Initially we introduce Classical Sensing and then give an overview of both Compressive Sensing and Group Testing. Finally, we discuss some sub-Nyquist sampling techniques.

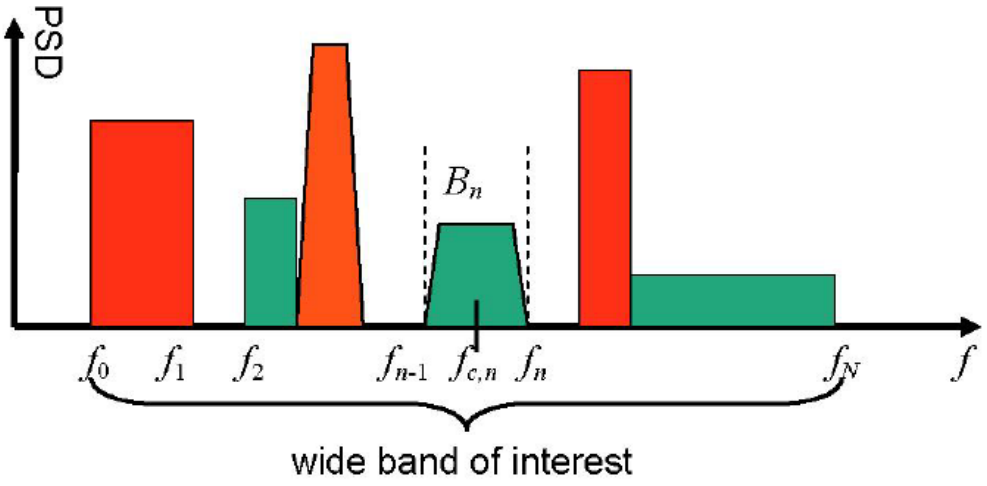

Figure 2.3: A digram of the Spectrum Sensing model [**Tian**]

2.4.1 Compressed Sensing

However, in practice many signals encountered 'in the wild' can be fully specified by much fewer bits than required by the sampling theorem above. For example, image compression algorithms can reduce the size of a stored image to about 1% of the size required by Nyquist sampling. If the reconstruction algorithm is able to reconstruct the image from this small amount of data, this raises the question: why collect all the data in the first place, when most of the information can be thrown away? Why not directly measure the part that will not end up being thrown away?

Compressed Sensing considers situations where the signal is *undersampled* i.e. situations in which the number of samples is much smaller than the dimension of the signal (or the number of samples required by classical sampling theory). This is equivalent to a system of linear equations which is under-determined. That is, this is a method of measuring the informative parts of a signal directly without acquiring unessential information at the same time (i.e. the parts of the signal that would be discarded in traditional compression applications). The questions then are how can we acquire these measurements in the first place, and how to 'decompress' them once they are obtained [**Donoho2006**].

To answer the first, note that signals have representations in which they are sparse (i.e. the most of the co-efficients in that representation are zero, or close to zero). For example,

1. A sine wave at frequency ω is defined as a single spike in the frequency domain yet has an infinite support in the time domain

2. An image will have values for every pixel, yet the wavelet decomposition of the image will typically only have a few non-zero coefficients

However, we may not be able to directly obtain those coefficients, as we may not posses an appropriate measuring device (or one may not exist). Yet we are able to measure correlations between the signal and the basis waveforms of the domain where the signal is sparse ϕ_k i.e.

$$y_k = \langle f, \phi_k \rangle \; k = 1 \dots m \tag{2.4.1.22}$$

for $f \in \mathbb{R}^n$ expanded in an orthonormal basis ψ s.t.

$$f(t) = \sum_{i=1}^{n} x_i \psi_i(t) \tag{2.4.1.23}$$

where the x_i are the coefficient sequence of f.

An example of a practical Compressive Sensing system is the single-pixel camera at Rice University [**Duarte2008**], [**wakin2006architecture**]. Typical camera devices obtain pixel samples by exposing a bank of photon detectors (one for each pixel) to the incident light field. This data is the processed into an image.

The single pixel camera takes pictures by first directing the incoming light field onto an array of tiny mirrors (one for each pixel). Each mirror can be either be oriented towards a single photon detector, or oriented away from the detector. In this setup, a measurement is taken as the sum of all the incident light beams. Afterwards, the mirrors are flipped to a new random configuration, and another measurement is taken. This process is repeated, until enough information has been collected to reconstruct the image. Figure 2.4 shows the operation of the single pixel camera.

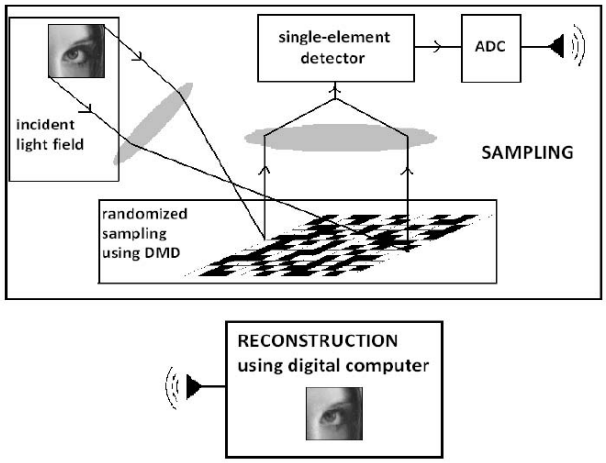

Figure 2.4: The operation of the single pixel camera at Rice University [**Thompson2011**], [**DavenportSinglePixel**]

The question all this raises is where do we do our sensing? In other words, given that we know a basis in which our signal is sparse, ϕ, how do we choose ψ? It's best to choose ψ so that the signal is 'spread out' relative to the signal's expansion in ϕ. Such pairs are said to be incoherent.

Definition 1. *A pair of bases is said to be incoherent if the largest projection of two elements between the sensing (ψ) and representation (ϕ) basis is in the set* $[1, \sqrt{n}]$*, where n is the dimension of the signal. The coherence of a pair of bases is denoted by* μ*.*

This implies that sensing with incoherent systems is good (in the sine wave example above it would be better to sample randomly in the time domain as opposed to the frequency domain), and efficient mechanisms ought to acquire correlations with random waveforms (e.g. white noise).

Theorem [**Candes2006**] Fix f $\in \mathbb{R}^n$ with a sparse coefficient basis, x_i in ψ. Then a reconstruction from m random measurements in ϕ is possible with probability $1-\delta$ if:

$$m \geq C\mu^2(\phi,\psi)S\log\left(\frac{n}{\delta}\right) \tag{2.4.1.24}$$

where $\mu(\phi,\psi)$ is the coherence of the two bases, and S is the number of non-zero entries on the support of the signal.

Then $f*$ (the proposed reconstruction) is given by $f^* = \psi x^*$ where x^* is the solutionn to the convex optimisation program (n.b. $\|x\|_{l_1} := \sum_i |x_i|$):

$$min\|\tilde{x}\|_{l_1} \text{ subject to } y_k = \left\langle \phi_k, \psi x^* \right\rangle \ \forall k \in M \subset [1 \dots n] \tag{2.4.1.25}$$

i.e. **CS: Sample non-adaptively in an incoherent domain and invoke linear programming after the acquisition step**

Thus the role of l_1 minimisation is to decompress the data. There are many ways to perform this operation: some popular methods are basis pursuit [**Chen1998**] and Greedy approaches such as Orthogonal Matching Pursuit [**Tropp2007**].

It may be remarked that the l_0 norm:

$$||l_0|| = \{\#i, x_i \neq 0\} \tag{2.4.1.26}$$

is a more appropriate functional to minimise. In fact under this norm, $m \geq 2k$ measurements will suffice to uniquely determine all k-sparse signals. However, this norm is not convex and so minimising it is an NP-hard problem. As we are seeking sparse solutions the l_1-norm will suffice [**Donoho2006a**]. This is because all vectors in a random k-dimensional subspace of an n-dimensional space are approximately Gaussian (in the sense that the components are distributed according to an approximate normal distribution). Such vectors have roughly equivalent norms, and so any solution to the l_1 minimisation problem will be the same solution to the l_1 minimisation problem for sufficiently sparse signals.

Incoherence, isometries and all that

To recover a sparse vector, we must make sure that the vectors are not in the null space of the sensing matrix (otherwise there would be no hope of recovery). We also require that any subset of S columns taken from the measurement matrix be nearly orthogonal w.r.t sparse vectors: i.e. all pairwise distances between S-sparse vecotrs be well preserved in the measurement space.

This can be summed up in the following inequality (Restricted Isometry Property) [**Emma**]:

Definition 2.4.1 (Restricted Isometry Property)**.**

$$(1-\delta)\,||x||_{l_2}^2 \leq ||Ax||_{l_2}^2 \leq (1+\delta)\,||x||_{l_2}^2 \tag{2.4.1.27}$$

Note, that the RIP isn't necessary for a complete theory of Compressive Sensing, [**candes2011probabilistic**], it's however a handy tool for analysing the performance of CS in a variety of situations. This is not the only fomrulation of the RIP possible, for a more comprehensive overview see [**foucart2013mathematical**].

We are also in a position to evaluate the meaning of the constant μ in 7.1. We are considering sampling within orthonormal systems (for example, Time and Frequency):

$$A * A = nI \tag{2.4.1.28}$$

so that each row or column has l_2 norm equal to $sqrtn$. A is any matrix satisfying this property (examples include the Fourier matrix and the Dirac matrix). Thus μ must be in the set $[1, \sqrt{n}]$. μ then, is a measure of how concentrated the rows of our measurement matrix is - i.e. how much information is spread across each vector. If $\mu = 1$ then the rows are 'flat' - and we need relatively fewer samples to reconstruct an S-sparse signal (i.e. each sample provides the same amount of information). However, if the rows contain all non-zero entries except for a single component, then $\mu^2 = n$ and we will need to observe all components to determine the non-zero one (i.e. we have no guarantees of recovery from limited samples) [**Candes2007**].

Noting that the measurements we take are projections from our orthonormal system (from example time) onto a sparsifying basis (i.e. frequency) we can see that:

$$\mu = max_{k,j}|\langle \phi_k, \psi_j \rangle| \tag{2.4.1.29}$$

So we need to choose a sensing basis, where the vectors will be 'spread out', and the degree of spreading is characterised by μ.

Short, Fat matrices

As remarked upon earlier: Compressive Sensing is equivalent to solving an under-determined linear system, with the constraint that we seek the sparsest solution. The content of the previous sections amounts to constraints on the number of rows of matrix of this linear system.

If we had an Oracle which could tell us where the non-zero components of our solution were, then we would need only as many rows of the matrix as there were non-zero components in the signal to fully specify the problem.

However, such and Oracle does not exist, and so we're left with the task of constructing a matrix to recover those components. Knowing that we're looking for k-sparse solutions, we need a matrix with at least 2k columns which are linearly independent. Equivalently, all images of k-sparse vectors under the operation of the sensing matrix Φ must be distinct. From this, any k-sparse signal can be reconstructed from Ax.

To prove this assume the opposite - then there are two vectors $x, x' \in \mathbb{R}^n$ such that $Ax = Ax'$. I.e. $A(x - x') = 0$. However, $(x - x')$ is 2k-sparse and so there is a linear dependence between 2k columns

of the sensing matrix A. We have a contradiction, and so 2k columns will suffice to reconstruct a k-sparse signal.

The problem with this is that we are trying to find the support of a k-sparse signal over a vector of length N, and so we would need to check all $\binom{N}{k}$ combinations of k-sparse signals which is prohibitively computationally expensive. Is there some way to gain the advantages of sparsity, without having to minimise a non-convex functional?

As it turns out, the answer is yes. If we take $m \geq C\mu^2(\phi,\psi)S\log(n)$ rows minimising the l_1 norm will find the sparsest solution. This is because the l_1 norm is an octahedron (in 3-dimensions, in higher dimensions it has an analogous spiky geometry), and solutions are more likely to intersect the norm at the points. Figure 4.3 shows this.

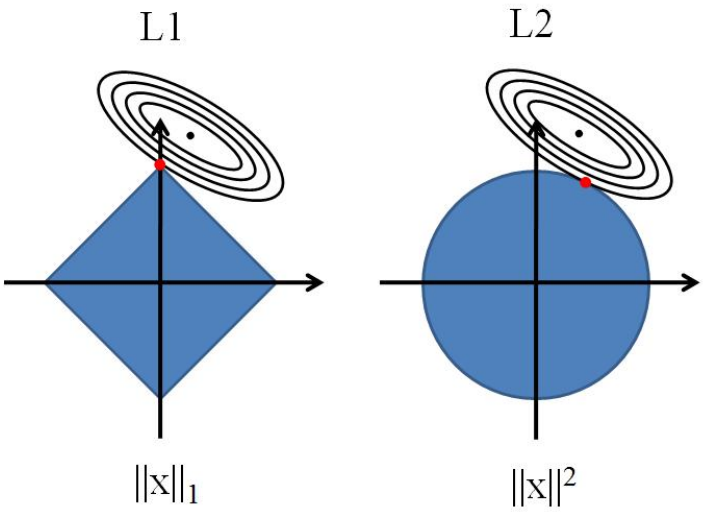

Figure 2.5: Solutions to the Compressive Sensing optimisation problem intersect the l_1 norm the points where all components (but one) of the vector are zero (i.e. it is sparsity promoting) [**Tibshirani1996**]

2.4.2 Reconstruction Algorithms

Compressive sensing places the computational load on reconstructing the Nyquist samples x, from the set of compressive samples y. There are many reconstruction algorithms, which fall into three broad classes: convex-optimisations/linear programming, greedy methods, and Bayesian statistical methods (discussed in detail in section 2.4.2). Convex optimisation methods offer better performance at the cost of greater computational complexity. Greedy methods are relatively simpler, but don't have the performance of convex algorithms. A recent greedy method, Approximate Message Passing (AMP), is a blend of both [**donoho2009message**].

Convex algorithms all relax the l-0 requirement for recovery [**tropp2006relax**], by instead solving the equivalent l-1 minimisation problem:

$$\arg\min_x ||x||_1 \text{ s.t } y = Ax \tag{2.4.2.30}$$

in the noiseless case, and

$$\arg\min_x \frac{1}{2}\left|\left|Ax-y\right|\right|_2^2+\lambda\left|\left|x\right|\right|_1 \tag{2.4.2.31}$$

in the noisy case.

The representative algorithm for this type of minimisation is Basis-Pursuit [**Chen1998a**].

Candes and Tao in [**candes2007dantzig**], propose an algorithm based on the following functional:

Greedy methods and iterative shrinkage algorithms, are another family of solutions to 2.4.1. They offer reduced computational complexity with correspondingly worse reconstruction quality and poorer guarantees on sparsity and undersampling than convex algorithms. Examples of this type are Orthogonal Matching Pursuit (OMP)[**tropp2007signal**] and Iterative Thresholding (IT) [**blumensath2009iterative**], [**Beck2009**].

The Greedy family of algorithms abandons exhaustive searches of the solution space in favour of locally optimal single term updates. They proceed by approximating the solution by some active set of columns from the sensinng matrix A and solving a restricted least-squares problem at each (in the case of OMP). This guarantees a maximal reduction in l-2 error in each iteration.

However, due to their greedy nature, these algorithms are not guaranteed to converge: in fact it can be shown that there exist k-sparse vectors and sensing matrices A such that OMP fails to converge in k iterations [**wen2013improved**].

Iterative shrinkage algorithms are similar to greedy algorithms, but replace the single term updates by a iteratively denoised gradient descent. The choice of the (component-wise) denoiser is dependent upon the regulaiser used in 2.4.1. These algorithms have an interpretation as Expectation-Maximisation [**figueiredo2003algorithm**] - where the E-step is performed as gradient descent, and the M-step is the application of the denoiser.

Belief propagation (BP) [**Yedidia2011**] is a popular iterative algorithm, offering improved reconstruction quality and undersampling performance. However, it is a computationally complex algorithm. It is also difficult to implement. Approximate message passing solves this issue by blending BP and (IT).

The algorithm proceeds like iterative thresholding, but computes an adjusted residual at each stage. The extra term comes from a first order approximation to the messages passed by BP [**metzler2014denoising**].

Bayesian Compressive Sensing

Based on the discussion above we can represent the compressive sensing measurements as:

$$\mathbf{g}=\Phi\mathbf{w} \tag{2.4.2.32}$$

where Φ is a $K\times N$ matrix which is the product of the measurement and sparse bases described earlier.

Note that the measurements may be noisy, with the measurement noise represented by a zero mean Gaussian distribution and unknown variance σ^2:

$$\mathbf{g}=\Phi\mathbf{w}+\mathbf{n} \tag{2.4.2.33}$$

Where **n** is the vector representing the vector of noise, and has the same support as the measurements.

Previous sections have shown how the weights w may be found through optimisation methods such as basis pursuit or greedy algorithms. Here, an alternative Bayesian model is described.

From 4.2.2 we have a Gaussian likelihood model:

$$p\left(\mathbf{g} \mid \mathbf{w}, \sigma^2\right) = (2\pi\sigma^2)^{-K/2} \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{g} - \Phi\mathbf{w}\|_2^2\right) \tag{2.4.2.34}$$

The above has converted the CS problem of inverting sparse weight **w** into a linear regression problem with a constraint (prior) that **w** is sparse.

To seek the full posterior distribution over **w** and σ^2, we can chose a sparsity promoting prior. A popular sparseness prior is the Laplace density functions:

$$p(w \mid \lambda) = \left(\frac{\lambda}{2}\right)^N exp - \lambda \sum_{i=1}^{N} |w_i| \tag{2.4.2.35}$$

Note that the solution the convex optimisation problem 2.4.1 corresponds to a maximum *a posteriori* estimate for w using this prior. I.e this prior is equivalent to using the l_1 norm as an optimisation function (see figure 4.4 [**Tibshirani1996**]).

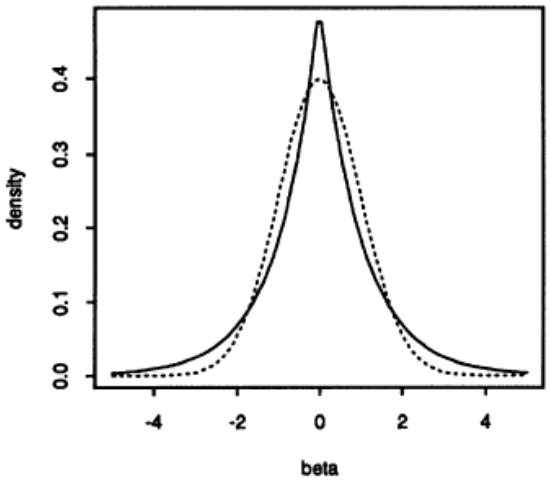

Figure 2.6: The Laplace (l_1-norm, bold line) and Normal (l_2-norm, dotted line) densities. Note that the Laplace density is sparsity promoting as it penalises solutions away from zero more than the Gaussian density. [**Tibshirani1996**]

The full posterior distribution on w and σ^2 may be realised, by using a hierarchical prior instead. To do this, define a zero-mean Gaussian prior on each element of w:

$$p(w \mid a) = \prod_{i=1}^{N} \mathbb{N}\left(w_i \mid 0, \alpha_i^- 1\right) \tag{2.4.2.36}$$

where α is the precision of the distribution. A gamma prior is then imposed on α:

$$p(\alpha \mid a,b) = \prod_{i=1}^{N} \Gamma(\alpha_i \mid a,b) \tag{2.4.2.37}$$

The overall prior is found by marginalising over the hyperparameters:

$$p(w \mid a,b) = \prod_{i=1}^{N} \int_0^{\infty} \mathbb{N}\left(w_i \mid 0, \alpha_i^{-}1\right) \Gamma(\alpha_i \mid a,b) \tag{2.4.2.38}$$

This integral can be done analytically and is a Student-t distribution. Choosing the parameters a, b appropriately we can make the Student-t distribution peak strongly around $w_i = 0$ i.e. sparsifying. This process can be repeated for the noise variance σ^2. The hierarchical model for this process is shown in 8.1. This model, and other CS models which not necessarily have closed form solutions, can be solved via belief-propagation [**Baron2010**]

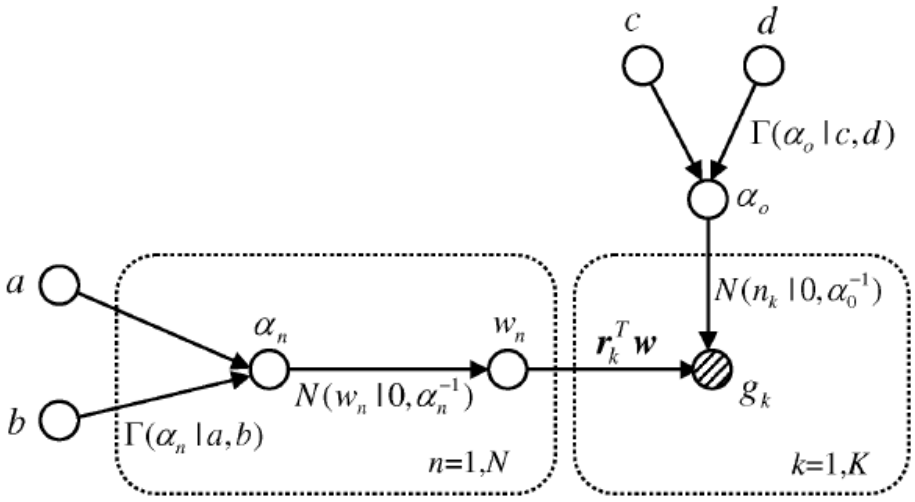

Figure 2.7: The hierarchical model for the Bayesian CS formulation [**Ji2008**]

CHAPTER 3

COMPRESSIVE SENSING

3.1 Introduction and Preliminaries

Compressive sensing is a modern signal acquisition technique in which randomness is used as an effective sampling strategy. In practice many signals encountered 'in the wild' can be fully specified by much fewer bits than required by the Nyquist sampling theorem. This is either a natural property of the signals, for example images have large areas of similar pixels, or as a conscious design choice, for example training sequences in communication transmissions. These signals are not statistically white, and so these signals may be compressed (to save on storage). For example, lossy image compression algorithms can reduce the size of a stored image to about 1% of the size required by Nyquist sampling.

Whilst this vein of research has been extraordinarily successful, it poses the question: if the reconstruction algorithm is able to reconstruct the signal from this compressed representation, why collect all the data in the first place, when most of the information can be thrown away?

Compressed Sensing answers these questions, by way of providing an alternative signal acquisition method to the Nyquist theorem. Specifically, situations are considered where fewer samples are collected than traditional sensing schemes. That is, in contrast to Nyquist sampling, Compressive Sensing is a method of measuring the informative parts of a signal directly without acquiring unessential information at the same time.

Signals which are compressible, are signals whose information content is smaller than the ambient dimension they are acquired in. Such signals have representations in which they are sparse (i.e. the most of the co-efficients in that representation are zero, or close to zero). For example,

1. A sine wave at frequency ω is defined as a single spike in the frequency domain yet has an infinite support in the time domain

2. An image will have values for every pixel, yet the wavelet decomposition of the image will typically only have a few non-zero coefficients

Informally, CS posits that for s-sparse signals $\in \mathbb{R}^n$ - signals with s non-zero amplitudes at unknown locations) - $\mathcal{O}(s \log n)$ measurements are sufficient to exactly reconstruct the signal.

In practice this can be far fewer samples than conventional sampling schemes. For example a megapixel image requires 1,000,000 Nyquist samples, but can be perfectly recovered from 96,000 compressive samples in the wavelet domain [**watkincandes**].

As in classical sensing, the measurements are acquired linearly:

$$y_i = \langle \alpha, \psi_i \rangle \tag{3.1.0.1}$$

where, y_i is the i^{th} sample, $\alpha \in \mathbb{R}^n$ is the signal, and ψ_i is the i^{th} sensing vector.

We require that sensing vectors satisfy two technical conditions (described in detail below): an Isotropy property, which means that components of the sensing vectors have unit variance and are uncorrelated, and an Incoherence property, which means that sensing vectors are almost orthogonal. Once the set of measurements have been taken, the signal may be reconstructed from a simple linear program.

3.1.1 RIP and Stable Embeddings

A high-dimensional signal is sparse, if most of the coefficients x_i in the linear expansion

$$\alpha = \sum_{i=1}^{n} x_i \phi_i \tag{3.1.1.2}$$

are zero, where $x \in \mathbb{R}^n$, $\alpha \in \mathbb{R}$, and ϕ_i are a set of basis functions of $\mathbb{R}^n$. We can make the notion of sparsity precise by defining Σ_s as the set of s-sparse signals in $\mathbb{R}^n$:

Definition 3.1.1.

$$\Sigma_s = \{x \in \mathbb{R}^n : |\text{supp}\,(x)| \leq s\} \tag{3.1.1.3}$$

where $\text{supp}\,(x)$ *is the set of indices on which* x *is non-zero.*

We may not be able to directly obtain these coefficients, as we may not posses an appropriate measuring device or one may not exist, or there is considerable uncertainty about where the non-zero coefficients are. Yet we still are able to measure correlations between the signal and some waveforms ψ_k i.e.

$$y_k = \left\langle \alpha, \psi_k \right\rangle \; k = 1 \ldots m \tag{3.1.1.4}$$

Given a signal $\alpha \in \mathbb{R}^n$, a matrix $A \in \mathbb{R}^{m \times n}$, with $m \ll n$, we can acquire the signal via the set of linear measurements:

$$y = Ax \tag{3.1.1.5}$$

where in this case A represents the sampling system (i.e the columns of A are the products of the two bases ψ, ϕ). In contrast to classical sensing, which requires that $m = n$ for there to be no loss of information, it is possible to reconstruct x from an under-determined set of measurements as long as x is sparse in some basis.

There are two conditions the matrix A needs to satisfy for recovery below Nyquist rates:

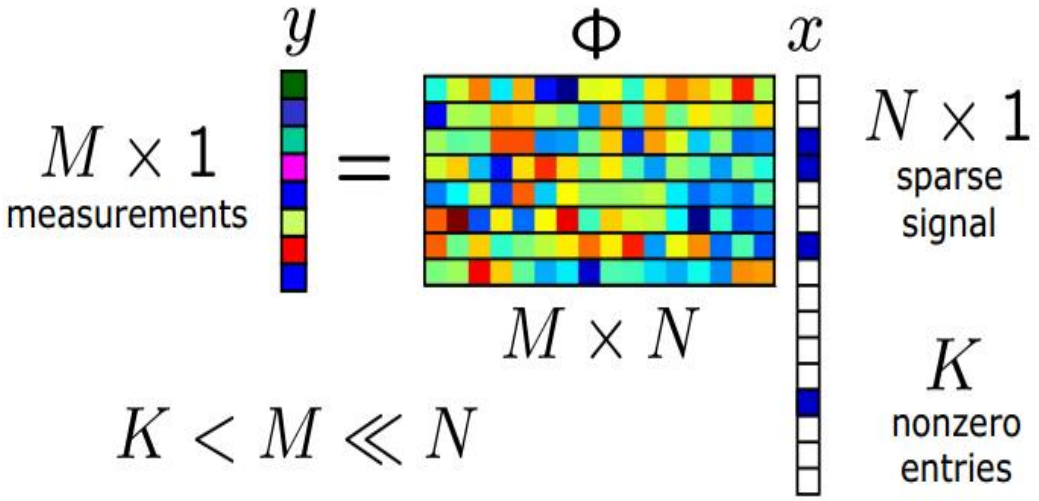

Figure 3.1: A visualisation of the Compressive Sensing problem as an under-determined system

1. Restricted Isometry Property.
2. Incoherence between sensing and signal bases.

Definition 3.1.2 (RIP)**.** *We say that a matrix A satisifes the RIP of order s if there exists a $\delta \in (0,1)$ such that for all $x \in \Sigma_s$:*

$$(1-\delta)\,||x||_2^2 \leq ||Ax||_2^2 \leq (1+\delta)\,||x||_2^2 \tag{3.1.1.6}$$

i.e. A approximately preserves the lengths of all s-sparse vectors in $\mathbb{R}^n$.

Remark 3.1.3. *Although the matrix A is not square, the RIP (7.2.2) ensures that $A^T A$ is close to the identity, and so A behaves approximately as if it were orthogonal. This is formalised in the following lemma from [***shalev2014understanding** *]:*

Lemma 3.1.4. *Let A be a matrix which satisfies the RIP of order $2s$ with RIP constant δ. Then for two disjoint subsets $I, J \subset [n]$ each of size at most s, and for any vector $u \in \mathbb{R}^n$:*

$$\langle Au_I, Au_J \rangle \leq \delta\,||u_I||_2^2\,\big|\big|u_J\big|\big|_2^2 \tag{3.1.1.7}$$

where u_I is the vector with component u_i if $i \in I$ and zero elsewhere.

Remark 3.1.5 (Information Preservation)**.** *A necessary condition to recover all s-sparse vectors from the measurements Ax is that $Ax_1 \neq Ax_2$ for any pair $x \not= x_2$, $x_1, x_2 \in \Sigma_s$, which is equivalent to $||A(x_1 - x_2)||_2^2 > 0$.*

This is guaranteed as long as A satisfies the RIP of order $2s$ with constant δ - as the vector $x_1 - x_2$ will hve at most $2s$ non-zero entries, and so will be distinguishable after multiplication with A. To complete the argument take $x = x_1 - x_2$ in definition (7.2.2), guaranteeing $||A(x_1 - x_2)||_2^2 > 0$, and requiring the RIP order of A to be $2s$.

Remark 3.1.6 (Stability)**.** *We also require that the dimensionality reduction of compressed sensing is the preservation of relative distances: that is if x_1 and x_2 are far apart in $\mathbb{R}^n$ then their projections Ax_1 and Ax_2 are far apart in $\mathbb{R}^m$. This will guarantee that the dimensionality reduction is robust to noise.*

A requirement on the matrix A that satisfies both of these conditions is the following:

Definition 3.1.7 (δ-stable embedding)**.** *We say that a mapping is a δ-stable embedding of $U, V \subset \mathbb{R}^n$ if*

$$(1-\delta)\,||u-v||_2^2 \leq ||Au-Av||_2^2 \leq (1+\delta)\,||u-v||_2^2 \tag{3.1.1.8}$$

for all $u \in U$ and $v \in V$.

Remark 3.1.8. *Note that a matrix A, satisfying the RIP of order $2s$ is a δ-stable embedding of Σ_s, Σ_s.*

Remark 3.1.9. *Definition 7.2.5 has a simple interpretation: the matrix A must approximately preserve Euclidean distances between all points in the signal model Σ_s.*

Given that we know a basis in which our signal is sparse, ϕ, how do we choose ψ, so that we can accomplish this sensing task? In classical sensing, we choose ψ_k to be the set of T_s-spaced delta functions (or equivalently the set of $1/T_s$ spaced delta functions in the frequency domain). A simple set of ψ_k would be to choose a (random) subset of the delta functions above.

In general, we seek waveforms in which the signals' representation would be dense.

Definition 3.1.10 (Incoherence)**.** *A pair of bases is said to be incoherent if the largest projection of two elements between the sensing (ψ) and representation (ϕ) basis is in the set $[1, \sqrt{n}]$, where n is the dimension of the signal. The coherence of a set of bases is denoted by μ.*

Examples of pairs of incoherent bases are:

- Time and Fourier bases: Let $\Phi = \mathbf{I}_n$ be the canonical basis and $\Psi = \mathbf{F}$ with $\psi_i = n^{-12} e^{i\omega k}$ be the Fourier basis, then $\mu(\phi, \psi) = 1$. This corresponds to the classical sampling strategy in time or space.
- Consider the basis Φ to have only entries in a single row, then the coherence between Φ and any fixed basis Ψ will be $\sqrt{n}$.
- Random matrices are incoherent with any fixed basis Ψ. We can choose Φ by creating n orthonormal vectors from n vectors sampled independently and uniformly on the unit sphere. With high probability $\mu = \sqrt{n \log n}$. This extends to matrices whose rows are created by sampling independent Gaussian or Bernoulli random vectors.

This implies that sensing with incoherent systems is good (in the sine wave example above it would be better to sample randomly in the time domain as opposed to the frequency domain), and efficient mechanisms ought to acquire correlations with random waveforms (e.g. white noise).

Theorem [**Candes2006**] Fix a signal $f \in \mathbb{R}^n$ with a sparse coefficient basis, x_i in ϕ. Then a reconstruction from m random measurements in ψ is possible with probability $1-\delta$ if:

$$m \geq C\mu^2(\phi, \psi) S \log\left(\frac{n}{\delta}\right) \tag{3.1.1.9}$$

where $\mu(\phi, \psi)$ is the coherence of the two bases, and S is the number of non-zero entries on the support of the signal.

3.1.2 Random Matrix Constructions

To construct matrices satisfying definition 7.2.5, given m, n we generate A by A_{ij} being i.i.d random variables from distributions with the following conditions [**davenport2010signal**]

Condition 1 (Norm preservation)**.** $\mathbb{E}A_{ij}^2 = \frac{1}{m}$

Condition 2 (sub-Gaussian)**.** $\mathbb{E}\left(e^{A_{ij}t}\right) \leq e^{C^2t^2/2}$

Random variables A_{ij} satisfying conditions (3) and (4) satisfy the following concentration inequality [**baraniuk2008simple**]:

Lemma 3.1.11 (sub-Gaussian)**.**

$$\mathbb{P}\left(|\,||Ax||_2^2 - ||x||_2^2\,| \geq \varepsilon\,||x||_2^2\right) \leq 2e^{-cM\varepsilon^2} \tag{3.1.2.10}$$

Then in [**baraniuk2008simple**] the following theorem is proved:

Theorem 3.1.12. *Suppose that m, n and $0 < \delta < 1$ are given. If the probability distribution generating A satisfies condition* (7.2.2.13)*, then there exist constants c_1, c_2 depending only on δ such that the RIP* (7.2.2) *holds for A with the prescribed δ and any $s \leq \frac{c_1 n}{\log n/s}$ with probability $\geq 1 - 2e^{-c_2 n}$*

For example, if we take $A_{ij} \sim \mathcal{N}(0, 1/m)$, then the matrix A will satisfy the RIP

3.1.3 Wishart Matrices

Let $\{X_i\}_{i=1}^r$ be a set of i.i.d $1 \times p$ random vectors drawn from the multivariate normal distribution with mean 0 and covariance matrix H.

$$X_i = \left(x_1^{(i)}, \ldots, x_p^{(i)}\right) \sim N(0, H) \tag{3.1.3.11}$$

We form the matrix X by concatenating the r random vectors into a $r \times p$ matrix.

Definition 3.1.13 (Wishart Matrix)**.** *Let*

$$W = \sum_{j=1}^{r} X_j X_j^T = XX^T \tag{3.1.3.12}$$

Then $W \in \mathbb{R}^{r \times r}$ has the Wishart distribution with parameters

$$W_r\left(H, p\right) \tag{3.1.3.13}$$

where p is the number of degrees of freedom.

Remark 3.1.14. *This distribution is a generalisation of the Chi-squared distribution: let $p = H = 1$.*

Theorem 3.1.15 (Expected Value)**.**

$$\mathbb{E}(W) = rH \tag{3.1.3.14}$$

Proof.

$$\begin{aligned}\mathbb{E}(W) &= \mathbb{E}\left(\sum_{j=1}^{r} X_j X_j^T\right) \\ &= \sum_{j=1}^{r} \mathbb{E}\left(X_j X_j^T\right) \\ &= \sum_{j=1}^{r} \left(\mathrm{Var}(X_j) + \mathbb{E}(X_j)\mathbb{E}(X_j^T)\right) \\ &= rH\end{aligned}$$

Where the last line follows as X_j is drawn from a distribution with zero mean. □

Remark 3.1.16. *The matrix $M = A^T A$, where A is constructed by the methods from section 7.2.2, will have a Wishart distribution. In particular, it will have $\mathbb{E}M = \frac{1}{m} I_n$*

The joint distribution of the eigenvalues is given by [**levequeMatrices**]:

$$p(\lambda_1, \ldots, \lambda_r) = c_r \prod_{i=1}^{r} e^{-\lambda_i} \prod_{i<j} \left(\lambda_i - \lambda_j\right)^2 \tag{3.1.3.15}$$

The eigenvectors are uniform on the unit sphere in $\mathbb{R}^r$.

3.1.4 Reconstruction Algorithms

Compressive sensing places the computational load on reconstructing the Nyquist samples x, from the set of compressive samples y. This is in contrast to traditional sensing, where the heavy lifting computationally is done by the process with discretises the continuous signal to create the digital samples.

Many recovery algorithms have been proposed, and all are based upon minimising some functional of the data. Generally, this is based upon two terms: a data fidelity term, minimising the discrepancy between the reconstruction and the true data, and regularisation term, biasing the reconstruction towards a class of solutions with desirable properties, for example sparsity. Typically the squared error $\frac{1}{2}\left|\left|y - Ax\right|\right|_2^2$ is chosen as the data fidelity term, whilst a number of regularisation terms have been introduced in the literature.

A particularly important functional is:

$$\arg\min_x ||x||_1 \text{ s.t } y = Ax \tag{3.1.4.16}$$

known as Basis Pursuit [**Chen1998a**], with the following program known as the LASSO [**tibshirani1996regression**] as a noisy generalisation:

$$\arg\min_x \frac{1}{2}\left|\left|Ax - y\right|\right|_2^2 + \lambda ||x||_1 \tag{3.1.4.17}$$

The statistical properties of LASSO have been well studied. The program performs, both regularisation and variable selection: the parameter λ trades of data fidelity and sparsity with higher values of λ leading to sparser solutions.

The LASSO shares several features with Ridge regression, and the Non-negative garotte (used for subset regression). It can be shown [**hastie2005elements**], that the solution to (3.1.4.17) can be written as:

$$\hat{x} = S_\lambda\left(x^{OLS}\right) = x^{OLS}\text{sign}\left(x_i - \lambda\right) \tag{3.1.4.18}$$

where $x^{OLS} = (A^T A)^{-1} A^T y$, whereas the solution to Ridge regression:

$$\arg\min_x \frac{1}{2}\left|\left|Ax - y\right|\right|_2^2 + \lambda\left|\left|x\right|\right|_2^2 \tag{3.1.4.19}$$

can be written as:

$$\hat{x} = (1+\lambda)^{-1} x^{OLS} \tag{3.1.4.20}$$

and the solution to the best subset regression:

$$\arg\min_x \frac{1}{2}\left|\left|Ax - y\right|\right|_2^2 + \lambda\left|\left|x\right|\right|_0 \tag{3.1.4.21}$$

where $||x||_0 = \{|i| : x_i \neq 0\}$, can be written as:

$$\hat{x} = H_\lambda\left(x^{OLS}\right) = x^{OLS}\mathbb{I}\left(|x^{OLS}| > \lambda\right) \tag{3.1.4.22}$$

where $\mathbb{I}$ is the indicator function. From (3.1.4.22) and (3.1.4.20), we can see that the solution to (3.1.4.17), (3.1.4.18), translates coefficients towards zero by a constant factor, and set coefficients to zero if they are too small; thus the LASSO is able to perform both model selection (choosing relevant covariates) and regularisation (shrinking model coefficients).

Figure (3.2), provides a graphical demonstration of why the LASSO promotes sparse solutions. (3.1.4.17) can also be thought of as the best convex approximation of the ℓ_0 problem (3.1.4.21), as the ℓ_1-norm is the convex hull of the points defined by $||x||_p$ for $p < 1$ as $p \to 0$.

Other examples of regularisers are:

- Elastic Net: This estimator is a blend of both (3.1.4.17) and (3.1.4.17), found by minimising:

$$\arg\min_x \frac{1}{2}\left|\left|Ax - y\right|\right|_2^2 + \lambda\left|\left|x\right|\right|_2^2 + \mu\left|\left|x\right|\right|_1 \tag{3.1.4.23}$$

  The estimate has the benefits of both Ridge and Lasso regression: feature selection from the LASSO, and regularisation for numerical stability (useful in the under-determined case we consider here) from Ridge regression. The Elastic-net will outperform the LASSO when there is a high degree of collinearity between coefficients of the true solution.

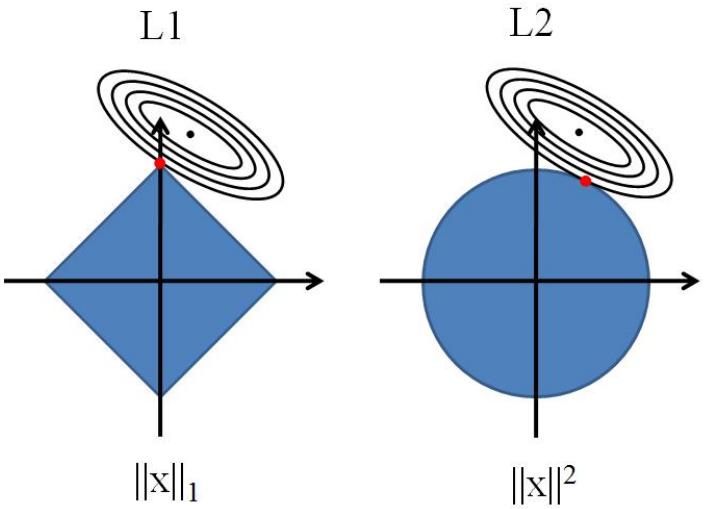

Figure 3.2: Solutions to the Compressive Sensing optimisation problem intersect the l_1 norm the points where all components (but one) of the vector are zero (i.e. it is sparsity promoting) [**Tibshirani1996**]

- TV regularisation

$$\arg\min_x \frac{1}{2}\left|\left|Ax-y\right|\right|_2^2 + \lambda\,||\nabla x||_1 \tag{3.1.4.24}$$

This type of regularisation is used when preserving edges whilst simultaneously de-noising a signal is required. It is used extensively in image processing, where signals exhibit large flat patches alongside large discontinuities between groups of pixels.

- Tree Lasso
- Candes and Tao in [**candes2007dantzig**], propose an alternative functional:

$$\min_{x\in\mathbb{R}^n} ||x||_1 \text{ s.t. } \left|\left|A^T(Ax-y)\right|\right|_\infty \leq t\sigma \tag{3.1.4.25}$$

with $t = c\sqrt{2\log n}$. Similarly to the LASSO this functional selects sparse vectors consistent with the data, in the sense that the residual $r = y - Ax$ is smaller than the maximum amount of noise present. In [**candes2007dantzig**] it was shown that the l_2 error of the solution is within a factor of $\log n$ of the ideal l_2 error. More recent work by Bikel, Ritov, and Tsybakov, [**bickel2009simultaneous**], has shown that the LASSO enjoys similar properties.

Broadly reconstruction algorithms fall into three classes: convex-optimisations/linear programming, greedy methods, and Bayesian methods. Convex optimisation methods offer better performance, measured in terms of reconstruction accuracy, at the cost of greater computational complexity. Greedy methods are relatively simpler, but don't have the reconstruction guarantees of

procedure IST($y, A, \mu, \tau, \varepsilon$)
    $x^0 = 0$
    while $\left|\left|x^t - x^{t-1}\right|\right|_2^2 \leq \varepsilon$ **do**
        $x^{t+1} \leftarrow S_{\mu\tau}\left(x^t + \tau A^T z^t\right)$
        $z^t \leftarrow y - Ax^t$
    end while
    return x^{t+1}
end procedure

Figure 3.3: The Iterative Soft Thresholding Algorithm

convex algorithms. Bayesian methods offer the best reconstruction guarantees, as well as uncertainty estimates about the quality of reconstruction, but come with considerable computational complexity.

A recent greedy method, Approximate Message Passing (AMP), is a blend of both greedy and Bayesian methods [**donoho2009message**].

Convex methods cast the optimisation objective either as a linear program with linear constraints, or as a second order cone program with quadratic constraints. Both of these types of program can be solved with first order interior point methods. However, their practical application to compressive sensing problems is limited due to their polynomial dependence upon the signal dimension and the number of constraints.

Compressive Sensing poses a few difficulties for convex optimisation based methods. In particular, many of the unconstrained objectives are non-smooth meaning methods based upon descent down a smooth gradient are inapplicable.

To overcome these difficulties, a series of algorithms originally proposed for wavelet-based image de-noising have been applied to CS, known as iterative shrinkage methods. These have the desirable property that they boil down to matrix-vector multiplications and component-wise thresholding.

Iterative shrinkage algorithms replace searching for a minimal facet of a complex polytope by a iteratively denoised gradient descent. The choice of the (component-wise) denoiser is dependent upon the regulaiser used in 3.1.4.17. These algorithms have an interpretation as Expectation-Maximisation [**figueiredo2003algorithm**] - where the E-step is performed as gradient descent, and the M-step is the application of the denoiser.

Greedy methods are another family of solutions to 3.1.4.17. They offer reduced computational complexity with correspondingly worse reconstruction quality and poorer guarantees on sparsity and undersampling than convex algorithms. Examples of this type are Orthogonal Matching Pursuit (OMP)[**tropp2007signal**], and Compressive Sensing Orthogonal Matching Pursuit (CoSAMP).

The Greedy family of algorithms abandons exhaustive searches of the solution space in favour of locally optimal single term updates. They proceed by approximating the solution by some active set of columns from the sensinng matrix A and solving a restricted least-squares problem at each (in the case of OMP). This guarantees a maximal reduction in l-2 error in each iteration.

Despite their computational simplicity, greedy algorithms have several drawbacks. Primarily they do not come with stable recovery guarantees, and they require a larger number of samples to recover the signal when compared to Bayesian and Convex recovery algorithms. Also, due to their greedy nature, these algorithms are not guaranteed to converge: in fact it can be shown that there exist k-sparse vectors and sensing matrices A such that OMP fails to converge in k iterations

procedure OMP(y, A, K, ε)
  $x^0 = 0$, $r = y$, $\Omega = \varnothing$, $i = 0$
  while $\left|\left|x^t - x^{t-1}\right|\right|_2^2 \leq \varepsilon$ **do**
    $i \leftarrow i + 1$
    $b \leftarrow A^T r$
    $\Omega \leftarrow \Omega \cup \text{supp}\,(H_1\,(b))$
    $x\restriction_\Omega \leftarrow A^T\restriction_\Omega x$
    $x\restriction_{\Omega^c} \leftarrow 0$
    $b \leftarrow y - Ax$
  end while
  return x
end procedure

Figure 3.4: The OMP recovery algorithm

procedure AMP(y, A, ε)
  $x^0 = 0$, $z^0 = A^T y$
  while $\left|\left|x^t - x^{t-1}\right|\right|_2^2 \leq \varepsilon$ **do**
    $x^{t+1} \leftarrow S_{\mu\tau}\left(x^t - \tau + A^T z^t\right)$
    $z^{t+1} \leftarrow y - Ax^t + \frac{||x||_0}{m} z^t$
  end while
  return x^{t+1}
end procedure

Figure 3.5: The AMP recovery algorithm

[**wen2013improved**].

Bayesian methods reformulate the optimisation problem into an inference problem. These methods come with a unified theory, and standard methods to produce solutions. The theory is able to handle hyper-parameters in a unified way, provides a flexible modelling framework, and is able to provide desirable statistical quantities such as the uncertainty inherent in the prediction.

Based on the discussion above we can represent the compressive sensing measurements as:

$$y = Ax \tag{3.1.4.26}$$

where A is a $K \times N$ matrix which is the product of the measurement and sparse bases described earlier.

Note that the measurements may be noisy, with the measurement noise represented by a zero mean Gaussian distribution and unknown variance σ^2:

$$y = Ax + n \tag{3.1.4.27}$$

Where **n** is the vector representing the vector of noise, and has the same support as the measurements.

Previous sections have shown how the weights x may be found through optimisation methods such as basis pursuit or greedy algorithms. Here, an alternative Bayesian model is described.

From 4.2.2 we have a Gaussian likelihood model:

$$p\left(y \mid z,\sigma^2\right) = (2\pi\sigma^2)^{-K/2} \exp\left(-\frac{1}{2\sigma^2}\|y - Ax|_2^2\right) \tag{3.1.4.28}$$

The above has converted the CS problem of inverting sparse weight **w** into a linear regression problem with a constraint (prior) that **w** is sparse.

To seek the full posterior distribution over **w** and σ^2, we can chose a sparsity promoting prior. A popular sparseness prior is the Laplace density functions:

$$p\left(x \mid \lambda\right) = \left(\frac{\lambda}{2}\right)^N exp{-\lambda \sum_{i=1}^{N} |x_i|} \tag{3.1.4.29}$$

Note that the solution the convex optimisation problem 2.4.1 corresponds to a maximum *a posteriori* estimate for w using this prior. I.e this prior is equivalent to using the l_1 norm as an optimisation function (see figure 4.4 [**Tibshirani1996**]).

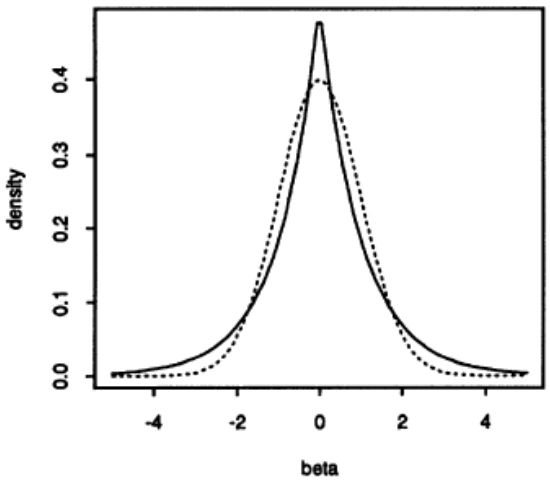

Figure 3.6: The Laplace (l_1-norm, bold line) and Normal (l_2-norm, dotted line) densities. Note that the Laplace density is sparsity promoting as it penalises solutions away from zero more than the Gaussian density. [**Tibshirani1996**]

The full posterior distribution on w and σ^2 may be realised, by using a hierarchical prior instead. To do this, define a zero-mean Gaussian prior on each element of w:

$$p\left(w \mid a\right) = \prod_{i=1}^{N} \mathbb{N}\left(w_i \mid 0, \alpha_i^- 1\right) \tag{3.1.4.30}$$

where α is the precision of the distribution. A gamma prior is then imposed on α:

(3.1.4.31) $$p(\alpha \mid a,b) = \prod_{i=1}^{N} \Gamma(\alpha_i \mid a,b)$$

The overall prior is found by marginalising over the hyperparameters:

(3.1.4.32) $$p(w \mid a,b) = \prod_{i=1}^{N} \int_0^{\infty} \mathbb{N}\left(w_i \mid 0, \alpha_i^{-}1\right) \Gamma(\alpha_i \mid a,b)$$

This integral can be done analytically and is a Student-t distribution. Choosing the parameters a, b appropriately we can make the Student-t distribution peak strongly around $w_i = 0$ i.e. sparsifying. This process can be repeated for the noise variance σ^2. The hierarchical model for this process is shown in 8.1. This model, and other CS models which not necessarily have closed form solutions, can be solved via belief-propagation [**Baron2010**], or via Monte-Carlo methods.

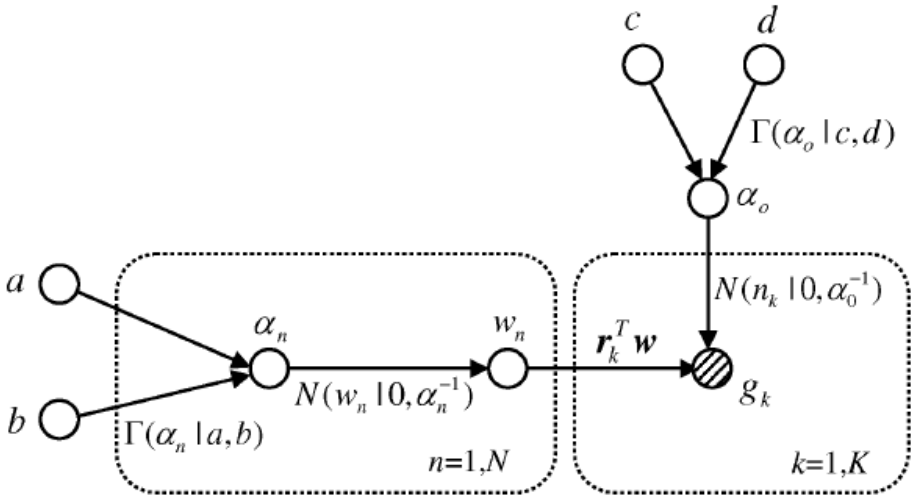

Figure 3.7: The hierarchical model for the Bayesian CS formulation [**Ji2008**]

However, as with all methodologies, Bayesian algorithms have their drawbacks. Most notable is the use of the most computationally complex recovery algorithms. In particular MCMC methods suffer in high dimensional settings, such as those considered in compressive sensing. There has been an active line of work to address this: most notably Hamiltonian Monte Carlo - an MCMC sampling method designed to follow the typical set of the posterior density.

Belief propagation (BP) [**Yedidia2011**] is a popular iterative algorithm, offering improved reconstruction quality and undersampling performance. However, it is a computationally complex algorithm. It is also difficult to implement. Approximate message passing (3.5) solves this issue by blending BP and (IT).

The algorithm proceeds like iterative thresholding, but computes an adjusted residual at each stage. The extra term comes from a first order approximation to the messages passed by BP [**metzler2014denoising**].

The choice of prior is key in Bayesian inference, as it encodes all knowledge about the problem. Penalising the least-squares estimate with the ℓ_1 norm,

3.2 Compressive Sensing Architechtures

3.2.1 Modulated Wideband Converter

We consider a radio environment with a single primary user (PU) and a network of J nodes collaboratively trying to sense and reconstruct the PU signal, either in a fully distributed manner (by local communication), or by transmitting measurements to a fusion centre which then solves the linear system.

We try to sense and reconstruct a wideband signal, divided into L channels. We have a (connected) network of J (= 50) nodes placed uniformly at random within the square $[0,1] \times [0,1]$. This is the same model, as in [**Zhang2011b**]. The calculations which follow are taken from [**Zhang2011b**] as well.

The nodes individually take measurements (as in [**mishali2010theory**]) by mixing the incoming analogue signal $x(t)$ with a mixing function $p_i(t)$ aliasing the spectrum. $x(t)$ is assumed to be bandlimited and composed of up to k uncorrelated transmissions over the L possible narrowband channels - i.e. the signal is k-sparse.

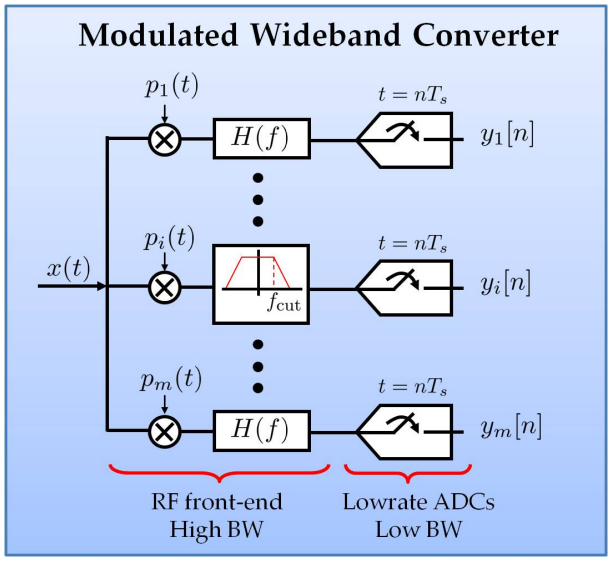

Figure 3.8: Mse vs SNR for the sensing model, with AWGN only, showing the performance of distributed and centralised solvers

The mixing functions - which are independent for each node - are required to be periodic, with period T_p. Since p_i is periodic it has Fourier expansion:

$$p_i(t) = \sum_{l=-\infty}^{\infty} c_{il} \exp\left(jlt\frac{2\pi}{T_p}\right) \tag{3.2.1.33}$$

The c_{il} are the Fourier coefficients of the expansion and are defined in the standard manner. The result of the mixing procedure in channel i is therefore the product xp_i, with Fourier transform (we denote the Fourier Transform of x by $X(\cdot)$):

$$X_i(f) = \int_{-\infty}^{\infty} x(t)\, p_i(t)\, dt$$

$$= \sum_{l=-\infty}^{\infty} c_{il} X(f - l f_p) \tag{3.2.1.34}$$

(We insert the Fourier series for p_i, then exchange the sum and integral). The output of this mixing process then, is a linear combination of shifted copies of $X(f)$, with at most $\lceil f_N Y Q / f_p \rceil$ terms since $X(f)$ is zero outside its support (we have assumed this Nyquist frequency exists, even though we never sample at that rate).

This process is repeated in parallel at each node so that each band in x appears in baseband.

Once the mixing process has been completed the signal in each channel is low-pass filtered and sampled at a rate $f_s \geq f_p$. In the frequency domain this is a ideal rectangle function, so the output of a single channel is:

$$Y_i\left(e^{j2\pi f T_s}\right) = \sum_{l=-L_0}^{+L_0} c_{il} X(f - l f_p) \tag{3.2.1.35}$$

since frequencies outside of $[-f_s/2, f_s/2]$ will filtered out. L_0 is the smallest integer number of non-zero contributions in $X(f)$ over $[-f_s/2, f_s/2]$ - at most $\lceil f_N Y Q / f_p \rceil$ if we choose $f_s = f_p$. These relations can be written in matrix form as:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w} \tag{3.2.1.36}$$

where $\mathbf{y}$ contains the output of the measurement process, and $\mathbf{A}$ is a product matrix of the mixing functions, their Fourier coefficients, a partial Fourier Matrix, and a matrix of channel coefficients. $\mathbf{x}$ is the vector of unknown samples of $x(t)$.

i.e. $\mathbf{A}$ can be written:

$$\mathbf{A}^{m \times L} = \mathbf{S}^{m \times L} \mathbf{F}^{L \times L} \mathbf{D}^{L \times L} \mathbf{H}^{L \times L} \tag{3.2.1.37}$$

The system 6.3.0.10 can then be solved (in the sense of finding the sparse vector $\mathbf{x}$ by convex optimisation via minimising the objective function:

$$\frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{x}\|_1 \tag{3.2.1.38}$$

where λ is a parameter chosen to promote sparsity. Larger λ means sparser $\mathbf{x}$.

3.2.2 Random Demodulator

We assume that the analogue signal $x(t)$ is comprised of a finite number of components from some arbitrary dictionary $\psi_n(t)$:

$$x(t) = \sum_{n=1}^{N} \alpha_n \psi_n(t) \tag{3.2.2.39}$$

The signal is said to be sparse when there are only a few non-zero α_n. The dictionary elements ψ_n may have a relatively high bandwidth, but the signal itself will have only a few degrees of freedom.

The signal acquisition method proposed consists of three stages (all analogue processing): demodulation, filtering and uniform sampling.

Initially, the signal is modulated by a pseudo-random sequence $p_c(t)$, which alternates at frequencies at (or above) the Nyquist frequency of $x(t)$. The signal is then filtered, through a filter with impulse response $h(t)$, before being sampled at rate $\mathscr{M}$ with a traditional ADC.

The output of this system, $y[m]$, can be related to the input $x(t)$ via a linear transformation of the coefficient vector α_n.

To find the transformation A, first consider the output of $y[m]$, which is the result of convolution and demodulation followed by sampling at rate $\mathscr{M}$:

$$y[m] = \int_{-\infty}^{\infty} x(\tau)\, p_c(\tau)\, h(t-\tau)\,|_{t=m\mathscr{M}}\, d\tau \tag{3.2.2.40}$$

and by expanding $x(t) = \sum_{n=1}^{N} \alpha_n \psi_n(t)$:

$$y[m] = \sum_{n=1}^{N} \alpha_n \int_{-\infty}^{\infty} \psi_n(t)\, p_c(\tau)\, h(m\mathscr{M}-\tau)\, d\tau \tag{3.2.2.41}$$

we see that the output can be written as:

$$y = Ax \tag{3.2.2.42}$$

with

$$A_{m,n} = \int_{-\infty}^{\infty} \psi_n(t)\, p_c(\tau)\, h(m\mathscr{M}-\tau)\, d\tau \tag{3.2.2.43}$$

CHAPTER 4

ADMM

4.1 Introduction

There is an almost ubiquitous growing demand for mobile and wireless data, with consumers demanding faster speeds and better quality connections in more places. Consequently 4G is now being rolled out in the UK and US and with 5G being planned for 2020 and beyond [**Dahlman2014**].

However, there is constrained amount of frequencies over which to transmit this information; and demand for frequencies that provide sufficient bandwidth, good range and in-building penetration is high.

Not all spectrum is used in all places and at all times, and judicious spectrum management, by developing approaches to use white spaces where they occur, would likely be beneficial.

Broadly, access to spectrum is managed in two, complementary ways, namely through licensed and licence exempt access. Licensing authorises a particular user (or users) to access a specific frequency band. Licence exemption allows any user to access a band provided they meet certain technical requirements intended to limit the impact of interference on other spectrum users.

A licence exempt approach might be particularly suitable for managing access to white spaces. Devices seeking to access white spaces need a robust mechanism for learning of the frequencies that can be used at a particular time and location. One approach is to refer to a database, which maps the location of white spaces based on knowledge of existing spectrum users. An alternative approach is for devices to detect white spaces by monitoring spectrum use.

The advantages of spectrum monitoring [**akan2009cognitive**] over maintaining a database of space-frequency data are the ability of networks to make use of low-cost low-power devices, only capable of making local (as opposed to national) communications, keeping the cost of the network low and opportunistic channel usage for bursty traffic, reducing channel collisions in dense networks.

The realisation of any Cognitive Radio standard (such as IEEE 802.22 [**stevenson2009ieee**]), requires the co-existence of primary (e.g. TV users) and secondary (everybody else who wants to use TVWS spectrum) users of the frequency spectrum to ensure proper interference mitigation and appropriate network behaviour. We note, that whereas TVWS bands are an initial step towards

dynamic spectrum access, the principles and approaches we describe are applicable to other frequency bands - in particular it makes ultra-wideband spectrum sensing possible.

The challenges of this technology are that Cognitive Radios (CRs) must sense whether spectrum is available, and must be able to detect very weak primary user signals. Furthermore they must sense over a wide bandwidth (due to the amount of TVWS spectrum proposed), which challenges traditional Nyquist sampling techniques, because the sampling rates required are not technically feasible with current RF or Analogue-to-Digital conversion technology.

Due to the inherent sparsity of spectral utilisation, Compressive Sensing (CS) [**Candes2006**] is an appropriate formalism within which to tackle this problem. CS has recently emerged as a new sampling paradigm allowing images to be taken from a single pixel camera for example. Applying this to wireless communication, we are able to reconstruct sparse signals at sampling rates below what would be required by Nyquist theory, for example the works [**mishali2010theory**], [**Mishali2010a**], [**Mishali2009**], [**Mishali2011**], and [**tropp2010beyond**] detail how this sampling can be achieved.

However, even with CS, spectrum sensing from a single machine will be costly as the proposed TVWS band will be over a large frequency range (for instance in the UK the proposed TVWS band is from 470 MHz to 790 MHz, requiring traditional sampling rates of ~600 MHz). CS at a single sensor would still require high sampling rates. In this report we propose a distributed model, which allows a sensing budget at each node far below what is required by centralised CS.

4.2 Wideband Spectrum Sensing

This section presents a new method of sensing sparse signals, and its application to the problem of sensing over wideband spectra in Cognitive Radios. Initially we introduce Classical Sensing and then give an overview of both Compressive Sensing and Group Testing. Finally, we discuss some sub-Nyquist sampling techniques.

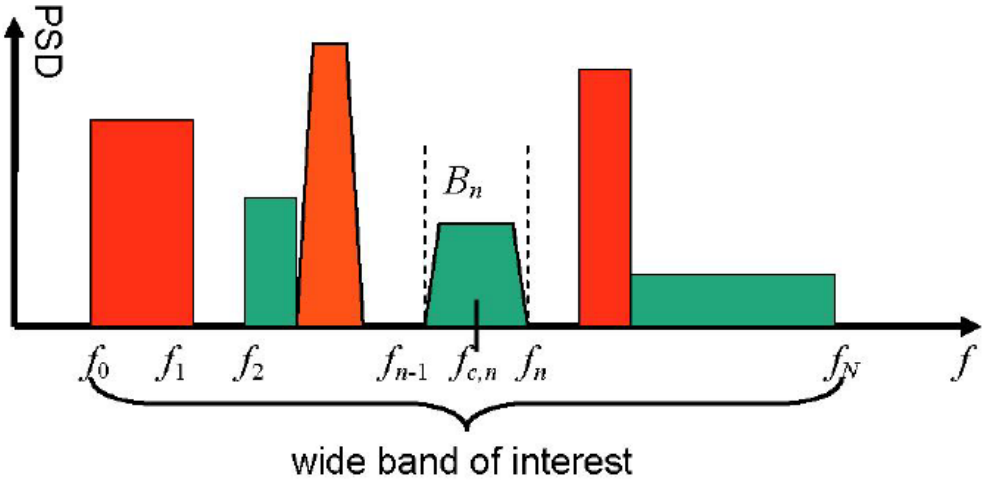

Figure 4.1: A digram of the Spectrum Sensing model [**Tian**]

4.2.1 Compressed Sensing

Compressive sensing is a modern signal acquisition technique in which randomness is used as an effective sensing strategy for classes of signals typically encountered in practice.

Informally, CS posits that for k-sparse signals $\in \mathbb{R}^n$ - signals with k non-zero amplitudes at unknown locations) - $k \log n$ measurements are sufficient to exactly reconstruct the signal. In other words we can sample at the information rate, without information loss.

For TVWS signals, this reasoning can be inverted: signals with very large bandwidth - but with a sparse spectrum - can be sampled randomly in time at rates below below those thought sufficient by the Nyquist theorem.

This work has been extended to cases where the signal isn't exactly sparse, and where the measurements are imperfect.

The central idea of CS is that randomness is an effective sensing strategy. We require that sensing vectors satisfy two technical conditions (described in detail below): an Isotropy property, which means that components of the sensing vectors have unit variance and are uncorrelated, and an Incoherence property, which means that sensing vectors are almost orthogonal. These conditions are summed up in the Restricted Isometry Property.

Once the set of measurements have been taken, the signal may be reconstructed from a simple linear program.

In practice many signals encountered 'in the wild' can be fully specified by much fewer bits than required by the Nyquist sampling theorem. This is either a natural property of the signals, for example images have large areas of similar pixels, or as a conscious design choice, for example training sequences in communication transmissions. These signals are not statistically white, and so these signals may be compressed (to save on storage). For example, lossy image compression algorithms can reduce the size of a stored image to about 1% of the size required by Nyquist sampling.

Whilst this vein of research has been extraordinarily successful, it poses the question: if the reconstruction algorithm is able to reconstruct the signal from this compressed representation, why collect all the data in the first place, when most of the information can be thrown away? Is it possible to directly measure the part that will not end up being thrown away?

Compressed Sensing answers these questions, by way of providing an alternative signal acquisition method to the Nyquist theorem. Specifically, situations are considered where fewer samples are collected than traditional sensing schemes.

That is, in contrast to Nyquist sampling, Compressive Sensing is a method of measuring the informative parts of a signal directly without acquiring unessential information at the same time.

Signals which are compressible, are signals whose information content is smaller than the ambient dimension they are acquired in. Such signals have representations in which they are sparse (i.e. the most of the co-efficients in that representation are zero, or close to zero). For example,

1. A sine wave at frequency ω is defined as a single spike in the frequency domain yet has an infinite support in the time domain

2. An image will have values for every pixel, yet the wavelet decomposition of the image will typically only have a few non-zero coefficients

We may not be able to directly obtain those coefficients, as we may not posses an appropriate measuring device or one may not exist, or there is considerable uncertainty about where the non-zero coefficients are. Yet we still are able to measure correlations between the signal and some

waveforms ϕ_k i.e.

$$y_k = \langle f, \phi_k \rangle \; k = 1 \dots m \tag{4.2.1.1}$$

for $f \in \mathbb{R}^n$ expanded in an orthonormal basis ψ s.t.

$$f(t) = \sum_{i=1}^{n} x_i \psi_i(t) \tag{4.2.1.2}$$

where the x_i are the coefficient sequence of f.

Given that we know a basis in which our signal is sparse, ϕ, how do we choose ψ, so that we can accomplish this sensing task? In classical sensing, we choose ψ_k to be the set of T_s-spaced delta functions (or equivalently the set of $1/T_s$ spaced delta functions in the frequency domain). A simple set of ψ_k would be to choose a (random) subset of the delta functions above.

In general, we seek waveforms in which the signals' representation would be dense.

Definition 2. *A pair of bases is said to be incoherent if the largest projection of two elements between the sensing (ψ) and representation (ϕ) basis is in the set* $[1, \sqrt{n}]$*, where n is the dimension of the signal.*

The coherence of a set of bases is denoted by μ.

This implies that sensing with incoherent systems is good (in the sine wave example above it would be better to sample randomly in the time domain as opposed to the frequency domain), and efficient mechanisms ought to acquire correlations with random waveforms (e.g. white noise).

Theorem [**Candes2006**] Fix a signal f $\in \mathbb{R}^n$ with a sparse coefficient basis, x_i in ϕ. Then a reconstruction from m random measurements in ψ is possible with probability $1-\delta$ if:

$$m \geq C\mu^2(\phi, \psi) S \log\left(\frac{n}{\delta}\right) \tag{4.2.1.3}$$

where $\mu(\phi, \psi)$ is the coherence of the two bases, and S is the number of non-zero entries on the support of the signal.

Once we have obtained the measurements m, we need to reconstruct the signal.

To recover a sparse vector, we must make sure that the vectors are not in the null space of the sensing matrix (otherwise there would be no hope of recovery). We also require that any subset of S columns taken from the measurement matrix be nearly orthogonal w.r.t sparse vectors: i.e. all pairwise distances between S-sparse vecotrs be well preserved in the measurement space.

This can be summed up in the following inequality (Restricted Isometry Property) [**Emma**]:

$$(1-\delta)\,||x||_{l_2}^2 \leq ||Ax||_{l_2}^2 \leq (1+\delta)\,||x||_{l_2}^2 \tag{4.2.1.4}$$

We are also in a position to evaluate the meaning of the constant μ in 7.1. We are considering sampling within orthonormal systems (for example, Time and Frequency):

$$A * A = nI \tag{4.2.1.5}$$

so that each row or column has l_2 norm equal to $sqrtn$. A is any matrix satisfying this property (examples include the Fourier matrix and the Dirac matrix). Thus μ must be in the set $\left[1, \sqrt{n}\right]$. μ

then, is a measure of how concentrated the rows of our measurement matrix is - i.e. how much information is spread across each vector. If $\mu = 1$ then the rows are 'flat' - and we need relatively fewer samples to reconstruct an S-sparse signal (i.e. each sample provides the same amount of information). However, if the rows contain all non-zero entries except for a single component, then $\mu^2 = n$ and we will need to observe all components to determine the non-zero one (i.e. we have no guarantees of recovery from limited samples) [**Candes2007**].

Noting that the measurements we take are projections from our orthonormal system (from example time) onto a sparsifying basis (i.e. frequency) we can see that:

$$\mu = max_{k,j} |\langle \phi_k, \psi_j \rangle| \tag{4.2.1.6}$$

So we need to choose a sensing basis, where the vectors will be 'spread out', and the degree of spreading is characterised by μ.

The correct functional to minimise would be:

$$min \|\tilde{x}\|_{l_0} \text{ subject to } y_k = \langle \phi_k, \psi x^* \rangle \ \forall k \in M \subset [1 \dots n] \tag{4.2.1.7}$$

where

$$||s||_0 = |s| \tag{4.2.1.8}$$

However, this norm is not convex and so minimising it is an NP-hard optimisation problem. As we are seeking sparse solutions the l_1-norm will suffice [**Donoho2006a**]. This is because all vectors in a random k-dimensional subspace of an n-dimensional space are approximately Gaussian (in the sense that the components are distributed according to an approximate normal distribution). Such vectors have roughly equivalent norms, and so any solution to the l_1 minimisation problem will be the same solution to the l_0 minimisation problem for sufficiently sparse signals.

Thus the role of l_1 minimisation is to decompress the data. There are many ways to perform this operation: some popular methods are basis pursuit [**Chen1998**] and Greedy approaches such as Orthogonal Matching Pursuit [**Tropp2007**].

Then f^* (the proposed reconstruction) is given by $f^* = \psi x^*$ where x^* is the solution to the convex optimisation program (n.b. $\|x\|_{l_1} := \sum_i |x_i|$):

$$min \|\tilde{x}\|_{l_1} \text{ subject to } y_k = \left\langle \phi_k, \psi x^* \right\rangle \ \forall k \in M \subset [1 \dots n] \tag{4.2.1.9}$$

In summary the **CS: Sample non-adaptively in an incoherent domain and invoke linear programming after the acquisition step to decompress the signal**

4.2.2 RIPless Theory

Short, Fat matrices

As remarked upon earlier: Compressive Sensing is equivalent to solving an under-determined linear system, with the constraint that we seek the sparsest solution. The content of the previous sections amounts to constraints on the number of rows of matrix of this linear system.

If we had an Oracle which could tell us where the non-zero components of our solution were, then we would need only as many rows of the matrix as there were non-zero components in the signal to fully specify the problem.

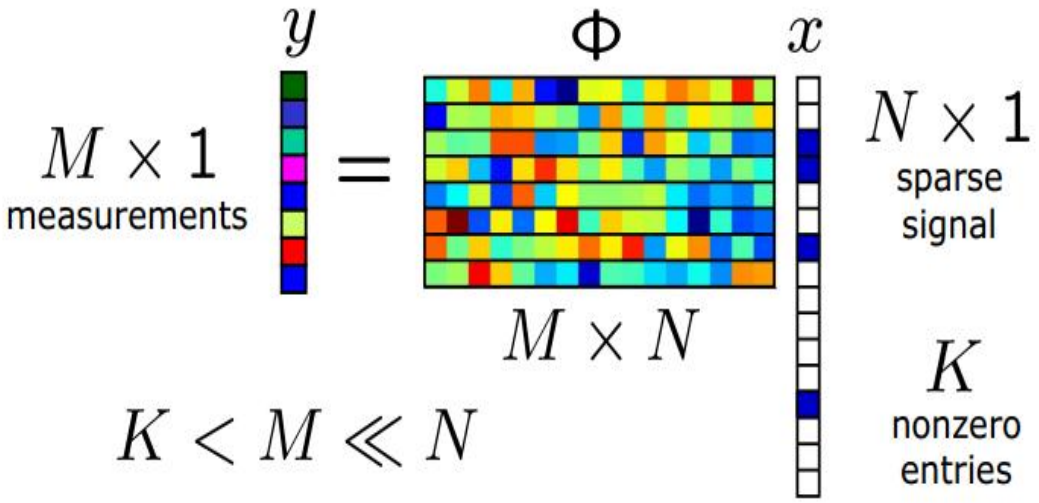

Figure 4.2: A visualisation of the Compressive Sensing problem as an under-determined system

However, such and Oracle does not exist, and so we're left with the task of constructing a matrix to recover those components. Knowing that we're looking for k-sparse solutions, we need a matrix with at least 2k columns which are linearly independent. Equivalently, all images of k-sparse vectors under the operation of the sensing matrix Φ must be distinct. From this, any k-sparse signal can be reconstructed from Ax.

To prove this assume the opposite - then there are two vectors $x, x' \in \mathbb{R}^n$ such that $Ax = Ax'$. I.e. $A(x - x') = 0$. However, $(x - x')$ is 2k-sparse and so there is a linear dependence between 2k columns of the sensing matrix A. We have a contradiction, and so 2k columns will suffice to reconstruct a k-sparse signal.

The problem with this is that we are trying to find the support of a k-sparse signal over a vector of length N, and so we would need to check all $\binom{N}{k}$ combinations of k-sparse signals which is prohibitively computationally expensive. Is there some way to gain the advantages of sparsity, without having to minimise a non-convex functional?

As it turns out, the answer is yes. If we take $m \geq C\mu^2(\phi,\psi)S\log(n)$ rows minimising the l_1 norm will find the sparsest solution. This is because the l_1 norm is an octahedron (in 3-dimensions, in higher dimensions it has an analogous spiky geometry), and solutions are more likely to intersect the norm at the points. Figure 4.3 shows this.

Bayesian Compressive Sensing

Based on the discussion above we can represent the compressive sensing measurements as:

$$\mathbf{g} = \Phi\mathbf{w} \tag{4.2.2.10}$$

where Φ is a $K \times N$ matrix which is the product of the measurement and sparse bases described earlier.

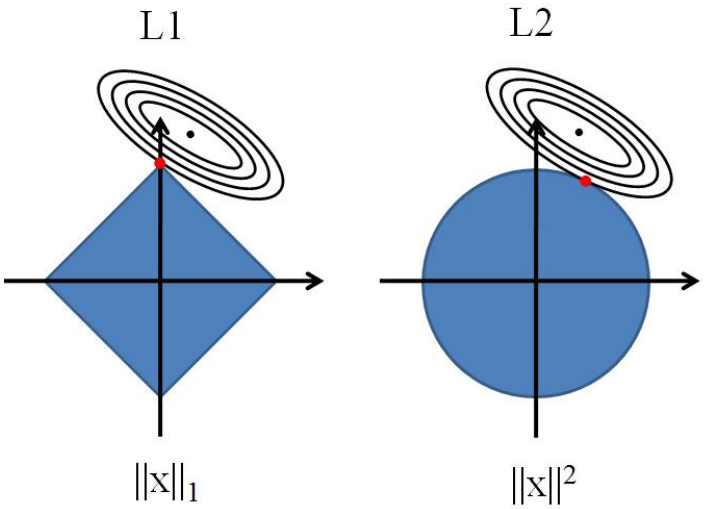

Figure 4.3: Solutions to the Compressive Sensing optimisation problem intersect the l_1 norm the points where all components (but one) of the vector are zero (i.e. it is sparsity promoting) [**Tibshirani1996**]

Note that the measurements may be noisy, with the measurement noise represented by a zero mean Gaussian distribution and unknown variance σ^2:

$$\mathbf{g} = \Phi\mathbf{w} + \mathbf{n} \tag{4.2.2.11}$$

Where **n** is the vector representing the vector of noise, and has the same support as the measurements.

Previous sections have shown how the weights w may be found through optimisation methods such as basis pursuit or greedy algorithms. Here, an alternative Bayesian model is described.

From 4.2.2 we have a Gaussian likelihood model:

$$p\left(\mathbf{g} \mid \mathbf{w},\sigma^2\right) = (2\pi\sigma^2)^{-K/2} \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{g}-\Phi\mathbf{w}\|_2^2\right) \tag{4.2.2.12}$$

The above has converted the CS problem of inverting sparse weight **w** into a linear regression problem with a constraint (prior) that **w** is sparse.

To seek the full posterior distribution over **w** and σ^2, we can chose a sparsity promoting prior. A popular sparseness prior is the Laplace density functions:

$$p\left(w \mid \lambda\right) = \left(\frac{\lambda}{2}\right)^N exp{-\lambda \sum_{i=1}^{N} |w_i|} \tag{4.2.2.13}$$

Note that the solution the convex optimisation problem 2.4.1 corresponds to a maximum *a posteriori* estimate for w using this prior. I.e this prior is equivalent to using the l_1 norm as an optimisation function (see figure 4.4 [**Tibshirani1996**]).

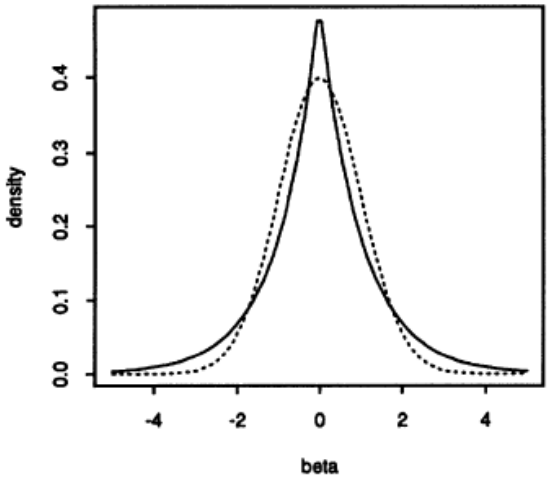

Figure 4.4: The Laplace (l_1-norm, bold line) and Normal (l_2-norm, dotted line) densities. Note that the Laplace density is sparsity promoting as it penalises solutions away from zero more than the Gaussian density. [**Tibshirani1996**]

The full posterior distribution on w and σ^2 may be realised, by using a hierarchical prior instead. To do this, define a zero-mean Gaussian prior on each element of w:

$$p(w \mid a) = \prod_{i=1}^{N} \mathbb{N}\left(w_i \mid 0, \alpha_i^{-}1\right) \tag{4.2.2.14}$$

where α is the precision of the distribution. A gamma prior is then imposed on α:

$$p(\alpha \mid a, b) = \prod_{i=1}^{N} \Gamma(\alpha_i \mid a, b) \tag{4.2.2.15}$$

The overall prior is found by marginalising over the hyperparameters:

$$p(w \mid a, b) = \prod_{i=1}^{N} \int_0^{\infty} \mathbb{N}\left(w_i \mid 0, \alpha_i^{-}1\right) \Gamma(\alpha_i \mid a, b) \tag{4.2.2.16}$$

This integral can be done analytically and is a Student-t distribution. Choosing the parameters a, b appropriately we can make the Student-t distribution peak strongly around $w_i = 0$ i.e. sparsifying. This process can be repeated for the noise variance σ^2. The hierarchical model for this process is shown in 8.1. This model, and other CS models which not necessarily have closed form solutions, can be solved via belief-propagation [**Baron2010**]

4.2.3 Sub-Nyquist Sampling techniques

This section presents some work on sampling methods for wide-band spectrum sensing

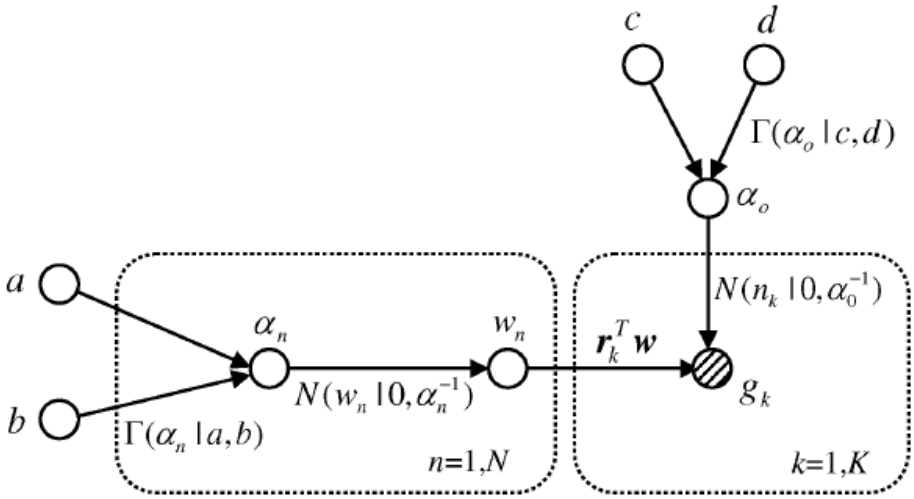

Figure 4.5: The hierarchical model for the Bayesian CS formulation [**Ji2008**]

Wideband Modulated Converter

The sampling scheme proposed in [**Mishali2010**] is capable of sampling wideband signals at rates below those predicted by Shannon-Nyquist sampling theory.

It works by mixing the incoming analogue signal $x(t)$ with a mixing function $p_i(t)$ aliasing the spectrum. $x(t)$ is assumed to be bandlimited and composed of up to N_sig uncorrelated transmissions (i.e. possible narrowband channels).

This process is repeated in parallel over M channels (unrelated to N_sig so that each band in x appears in baseband. The mixing functions are required to be periodic, with period T_p. Since p_i is periodic it has Fourier expansion:

$$p_i(t) = \sum_{l=-\infty}^{\infty} c_{il} expjlt\frac{2\pi}{T_p} \tag{4.2.3.17}$$

The c_{il} are the Fourier coefficients of the expansion and are defined in the standard manner. The result of the mixing procedure in channel i is therefore xp_i, with Fourier transform:

$$X_i(f) = \int_{-\infty}^{\infty} x(t)\, p_i(t)\, dt \tag{4.2.3.18}$$

$$= \sum_{l=-\infty}^{\infty} c_{il} X(f - l f_p) \tag{4.2.3.19}$$

(insert the Fourier series for p_i, then exchange the sum and integral). The output of this mixing process then, is a linear combination of shifted copies of $X(f)$, with at most $\lceil f_N YQ/f_p \rceil$ terms since $X(f)$ is zero outside it's support (we have assumed this Nyquist frequency exists, even though we never sample at that rate).

Once the mixing process has been completed the signal in each channel is low-pass filtered and sampled at a rate $f_s \geq f_p$. In the frequency domain this is a ideal rectangle function, so the output of a single channel is:

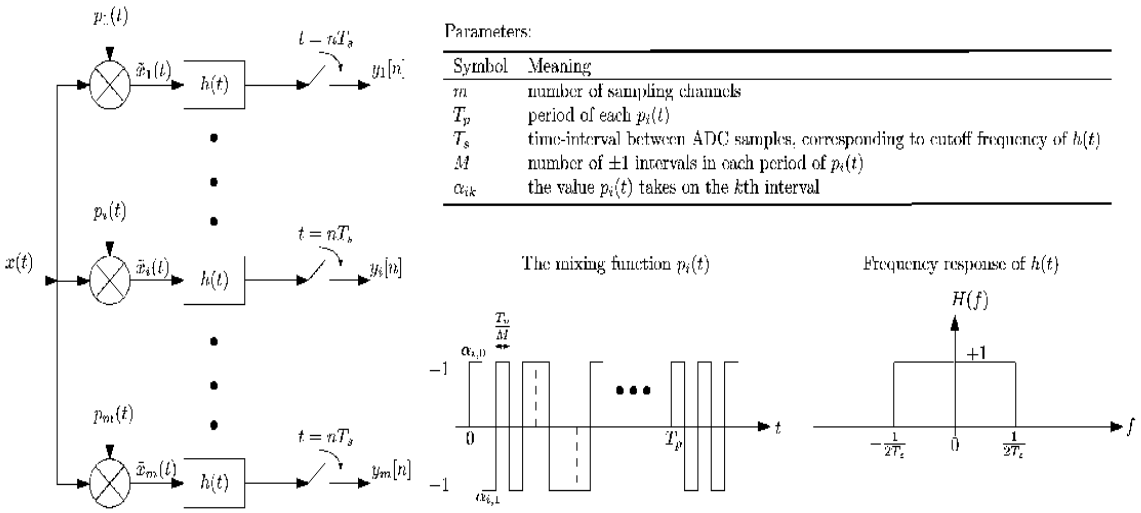

Figure 4.6: The operation of the Modulated Wideband Converter [**mishali2010theory**]

(4.2.3.20) $$Y_i\left(e^{j2\pi f T_s}\right) = \sum_{l=-L_0}^{+L_0}$$

since frequencies outside of $[-f_2/2, f_s/2]$ will filtered out. L_0 is the smallest integer number of non-zero contributions in $X\left(f\right)$ over $[-f_2/2, f_s/2]$ - at most $\lceil f_N Y Q/f_p \rceil$ if we choose $f_s = f_p$. These relations can be written in matrix form as:

(4.2.3.21) $$\mathbf{y} = \_\mathbf{A}\mathbf{x}$$

where **y** contains the output of the WMC process, **_A** contains the Fourier coefficients of the mixing functions, and **x** is the vector of unknown samples of $x(t)$.

4.3 Results and Simulations

To compare the efficacy of Group Testing and Compressive Sensing, Hwang's algorithm and the algorithm presented in [**Aldrouobi**] were simulated for a problem size of N=1024 and K=10. The problem was simulated 100 times and the cumulative distribution found - i.e. after how many tests or measurements were the respective problems solved? This allows the number of tests required by Group Testing to be compared to the number of measurements in Compressive Sensing. Figure 4.7 shows the results:

Note that both the algorithms meet their respective asymptotic bounds ($\log_2 \binom{N}{K}$ in the case of GT and $k \log N$ for CS). The main point of interest is that Group Testing requires roughly $\frac{2}{3}$ of the tests required by Compressive Sensing. This is encouraging: despite there being 'less' information - in the sense that the result is a binary number as opposed to a real one - GT outperforms CS. Intuitively, one would conjecture the opposite - more information should allow you to locate the non-zero components faster. This justifies our interest in the problem, as the performance increase is substantial.

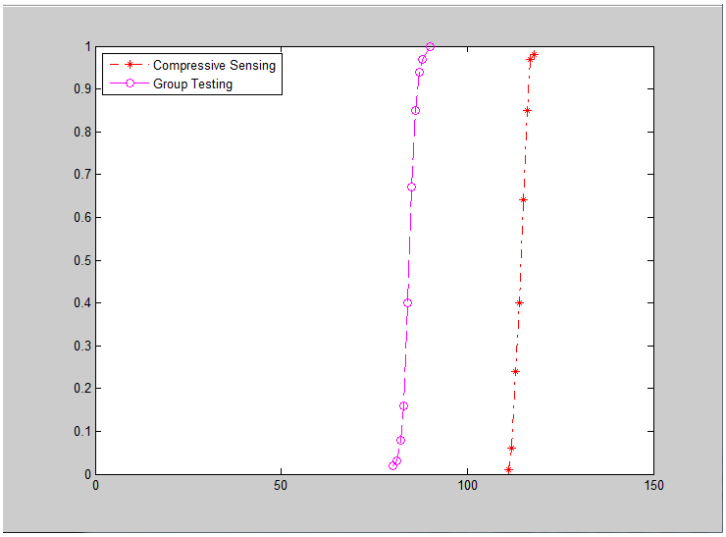

Figure 4.7: Group Testing vs Compressive Sensing

CHAPTER 5

OPTIMISATION ON GRAPHS

The contributions of this report are that we propose a distributed model and solver pair which obviates the need for a Fusion Centre (centralised node) as in [**Zhang2011b**]) to do any data processing. That is the solution is found in a distributed manner, by local computations and communications in with one-hop neighbours. This can be applied to other models which previously required central processing.

Moreover, our algorithm is simple to understand (it is an extension of the multi-block ADMM [**mota2013d**]) and can be applied to other composite optimisation problems.

We also give new proofs of ideas found in [**mota2013d**].

The structure of the report is as follows: in section 6.3 we introduce the sensing model, in section 6.4 we describe the distributed reconstruction algorithm [**mota2013d**], and finally in section 6.5 we show some results of the reconstruction quality of this model.

5.1 ADMM

The alternating direction method of multipliers (ADMM) is a convex optimisation algorithm which splits large, and typically non-smooth, problems into smaller pieces which are easier to handle. At heart, the algorithm involves finding a zero of a convex objective function, by evaluating proximal operators [**parikh2014proximal**].

ADMM has a long history, first being investigated by Rockarfellar [**rockafellar1976monotone**], in the context of minimisation in Hiilbert spaces, and Douglas and Rachford [**douglas1956numerical**] for solving the heat equation. Subsequently the theory was generalised and extended by Eckstein and Bersekas [**eckstein1992douglas**].

ADMM has found applications in Sparse Coding [**Bristow2014**], Compressive Radar Imaging [**heredia2015consensus**], Medical Imaging [**sawatzky2014proximal**], and Optimal Control [**o2013splitting**]. This is because many optimisation objective functions can be formulated as a sum of simpler convex functions, and casting the problem in an alternating minimisation framework is straightforward for practitioners.

ADMM also has explicit (linear) convergence rates, (see [**Shi2013**], and [**nishihara2015general**]), as well as well understood methodologies for tuning and acceleration [**ghadimi2015optimal**] and

[**goldstein2014fast**]. However, care must be taken when the problem is posed as the sum of more than two convex functions - the direct extension is not necessarily convergent [**chen2016direct**].

Given a set of measurements of the form

$$y = Ax + n \tag{5.1.0.1}$$

where $x \in \mathbb{R}^n$ is an s-sparse vector we wish to recover, $y \in \mathbb{R}^m$ is a set of noisy measurements, $A \in \mathbb{R}^{m \times n}$ is a design or measurement matrix s.t. x is not in the null-space of A, and $z \in \mathbb{R}^m$ is AGWN. The signal x can be recovered by algorithms minimising the objective function:

$$L = \frac{1}{2} \left|\left| Ax - y \right|\right|_2^2 + \lambda \left|\left| x \right|\right|_1 \tag{5.1.0.2}$$

where λ is a parameter which trades off the reconstruction accuracy and sparsity of x: larger λ means sparser x.

We also consider the problem:

$$L = \frac{1}{2} \left|\left| Ax - y \right|\right|_2^2 + \lambda \left|\left| x \right|\right|_0 \tag{5.1.0.3}$$

where $||t||_0 = \{|t| \mid t_i \neq 0\}$.

One such algorithm is the alternating direction method of multipliers [**Boyd2010a**], (ADMM). This algorithm solves (constrained) problems of the form

$$\begin{aligned} &\arg\min_x f(x) + g(z) \\ &\text{s.t } Ux + Vz = c \end{aligned} \tag{5.1.0.4}$$

where f and g are assumed to be convex function with range in $\mathbb{R}$, $U \in \mathbb{R}^{p \times n}$ and $V \in \mathbb{R}^{p \times m}$ are matrices (not assumed to have full rank), and $c \in \mathbb{R}^p$.

ADMM consists of iteratively minimising the augmented Lagrangian

$$L_p(x, z, \eta) = f(x) + g(z) + \eta^T (Ux + Vz - c) + \frac{\rho}{2} \|Ux + Vz - c\|_2^2$$

(η is a Lagrange multiplier), and ρ is a parameter we can choose to make $g(z)$ smooth [**nesterov2005smooth**], with the following iterations:

$$x^{k+1} := \arg\min_x L_\rho \left(x, z^k, \eta^k\right) \tag{5.1.0.5}$$

$$z^{k+1} := \arg\min_z L_\rho \left(x^{k+1}, z, \eta^k\right) \tag{5.1.0.6}$$

$$\eta^{k+1} := \eta^k + \rho \left(Ux^{k+1} + Vz^{k+1} - c\right) \tag{5.1.0.7}$$

The alternating minimisation works because of the decomposability of the objective function: the x minimisation step is independent of the z minimisation step and vice versa.

We illustrate an example, relevant to the type of problems encountered in signal processing.

ADMM can be formulated as an iterative MAP estimation procedure for the problem (5.1.0.2). We can write (5.1.0.2) in constrained form as:

$$\frac{1}{2}\|Ax-b\|_2^2+\lambda\|z\|_1 \tag{5.1.0.8}$$

$$\text{s.t } z=x \tag{5.1.0.9}$$

i.e this is of the form (5.1.0.4) with $f(x)=\|Ax-y\|_2^2$, $g(z)=\lambda\|z\|_1$, $U=I$, $V=-I$, and $c=0$.

The associated (augmented) Lagrangian is:

$$L_\rho=\frac{1}{2}\|Ax-b\|_2^2+\lambda\|z\|_1+\eta(x-z)+\frac{\rho}{2}\|x-z\|^2 \tag{5.1.0.10}$$

The ADMM iterations for LASSO, which can be found by alternately differentiating (5.1.0.10) with respect to x,z and η, are (in closed form):

$$x^{k+1}:=\left(A^TA+\rho I\right)^{-1}\left(A^Ty+\rho\left(z^k-\eta^k/\rho\right)\right) \tag{5.1.0.11}$$

$$z^{k+1}:=S_{\lambda/\rho}\left(x^{k+1}+\eta^k/\rho\right) \tag{5.1.0.12}$$

$$\eta^{k+1}:=\eta^k+\rho\left(x^{k+1}-z^{k+1}\right) \tag{5.1.0.13}$$

where $S_{\lambda/\rho}(\circ)$ is the soft thresholding operator: $S_\gamma(x)_i=\text{sign}(x_i)\left(|x_i|-\gamma\right)^+$.

These can be found differentiating (5.1.0.10) with respect to x and z as follows:

$$\frac{\partial L}{\partial x}=-A^T\left(y-Ax\right)+\rho(x-z)+\eta$$

as

$$\frac{\partial}{\partial x}||F(x)||_2^2=2\left(\frac{\partial}{\partial x}F(x)\right)F(x) \tag{5.1.0.14}$$

by the chain rule, and $\partial/\partial x(Ax)=-A^T$ (see the Matrix Cookbook) as differentiation exchanges a linear operator with its adjoint.

Setting (5.1.0.14) to zero and collecting like terms:

$$\left(A^TA+\rho I\right)x=A^Ty+\rho z-\eta \tag{5.1.0.15}$$

so we find the optimal x is:

$$x=\left(A^TA+\rho I\right)^{-1}\left(A^Ty+\rho\left(z-\eta/\rho\right)\right) \tag{5.1.0.16}$$

note that this estimator is a weighted average of the ordinary least squares estimate (A^Ty) and a Gaussian prior. This is to be expected, as the minimisation problem w.r.t x is an l_2-regularised MAP problem.

for $z > 0$

$$\frac{\partial L}{\partial z} = \lambda + \rho(x - z) - \eta \tag{5.1.0.17}$$

from which we obtain:

$$z = x + \frac{1}{\rho}(\eta - \lambda)$$

since $z > 0$ then $x + \frac{1}{\rho}(\eta - \lambda I) > 0$ when $x + \frac{\eta}{\rho} > \frac{\lambda}{\rho}$.
Similarly for $z < 0$:

$$\frac{\partial L}{\partial z} = -\lambda + \rho(x - z) \tag{5.1.0.18}$$

setting (5.1.0.18) to zero we obtain:

$$z = x + \frac{1}{\rho}(\eta + \lambda)$$

since $z < 0$ then $x + \frac{1}{\rho}(\eta + \lambda) < 0$ when $x + \frac{\eta}{\rho} < -\frac{\lambda}{\rho}$.
at $z = 0$ we find:

$$-\frac{\lambda}{\rho} \leq x + \frac{\eta}{\rho} \leq \frac{\lambda}{\rho}$$

i.e.

$$| x + \frac{\eta}{\rho} | \leq \frac{\lambda}{\rho} \tag{5.1.0.19}$$

combining (5.1.0.18), (5.1.0.17), (5.1.0.19) together we find the optimal z is:

$$z = \text{sign}(x + \frac{\eta}{\rho}) \max\left(| x + \frac{\eta}{\rho} | - \frac{\lambda}{\rho}, 0 \right) \tag{5.1.0.20}$$

Together (5.1.0.16), (5.1.0.20) and the third step of (5.1.0.13) constitute the steps of the ADMM algorithm.

This algorithm has a nice statistical interpretation: it iteratively performs ridge regression, followed by shrinkage towards zero. This is the MAP estimate for x under a Laplace prior.

The soft-thresholding operator can be derived by considering the MAP estimate of the following model:

$$y = x + w \tag{5.1.0.21}$$

where x is some (sparse) signal, and w is additive white Gaussian noise. We seek

$$\hat{x} = \arg\max_x \mathbb{P}_{x|y}\big(x|y\big) \tag{5.1.0.22}$$

This can be recast in the following form by using Bayes rule, noting that the denominator is independent of x and taking logarithms:

$$\hat{x} = \arg\max_x \left[\log \mathbb{P}_w\left(y-x\right) + \log \mathbb{P}(x)\right] \tag{5.1.0.23}$$

The term $\mathbb{P}_n\left(y-x\right)$ arises because we are considering $x+w$ with w zero mean Gaussian, with variance σ_n^2. So, the conditional distribution of y (given x) will be a Gaussian centred at x.

We will take $\mathbb{P}(x)$ to be a Laplacian distribution:

$$\mathbb{P}(x) = \frac{1}{\sqrt{2}\sigma} \exp -\frac{\sqrt{2}}{\sigma}|x| \tag{5.1.0.24}$$

Note that $f(x) = \log \mathbb{P}_x(x) \ -\frac{\sqrt{2}}{\sigma}|x|$, and so by differentiating $f'(x) = -\frac{\sqrt{2}}{\sigma}\text{sign}(x)$

Taking the maximum of 5.1.0.23 we obtain:

$$\frac{y-\hat{x}}{\sigma_n^2} - \frac{\sqrt{2}}{\sigma} sign(x) = 0 \tag{5.1.0.25}$$

Which leads the soft thresholding operation defined earlier, with $\gamma = \frac{\sqrt{2}\sigma_n^2}{\sigma}$ as (via rearrangement):

$$y = \hat{x} + \frac{\sqrt{2}\sigma_n^2}{\sigma}\text{sign}(x)$$

or

$$\hat{x}\left(y\right) = \text{sign}(y)\left(y - \frac{\sqrt{2}\sigma_n^2}{\sigma}\right)_+$$

i.e $S_\gamma(y)$.

5.1.1 The Proximity Operator

The Proximity Operator for a closed, convex, and proper function f (the set of all such functions will be denoted Γ in a Hilbert space $\mathscr{H}$ is defined as [**moreau1965proximite**]:

Definition 5.1.1 (Proximity Operator)**.**

$$\text{Prox}_f\left(y\right) := \arg\min_{y\in\mathscr{H}} f\left(y\right) + \frac{1}{2}\left|\left|y-x\right|\right|^2 \tag{5.1.1.26}$$

Intuitively the Proximity Operator approximates a point x by another point y, that is close in the mean-square sense under the penalty f.

The Prox$(\circ)$ operator exists for closed and convex f as $\left(y\right) + \frac{1}{2}\left|\left|y-x\right|\right|^2$ is closed with compact level sets and is unique as $\left(y\right) + \frac{1}{2}\left|\left|y-x\right|\right|^2$ is strictly convex.

The corresponding Moreau envelope is defined as

Definition 5.1.2 (Moreau Envelope)**.**

$$\mathrm{M}_f\left(y\right) := \min_{y \in \mathscr{H}} f\left(y\right) + \frac{1}{2}\left|\left|y - x\right|\right|^2 \tag{5.1.1.27}$$

The Moreau envelope is a strict generalisation of the squared distance function. M_f is real valued - even when f takes the value ∞, whilst Prox_f is $\mathscr{H}$-valued.

Properties

Theorem 5.1.3 (Moreau '65)**.** *Let $f \in \Gamma$ and f^* be its Fenchel conjugate. Then the following are equivalent:*

- $z = x + y, y \in \partial f\left(x\right)$
- $x = \mathrm{Prox}_f\left(z\right), y = \mathrm{Prox}_{f^*}\left(z\right)$

Theorem 5.1.4 ([**moreau1965proximite**])**.** *Let $f \in \Gamma$. Then for all $z \in \mathscr{H}$*

- $\mathrm{Prox}_f\left(z\right) + \mathrm{Prox}_{f^*}\left(z\right) = z$
- $\mathrm{M}_f\left(z\right) + \mathrm{M}_{f^*}\left(z\right) = \frac{1}{2}\left|\left|z\right|\right|^2$

Theorem 5.1.5 ([**moreau1965proximite**])**.** *The Moreau envelope is (Frechet) differentiable, with*

$$\nabla \mathrm{M}_f = Id - \mathrm{Prox}_f = \mathrm{Prox}_{f^*} \tag{5.1.1.28}$$

Theorem 5.1.6 ([**moreau1965proximite**])**.** $\mathrm{Prox}_f : (\mathscr{H}, ||\circ||) \leftarrow (\mathscr{H}, ||\circ||)$ *is 1-Lipchitz continuous.*

Motivation

We are solving problems of the following form:

$$\min_{x \in \mathscr{H}} f\left(x\right) + g\left(z\right) \tag{5.1.1.29}$$

$$\text{s.t } x - z = 0 \tag{5.1.1.30}$$

with $f, g \in \Gamma$. To solve this problem we form the augmented Lagrangian:

$$L_p\left(x, z, \eta\right) = f\left(x\right) + g\left(z\right) + \eta^T\left(Ux + Vz - c\right) + \frac{\rho}{2}\|Ux + Vz - c\|_2^2$$

and then performing the following iterative minimisation:

$$x^{k+1} := \arg\min_x L_\rho\left(x, z^k, \eta^k\right) \tag{5.1.1.31}$$

$$z^{k+1} := \arg\min_z L_\rho\left(x^{k+1}, z, \eta^k\right) \tag{5.1.1.32}$$

$$\eta^{k+1} := \eta^k + \rho\left(x^{k+1} + z^{k+1}\right) \tag{5.1.1.33}$$

i.e.

$$x^{k+1} := \arg\min_x \left(f(x) + \eta^{kT} x + \frac{\rho}{2} \left|\left| x - z^k \right|\right|^2 \right) \tag{5.1.1.34}$$

$$z^{k+1} := \arg\min_z \left(g(z) - \eta^{kT} z + \frac{\rho}{2} \left|\left| x^{k+1} - z \right|\right|^2 \right) \tag{5.1.1.35}$$

$$\eta^{k+1} := \eta^k + \rho \left(x^{k+1} + z^{k+1} \right) \tag{5.1.1.36}$$

pulling the linear terms into the quadratic ones we get:

$$x^{k+1} := \arg\min_x \left(f(x) + \frac{\rho}{2} \left|\left| x - z^k + (1/\rho)\eta^k \right|\right|^2 \right) \tag{5.1.1.37}$$

$$z^{k+1} := \arg\min_z \left(g(z) + \frac{\rho}{2} \left|\left| x^{k+1} - z - (1/\rho)\eta^k \right|\right|^2 \right) \tag{5.1.1.38}$$

$$\eta^{k+1} := \eta^k + \rho \left(x^{k+1} + z^{k+1} \right) \tag{5.1.1.39}$$

i.e.

$$x^{k+1} := \text{Prox}_f \left(z^k - u^k \right) \tag{5.1.1.40}$$

$$z^{k+1} := \text{Prox}_f \left(x^{k+1} + u^k \right) \tag{5.1.1.41}$$

$$u^{k+1} := u^k + \left(x^{k+1} + z^{k+1} \right) \tag{5.1.1.42}$$

with $u^k = (1/\rho)\eta^k$.

The motivation for the Proximal operator should now be clear: to perform the minimisation we simply calculate the proximal operator of each of the functions at each step. For many functions found in Statistics (e.g. the l_p norms, this can be found in closed form, and so ADMM presents a particularly attractive method for finding MAP solutions to regularised statistical problems.

Examples

Example 5.1.7 (Indicator)**.** *From the definition*

$$\text{Prox}_I(x) := \arg\min_y I_C(y) + \frac{1}{2} \left|\left| y - x \right|\right|^2 \tag{5.1.1.43}$$

$$= \arg\min_{y \in C} \frac{1}{2} \left|\left| y - x \right|\right|^2 \tag{5.1.1.44}$$

$$= P_C(x) \tag{5.1.1.45}$$

where $I_C(y)$ is the indicator of some set C and P_C is the projection operator onto that set.

Example 5.1.8 (l_2 norm)**.** *For $f(y) = \frac{\mu}{2} \left|\left| y \right|\right|^2$ the* Prox *operator is:*

$$\text{Prox}_f(x) := \arg\min_y \frac{\mu}{2} \left|\left| y \right|\right|^2 + \frac{1}{2} \left|\left| y - x \right|\right|^2 \tag{5.1.1.46}$$

$$= \frac{1}{1+\mu} x \tag{5.1.1.47}$$

Example 5.1.9 (l_1 norm)**.** $f = \|x\|_1$

$$\text{Prox}_f(x) := \text{sign}(x_i)\left(|x_i| - \gamma\right)^+ = S_\gamma(x)_i \tag{5.1.1.48}$$

Example 5.1.10 (Elastic Net)**.** *Consider*

$$f(x) = \lambda\|x\|_1 + \mu||x|| \tag{5.1.1.49}$$

$$\text{Prox}_f(x) := \frac{\lambda}{1+\mu} S_\gamma(x)_i \tag{5.1.1.50}$$

Example 5.1.11 (Fused Lasso)**.** *Consider*

$$f(x) = \|x\|_1 + \sum_{i=1}^{d-1}(x_i - x_{i-1}) \tag{5.1.1.51}$$

i.e the sum of the l_1 and TV norms

$$\text{Prox}_f(x) := \text{Prox}_{l_1} \circ \text{Prox}_{TV} = S_\gamma(\text{Prox}_{TV})_i \tag{5.1.1.52}$$

Example 5.1.12 (Consensus)**.** *Suppose we want to solve a problem such as:*

$$\underset{x}{\text{minimize}} \quad \sum_i f_i(x)$$

this could arise in statistical computing where f_i would be the loss function for the i^th block of training data. We can write the problem for distributed optimisation as:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \sum_i f_i(x_i) \\ \text{subject to} \quad & x_i - z = 0 \end{aligned}$$

where x_i are local variables (for example local to each node in a spectrum sensing) and $x_i - z = 0$ are the consensus constraints. Consensus and regularisation can be achieved by adding a regularisation term $g(z)$ - for example $g(z) = \lambda||x||_1$ corresponds to the LASSO, and the f_i would be $f_i = ||A_i x_i - b||_2^2$.

As per the previous sections, we form the Augmented Lagrangian:

$$L_\rho(x, y) = \sum_i^n \left(f_i(x_i) + y_i^T(x_i - z) + \frac{\rho}{2}\|x_i - z\|_2^2 \right) \tag{5.1.1.53}$$

The ADMM iterations for this Lagrangian are:

$$x_i^{k+1} := \arg\min x_i \left(f_i(x_i) + y_i^{kT}(x_i - z) + \frac{\rho}{2}\|x_i - z\|_2^2 \right) \tag{5.1.1.54}$$

$$z^{k+1} := \frac{1}{n}\sum_i^n \left(x_i^{k+1} + (1\rho)\, y_i^k \right) \tag{5.1.1.55}$$

$$y_i^{k+1} := y_i^k + \rho\left(x_i^{k+1} - z^{k+1} \right) \tag{5.1.1.56}$$

The z^{k+1} iteration is analytic as we're minimising the squared norm of $x_i - z$ - so we average. With $\|x\|_1$ regularisation we perform soft-thresholding after the z update.

At each iteration the sum of the dual variables y_i is zero, so the algorithm can be simplified to:

$$x_i^{k+1} := \arg\min x_i \left(f_i(x_i) + y_i^{kT}\left(x_i - \bar{x}^k \right) + \frac{\rho}{2}\|x_i - \bar{x}^k\|_2^2 \right) \tag{5.1.1.57}$$

$$y_i^{k+1} := y_i^k + \rho\left(x_i^{k+1} - z^{k+1} \right) \tag{5.1.1.58}$$

where

$$\bar{x}^k = \frac{1}{n}\sum_i^n x_i^k \tag{5.1.1.59}$$

This algorithm can be summarised as follows: in each iteration

- *gather x^k and average to get $\bar{x}^k$*
- *scatter the average to nodes*
- *update y_i^k locally*
- *update x_i locally*

Each agent is minimising it's own function, plus a quadratic term (the squared norm) which penalises the agent from moving too far from the previous average.

Note that the 'gather' stage doesn't require a central processor - this can be done in a distributed manner also.

5.1.2 Statistical Interpretation

At each step k of the algorithm each agent is minimising it's own loss function, plus a quadratic. This has a simple interpretation: we're doing MAP estimation under the prior $\mathcal{N}\left(\bar{x}^k + (1\rho)\, y_i^k, \rho I\right)$. I.e. the prior mean is the previous iteration's consensus shifted by node i disagreeing with the previous consensus.

5.1.3 Acceleration

5.2 Constrained Optimisation on Graphs

We model the network of sensors as an undirected graph $G = (V, E)$, where $V = \{1 \dots J\}$ is the set of vertices, and $E = V \times V$ is the set of edges. An edge between nodes i and j implies that the two sensors can communicate. The set of nodes that node i can communicate with is written $\mathcal{N}_i$ and the degree of node i is $D_i = |\mathcal{N}_i|$.

Individually nodes make the following measurements (as discussed in section 6.3):

$$\mathbf{y}_p = \mathbf{A}_p \mathbf{x} + \mathbf{n}_p \tag{5.2.0.60}$$

where $\mathbf{A}_p$ is the p^{th} row of the sensing matrix from (6.3.0.10), and the system (6.3.0.10) is formed by concatenating the individual nodes' measurements together.

We assume that a proper colouring of the graph is available: that is, each node is assigned a number from a set $C = \{1 \dots c\}$, and no node shares a colour with any neighbour. This is so that nodes may communicate in colour order, as opposed to communicating individually thus reducing the total number of communication rounds required.

To find the $\mathbf{x}$ we are seeking (the solution to the linear system, 6.3.0.10), to each node we give a copy of $\mathbf{x}, \mathbf{x}_p$ and we constrain the copies to be identical across all edges in the network. Each node, thus has a separate optimisation to solve, subject to the constraint that it is consistent with its neighbours.

The problem then is to solve:

$$\begin{aligned} \underset{\bar{x}}{\operatorname{argmin}} \sum_{c=1}^{C} \sum_{j \in c} f\left(x_j\right) + \frac{\lambda}{J} g\left(x_j\right) \\ \text{and } x_i = x_j \text{ if } \{i, j\} \in E \\ \text{and } x_i = z_i \ \forall i \in \{1, \dots, C\} \end{aligned} \tag{5.2.0.61}$$

with a particular special case being:

$$\begin{aligned} \underset{\bar{x}}{\operatorname{argmin}} \sum_{c=1}^{C} \sum_{j \in c} \| A_j x_j - y_j \|_2^2 + \frac{\lambda}{J} \| z \|_1 \\ \text{and } x_i = x_j \text{ if } \{i, j\} \in E \\ \text{and } x_i = z_i \ \forall i \in \{1, \dots, C\} \end{aligned} \tag{5.2.0.62}$$

i.e. $f = ||x||_2^2$ and $g = ||x||_1$.

That is, at each node we minimise a Lasso functional constrained to be consistent across edges but that is separable in the l_2 and l_1 norms.

We can write the global optimisation variable as $\bar{x}$, which collects together C copies of a $n \times 1$ vector $\mathbf{x}$:

Definition 3. *We define vectors x_c, where $c = 1, \dots, C$ and write the vector of length nJ:*

$$\bar{x} = \sum_{c=1}^{C} w_c \otimes x_c = \left[x_{c(1)}^T, \dots, x_{c(J)}^T \right]^T \tag{5.2.0.63}$$

where $w_{c(i)} = \mathbb{I}(c(i) = c)$, $\mathbb{I}$ is the indicator function, and we have written $c(i)$ for the colour of the ith node.

These constraints can be written more compactly by introducing the node-arc incidence matrix B: a V by E matrix where each column is associated with an edge $(i, j) \in E$ and has 1 and -1 in the ith and jth entry respectively. Figures (5.1) and (5.2) show examples of a network and it's associated incidence matrix.

The constraint $x_i = x_j$ if $\{i, j\} \in E$ can now be written

$$\sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_c = 0 \tag{5.2.0.64}$$

note that $\left(B^T \otimes I_n\right) \in \mathbb{R}^{nE \times nJ}$. Together (6.4.0.13) and (6.4.0.15), suggests that the problem (6.4.0.14) can be re-written as:

$$\begin{aligned} \arg\min_{\bar{x}} & \sum_{c=1}^{C} \sum_{j \in C_c} f\left(x_j\right) + \frac{\lambda}{J} g\left(z_j\right) \\ & \text{s.t.} \sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_c = 0 \\ & \text{and } \bar{x}_c - \bar{z}_c = 0 \end{aligned} \tag{5.2.0.65}$$

where $\beta = \frac{\lambda}{J}$.

The global Augmented Lagrangian [**Boyd2010a**] for the problem (6.4.0.16) can be written down as:

$$\begin{aligned} L_\rho = \sum_{c=1}^{C} (& \sum_{j \in c} f\left(x_j\right) + \frac{\lambda}{J} g\left(z_j\right) + \\ & + \theta^T \left(\bar{x}_j - \bar{z}_j\right) + \frac{\rho}{2} \left|\left|\bar{x}_j - \bar{z}_j\right|\right|_2^2) + \\ & + \eta^T \left(B_c^T \otimes I_n\right) \bar{x}_c + \frac{\rho}{2} \left|\left| \sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_c \right|\right|_2^2 \end{aligned} \tag{5.2.0.66}$$

This is, superficially, similar to the Augmented Lagrangian for the Lasso problem [**Boyd2010a**][Section 6.4]. That is, the terms indexed by j are a straightforward Lasso problem, constrained by edge-wise variables (indexed by c) forcing consistency across the network. However, the problem (as currently written) is not separable across the edges of the network as the final and penultimate term represent the constraint that the nodes agree on their estimates across edges.

To make it possible that 6.4.0.17 can be posed as a constrained optimisation problem at each node, we introduce the following variable (so that the the final term of 6.4.0.17 is separable across edges of the graph):

Definition 4.

$$\begin{aligned} u &:= \left(B^T \otimes I_n\right) \bar{x} \\ &= \left(B^T \otimes I_n\right) \sum_{c=1}^{C} w_c \otimes x_c \\ &= \sum_{c=1}^{C} B_c^T \otimes x_c \end{aligned}$$

where we have used the definition (6.4.0.13) *in the second line, and the property of Kronecker products* $(A \otimes C)(B \otimes D) = (AB \otimes CD)$ *between the second and third lines, and we write* $B_c = w_c^T B$.

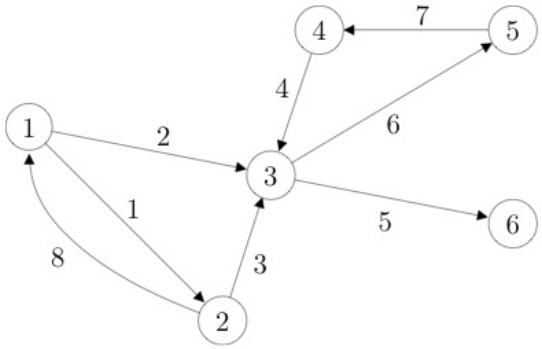

Figure 5.1: An example of a network

$$\mathsf{B} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & -1 & -1 & -1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix}.$$

Figure 5.2: The incidence matrix associated with Figure (5.1)

The terms $\| \sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_c \|^2$ and $\eta^T \left(B_c^T \otimes I_n\right) \bar{x}_c$ of (6.4.0.17), can be decomposed across edges, using the following lemma:

Lemma 5.2.1 (Edge Decomposition)**.**

$$\left|\left| \sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_c \right|\right|^2 = \sum_{j \in C_1} \left(D_j \left|\left| x_j \right|\right|_2^2 - \sum_{k \in N_j} x_j^T x^k \right) \tag{5.2.0.67}$$

and

$$\eta^T \sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_1 = \sum_{l \in C_c} \sum_{m \in N_l} sign(m-l)\, \eta_{ml}^T x_l \tag{5.2.0.68}$$

where η is decomposed edge-wise: $\eta = (\ldots, \eta_{ij}, \ldots)$, such that $\eta_{i,j} = \eta_{j,i}$, and is associated with the constraint $x_i = x_j$.

Proof.

$$\begin{aligned} u^T u &= \sum_{c_1=1}^{C} \sum_{c_2=1}^{C} \left(B_{c_1} \otimes x_{c_1}^T\right)\left(B_{c_2}^T \otimes x_c\right) \\ &= \sum_{c_1,c_2} B_{c_1} B_{c_2}^T \otimes x_{c_1}^T x_{c_2} \end{aligned}$$

BB^T is a $J \times J$ matrix, with the degree of the nodes on the main diagonal and -1 in position (i, j) if nodes i and j are neighbours (i.e BB^T is the graph Laplacian). Hence, since we can write $B_{c_1} B_{c_2}^T = w_{c_1}^T BB^T w_{c_2}$, the trace of $B_{c_1} B_{c_1}^T$ is simply the sum of the degrees of nodes with colour 1.

For $c_1 \neq c_2$, $B_{c_1} B_{c_2}^T$ corresponds to an off diagonal block of the graph Laplacian, and so counts how many neighbours each node with colour 1 has.

Finally, note that $\eta \in \mathbb{R}^{nE}$ and can be written:

$$\eta = \sum_{c=1}^{C} w_c \otimes \eta_c \tag{5.2.0.69}$$

where η_c is the vector of Lagrange multipliers associated across edges from colour c. Now

$$\eta^T u = \sum_{c_1=1}^{C} \sum_{c_2=1}^{C} w_{c_1} B w_{c_2} \otimes \eta_{c_1}^T x_c$$

by the properties of Kronecker products, and the definition of B_c. For $c_1 = c_2$, $\eta^T u$ is zero, as there are no edges between nodes of the same colour b definition. For $c_1 \neq c_2$, $\eta^T u$ counts the edges from c_1 to c_2, with the consideration that the edges from c_2 to c_1 are counted with opposite parity. □

Adding together this with the lemma, lets us write (6.4.0.17) as:

$$\begin{aligned} L_\rho = \sum_{c=1}^{C} \sum_{j \in C_c} &\left(f\left(x_j\right) + \beta g\left(z_j\right)\right) + v^T x_j \\ &\theta\left(x_j - z_j\right) + \frac{\rho}{2} D_i \left|\left|x_j\right|\right|^2 + \frac{\rho}{2} \|x_j - z_j\|^2 \end{aligned} \tag{5.2.0.70}$$

where we have defined:

$$v_i = \left(\sum_{k \in \mathcal{N}_i} sign(k-i)\, \eta_{\{i,k\}} - \rho x_k \right) \tag{5.2.0.71}$$

this is a rescaled version of the Lagrange multiplier, η, which respects the graph structure.

Then by differentiating (6.4.0.21) with respect to x_j and z_j we can find closed forms for the updates as:

Theorem 1.

$$x_j^{k+1} := \left(A_j^T A_j + (\rho D_J + 1) I\right)^{-1} \left(A_j^T y_j + z^k - \nu^{kT}\right) \tag{5.2.0.72}$$

$$z_j^{k+1} := S_{\beta/\rho}\left(x_j^{k+1}\right) \tag{5.2.0.73}$$

$$\theta_j^{k+1} := \theta_j^k + \rho\left(x^{k+1} - z^{k+1}\right) \tag{5.2.0.74}$$

$$\eta_j^{k+1} := \eta_j^k + \rho\left(\sum_{m \in N_j} z_m^k - z_j^k\right) \tag{5.2.0.75}$$

This algorithm can be thought of as follows: each node performs an iteration of (non multi-block) ADMM - i.e. each node solves an approximate Gaussian least-squares problem and then soft-thresholds - and then exchanges the result of this computation with its one-hop neighbours. This explains the inclusion of an extra Lagrange multiplier: the multiplier θ controls how far each node moves from its previous estimate in each iteration, whilst the multiplier η enforces consistency between nodes. Note that there is no communication of data between the nodes - only the result the computation in each round.

5.3 Joint Space-Frequency Model

We write the power spectral density (psd) of the sth transmitter as:

$$\phi_s = \sum_b \beta_{bs} \psi_b\left(f\right) \tag{5.3.0.76}$$

This model expresses in psd of the transmitter in a suitable basis - for example $\psi_b\left(f\right)$ could be zero everywhere except for the set of frequencies where $f = b$ i.e. ψ is a rectangular function with height β_{bs} and support f. Other candidates for ψ include splines (e.g. raised cosines), and complex exponentials.

Given this, the psd at the rth receiver is:

$$\phi_r = \sum_s g_{sr} \phi_s = \sum_s g_{sr} \sum_b \beta_{bs} \psi_b\left(f\right) \tag{5.3.0.77}$$

where

$$g_{sr} = \exp\left(-||x_r - x_s||_2^{\alpha}\right) \tag{5.3.0.78}$$

is the channel response between the sth transmitter and the rth reciver.

This model can be summarised using Kronecker products as follows:

Let $\tilde{G} = g_s r^T$, e_r, e_b be unit vectors i.e. they are 1 for the i^{th} receiver or frequency band respectively.

The received power at a receiver (when only a single transmitter is transmitting) can be written:

$$y_r = \left(e_r^T \bigotimes I_{n_b}\right) y \tag{5.3.0.79}$$

with,

$$y = \left(\tilde{G} \bigotimes I_{n_b}\right) \phi \tag{5.3.0.80}$$

Now, we have

$$\phi = e_s \bigotimes \phi_s \tag{5.3.0.81}$$

so,

$$y = \left(\tilde{G} \bigotimes I_{n_b}\right) \left(e_s \bigotimes \phi_s\right) \tag{5.3.0.82}$$

finally we have,

$$y_r = \left(e_r^T \bigotimes I_{n_b}\right) \left[\left(\tilde{G} \bigotimes I_{n_b}\right) \left(e_s \bigotimes \phi_s\right)\right] \tag{5.3.0.83}$$

$\beta_{bs} \in \mathbb{R}^{1 \times n_b}$, $g_{sr} \in \mathbb{R}^{n_r \times n_s}$ and $\psi_{kb} \in 1 \times n_k n_b$ where n_k is the number of frequency bands (in this example $n_k = n_b$.

In the absence of knowledge of the location of the transmitters we introduce a grid of *candidate* locations, to make the above model linear. s now runs over the set of these candidate locations.

The problem of estimating the coefficients, β, from noisy observations $y = \phi_r + N(0,1)$ is now one that can be tackled by linear regression/convex optimisation.

5.4 Results

The model described in section (6.3), equation (6.3.0.10) was simulated, with a wideband signal of 201 channels and a network of 50 nodes (i.e. the signal will be sampled at a 1/4 of rate predicted by Nyquist theory). The mixing patterns were generated from iid Gaussian sources (i.e the matrix S had each entry drawn from an iid Gaussian source). Monte Carlo simulations were performed at SNR values ranging from 5 to 20, and the expected Mean Squared Error (MSE) of solutions of a centralised solver (spgl1) and a distributed solver (ADMM) were calculated over 10 simulations per SNR value. The results can be seen in fig (5.4).

The MSE was calculated as follows:

$$\frac{\left|\left|Z^k - Z*\right|\right|}{||Z*||} \tag{5.4.0.84}$$

where Z^k is the result of the algorithm at iteration k, and Z^* is the optimal solution.

These results indicate that for both centralised and distributed solvers, adding noise to the system results in a degrading of performance. Interestingly note, that the distributed solver seems to (slightly) outperform the centralised solver at all SNRs. This is counter-intuitive, as it would be expected that centralised solvers knowing *all* the available information would outperform distributed solutions. We conjecture that the updates described in section (6.4), take into account differences in noise across the network. The distributed averaging steps, which form the new prior

for each node, then penalise updates from relatively more noisy observations. This corroborates observations from [**bazerque2008**].

This observation is (partially) confirmed in figure (**??**), which plots the progress of the centralised and distributed solvers (as a function of iterations) towards the optimum solution. The SNR is 0.5 (i.e the signal is twice as strong as the noise). Note that after around 300 iterations, the MSE of the distributed solver is consistently below that of the centralised solver.

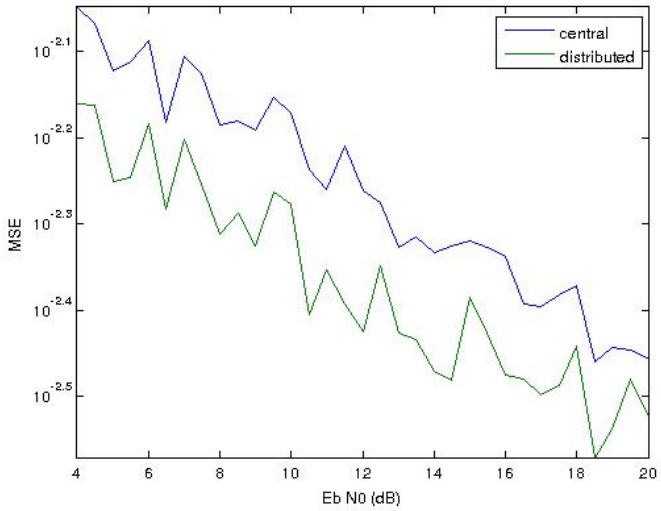

Figure 5.3: Mse vs SNR for the sensing model, with AWGN only, showing the performance of distributed and centralised solvers

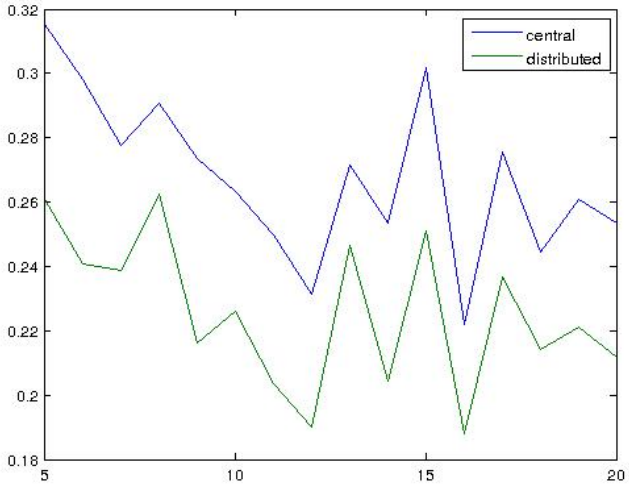

Figure 5.4: Mse vs SNR for the sensing model, showing the performance of distributed and centralised solvers

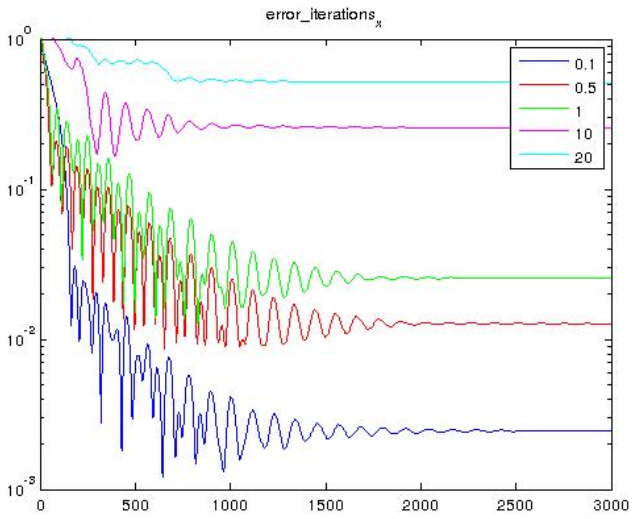

Figure 5.5: The progress of the distributed solver as a function of the number of iterations, with different values of the regression parameter λ

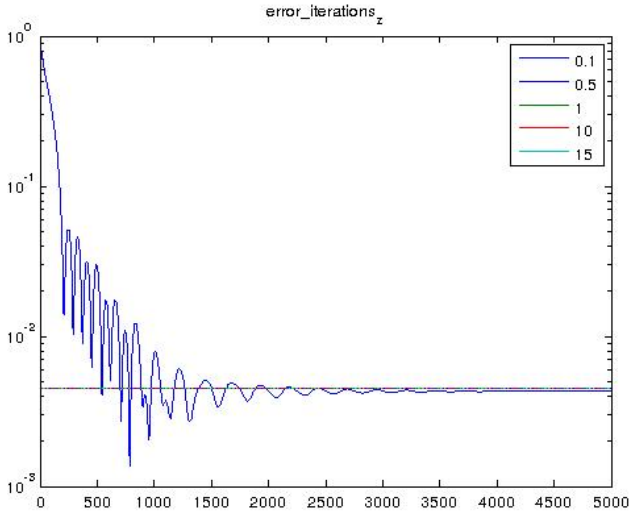

Figure 5.6: The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$

5.5 Conclusions

We have demonstrated an alternating direction algorithm for distributed optimisation with closed forms for the computation at each step, and discussed the statistical properties of the estimation.

We have simulated the performance of this distributed algorithm for the distributed estimation of frequency spectra, in the presence of additive (white, Gaussian) and multiplicative (frequency flat) noise. We have shown that the algorithm is robust to a variety of SNRs and converges to the same solution as an equivalent centralised algorithm (in relative mean-squared-error).

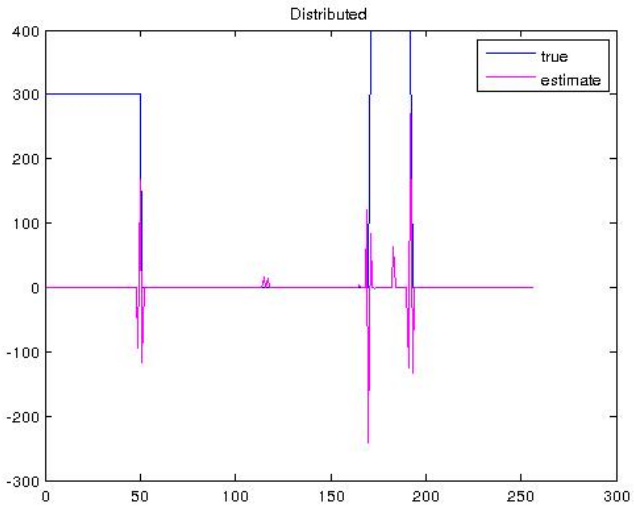

Figure 5.7: The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$

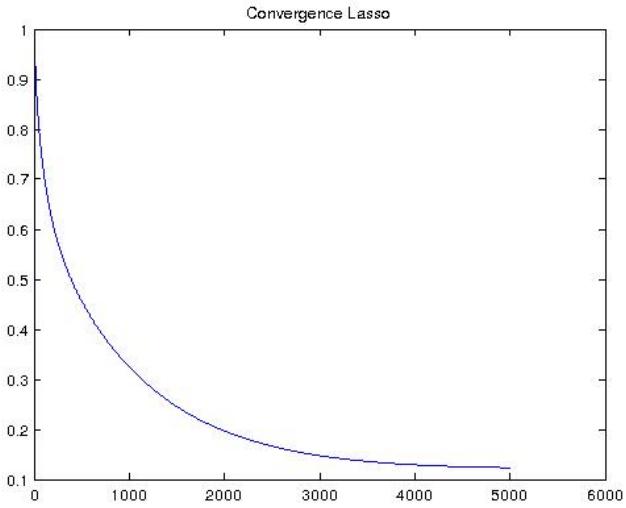

Figure 5.8: The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$

We plan to work on larger, more detailed, models for the frequency spectra and to accelerate the convergence via Nesterov type methods to smooth the convergence of the distributed algorithm [**goldstein2014fast**]. Specifically, we seek to dampen the ringing seen in Figure 5.11

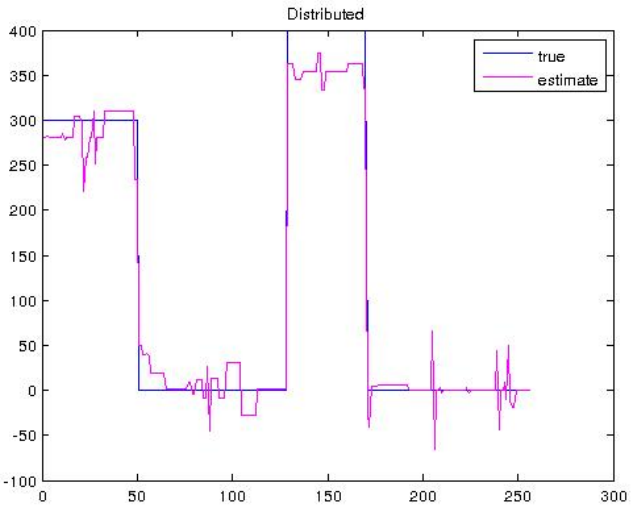

Figure 5.9: The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$

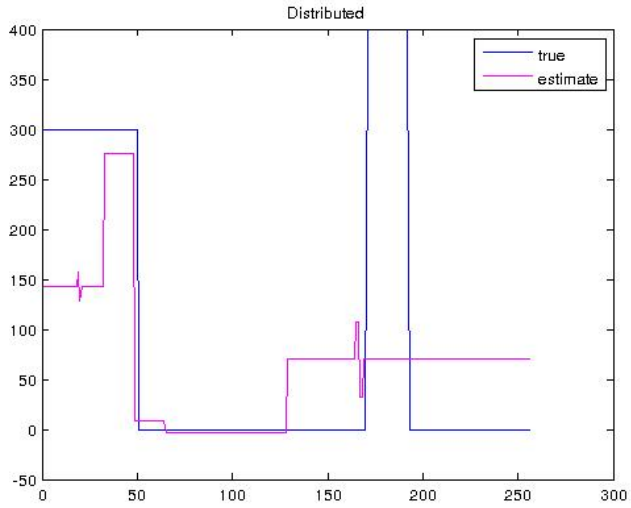

Figure 5.10: The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$

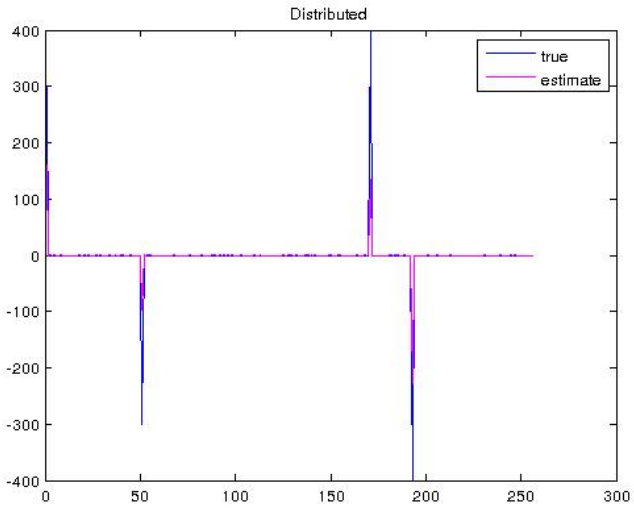

Figure 5.11: The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$

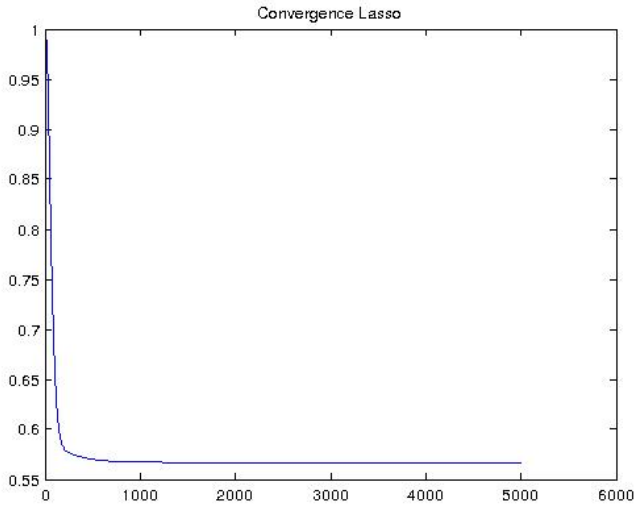

Figure 5.12: The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$

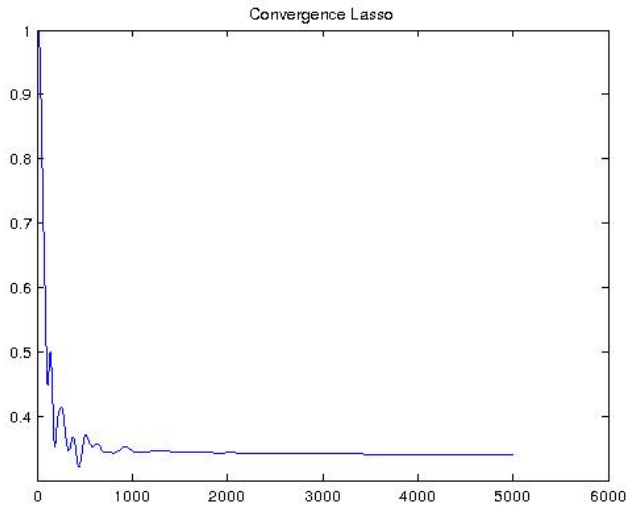

Figure 5.13: The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$

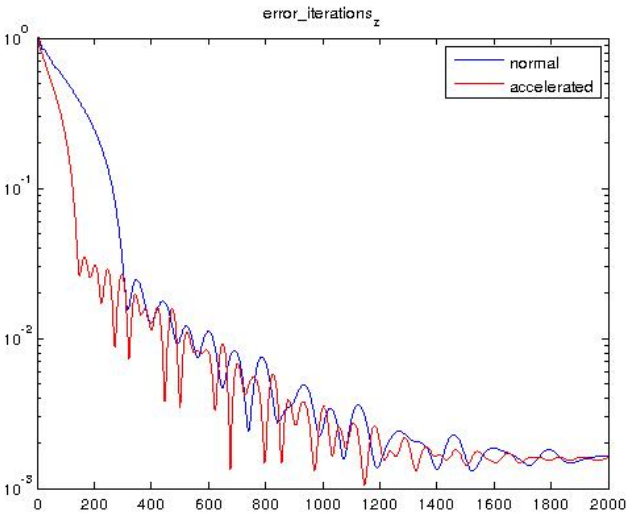

Figure 5.14: The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$

CHAPTER 6

SENSING WITH HEAVYSIDE BASIS

6.1 Introduction

Spectrum Sensing is a key technology for Cognitive Radio. The initial task of any cognitive device, before any kind of dynamic spectrum management will be to accurately sense and classify spectral bands for availability. Dynamic management holds the promise of satisfying the almost ubiquitous growing demand for mobile and wireless data, with consumers demanding faster speeds and better quality connections in more places. However, there is constrained amount of frequencies over which to transmit this information; and demand for frequencies that provide sufficient bandwidth, good range and in-building penetration is high. Not all spectrum is used in all places and at all times, and judicious spectrum management, by developing approaches to use white spaces where they occur, would likely be beneficial.

Devices seeking to access white spaces need a robust mechanism for learning of the frequencies that can be used at a particular time and location. One approach is to refer to a database, which maps the location of white spaces based on knowledge of existing spectrum users. An alternative approach is for devices to detect white spaces by monitoring spectrum use.

The advantages of spectrum monitoring [**akan2009cognitive**] over persisting a database of space-frequency data are the ability of networks to make use of low-cost low-power devices, only capable of making local (as opposed to national) communications, keeping the cost of the network low and opportunistic channel usage for bursty traffic, reducing channel collisions in dense networks. The main technical difficulty preventing spectrum sensing currently, are the high sampling rates required by wideband spectra such as TV white spaces (TVWS). However such spectra are typically sparse, in that transmissions use far fewer frequencies that the available bandwidth.

Compressive Sensing (CS) [**Candes2006**] has recently emerged as a new sampling paradigm, for acquiring sparse signals. Applying this to wireless communication, we are able to reconstruct sparse signals at sampling rates below what would be required by Nyquist theory, for example the works [**mishali2010theory**], [**polo2009compressive**], and [**tropp2010beyond**] detail how this sampling can be achieved.

However, even with CS, spectrum sensing from a single machine will be costly as the proposed TVWS band will be over a large frequency range CS at a single sensor would still require high

sampling rates [**Zhang2011b**]. In this paper we propose a distributed model, which allows a sensing budget at each node far below what is required by centralised CS. The advantages of such a network are that is should be able to average out noise process across some geographic area by making distributed observations, and make use of cheaper sensors due to the lowered sensing budget required per node.

We cannot always guarantee that the frequency spectrum will always be sparse: for example, should TVWS become widely utilised, the spectra will not be sparse. However, even for highly occupied spectra, the gradient of the spectrum will be sparse. This has previously been exploited by [**tian2006wavelet**].

Reconstructing the spectrum from compressive measurements could take place at a fusion centre, but such communications are expensive. It is more efficient therefore to design distributed algorithms where CRs communicate with their neighbours to reach consensus on the reconstruction, given each nodes' private data. However, regularising the reconstruction process would require global co-ordination if Total Variation (the l_1 norm of the gradient of the signal) regularisation was chosen, as.

In this paper we propose a different model for sensing the gradient of the frequency spectrum to [**tian2006wavelet**] - a model which doesn't require Total Variation regularisation of the objective function.

We also propose a decentralised algorithm to solve the LASSO by consensus optimisation. This allows us to design an algorithm which requires no global co-ordination whilst reconstructing the gradient of the spectrum. We choose a convex approach, as convex algorithms require no knowledge of the signal statistics (such as sparsity), and are guaranteed to converge. We are able to find exact, closed form, expressions for the Distributed Lasso, reducing the computational load per iteration whilst obviating the need to approximate the objective function [**ling2015dlm**], [**mokhtari2015dqm**].

The structure of the paper is as follows: in section 6.3 we introduce the sensing model, in section 6.4 we describe the distributed reconstruction algorithm [**mota2013d**], and finally in section 6.5 we show some results of the reconstruction quality of this model.

6.2 Signal Model

Not all signals are sparse in an orthogonal basis: for example, many images are sparse in an over-complete dictionary (set of bases). In particular, frequency spectra for TVWS may no longer be sparse once opportunistic radios begin operating in these frequency bands.

Instead we aim to reconstruct the gradient of the spectrum, as we assume that transitions are constant within a band. Consider the basis defined by the function:

$$l_i(x) = \begin{cases} 1 & \text{if } x \leq i \\ 0 & \text{otherwise} \end{cases} \tag{6.2.0.1}$$

That is, l_i is a left-hand step function.

The basis (7.3.2.42) can be expressed as a matrix in $\mathbb{R}^{n\times n}$ as:

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 & 0\dots 0 \\ 1 & 1 & 0 & 0 & 0\dots 0 \\ 1 & 1 & 1 & 0 & 0\dots 0 \\ \dots & & & & \\ 1 & 1 & 1 & 1 & 1\dots 1 \end{pmatrix} \tag{6.2.0.2}$$

By direct computation, this inverse of L is:

$$D = \begin{pmatrix} 1 & 0 & 0 & 0 & 0\dots 0 \\ -1 & 1 & 0 & 0 & 0\dots 0 \\ 0 & -1 & 0 & 0 & 0\dots 0 \\ \dots & & & & \\ 0 & 0 & 0 & 0\dots -1 & 1 \end{pmatrix} \tag{6.2.0.3}$$

We model our PSD signal g as a linear combination of the basis functions (7.3.2.42):

$$g(x) = \sum_i a_i l_i(x) = L^T a \tag{6.2.0.4}$$

where $a = (a_1, \dots, a_n$ are the coefficients in this basis expansion, and l_i are the rows of L. Note that as defined, g is a column vector.

Proposition 6.2.1.

$$D^T g = a \tag{6.2.0.5}$$

Proof.

$$D^T g = D^T L^T a \tag{6.2.0.6}$$

$$= (LD)^T a \tag{6.2.0.7}$$

$$= a \tag{6.2.0.8}$$

as $LD = I$.

□

6.3 Sensing Model

We consider a radio environment with a single primary user (PU) and a network of J nodes collaboratively trying to sense and reconstruct the PU signal in a fully distributed manner by local communication and regularisation only.

We try to sense and reconstruct a wideband signal, using a network of J (= 50) nodes placed uniformly at random within the square $[0,1] \times [0,1]$.

We consider the frequency domain measurements, formed by each node mixing the signal with a random Gaussian signal $A_j \in \mathbb{R}^n$. The measurements taken at node j are:

(6.3.0.9) $$y_j = A_j H_j g + w_j$$

where $H_j \in \mathbb{R}$ is the scalar channel gain, and $w_j \sim \mathcal{N}(0, \sigma_n^2) \in \mathbb{R}$ is additive white Gaussian noise. For the purposes of comparison in section (6.5), this corresponds to the concatenated system:

(6.3.0.10) $$y = AHg + w$$

where $H \in \mathbb{R}^{n \times n}$ is a block diagonal matrix of channel gains.

The system 6.3.0.10 can then be solved (in the sense of finding the sparse vector a (7.3.2.42) by convex optimisation via minimising the objective function:

(6.3.0.11) $$\hat{a} = \arg\min_a \frac{1}{2} \| AHL^T a - y \|_2^2 + \lambda \| a \|_1$$

where λ is a parameter chosen to promote sparsity. Larger λ means sparser a.

6.4 Constrained Optimisation on Graphs

We model the network of sensors as an undirected graph $G = (V, E)$, where $V = \{1 \dots J\}$ is the set of vertices, and $E = V \times V$ is the set of edges. An edge between nodes i and j implies that the two sensors can communicate. The set of nodes that node i can communicate with is written $\mathcal{N}_i$ and the degree of node i is $D_i = |\mathcal{N}_i|$.

We assume that a proper colouring of the graph is available: that is, each node is assigned a number from a set $C = \{1 \dots c\}$, and no node shares a colour with any neighbour. This is so that nodes may communicate in colour order, as opposed to communicating individually thus reducing the total number of communication rounds required.

Individually nodes make the following measurements (as discussed in section 6.3):

(6.4.0.12) $$\mathbf{y}_j = \mathbf{M}_j \mathbf{x} + \mathbf{n}_j$$

where $\mathbf{M}_j = \left(\mathbf{AHL^T}\right)_j$ is the p^{th} row of the sensing matrix from (6.3.0.10).

To find the $\mathbf{x}$ we are seeking (the solution to (6.3.0.11), to each node we give a copy of $\mathbf{x}$, $\mathbf{x}_j \in \mathbb{R}^n$, and we constrain the copies to be identical across all edges in the network. To separate the minimisation of the ℓ_2 and ℓ_1 norms, we also introduce a dummy variable $\mathbf{z}_j \in \mathbb{R}^n$ to each node. Each node, thus has a separate optimisation to solve, subject to the constraint that it is consistent with its neighbours.

We write the global optimisation variable as $\bar{x}$, which collects together C copies of a $n \times 1$ vector $\mathbf{x}$:

Definition 5. *We define vectors x_c which represent the subset of nodes with colour c, where $c = 1, \dots, C$, and write the vector of length nJ:*

(6.4.0.13) $$\bar{x} = \sum_{c=1}^{C} w_c \otimes x_c = \left[x_{c(1)}^T, \dots, x_{c(J)}^T \right]^T$$

where $w_{c(i)} = \mathbb{I}(c(i) = c)$, $\mathbb{I}$ is the indicator function, and we have written $c(i)$ for the colour of the ith node.

The problem then is to solve:

$$\begin{aligned} \arg\min_{\bar{x}} \sum_{c=1}^{C} \sum_{j \in c} \|M_j x_j - y_j\|_2^2 + \frac{\lambda}{J}\|z\|_1 \\ \text{and } x_i = x_j \text{ if } \{i,j\} \in E \\ \text{and } x_i - z_i = \ \forall i \in \{1,\ldots,C\} \end{aligned} \tag{6.4.0.14}$$

That is, at each node we minimise a Lasso functional constrained to be consistent across edges, but that is separable in the ℓ_2 and ℓ_1 norms.

The first set of constraints (edge-agreement) can be written more compactly by introducing the node-arc incidence matrix B: a V by E matrix where each column is associated with an edge $(i,j) \in E$ and has 1 and -1 in the ith and jth entry respectively. We require that $Bx_j = 0$ for all nodes $j = 1,\ldots,J$. The global constraint is simply $(B \otimes I_n)\bar{x} = 0$, and using definition (6.4.0.13) the constraint $x_i = x_j$ if $\{i,j\} \in E$ can now be written:

$$\sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_c = 0 \tag{6.4.0.15}$$

note that $\left(B^T \otimes I_n\right) \in \mathbb{R}^{nE \times nJ}$.

Together (6.4.0.13) and (6.4.0.15), suggests that the problem (6.4.0.14) can be re-written as:

$$\begin{aligned} \arg\min_{\bar{x}} \sum_{c=1}^{C} \sum_{j \in C_c} \|M_j x_j - y_j\|_2^2 + \beta\|z_j\|_1 \\ \text{s.t. } \sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_c = 0 \\ \text{and } \bar{x}_c - \bar{z}_c = 0 \end{aligned} \tag{6.4.0.16}$$

where $\beta = \frac{\lambda}{J}$.

The global Augmented Lagrangian [**Boyd2010a**] for the problem (6.4.0.16) can be written down as:

$$\begin{aligned} L_\rho = \sum_{c=1}^{C} \Big(\sum_{j \in c} \|M_j x_j - y_j\|_2^2 + \beta\|z_j\|_1 + \\ + \theta^T \left(\bar{x}_j - \bar{z}_j\right) + \frac{\rho}{2} \left|\left|\bar{x}_j - \bar{z}_j\right|\right|_2^2 \Big) + \\ + \eta^T \left(B_c^T \otimes I_n\right) \bar{x}_c + \frac{\rho}{2} \left|\left| \sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_c \right|\right|_2^2 \end{aligned} \tag{6.4.0.17}$$

This is, superficially, similar to the Augmented Lagrangian for the Lasso problem [**Boyd2010a**][Section 6.4]. That is, the terms indexed by j are a straightforward Lasso problem, constrained by edge-wise variables (indexed by c) forcing consistency across the network. However, the problem (as currently written) is not separable across the edges of the network as the final and penultimate term represent the constraint that the nodes agree on their estimates across edges.

To make it possible that 6.4.0.17 can be posed as a constrained optimisation problem at each node, we introduce the following variable:

Definition 6 (Edge-equality vector)**.**

$$\begin{aligned} u &:= \left(B^T \otimes I_n\right) \bar{x} \\ &= \left(B^T \otimes I_n\right) \sum_{c=1}^{C} w_c \otimes x_c \\ &= \sum_{c=1}^{C} B_c^T \otimes x_c \end{aligned}$$

where we have used the definition (6.4.0.13) *in the second line, the property of Kronecker products* $(A \otimes C)(B \otimes D) = (AB \otimes CD)$ *between the second and third lines, and we write* $B_c = w_c^T B$*.*

The terms $\| \sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_c \|^2$ and $\eta^T \left(B_c^T \otimes I_n\right) \bar{x}_c$ of (6.4.0.17), can be decomposed across edges, using the following lemma:

Lemma 6.4.1 (Edge Decomposition)**.**

$$\left|\left| \sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_c \right|\right|^2 = \sum_{j \in C_c} \left(D_j \left|\left| x_j \right|\right|_2^2 - \sum_{k \in \mathcal{N}_j} x_j^T x^k \right) \tag{6.4.0.18}$$

and

$$\eta^T \sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_1 = \sum_{l \in C_c} \sum_{m \in \mathcal{N}_l} \operatorname{sign}(m-l)\, \eta_{ml}^T x_l \tag{6.4.0.19}$$

where η *is decomposed edge-wise:* $\eta = \left(\ldots, \eta_{ij}, \ldots\right)$*, such that* $\eta_{i,j} = \eta_{j,i}$*, and is associated with the constraint* $x_i = x_j$*.*

Proof. For the first part, note that

$$\begin{aligned} u^T u &= \left|\left| \sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_c \right|\right|^2 \\ &= \sum_{c_1=1}^{C} \sum_{c_2=1}^{C} \left(B_{c_1} \otimes x_{c_1}^T\right) \left(B_{c_2}^T \otimes x_c\right) \\ &= \sum_{c_1, c_2} B_{c_1} B_{c_2}^T \otimes x_{c_1}^T x_{c_2} \end{aligned}$$

BB^T is a $J \times J$ matrix, with the degree of the nodes on the main diagonal and -1 in position (i, j) if nodes i and j are neighbours (i.e BB^T is the graph Laplacian). Hence, since we can write $B_{c_1} B_{c_2}^T = w_{c_1}^T B B^T w_{c_2}$, the trace of $B_{c_1} B_{c_1}^T$ is simply the sum of the degrees of nodes with colour 1. Similar reasoning applies to all other colours.

For $c_1 \neq c_2$, $B_{c_1} B_{c_2}^T$ corresponds to an off diagonal block of the graph Laplacian, and so counts how many neighbours each node with colour 1 has.

For the second part note that $\eta \in \mathbb{R}^{nE}$ and can be written:

$$\eta = \sum_{c=1}^{C} w_c \otimes \eta_c \tag{6.4.0.20}$$

procedure DADMM(y_j, M_j, ε)
  $x^0 = 0, z^0 = 0, \theta^0 = 0, \eta^0 = 0,$
$Q = \left(M_j^T M_j + (\rho D_J + 1) I\right)^{-1}$, $w_j = M_j^T y_j$
  while $\left|\left|z^{k+1} - z^k\right|\right| \leq \varepsilon$ **do**
    for $c = 1, \ldots, C$ **do**
      $x^{k+1} \leftarrow Q\left(w_j + z^k - \theta^{kT} - \nu^{kT}\right)$
      $z^{k+1} \leftarrow S_{\beta/\rho}\left(x_j^{k+1}\right)$
      $\theta^{k+1} \leftarrow \theta_j^k + \rho\left(x^{k+1} - z^{k+1}\right)$
    end for
Each node transmits x^{k+1} in $\mathcal{N}_j$ and calculates
    $\nu_j^{k+1} \leftarrow \nu_j^k + \rho\left(\sum_{m \in N_j} z_m^k - z_j^k\right)$
  end while
  return z^{k+1}
end procedure

Figure 6.1: The algorithm at Node j

where η_c is the vector of Lagrange multipliers associated across edges from colour c. Now

$$\eta^T u = \sum_{c_1=1}^{C} \sum_{c_2=1}^{C} w_{c_1}^T B w_{c_2} \otimes \eta_{c_1}^T x_{c_2}$$

where we have repeated the reasoning from the previous part: using the properties of Kronecker products, and the definition of B_c. For $c_1 = c_2$, $\eta^T u$ is zero, as there are no edges between nodes of the same colour by definition. For $c_1 \neq c_2$, $\eta^T u$ counts the edges from c_1 to c_2, with the consideration that the edges from c_2 to c_1 are counted with opposite parity. I.e. for a node l with colour C, $w_{c_1}^T B_{c_2}$ counts the edges to the neighbours of node l, and the edges from the neighbours of node l to node l with opposite parity - $\sum_{l \in C_c} \sum_{m \in N_l} \text{sign}(m - l)$. □

Adding together this with the lemma, lets us write (6.4.0.17) as:

$$\begin{aligned} L_\rho = \sum_{c=1}^{C} \Big(& \sum_{j \in C_c} \|M_j x_j - y_j\|_2^2 + \beta \|z_j\|_1 + \nu^T x_j \\ & + \theta\left(x_j - z_j\right) + \frac{\rho}{2} D_i \left|\left|x_j\right|\right|^2 + \frac{\rho}{2} \left|\left|x_j - z_j\right|\right|^2 \Big) \end{aligned} \tag{6.4.0.21}$$

where we have defined:

$$\nu_i = \left(\sum_{k \in \mathcal{N}_i} sign(k - i)\, \eta_{\{i,k\}} - \rho x_k \right) \tag{6.4.0.22}$$

which is a rescaled version of the Lagrange multiplier, η, which respects the graph structure.

Then by differentiating (6.4.0.21) with respect to x_j and z_j we can find closed forms for the updates as:

$$x_j^{k+1} := \left(M_j^T M_j + (\rho D_J + 1) I\right)^{-1} \left(M_j^T y_j + z^k - \theta^{kT} - \nu^{kT}\right) \tag{6.4.0.23}$$

$$z_j^{k+1} := S_{\beta/\rho}\left(x_j^{k+1}\right) \tag{6.4.0.24}$$

$$\theta_j^{k+1} := \theta_j^k + \rho\left(x^{k+1} - z^{k+1}\right) \tag{6.4.0.25}$$

$$\nu_j^{k+1} := \nu_j^k + \rho\left(\sum_{m \in \mathcal{N}_j} z_m^k - z_j^k\right) \tag{6.4.0.26}$$

Where we have defined

Definition 7 (Soft-thresholding)**.**

$$S_\tau\left(y\right) := \text{sign}(y)\max(y - |\tau|, 0) \tag{6.4.0.27}$$

Remark 6.4.2. *This algorithm can be thought of as a distributed EM algorithm with memory: each node places a Gaussian prior with variance proportional to its degree on its private data, and solves a posterior least-squares problem. Each node then soft thresholds and then exchanges the result of this computation with its one-hop neighbours.*

This explains the inclusion of an extra Lagrange multiplier: the multiplier θ controls how far each node moves from its previous estimate in each iteration, whilst the multiplier η enforces consistency between nodes by integrating past disagreements between neighbouring nodes. Note that there is no communication of data between the nodes - only the result the computation in each round.

6.5 Results

The model described in section (6.3), equation (6.3.0.10) was simulated. The signal $g \in \mathbb{R}^{300}$ was composed of 3 rectangular pulses, mimicking primary user signals in TVWS, as shown in figure (6.2) (a). The signal was put through a Rayleigh channel, before being sensed by the nodes. The network was generated as a random geometric graph in $[0,1] \times [0,1]$, with 50 nodes. If the network wasn't connected, it was redrawn. 200 mixing patters were drawn i.i.d from a $\mathcal{N}\left(0, \sigma^2 I_{300}\right)$ distribution, with $\sigma^2 = 1/200$, to from the matrix $A \in \mathbb{R}^{200\times 300}$.

Monte Carlo simulations were performed at 18 σ_n^2 values ranging from 1 to 10 and the expected Mean Squared Error (MSE) of solutions of a centralised ADMM solver and a our distributed solver were calculated over 500 repetitions with 1200 iterations (k) per repetition.

The MSE was calculated as follows:

$$\frac{\left|\left|L^t z^k - g^*\right|\right|}{\left|\left|g^*\right|\right|} \tag{6.5.0.28}$$

where z^k is the result of the algorithm at iteration k, and g^* is the optimal solution.

The SNR for each repetition was calculated as

$$\frac{\left|\left|g^*\right|\right|}{||w||} \tag{6.5.0.29}$$

and averaged over the 500 repetitions. The results are shown in figure (6.3). Following [**Chen1998**], for each repetition we chose

$$\lambda = \sqrt{2\sigma_n^2 \log n} \tag{6.5.0.30}$$

The error bars indicate the empirical variance across the 500 repetitions.

These results indicate that for both the centralised and distributed solvers, their performance degrades as the noise power increases in a roughly log-linear fashion. The performance of the distributed algorithm is consistently worse than the centralised version, this contrasts with results from [**bazerque2008**]; this is due to the differing sparsity models: [**bazerque2008**] use a joint space and frequency model for the sparsity, and as such observe an spatial averaging out of noise when using a distributed solver. The performance of DADMM is within the error bars of the centralised version at low SNR, and gap in performance between the two versions is no more than 10^{-2}. Even at relatively lower SNRs both solvers reach a solution within 10^{-1} of the optimal (as measured by normalised MSE), which will be adequate for the task of spectrum sensing. For example the reconstructions in figures (6.2) (c) and (d) show realisations of the reconstruction from DADMM with $\sigma_n^2 = 5$ and $\sigma_n^2 = 20$ respectively. It is still possible to distinguish the occupied bands from unoccupied frequencies for both reconstructions.

The distributed algorithm has consistently larger variance, than the centralised solver at all SNRs. This is due to individual nodes only having access to a subset of the data to perform calculations on: the variance will be proportional to the square-root of number of data samples at each node, which are fewer than the total number of samples available to the centralised solver.

In figure (6.4), we plot the progress of DADMM along the solution path for a variety of regularisation parameters λ. The y-axis is the relative (unormalised) MSE between the optimal solution and the current iteration, and the x-axis is the iteration number. We note that for a fixed λ there is a single unique optimal solution, which DADMM converges to (in the sense of stationary error between consecutive iterations). This solution may not be attained in the allotted number of iterations, as the rate of convergence is determined by λ, ρ and the eigenvalues of the Laplacian of G. The paper [**shi2014linear**], proves linear convergence for DADMM, with explicit expressions for the rate. In particular the rate convergence of DADMM is affected by the choice of λ: smaller λ corresponds to slower convergence - this is intuitive as solutions with fewer non-zero components should require fewer iterations to fully specify. Notice that for some λs the solution path exhibits phenomenological behaviour similar to damped oscillations: this phenomena has been explored in [**nishihara2015general**] and [**su2014differential**].

6.6 Conclusions

We have demonstrated an alternating direction algorithm for distributed optimisation with exact (as opposed to linear or quadratic approximations to the objective as in [**mokhtari2015dqm**] and [**ling2015dlm**]) closed form expressions for the computation at each iteration, and discussed the statistical properties of the estimation.

We have simulated the performance of this distributed algorithm for the distributed estimation of frequency spectra, in the presence of additive (white, Gaussian) and multiplicative noise. We have shown that the algorithm is robust to a variety of SNRs and converges to a similar solution as an equivalent centralised algorithm (in relative mean-squared-error).

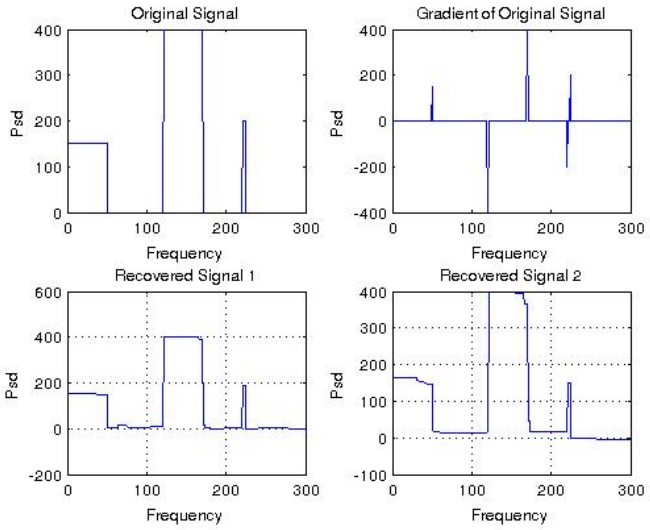

Figure 6.2: Left to right: (a) The original signal. (b) The gradient (6.2.1) of the original signal. (c) Recovery using DADMM, 1000 iterations, $\sigma_n^2 = 5$. (d) Recovery using DADMM, 1000 iterations, $\sigma_n^2 = 20$

We plan to work on larger, more detailed, models for the frequency spectra, to extend our regression framework to solve the MMV problem, to accelerate the convergence via Nesterov type methods to smooth the convergence of the distributed algorithm [**goldstein2014fast**], and to incorporate spatial variation into our model to further promote sparsity. We also plan to automate the choice of λ via continuation methods, and study how the choice of λ and ρ affect the rate of convergence.

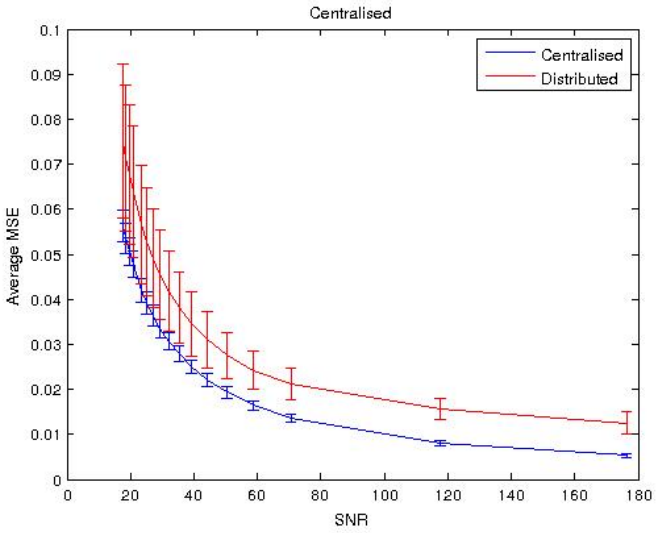

Figure 6.3: MSE vs SNR for the sensing model showing the performance of distributed and centralised solvers. The performance of DADMM is consistently within 10^{-2} of ADMM, and within the error bars of ADMM at low SNRs. The variance of estimates produced by DADMM is larger than ADMM, due to nodes performing computations on a subset of data. Both estimates are consistently within 10^{-1} of the optimal solution, which is sufficient to classify occupied bands.

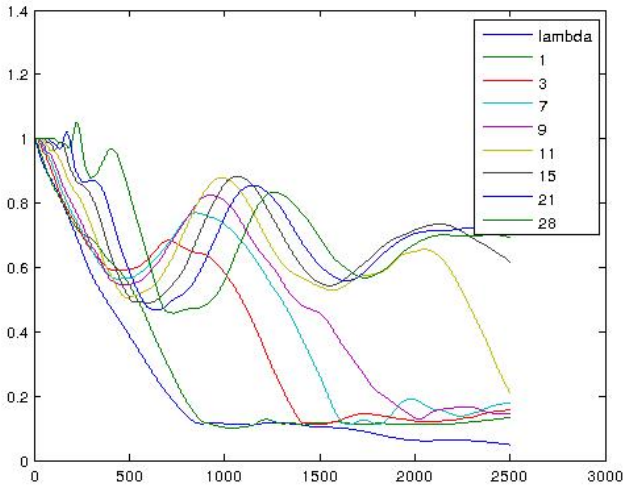

Figure 6.4: The progress of the distributed solver as a function of the number of iterations, with different values of the regression parameter λ. For a fixed λ there is a single unique optimal solution, with higher λ favouring sparser solutions. The convergence of DADMM is slowed by smaller λ. This is intuitive: solutions with fewer non-zero components should be identified in fewer iterations.

CHAPTER 7

COMPRESSIVE INFERENCE

7.1 Introduction

The recent work of Candes and Tao [**Candes2006**] and Donoho [**Donoho2006**] has established that many real-world signals can be effectively captured via a small number of random projections relative to the dimension of the signal. For example, a 5 megapixel image can be thought of as a vector in $\mathbb{R}^{5,000,000}$. However, it is well known that images have relatively few wavelet coefficients; this is exploited by the JPEG-2000 standard, which can represent the image as a 64-kb file (i.e a point in $\mathbb{R}^{64,000}$.

Classically, for perfect signal reconstruction, we must sample a signal such that the sampling rate must be at least twice the maximum frequency in the bandlimited signal. The continuous time signal can then be recovered using an appropriate reconstruction filter (e.g. a sinc filter). For example, we can represent a sampled continuous signal as a multiplication of the signal with a train of Dirac delta functions at multiples of the sampling period T.

$$x(nT) = \text{Ш}(t-nT)\,x(t) \tag{7.1.0.1}$$

where

$$\text{Ш}(t-nT) = \sum_{k=-\infty}^{\infty} \delta(t-kT) \tag{7.1.0.2}$$

Working the frequency domain, this multiplication becomes convolution (which is equivalent to shifting):

$$\hat{X}_s(f) = \sum_{k=-\infty}^{\infty} x(t-kT) \tag{7.1.0.3}$$

Thus if the spectrum of the frequency is supported on the interval $(-B,B)$ then sampling at intervals $\frac{1}{2B}$ will contain enough information to reconstruct the signal $x((t)$. Multiplying the spectrum by a rectangle function (low-pass filtering), to remove any images caused by the periodicity of the function, and the signal $x((t)$ can be reconstructed from its samples:

$$x(t) = \sum_{n=-\infty}^{\infty} x(nT)\, sinc\left(\frac{t_n T}{T}\right) \tag{7.1.0.4}$$

In contrast Compressive Sampling suggests that by adding randomness into the measurement process, a sparse (or compressible signal) may be accurately sensed with far fewer measurements: $y = Ax + w$

where $A \in \mathbb{R}^{m \times n}$ is a matrix with random entries, $x \in \mathbb{R}^n$ is the signal we capture, $y \in \mathbb{R}^m$ is the result of the measurement process and $w \sim N(0,1) \in \mathbb{R}^m$ is additive white Gaussian noise, $m < n$.

Some technical conditions on the matrix A have to satisfied for it: namely the transformation defined by A must behave like an approximate Isometry, and it must be incoherent.

Definition 7.1.1 (RIP)**.** *We say that a matrix A satisifes the RIP of order δ if $\exists$ a $\delta \in (0,1)$ such that:*

$$(1-\delta)\,||x||_2^2 \leq ||Ax||_2^2 \leq (1+\delta)\,||x||_2^2 \tag{7.1.0.5}$$

i.e. A approximately preserves the lengths of all s-sparse vectors in $\mathbb{R}^n$.

Definition 7.1.2 (Coherence)**.** *The mutual coherence of a matrix A is the absolute normalised inner product between different columns from A. Denoting the k-th column in A by a_k, the mutual coherence is given by:*

$$\mu(A) = \max_{1 \leq i,j \leq n, i \neq j} \frac{|\langle a_i^T, a_j \rangle|}{||a_i||_2 \left|\left|a_j\right|\right|_2} \tag{7.1.0.6}$$

This implies that sensing with incoherent systems is good, and efficient mechanisms ought to acquire correlations with random waveforms (e.g. white noise).

Theorem [**Candes2006**] Fix x $\in \mathbb{R}^n$ with a sparse coefficient basis, x_i in ψ. Then a reconstruction from m random measurements is possible with probability $1-\delta$ if:

$$m \geq C\mu^2(A) S \log\left(\frac{n}{\delta}\right) \tag{7.1.0.7}$$

where $\mu(A)$ is the coherence of the two bases, and S is the number of non-zero entries on the support of the signal.

In this new sensing paradigm, the complexity is shifted to the reconstruction process, where with high probability Donoho proved [**donoho2004neighborly**], that the minimiser of the program:

$$\arg\min_x \frac{1}{2}\left|\left|y - Ax\right|\right|_2^2 + \lambda\,||x||_1 \tag{7.1.0.8}$$

coincides with the sparsest solution to the under-determined system of linear equations. Thus we are able to sense sparse signals with random waveforms, and reconstruct them via linear programming.

However, signal reconstruction is not the only interesting signal processing task. Filtering, classification, detection, and estimation are also required in real world systems. For these tasks it was thought that signal reconstruction must be performed first, and then classical signal processing techniques could be brought to bear on the reconstruction.

There is some tension in this idea however: since the measurement matrix is an approximate isometry, some (as yet unspecified) operations on the measurements y should correspond to inference tasks (such as filtering and estimation) on x. This means that performing inference needn't require the reconstruction of the signal.

The papers [**davenport2010signal**] and [**davenport2007smashed**] provide an introductory answer for the cases of filtering, detection, classification and estimation.

There has been a little work applying this line of work: [**schnelle2012compressive**] shows how to design a Phase-Locked loop directly in the compressive domain and applies it to demodulating FM signals, and [**davenport2010wideband**] extends this idea to a wideband compressive radio receiver. Matched filtering from compressive samples is performed in [**eftekhari2013matched**], and

The structure of this document is as follows: sections (7.2), is a literature review of relevant material from compressed sensing, Wishart matrices, and maximum likelihood estimation of uncompressed signals in noise. Section (7.3) gives an overview of the problem of estimating a signal from a known set of basis functions.

7.2 Preliminaries

7.2.1 RIP and Stable Embeddings

Given a signal $x \in \mathbb{R}^n$, a matrix $A \in \mathbb{R}^{m \times n}$ we can acquire the signal via the set of linear measurements:

$$y = Ax \tag{7.2.1.9}$$

where in this case A represents the sampling system. In contrast to classical sensing, which requires that $m = n$ for there to be no loss of information, it is possible to reconstruct x from an under-determined set of measurements as long as x is sparse in some basis.

To make this precise, we define Σ_s as the set of s-sparse signals in $\mathbb{R}^n$:

Definition 7.2.1.

$$\Sigma_s = \{x \in \mathbb{R}^n : \text{supp}\,(x) \leq s\} \tag{7.2.1.10}$$

where $\text{supp}\,(x)$ *is the set of indices on which* x *is non-zero.*

Definition 7.2.2 (RIP)**.** *We say that a matrix* A *satisifes the RIP of order* s *if there exists a* $\delta \in (0,1)$ *such that for all* $x \in \Sigma_s$*:*

$$(1-\delta)\,||x||_2^2 \leq ||Ax||_2^2 \leq (1+\delta)\,||x||_2^2 \tag{7.2.1.11}$$

i.e. A approximately preserves the lengths of all s*-sparse vectors in* $\mathbb{R}^n$*.*

Remark 7.2.3 (Information Preservation)**.** *A necessary condition to recover all* s*-sparse vectors from the measurements* Ax *is that* $Ax_1 \neq Ax_2$ *for any pair* $x \neq x_2$*,* $x_1, x_2 \in \Sigma_s$*, which is equivalent to* $||A(x_1 - x_2)||_2^2 > 0$*.*

This is guaranteed as long as A *satisfies the RIP of order* $2s$ *with constant* δ *- as the vector* $x_1 - x_2$ *will hve at most* $2s$ *non-zero entries, and so will be distinguishable after multiplication with* A*. To complete the argument take* $x = x_1 - x_2$ *in definition* (7.2.2)*, guaranteeing* $||A(x_1 - x_2)||_2^2 > 0$*, and requiring the RIP order of* A *to be* $2s$*.*

Remark 7.2.4 (Stability)**.** *We also require that the dimensionality reduction of compressed sensing is the preservation of relative distances: that is if x_1 and x_2 are far apart in $\mathbb{R}^n$ then their projections Ax_1 and Ax_2 are far apart in $\mathbb{R}^m$. This will guarantee that the dimensionality reduction is robust to noise.*

A requirement on the matrix A that satisfies both of these conditions is the following:

Definition 7.2.5 (δ-stable embedding)**.** *We say that a mapping is a δ-stable embedding of $U, V \subset \mathbb{R}^n$ if*

$$(1-\delta)\,||u-v||_2^2 \leq ||Au-Av||_2^2 \leq (1+\delta)\,||u-v||_2^2 \tag{7.2.1.12}$$

for all $u \in U$ and $v \in V$.

Remark 7.2.6. *Note that a matrix A, satisfying the RIP of order $2s$ is a δ-stable embedding of Σ_s, Σ_s.*

Remark 7.2.7. *Definition 7.2.5 has a simple interpretation: the matrix A must approximately preserve Euclidean distances between all points in the signal model Σ_s.*

7.2.2 Random Matrix Constructions

To construct matrices satisfying definition 7.2.5, given m, n we generate A by A_{ij} being i.i.d random variables from distributions with the following conditions [**davenport2010signal**]

Condition 3 (Norm preservation)**.** $\mathbb{E}A_{ij}^2 = \frac{1}{m}$

Condition 4 (sub-Gaussian)**.** $\mathbb{E}\left(e^{A_{ij}t}\right) \leq e^{C^2t^2/2}$

Random variables A_{ij} satisfying conditions (3) and (4) satisfy the following concentration inequality [**baraniuk2008simple**]:

Condition 5 (sub-Gaussian)**.**

$$\mathbb{P}\left(|\,||Ax||_2^2 - ||x||_2^2\,| \geq \varepsilon\,||x||_2^2\right) \leq 2e^{-cM\varepsilon^2} \tag{7.2.2.13}$$

Then in [**baraniuk2008simple**] the following theorem is proved:

Theorem 7.2.8. *Suppose that m, n and $0 < \delta < 1$ are given. If the probability distribution generating A satisfies condition* (7.2.2.13)*, then there exist constants c_1, c_2 depending only on δ such that the RIP* (7.2.2) *holds for A with the prescribed δ and any $s \leq \frac{c_1 n}{\log n/s}$ with probability $\geq 1-2e^{-c_2 n}$*

For example, if we take $A_{ij} \sim \mathcal{N}(0, 1/m)$, then the matrix A will satisfy the RIP

7.2.3 Wishart Matrices

Let $\{X_i\}_{i=1}^r$ be a set of i.i.d $1 \times p$ random vectors drawn from the multivariate normal distribution with mean 0 and covariance matrix H.

$$X_i = \left(x_1^{(i)}, \ldots, x_p^{(i)}\right) \sim N(0, H) \tag{7.2.3.14}$$

We form the matrix X by concatenating the r random vectors into a $r \times p$ matrix.

Definition 7.2.9 (Wishart Matrix)**.** *Let*

$$W = \sum_{j=1}^{r} X_j X_j^T = X X^T \tag{7.2.3.15}$$

Then $W \in \mathbb{R}^{r\times r}$ has the Wishart distribution with parameters

$$W_r\left(H, p\right) \tag{7.2.3.16}$$

where p is the number of degrees of freedom.

Remark 7.2.10. *This distribution is a generalisation of the Chi-squared distribution: let $p = H = 1$.*

Theorem 7.2.11 (Expected Value)**.**

$$\mathbb{E}(W) = rH \tag{7.2.3.17}$$

Proof.

$$\begin{aligned}
\mathbb{E}(W) &= \mathbb{E}\left(\sum_{j=1}^{r} X_j X_j^T\right) \\
&= \sum_{j=1}^{r} \mathbb{E}\left(X_j X_j^T\right) \\
&= \sum_{j=1}^{r} \left(\mathrm{Var}(X_j) + \mathbb{E}(X_j)\mathbb{E}(X_j^T)\right) \\
&= rH
\end{aligned}$$

Where the last line follows as X_j is drawn from a distribution with zero mean. □

Remark 7.2.12. *The matrix $M = A^T A$, where A is constructed by the methods from section 7.2.2, will have a Wishart distribution. In particular, it will have $\mathbb{E}M = \frac{1}{m} I_n$*

The joint distribution of the eigenvalues is given by [**levequeMatrices**]:

$$p(\lambda_1, \ldots, \lambda_r) = c_r \prod_{i=1}^{r} e^{-\lambda_i} \prod_{i<j} \left(\lambda_i - \lambda_j\right)^2 \tag{7.2.3.18}$$

The eigenvectors are uniform on the unit sphere in $\mathbb{R}^r$.

7.2.4 Maximum Likelihood estimation: non-compressive case

Consider a received signal $y \in \mathbb{R}^n$, composed of a deterministic signal $\bar{s} \in \mathbb{R}^n$ corrupted by noise $n \in \mathbb{R}^n$ (assumed to have zero mean and unit variance), i.e.

$$y = s + n \tag{7.2.4.19}$$

We assume $s(\Theta)$ comes from a fixed class of signals, with parameters indexed by a set Θ. For example

- The signal s is composed of a single frequency with unit amplitude $s = e^{i\omega_0 k}$. In this case $\Theta = \{\omega_j\}_{j=1}^n$ is the set of possible frequencies the signal may take on.
- The signal s is a scaled and shifted version of a model signal f: $s(t) = Cf(t-\tau)$. In this case $\Theta = (C, \tau)$.
- The signal s can be expanded in some orthonormal basis, and that we have access to the basis functions $\{\phi_i\}_{i=1}^n$:

$$s = \sum_{i=1}^{n} \alpha_i \phi_i \tag{7.2.4.20}$$

  In this case $\Theta = \{\alpha_i, \phi_i\}_{i=1}^n$

We can write the likelihood for y down as, for a given Θ, s is deterministic. Therefore y is a Gaussian random variable with mean s:

$$f\left(y \mid s\right) = \left(\frac{1}{\sqrt{2\pi}}\right)^n \exp\left(-\frac{\left(y - s(\Theta)\right)^T \left(y - s(\Theta)\right)}{2}\right) \tag{7.2.4.21}$$

Maximising this is equivalent to maximising:

$$\ln f\left(\Theta\right) = -\left|\left|y\right|\right|_2^2 + 2\langle y, s\left(\Theta\right)\rangle - ||s||_2^2 \tag{7.2.4.22}$$

Since the terms $\left|\left|y\right|\right|_2^2$ and $||s||_2^2$ do not change, we can write the ML estimate of Θ as:

$$\hat{\Theta} = \arg\max_{\Theta} \langle y, s\left(\Theta\right)\rangle \tag{7.2.4.23}$$

So, for example:

- If we receive a single tone corrupted by noise $y = e^{i\omega_0 k} + n$ then we can estimate ω_0 by calculating $\arg\max_{\omega_i} \langle y, e^{i\omega_j k}\rangle = e^{i\omega_0 k}$ as the functions $e^{i\omega_0 jk}$ are orthonormal: $\langle e^{i\omega_l k}, e^{i\omega_r k}\rangle = \delta_{lr}$.
- If we receive a signal composed of a sum of orthonormal basis functions we can estimate the coefficients α_i as:

$$\langle y, \phi_i\rangle = \sum_{j=1}^{n} \alpha_j \phi_j^T \phi_i + n^T \phi_i \tag{7.2.4.24}$$

$$= \alpha_i + \varepsilon_i \tag{7.2.4.25}$$

  where $\varepsilon_i = \langle n^T, \phi_i\rangle$ is some small error. Thus the maximum likelihood estimate of s is:

$$\hat{s} = \sum_{i=1}^{n} \hat{\alpha}_i \phi_i \tag{7.2.4.26}$$

  where

$$\hat{\alpha} = \langle y, \phi_i\rangle \tag{7.2.4.27}$$

7.3 Compressive Estimation

In this section, we develop some intuition into constructing estimators for the signal s directly on the compressive measurements:

$$y = A(s+n) \tag{7.3.0.28}$$

where $A \in \mathbb{R}^{m\times n}$, $A_{ij} \sim \mathcal{N}(0, 1/m)$, and $n \in \mathbb{R}^n$ is AWGN. We again assume that s comes from a fixed set of models, parametrised by some set Θ.

The likelihood for this model is, (as y is a normal random variable):

$$f(y \mid s) = \left(\frac{1}{\sqrt{2\pi}}\right)^m \exp\left(-\frac{(y-As)^T(y-As)}{2}\right) \tag{7.3.0.29}$$

Taking the logarithm and expanding, we find

$$\ln f = -y^T y - s^T A^T As + 2\langle y, As\rangle \tag{7.3.0.30}$$

which is equal to:

$$lnf = -\left|\left|y\right|\right|_2^2 - ||As||_2^2 + 2\langle y, As\rangle \tag{7.3.0.31}$$

The first term of (7.3.0.31) is constant, for the same reasons as in section (7.3). The term

$$||As||_2^2 = \langle As, As\rangle \tag{7.3.0.32}$$

can be written as

$$\langle A^T As, s\rangle \tag{7.3.0.33}$$

We will replace this with its expectation $\mathbb{E}\big(\langle A^T As, s\rangle\big)$

$$\begin{aligned}
\mathbb{E}\big(\langle A^T As, s\rangle\big) &= \mathbb{E}\sum_{i=1}^{n}(A^T As)_i^T s_i \\
&= \sum_{i=1}^{n}\mathbb{E}(A^T As)_i s_i \\
&= \sum_{i=1}^{n}(\frac{1}{m}e_i s_i)_i^T s_i \\
&= \frac{1}{m}\langle s, s\rangle
\end{aligned}$$

because

(7.3.0.34) $$\mathbb{E}A^T A = \frac{1}{m} I$$

as it is a Wishart matrix (see section7.2).
So we can further approximate (7.3.0.31):

(7.3.0.35) $$\ln f = -\left|\left|y\right|\right|_2^2 + \frac{1}{m}||s||_2^2 + 2\langle y, As\rangle$$

The only, non-constant part of (7.3.0.35) is the third term and so we define the estimator:

(7.3.0.36) $$\hat{s} = \arg\max_{\Theta} \langle y, As(\Theta)\rangle$$

For the case where s can be expanded in an orthonormal basis $s = \sum_{i=1}^{n} \alpha_i \phi_i$, the maximum likelihood estimator is:

(7.3.0.37) $$\hat{s} = \sum_{i=1}^{n} m\langle y, A\phi_i\rangle \phi_i$$

Consider the case where $y = As$ (no noise). Then

$$y^T A\phi_j = \sum_i \alpha_i \phi_i^T A^T A\phi_j$$

So

$$y^T A\phi_j = \sum_i \alpha_i \phi_i^T A^T A\phi_j \sim \frac{\alpha_i}{m}\delta_{ij}$$

giving

(7.3.0.38) $$\hat{\alpha_i} = m\left(y^T A\phi_j\right)$$

Remark 7.3.1. *The matrix $M = A^T A$ is the projection onto the row-space of A. It follows that $||Ms||_2^2$ is simply the norm of the component of s which lies in the row-space of A. This quantity is at most $||s||_2^2$, but can also be* 0 *if s lies in the null space of A. However, because A is random, we can expect that $||Ms||_2^2$ will concentrate around $\sqrt{m/n}\,||s||_2^2$ (this follows from the concentration property of sub-Gaussian random variables* (7.2.2.13)*).*

7.3.1 Example: Single Spike

We illustrate these ideas with a simple example: estimate which of n frequencies s is composed of.

A signal $s \in \mathbb{R}^{300}$ composed of a single (random) delta function, with coefficients drawn from a Normal distribution (with mean 100, and variance 1) i.e

$$s = \alpha_i \delta_i \tag{7.3.1.39}$$

with

$$a_i \sim \mathcal{N}(100, 1) \tag{7.3.1.40}$$

and the index i chosen uniformly at random from $[1, n]$.
The signal was measured via a random Gaussian matrix $A \in \mathbb{R}^{100\times 300}$, with variance $\sigma^2 = 1/100$ and the inner product between $y = As$ and all 300 delta functions projected onto $\mathbb{R}^{100}$ was calculated:

$$\hat{\alpha}_j = m\langle (A\alpha_i\delta_i), A\delta_j \rangle \tag{7.3.1.41}$$

We plot the $\hat{\alpha_j}$ below, figure 7.1, (red circles), with the original signal (in blue, continuous line). Note how the maximum of the $\hat{\alpha_j}$, coincides with the true signal.

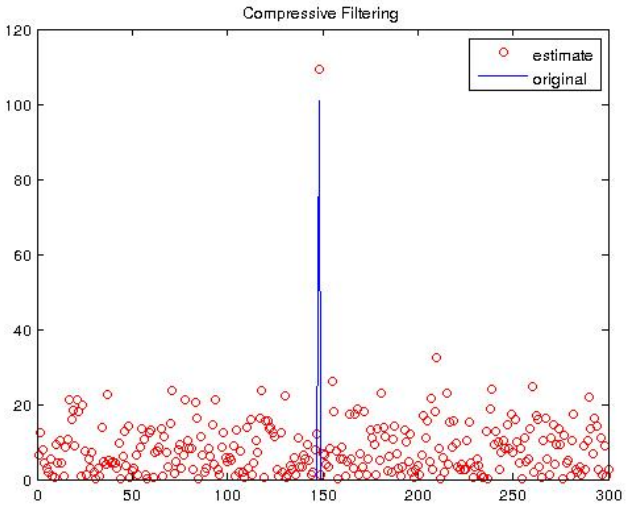

Figure 7.1:

7.3.2 Estimating a single rectangle

We show how to estimate the signal, composed of a single rectangle (7.2) expanded in the following basis

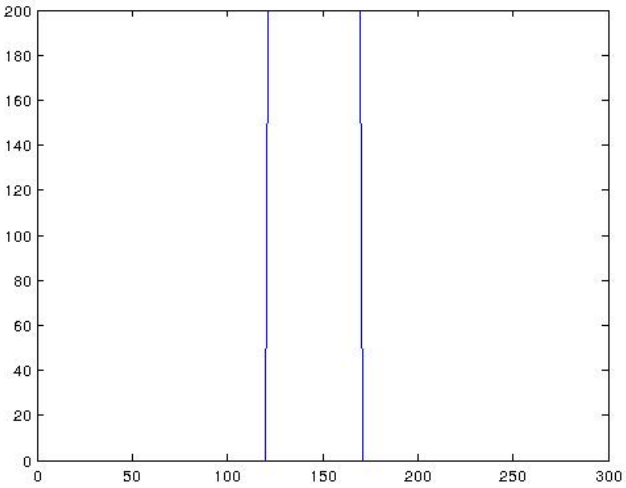

Figure 7.2:

$$f_i(x) = \begin{cases} 1 & \text{if } x \le i \\ 0 & \text{otherwise} \end{cases} \tag{7.3.2.42}$$

We model our signal g as a linear combination of the basis functions (7.3.2.42):

$$g(x) = \sum_i a_i f_i \tag{7.3.2.43}$$

To find the a_i, we correlate (take the inner product of) the signal against the basis (7.3.2.42).

Definition 7.3.2.

$$h_j = \langle g, f_j \rangle \tag{7.3.2.44}$$

$$= \sum_j g(x) f_j(x) \tag{7.3.2.45}$$

$$= \sum_j a_i f_i(x) f_j(x) \tag{7.3.2.46}$$

$$= a_i \langle f_i, f_j \rangle \tag{7.3.2.47}$$

$$\left(= \sum_{x=1}^{j} g(x) \right) \tag{7.3.2.48}$$

As per the previous sections, we take compressive measurements:

$$y = Ag \tag{7.3.2.49}$$

where $A \in \mathbb{R}^{m \times n}$, $A_{ij} \sim \mathcal{N}(0, 1/m)$, and then compute

$$\langle y, Af_i \rangle = a_j f_j^T A^t A f_j \sim \frac{a_j}{m} \langle f_j, f_i \rangle \tag{7.3.2.50}$$

for the set of basis vectors $f_1 \dots f_n$ i.e. the estimator from the previous section, corresponding to this set of basis functions (7.3.0.36).

We then form the vector

$$\hat{h} = m \sum_i \langle y, A f_i \rangle f_j \sim h \tag{7.3.2.51}$$

An example can be seen in figure 7.3, for a matrix $A \in \mathbb{R}^{200\times 300}$.

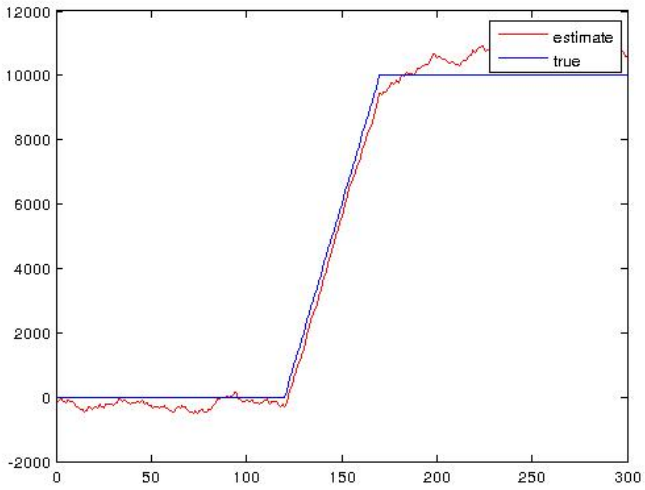

Figure 7.3:

7.3.3 Estimating Frequency spectra

Continuing from the previous (sub)-section we can create an estimate of g, by the following procedure:

- Estimate the coefficients of the basis $\hat{a}$ using $\hat{a} = F^{-1}\hat{h}$
- Choose the k largest (for some k to be determined later).
- Between the indices of the k $\hat{a}$ take the average of the signal $L^{-1}\hat{h}$.

Some examples of the output of this procedure are shown below, for synthetic and real (Ofcom) data.

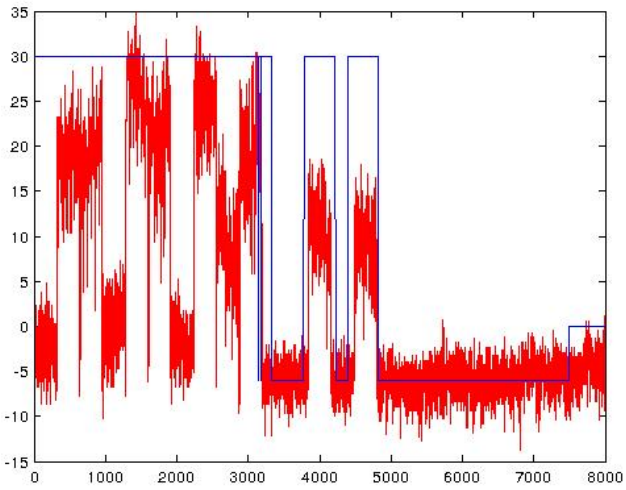

Figure 7.4: Example of classification with OFCOM data, 35 changepoints

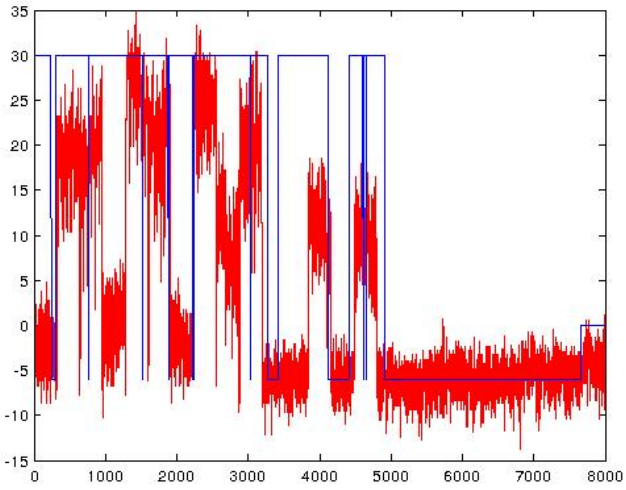

Figure 7.5: Example of classification with OFCOM data, 55 changepoints

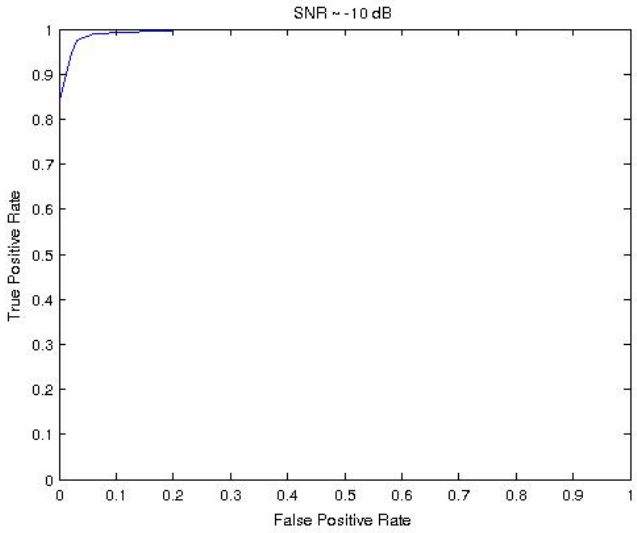

Figure 7.6: ROC for synthetic data, midly noisy

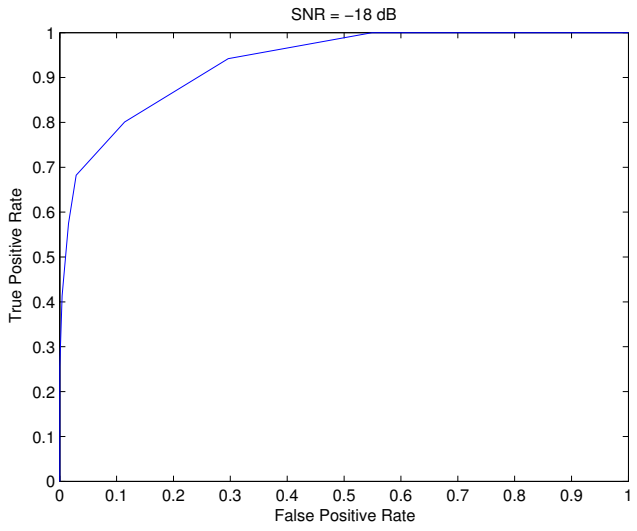

Figure 7.7: ROC for synthetic data, very noisy

CHAPTER 8

GROUP TESTING

8.1 Introduction and notation

8.1.1 Group Testing

Group Testing originated in the second world war because of the need to test all incoming conscriptees for syphilis. It would have been inefficient and expensive to test each soldier individually, as the rate for syphilis was only 10 per 100,000. Dorfman [**Dorfman1943**], considered the idea of pooling blood samples and testing the pooled samples for syphilis and only further testing the pools which come up positive.

A typical problem that can be solved by Group Testing is finding a counterfeit coin in a group of otherwise identical coins by weighing groups of coins on a pan balance. For example, given 80 coins known to contain a single counterfeit, which is lighter than the others, what is the minimum number of weighings needed to determine the counterfeit with certainty? You may get lucky and pick the counterfeit in the for the first go: but there's only a $\frac{1}{80}$ chance of that happening. There's also no need to check all $\binom{80}{1}$ combinations of pairs of coins. However, putting more than one coin on a pan reveals the same information - it's better to weigh groups of coins against each other.

Choose the groups so that each weighing can distinguish between the hypothesis that the pans will balance, or than there will be a heavier pan i.e. split the initial group into 3 (27, 27, 26). Continue this process recursively, splitting the remaining group into 3 each time, until you have found the counterfeit. If the two groups of 27 balance initially, take a coin from one of those groups and add it to the group of 26 to make a power of 3. This won't add any new information (you know this coin is not counterfeit) and so won't affect the inference.

The Group Testing problem can be formalised as follows: a set of items is given, along with an upper bound on the number of defectives. The set is described as a vector, where if an item is 0 it is not defective and 1 if it is defective. Before the tests are run, the position of the 1's is unknown.

To find the defective items, a query is run against a subset of $[n]$, where the answer is defined as follows:

(8.1.1.1) $$A(S) = 1\sum_i x_i \geq 1$$

Note that the addition is the binary-or in the above summation. The goal of Group Testing is to minimise the number of tests required to reconstruct the defective set.

Algorithms

An initial algorithm to consider is a simple binary search of the set to be tested. That is, given a set of size $N = 2^r$ i.e a power of 2, we can recover a single defective in $\lceil \log_2 N \rceil$ tests.

To do this, create a new set of size $S = 2^{\lceil \log_2 n \rceil}$ which is guaranteed to contain a defective. Label the items with integers, and test the items in the sets $1, 2, \ldots, S/2$ and $S/2+1, \ldots, S/2$ separately. Then repeat the procedure on any groups which have a positive test.

To see why this testing procedure takes at most $\lceil \log_2 n \rceil$ tests, note that the procedure defines a binary tree over subsets of the N items, and so the depth of this tree is $\lceil \log_2 N \rceil$.

For input sets with more than a single defective (say K defectives) the binary search algorithm can be repeated, and each time a defective is found it is removed from the set. The binary search is then repeated, but on a set of size $N-1$. Using this procedure we are guaranteed to find all the defectives in

(8.1.1.2) $$K\lceil \log_2 N \rceil \leq K \log_2 N + K$$

tests. However, this is a very inefficient algorithm: early sets are large and so are likely to contain a defective.

The above algorithms return, with certainty, after at most $K\lceil \log_2 \binom{N}{K} \rceil$ tests, the defective set. Much work has gone into combinatorial search algorithms, often more complex than those described above.

This has been motivated by the analogy that the Group Testing problem can be considered a decoding problem where an experimenter receives a binary vector:

(8.1.1.3) $$y = \mathbf{A}x$$

$y \in \{0,1\}^K$, $x \in \{0,1\}^N$, and wishes to decode the vector x to recover the defective set, subject to the constraints of the testing matrix $\mathbf{A}$. The matrix has to satisfy the property that the Boolean sum of any t columns was unique, and did not contain any other column in the matrix. These properties are known as seperability and disjunctness.

See [**du**] for more a detailed introduction and analysis of the requisite algorithms.

Hwang's Algorithm

In modern Coding Theory, there has been a move away from exlicit combinatorial algorithms which return the codeword with certainty, towards probabalisitic algorithms which return the correct codeword with an associated probability. The advantage of this has been the development of algorithms which can decode codes close to the Shannon capacity of the channel.

Similarly in Group Testing, the state of the art considers probabilistic algorithms instead of explicit combinatorial designs.

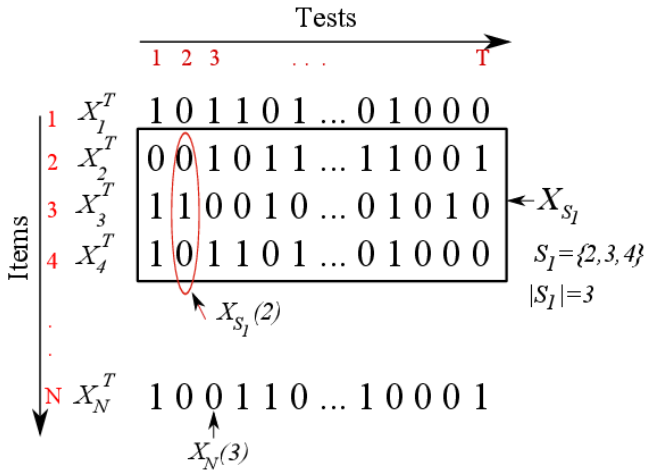

Figure 8.1: The Group Testing model: multiplication with a short, fat matrix [**atia2**]

The problems with the binary search algorithm (that initial groups are very large and are highly likely to contain a defective) above can be overcome by instead considering groups whose size is chosen so that the probability that the group will have a positive test is close to half. Equivalently, given a set of size $N = 2^r$ which are known to contain K defectives, in expectation a group of size $\frac{K}{N}$ should contain a defective.

Thus we can use fewer tests than predicted by simple repeated binary search, by testing 'pilot' groups of size roughly $\frac{K}{N}$. Hwang [**Hwang1972**] gives such an algorithm, and provides an upper bound on the number of tests required to recover the defective set.

The steps for the algorithm are:

1. If $n \leq 2d-2$ then test every item individually. Otherwise set $l = n-d+1$ and define $\alpha := \log \lceil \frac{l}{d} \rceil$.

2. Test a group of size 2^α. If the outcome is negative, the groups is good. Set $n := n-2^\alpha$ and go to 1. If the outcome is positive, then use binary splitting on the group to identify a defective and x good items. Set $n := n-1-x$ and $d := d-1$ and go to 1.

The upper bound on the number of tests is given by the following argument: as there are $\binom{n}{k}$ possible sets of defectives, and in t tests at most 2^t cases can be differentiated, $\lceil \log_2 \binom{n}{k} \rceil$ tests are needed.

Bounds

It has been previously believed that the success probability to recover the defective set given T tests was:

(8.1.1.4)
$$P(Success) \leq \frac{T}{\log_2 \binom{N}{K}}$$

However, a tighter upper bound has recently been found [**Aldridge2013**] at:

$$P(Success) \leq \frac{2^T}{\binom{N}{K}} \tag{8.1.1.5}$$

i.e. the probability of success increases exponentially with the number tests, opposed to linearly.

In the Group Testing literature there exists an 'adaptivity gap' - it seems that adaptive algorithms *do* give a performance improvement over non-adaptive algorithms, in terms of the number of tests required to recover the defective set. This is discussed here, using Hwang's algorithm as a test bed.

Hwang's algorithm is guaranteed to succeed in:

$$T = \log_2 \binom{N}{K} + K \tag{8.1.1.6}$$

tests. The Combinatorial Orthogonal Matching Pursuit algorithm, considered in [**Chan2011**], is guaranteed to recover the defective set with probability $N^{-\delta}$ in

$$T = ((1+\delta)\, e)\, K \ln N \tag{8.1.1.7}$$

tests. For all N and K) we have:

$$K \log_2 \frac{N}{K} \leq \log_2 \binom{N}{K} \leq K \log_2 \frac{Ne}{K} \tag{8.1.1.8}$$

which follows from well-know bounds on binomial coefficients. This allows a contrast between the asymptotic bounds of previous algorithms to be considered in this section. We see that, the regime where $K = N^{1-\beta}$, Hwang's algorithm succeeds with:

$$T = \beta K \log_2 N + K\left(\log_2 e + 1\right) \tag{8.1.1.9}$$

tests, whilst the COMP algorithm succeeds with:

$$1.88\,(1+\delta)\, K \log_2 N \tag{8.1.1.10}$$

tests. It's worthwhile to contrast these results, to gain some insight into the problem. 8.1.1 suggests that for very sparse problems (β tending towards 1) that Hwang's adaptive algorithm will outperform a simmilar non-adaptive algorithm. Even though the two procedures have the same complexity, they have different constants (1 v,s 1.88 in the sparse case). Thus, there are asymptotic gains (in terms of the number of tests required to recover the defective set) which are offered by adaptive algorithms, and not by non-adaptive algorithms.

These ideas can be summarised in the idea of a *capacity* for Group Testing [**Baldassini2013**]. That is, there is a constant C such that a sequence of Group Testing algorithms with $K = \omega(N)$ will succeed with probability tending to 1. This allows different noise, and dilution models to be considered so that a more complete characterisation of the structural properties of Group Testing is revealed.

Comparison to Compressive Sensing

The goal of Coding Theory is given a vector $x \in \mathbb{F}^m$, where $\mathbb{F}$ is some finite-field, is to construct a 'code-book' C which produces a vector $y \in \mathbb{F}^n$, $n > m$, so that the original vector may be transmitted over a noisy-channel with vanishing error probability. This problem is structurally similar to the Compressive Sensing and Group Testing problems, but in reverse. In CS and GT we're given 'short' vector, and we wish to infer the 'longer' one satisfying the constraint that we seek the sparsest vector, under some conditions on the matrix Φ. This suggests that there may be some Information-theoretic framework uniting all three disciplines.

In [**Emma**] Tao and Candes consider the CS problem as one of error correction of a linear code: however in this case the codewords are drawn from $\mathbb{R}^m$ as opposed to a finite alphabet more common in Coding Theory. This is done by considering Φ as the parity check matrix of a linear code and the signal x as the error pattern. Linear programming can then be viewed as a method for decoding.

Group testing is a combinatorial variant of Compressive Sensing, where the sensing matrix is a binary matrix. The matrix represents combinations (or pools) of items, such that a 1 in the i^{th} row and j^{th} column means that the i^{th} item is tested in the j^{th} pool. The goal of Group testing can then be seen as designing testing pools so to accurately reconstruct the sparse set of interesting items.

In Group Testing, instead of the sensing matrix being to subject to coherence constraints such as those above, the sensing matrices have the property that the support of any column is not contained in the union of the supports of any t other columns. Thus a t-disjunct matrix defines a group testing scheme which an identify any defective set up to size t.

The analogue between Group Testing and Coding is even closer, as GT explicitly considers signals and matrices from Binary alphabets. That is, Group Testing is a closer cousin of Coding Theory than Compressive Sensing, in a sense the inverse problem as in both Coding and Group Testing we are working over a finite field. This is encouraging, as it could allow the reconstruction of the defective set via methods developed in Coding Theory. There has been some work done on this, [**Sejdinovic2010**] considers the noisy Group Testing problem and the reconstruction of the defective set via belief propagation whilst [**Wadayama**] gives explicit theorems on conditions for the recovery of the defective set for the case of the binary symmetric channel. [**Baldassini2013**] takes this further and finds the capacity of Group Testing for a number of cases.

8.1.2 The Probabilistic group testing problem

Group testing is a sparse inference problem, first introduced by Dorfman [**dorfman**] in the context of testing for rare diseases. Given a large population of items $\mathscr{P}$, indexed by $\{1,\dots N\}$, where some small fraction of the items are interesting in some way, how can we find the interesting items efficiently?

We perform a sequence of T pooled tests defined by test sets $\mathscr{X}_1,\dots,\mathscr{X}_T$, where each $\mathscr{X}_i \subseteq \mathscr{P}$. We represent the interesting ('defective') items by a random vector $\mathbf{U} = (U_1,\dots,U_N)$, where U_i is the indicator of the event that item i is defective. For each test i, we jointly test all the items in $\mathscr{X}_i$, and the outcome y_i is 'positive' ($y_i = 1$) if and only if any item in $\mathscr{X}_i$ is defective. In other words, $y_i = \mathbb{I}\left(\sum_{j\in\mathscr{X}_i} U_j\right)$, since for simplicity we are considering the noiseless case. Further, in this paper, we restrict our attention to the adaptive case, where we choose test set $\mathscr{X}_i$ based on a knowledge of sets $\mathscr{X}_1,\dots,\mathscr{X}_{i-1}$ and outcomes $y_1,\dots,y_{i-1}$. The group testing problem requires us to infer $\mathbf{U}$ with high probability given a low number of tests T.

Since Dorfman's paper [**dorfman**], there has been considerable work on the question of how to design the sets $\mathscr{X}_i$ in order to minimise the number of tests T required. In this context, we briefly mention so-called combinatorial designs (see [**du**, **malyutov**] for a summary, with [**malyutov**] giving invaluable references to an extensive body of Russian work in the 1970s and 1980s). Such designs typically aim to ensure that set-theoretic properties known as disjunctness and separability occur. In contrast, for simplicity of analysis, as well as performance of optimal order, it is possible to consider random designs. Here sets $\mathscr{X}_i$ are chosen at random, either using constructions such as independent Bernoulli designs [**atia**, **johnsonc8**, **johnson33**] or more sophisticated random designs based on LDPC codes [**Wadayama**].

Much previous work has focussed on the Combinatorial group testing problem, where there are a fixed number of defectives K, and the defectivity vector $\mathbf{U}$ is chosen uniformly among all binary vectors of weight K. In contrast, in this paper we study a Probabilistic group testing problem as formulated for example in the work of Li et al. [**li5**], in that we suppose each item is defective independently with probability p_i, or equivalently take U_i to be independent Bernoulli(p_i).

This Probabilistic framework, including non-uniform priors, is natural for many applications of group testing. For example, see [**atia2**], the cognitive radio problem can be formulated in terms of a population of communication bands in frequency spectra with some (unknown) occupied bands you must not utilise. Here, the values of p_i may be chosen based on some database of past spectrum measurements or other prior information. Similarly, as in Dorfman's original work [**dorfman**] or more recent research [**shental**] involving screening for genetic conditions, values of p_i might summarise prior information based on a risk profile or family history.

8.1.3 Group testing capacity

It is possible to characterize performance tradeoffs in group testing from an information-theoretic point of view – see for example [**atia**, **johnsonc10**, **johnson33**, **tan**]. These papers have focussed on group testing as a channel coding problem, with [**atia**, **tan**] explicitly calculating the mutual information. The paper [**johnsonc10**] defined the capacity of a Combinatorial group testing procedure, which characterizes the number of bits of information about the defective set which we can learn per test. We give a more general definition here, which covers both the Combinatorial and Probabilistic cases.

Definition 8.1.1. *Consider a sequence of group testing problems where the i th problem has defectivity vector $\mathbf{U}^{(i)}$, and consider algorithms which are given $T(i)$ tests. We refer to a constant C as the (weak) group testing capacity if for any $\epsilon > 0$:*

1. *any sequence of algorithms with*

$$\liminf_{i\to\infty} \frac{H(\mathbf{U}^{(i)})}{T(i)} \geq C + \epsilon, \tag{8.1.3.11}$$

   has success probability $\mathbb{P}(suc)$ bounded away from 1,

2. *and there exists a sequence of algorithms with*

$$\liminf_{i\to\infty} \frac{H(\mathbf{U}^{(i)})}{T(i)} \geq C - \epsilon \tag{8.1.3.12}$$

   with success probability $\mathbb{P}(\text{suc}) \to 1$.

Remark 8.1.2. *In the Combinatorial case of K defective items with all defective sets equally likely, $H(\mathbf{U}) = \log_2 \binom{N}{K}$, which is the term found in the denominator in [***johnsonc10** *]. In the Probabilistic case (as in [***li5** *]) we know $H(\mathbf{U}) = -\sum_{i=1}^{N} h(p_i)$ where $h(t) = -t\log_2 t - (1-t)\log_2(1-t)$ is the binary entropy function.*

Remark 8.1.3. *If for $\liminf_{i\to\infty} \frac{H(\mathbf{U}^{(i)})}{T(i)} \geq C + \epsilon$, the success probability $\mathbb{P}(\text{suc}) \to 0$ we say that C is the strong group testing capacity, following standard terminology in information theory. Such a result is referred to as a strong converse.*

8.1.4 Main results

The principal contribution of [**johnsonc10**] was the following result:

Theorem 8.1.4 ([**johnsonc10**])**.** *The strong capacity of the adaptive noiseless Combinatorial group testing problem is $C = 1$, in any regime such that $K/N \to 0$.*

This argument came in two parts. First, in [**johnsonc10**] the authors proved a new upper bound on success probability

$$\mathbb{P}(\text{suc}) \leq \frac{2^T}{\binom{N}{K}}, \tag{8.1.4.13}$$

which implied a strong converse ($C \leq 1$). This was complemented by showing that, in the Combinatorial case, an algorithm based on Hwang's Generalized Binary Splitting Algorithm (HGBSA) [**hwang**], [**du**] is essentially optimal in the required sense, showing that $C = 1$ is achievable.

It may be useful to characterize the Probabilistic group testing problem in terms of the effective sparsity $\mu^{(N)} := \sum_{i=1}^{N} p_i$. In particular, if the p_i are (close to) identical, we would expect performance similar to that in the Combinatorial case with $K = \mu^{(N)}$ defectives. As in [**johnsonc10**], we focus on asymptotically sparse cases, where $\mu^{(N)}/N \to 0$ (in contrast, Wadayama [] considered a model where p_i are identical and fixed). The main result of the present paper is Theorem 8.3.9, stated and proved in Section 8.3.5 below, which implies the following Probabilistic group testing version of Theorem 8.1.4.

Corollary 8.1.5. *In the case where $p_i \equiv p$, the weak capacity of the adaptive noiseless Probabilistic group testing problem is $C = 1$, in any regime such that $\mu^{(N)}/N \to 0$ and $\mu^{(N)} \to \infty$.*

Again we prove our main result Theorem 8.3.9 using complementary bounds on both sides. First in Section 8.2.1 we recall a universal upper bound on success probability, Theorem 8.2.1, taken from [**li5**], which implies a weak converse. In [**li5**], Li et al. introduce the Laminar Algorithm for Probabilistic group testing. In Section 8.2.3 we propose a refined version of this Laminar Algorithm, based on Hwang's HGBSA [**hwang**], which is analysed in Section 8.3.5, and shown to imply performance close to optimal in the sense of capacity.

8.2 Algorithms and existing results

8.2.1 Upper bounds on success probability

Firstly [**li5**] can be restated to give the following upper bound on success probability:

Theorem 8.2.1. *Any Probabilistic group testing algorithm using T tests with noiseless measurements has success probability satisfying*

$$\mathbb{P}(\text{suc}) \leq \frac{T}{H(\mathbf{U})}.$$

Rephrased in terms of Definition 8.1.1, this tells us that the weak capacity of noiseless Probabilistic group testing is ≤ 1. The logic is as follows; if the capacity were $1+2\epsilon$ for some $\epsilon > 0$, then there would exist a sequence of algorithms with $H(\mathbf{U}^{(i)})/T(i) \geq 1+\epsilon$ with success probability tending to 1. However, by Theorem 8.2.1, any such algorithms have $\mathbb{P}(\text{suc}) \leq 1/(1+\epsilon)$, meaning that we have established that a weak converse holds.

Remark 8.2.2. *It remains an open and interesting problem to prove an equivalent of* (8.1.4.13) *as in [***johnsonc10** *]. That is we hope to find an upper bound on success probability in a form which implies a strong converse, and hence that the strong capacity of Probabilistic group testing is equal to 1.*

8.2.2 Binary search algorithms

The main contribution of this work is to describe and analyse algorithms that will find the defective items. In brief, we can think of Hwang's HGBSA algorithm as dividing the population $\mathscr{P}$ into search sets $\mathscr{S}$. First, all the items in a search set $\mathscr{S}$ are tested together, using a test set $\mathscr{X}_1 = \mathscr{S}$. If the result is negative ($y_1 = 0$), we can be certain that $\mathscr{S}$ contains no defectives. However, if the result is positive ($y_1 = 1$), $\mathscr{S}$ must contain at least one defective.

If $y_i = 1$, we can be guaranteed to find at least one defective, using the following binary search strategy. We split the set $\mathscr{S}$ in two, and test the 'left-hand' set, say $\mathscr{X}_2$. If $y_2 = 1$, then we know that $\mathscr{X}_2$ contains at least one defective. If $y_2 = 0$, then $\mathscr{X}_2$ contains no defective, so we can deduce that $\mathscr{S} \setminus \mathscr{X}_2$ contains at least one defective. By repeated use of this strategy, we are guaranteed to find a succession of nested sets which contain at least one defective, until $\mathscr{X}_i$ is of size 1, and we have isolated a single defective item.

However this strategy may not find every defective item in $\mathscr{S}$. To be specific, it is possible that at some stage both the left-hand and right-hand sets contain a defective. The Laminar Algorithm of [**li5**] essentially deals with this by testing both sets. However, we believe that this is inefficient, since typically both sets will not contain a defective. Nonetheless, the Laminar Algorithm satisfies the following performance guarantees proved in [**li5**]:

Theorem 8.2.3. *The expected number of tests required by the Laminar Algorithm [***li5** *] is $\leq 2H(\mathbf{U})+2\mu$. Under a technical condition (referred to as non-skewedness), the success probability can be bounded by $\mathbb{P}(\text{suc}) \geq 1-\epsilon$ using $T = (1+\delta)(2^{\Gamma+\log_2 3}+2)H(\mathbf{U})$ tests, where Γ is defined implicitly in terms of ϵ, and $\delta \geq 2e-1$.*

Ignoring the Γ term, and assuming the non-skewedness condition holds, this implies that (using the methods of [**li5**]) $T = 2e(3+2)H(\mathbf{U}) = 10eH(\mathbf{U})$ tests are required to guarantee convergence to 1 of the success probability. In our language, this implies a lower bound of $C \geq 1/(10e) = 0.0368$. Even ignoring the analysis of error probability, the fact that the expected number of tests is $\leq 2H(\mathbf{U})+2\mu$ suggests that we cannot hope to achieve $C > 1/2$ using the Laminar Algorithm.

A Set S of $|S| = n$ items, μ of which are actually defective in expectation, a probability vector $\mathbf{p}^{(n)}$ describing each item's independent probability of being defective, and a cutoff θ
ReturnsThe set of defective items Discard items with $p_i \leq \theta$
Sort the remaining items into B bins, collecting items together with $p_i \in \left[1/2C^r, 1/2C^{r-1}\right)$ in bin r.
Sort the items in each bin into sets s.t. the (normalised) probability of each set is less than 1/2.
Test each set in turn

if The test is positive **then**Arrange the items in the set on a Shannon-Fano/Huffman Tree and search the set for all the defectives it contains
end if

Figure 8.2: Algorithm for the non-iid group testing problem

8.2.3 Summary of our contribution

The main contribution of our paper is a refined version of the Laminar Algorithm, summarised above, and an analysis resulting in tighter error bounds as formulated in Proposition 8.3.7 (in terms of expected number of tests) and Theorem 8.3.9 (in terms of error probabilities). The key ideas are:

1. To partition the population $\mathscr{P}$ into search sets $\mathscr{S}$ containing items which have similar probabilities, expressed through the Bounded Ratio Condition 6. This is discussed in Section 8.3.1, and optimised in the proof of Proposition 8.3.7.

2. The way in which we deal with sets $\mathscr{S}$ which contain more than one defective, as discussed in Remark 8.3.2 below. Essentially we do not backtrack after each test by testing both left- and right-hand sets, but only backtrack after each defective is found.

3. To discard items which have probability below a certain threshold, since with high probability none of them will be defective. This is an idea introduced in [**li5**] and discussed in Section 8.3.2, with a new bound given in Lemma 8.3.4.

4. Careful analysis in Section 8.3.4 of the properties of search sets $\mathscr{S}$ gives Proposition 8.3.7, which shows that the expected number of tests required can be expressed as $H(\mathbf{U})$ plus an error term. In Section 8.3.5, we give an analysis of the error probability using Bernstein's inequality, Theorem 8.3.8, allowing us to prove Theorem 8.3.9.

8.2.4 Wider context: sparse inference problems

Recent work [**aksoylar**, **tan**] has shown that many arguments and bounds hold in a common framework of sparse inference which includes group testing and compressive sensing.

Digital communications, audio, images, and text are examples of data sources we can compress. We can do this, because these data sources are sparse: they have fewer degrees of freedom than the space they are defined upon. For example, images have a well known expansion in either the Fourier or Wavelet bases. The text of an English document will only be comprised of words from the English dictionary, and not all the possible strings from the space of strings made up from the characters $\{a, \ldots, z\}$.

Often, once a signal has been acquired it will be compressed. However, the compressive sensing paradigm introduced by [**Candes2006**, **donoho2**] shows that this isn't necessary. In those papers it

was shown that a 'compressed' representation of a signal could be obtained from random linear projections of the signal and some other basis (for example White Gaussian Noise). The question remains, given this representation how do we recover the original signal? For real signals, a simple linear programme suffices. Much of the work in this area has been couched in terms of the sparsity of the signal and the various bases the signal can be represented in (see for example [**Candes2006**, **donoho2**]).

8.3 Analysis and new bounds

8.3.1 Searching a set of bounded ratio

Recall that we have a population $\mathscr{P}$ of items to test, each with associated probability of defectiveness p_i. The strategy of the proof is to partition $\mathscr{P}$ into search sets $\mathscr{S}_1, \ldots, \mathscr{S}_G$, each of which contains items which have comparable values of p_i.

Condition 6 (Bounded Ratio Condition)**.** *Given $C \geq 1$, say that a set $\mathscr{S}$ satisfies the Bounded Ratio Condition with constant C if*

$$\max_{i,j \in \mathscr{S}} \frac{p_j}{p_i} \leq C. \tag{8.3.1.14}$$

(For example clearly if $p_i \equiv p$, any set $\mathscr{S}$ satisfies the condition for any $C \geq 1$).

Lemma 8.3.1. *Consider a set $\mathscr{S}$ satisfying the Bounded Ratio Condition with constant C and write $P_{\mathscr{S}} = \sum_{j \in \mathscr{S}} p_j$. In a Shannon–Fano tree for the probability distribution $\overline{p}_i := p_i / P_{\mathscr{S}}$, each item has length $\ell_i^{(\mathscr{S})}$ bounded by*

$$\ell_i^{(\mathscr{S})} \leq \ell_{\max}^{(\mathscr{S})} := \frac{h(\mathscr{S})}{P_{\mathscr{S}}} + \log_2 C + \log_2 P_{\mathscr{S}} + 1, \tag{8.3.1.15}$$

where we write $h(\mathscr{S}) := -\sum_{j \in \mathscr{S}} p_j \log_2 p_j$.

Proof. Under the Bounded Ratio Condition, for any i and j, we know that by taking logs of (8.3.1.14)

$$-\log_2 p_i \leq -\log_2 p_j + \log_2 C.$$

Multiplying by p_j and summing over all $j \in \mathscr{S}$, we obtain that

$$-P_{\mathscr{S}} \log_2 p_i \leq h(\mathscr{S}) + P_{\mathscr{S}} \log_2 C. \tag{8.3.1.16}$$

Now, the Shannon–Fano length of the ith item is

$$\begin{aligned} \ell_i^{(\mathscr{S})} = \lceil -\log_2 \overline{p}_i \rceil &\leq -\log_2 p_i + \log_2 P_{\mathscr{S}} + 1 \\ &\leq \left(\frac{h(\mathscr{S})}{P_{\mathscr{S}}} + \log_2 C \right) + \log_2 P_{\mathscr{S}} + 1. \end{aligned} \tag{8.3.1.17}$$

and the result follows by (8.3.1.16). □

Next we describe our search strategy:

Remark 8.3.2. *Our version of the algorithm will find every defective in a set $\mathscr{S}$. We start as before by testing every item in $\mathscr{S}$ together. If this test is negative, we are done. Otherwise, if it is positive, we can perform binary search as above to find one defective item, say d_1. Now, test every item in $\mathscr{S} \setminus \{d_1\}$ together. If this test is negative, we are done, otherwise we repeat the search step on this smaller set, to find another defective item d_2, then we test $\mathscr{S} \setminus \{d_1, d_2\}$ and so on.*

We think of the algorithm as repeatedly searching a binary tree. Clearly, if the tree has depth bounded by ℓ, then the search will take $\leq \ell$ tests to find one defective. In total, if the set contains U defectives, we need to repeat U rounds of searching, plus the final test to guarantee that the set contains no more defectives, so will use $\leq \ell U + 1$ tests.

Lemma 8.3.3. *Consider a search set $\mathscr{S}$ satisfying the Bounded Ratio Condition and write $P_{\mathscr{S}} = \sum_{j\in\mathscr{S}} p_j$. If (independently) item i is defective with probability p_i, we can recover all defective items in the set using $T_{\mathscr{S}}$ tests, where $\mathbb{E}\, T_{\mathscr{S}} \leq T_{\mathrm{bd}}(\mathscr{S})$ for*

$$T_{\mathrm{bd}}(\mathscr{S}) := h(\mathscr{S}) + P_{\mathscr{S}} \log_2 C + P_{\mathscr{S}} \log_2 P_{\mathscr{S}} + P_{\mathscr{S}} + 1. \tag{8.3.1.18}$$

Proof. Using the algorithm of Remark 8.3.2, laid out on the Shannon-Fano tree constructed in Lemma 8.3.1, we are guaranteed to find every defective. The number of tests to find one defective thus corresponds to the depth of the tree, which is bounded by $\ell_{\max}^{(\mathscr{S})}$ given in (8.3.1.15).

Recall that we write U_i for the indicator of the event that the ith item is defective, $U_{\mathscr{S}} = \sum_{i\in\mathscr{S}} U_i$ and $l_i^{(\mathscr{S})}$ for the length of the word in the Shannon Fano tree. As discussed in Remark 8.3.2 this search procedure will take

$$\begin{aligned} T_{\mathscr{S}} &= 1 + \sum_{i\in\mathscr{S}} U_i \ell_i^{(\mathscr{S})} \\ &= \sum_{i\in\mathscr{S}} p_i \ell_i^{(\mathscr{S})} + 1 + \sum_{i\in S} \ell_i^{(\mathscr{S})} (U_i - p_i) \\ &\leq \sum_{i\in\mathscr{S}} p_i \ell_{\max}^{(\mathscr{S})} + 1 + \sum_{i\in\mathscr{S}} V_i^{(\mathscr{S})} \\ &= P_{\mathscr{S}} \ell_{\max}^{(\mathscr{S})} + 1 + \sum_{i\in S} V_i^{(\mathscr{S})} \\ &\leq T_{\mathrm{bd}}(\mathscr{S}) + \sum_{i\in S} V_i^{(\mathscr{S})} \quad \text{tests.} \end{aligned} \tag{8.3.1.19}$$

Here we write $V_i^{(\mathscr{S})} = \ell_i^{(\mathscr{S})}(U_i - p_i)$, which has expectation zero, and (8.3.1.19) follows using the expression for $\ell_{\max}^{(\mathscr{S})}$ given in Lemma 8.3.1. □

8.3.2 Discarding low probability items

As in [**li5**], we use a probability threshold θ, and write $\mathscr{P}^*$ for the population having removed items with $p_i \leq \theta$. If an item lies in $\mathscr{P} \setminus \mathscr{P}^*$ we do not test it, and simply mark it as non-defective. This truncation operation gives an error if and only if some item in $\mathscr{P} \setminus \mathscr{P}^*$ is defective. By the union bound, this truncation operation contributes a total of $\mathbb{P}(\mathscr{P} \setminus \mathscr{P}^* \text{ contains a defective}) \leq \rho := \sum_{i=1}^n p_i \mathbb{I}(p_i \leq \theta)$ to the error probability.

Lemma 8.3.4. *Choosing $\theta(P_e)$ such that*

$$-\log_2 \theta(P_e) = \min\left(\log_2\left(\frac{2n}{P_e}\right), \frac{2H(\mathbf{U})}{P_e} \right) \tag{8.3.2.20}$$

ensures that

$$\mathbb{P}(\,\mathscr{P} \setminus \mathscr{P}^* \textit{ contains a defective}) \leq P_e/2. \tag{8.3.2.21}$$

Proof. The approach of [**li5**] is essentially to bound $\mathbb{I}(p_i \le \theta) \le \theta / p_i$ so that $\rho = \sum_{i=1}^n p_i \mathbb{I}(p_i \le \theta) \le \sum_{i=1}^n p_i(\theta/p_i) = n\theta$. Hence, choosing a threshold of $\theta = P_e/(2n)$ guarantees the required bound on ρ.

We combine this with another bound, constructed using a different function: $\mathbb{I}(p_i \le \theta) \le (-\log_2 p_i)/(-\log_2 \theta)$, so that

$$\rho = \sum_{i=1}^n p_i \mathbb{I}(p_i \le \theta) \le \sum_{i=1}^n p_i \left(\frac{-\log_2 p_i}{-\log_2 \theta} \right) \le \frac{H(\mathbf{U})}{-\log_2 \theta},$$

so we deduce the result. □

8.3.3 Searching the entire set

Having discarded items with p_i below this probability threshold θ and given bounding ratio C, we create a series of bins. We collect together items with probabilities $p \in [1/2, 1]$ in bin 0, $p \in [1/(2C), 1/2)$ in bin 1, items with probabilities $p \in [1/(2C^2), 1/(2C))$ in bin 2, ..., and items with probabilities $p \in [1/(2C^B), 1/(2C^{B-1}))$ in bin B.

The probability threshold θ means that there will be a finite number of such bins, with the index B of the last bin defined by the fact that $1/(2C^B) \le \theta < 1/(2C^{B-1})$, meaning that $(B-1)\log_2 C < -\log_2(2\theta)$, so

$$B \le \frac{-\log_2(2\theta)}{\log_2 C} + 1. \tag{8.3.3.22}$$

We split the items in each bin into search sets $\mathcal{S}_i$, motivated by the following definition:

Definition 8.3.5. *A set of items $\mathcal{S}$ is said to be full if $P_{\mathcal{S}} = \sum_{i \in \mathcal{S}} p_i \ge \frac{1}{2}$.*

Our splitting procedure is as follows: we create a list of possible sets $\mathcal{S}_1, \mathcal{S}_2, \ldots$. For i increasing from 0 to B, we place items from bin i into sets $\mathcal{S}_{b_i+1}, \ldots, \mathcal{S}_{b_{i+1}}$, for some b_i, where $b_0 = 0$. Taking the items from bin i, while $\mathcal{S}_{b_i+1}$ is not full (has total probability $< \frac{1}{2}$) we will place items into it. Once enough items have been added to fill $\mathcal{S}_{b_i+1}$, we will proceed in the same way to fill $\mathcal{S}_{b_i+2}$, and so on until all the items in bin i have been assigned to sets $\mathcal{S}_{b_i+1}, \ldots, \mathcal{S}_{b_{i+1}}$, where $\mathcal{S}_{b_{i+1}}$ may remain not full.

Proposition 8.3.6. *This splitting procedure will divide $\mathcal{P}^*$ into search sets $\mathcal{S}_1, \ldots, \mathcal{S}_G$, where the total number of sets is*

$$G \le 2\mu + B \le 2\mu + \left(\frac{-\log_2(2\theta)}{\log_2 C} + 1 \right).$$

Each set $\mathcal{S}_j$ satisfies the Bounded Ratio Condition and has total probability $P_j := P_{\mathcal{S}_j} \le 1$.

Proof. First, note that the items from bin 0 each lie in a set $\mathcal{S}$ on their own. These sets will be full, trivially satisfy the Bounded Ratio Condition 6 with constant C, and have probability satisfying $P_j \le 1$. For each of bins $1, \ldots, B$:

1. For each bin i, it is possible that the last set $\mathcal{S}_{b_i+1}$ will not be full, but every other set corresponding to that bin will be full. Hence, there are no more than B sets which are not full.

2. For each resulting set $\mathcal{S}_j$, the total probability $P_j \le 1$ (since just before we add the final item, $\mathcal{S}_j$ is not full, so at that stage has total probability $\le 1/2$, and each element in bins $1, \ldots, B$ has probability $\le 1/2$).

3. Since each set $\mathscr{S}_j$ contains items taken from the same bin, it will satisfy the Bounded Ratio Condition with constant C.

Note that the number of full sets is $\leq 2\mu$, since

$$\mu = \sum_{i\in\mathscr{P}} p_i \geq \sum_{i\in\mathscr{P}^*} p_i = \sum_{j=1}^{G} P_j \geq \sum_{j:\,\mathscr{S}_j \text{ full}} P_j \geq \left|\mathscr{S}_j \text{ full}\right| \frac{1}{2}. \tag{8.3.3.23}$$

Since, as discussed in point 1) above, the total number of sets is bounded by the number of full sets plus B, the result follows using Equation (8.3.3.22). □

8.3.4 Bounding the expected number of tests

We allow the algorithm to work until all defectives in $\mathscr{P}^*$ are found, and write T for the (random) number of tests this takes.

Proposition 8.3.7. *Given a population $\mathscr{P}$ where (independently) item i is defective with probability p_i, we recover all defective items in $\mathscr{P}^*$ in T tests with $\mathbb{E}T \leq T_{\text{bd}}$, where*

$$T_{\text{bd}} := \big(H(\mathbf{U}) + 3\mu + 1\big) + 2\sqrt{\mu\big(-\log_2(2\theta)\big)}. \tag{8.3.4.24}$$

Proof. Given a value of C, Proposition 8.3.6 shows that our splitting procedure divides $\mathscr{P}^*$ into G sets $\mathscr{S}_1,\ldots,\mathscr{S}_G$, such that each set $\mathscr{S}_j$ satisfies the Bounded Ratio Condition with constant C and has total probability $P_j \leq 1$. Using the notation of Lemma 8.3.3, $T = \sum_{j=1}^{G} T_{\mathscr{S}_j}$, where $\mathbb{E}T_{\mathscr{S}_j} \leq T_{\text{bd}}(\mathscr{S}_j)$.

Adding this bound over the different sets, since $P_j \leq 1$ means that $P_j \log_2 P_j \leq 0$, we obtain

$$\begin{aligned}
&\sum_{j=1}^{G} T_{\text{bd}}(\mathscr{S}_j) \\
&\quad\leq \sum_{j=1}^{G} \big(h(\mathscr{S}_j) + P_j(\log_2 C + 1) + 1\big) \\
&\quad= \sum_{j\in\mathscr{P}^*} -p_j \log_2 p_j + \mu(\log_2 C + 1) + G \\
&\quad\leq \sum_{j\in\mathscr{P}^*} h(p_j) + 3\mu + 1 + \left(\frac{-\log_2(2\theta)}{\log_2 C} + \mu\log_2 C\right) \\
&\quad\leq \big(H(\mathbf{U}) + 3\mu + 1\big) + \left(\frac{-\log_2(2\theta)}{\log_2 C} + \mu\log_2 C\right). \qquad (8.3.4.25)
\end{aligned}$$

This follows by the bound on G in Proposition 8.3.6, as well as the fact that $0 \leq p_j \leq 1$ means that for any i, $-p_j\log_2 p_j = (1-p_j)\log_2(1-p_j) + h(p_j) \leq h(p_j)$.

Finally, we choose $C > 1$ to optimize the second bracketed term in Equation (8.3.4.25). Differentiation shows that the optimal C satisfies $\log_2 C = \sqrt{-\log_2(2\theta)/\mu}$, meaning that the bracketed term

$$\frac{-\log_2(2\theta)}{\log_2 C} + \mu\log_2 C = 2\sqrt{\mu\big(-\log_2(2\theta)\big)},$$

and the result follows. □

8.3.5 Controlling the error probabilities

Although Section 8.3.4 proves that $\mathbb{E}T \le T_{\rm bd}$, to bound the capacity, we need to prove that with high probability T is not significantly larger than $T_{\rm bd}$. This can be done using Bernstein's inequality (see for example Theorem 2.8 of [**petrov**]):

Theorem 8.3.8 (Bernstein)**.** *For zero-mean random variables V_i which are uniformly bounded by $|V_i| \le M$, if we write $L := \sum_{j=1}^n \mathbb{E}V_j^2$ then*

$$\mathbb{P}\left(\sum_{j=1}^n V_j \ge t\right) \le \exp\left(-\frac{t^2}{4L}\right), \text{ for any } 0 \le t \le \tfrac{L}{M}. \tag{8.3.5.26}$$

We deduce the following result:

Theorem 8.3.9. *Write $L = \sum_{j\in\mathscr{P}^*} l_j^2 p_j(1-p_j)$, $M = -\log_2\theta + 1$ and $\psi = (L/(4M^2))^{-1/3}$. Define*

$$T_{\rm nec} = T_{\rm bd} + \psi H(\mathbf{U}), \tag{8.3.5.27}$$

where $T_{\rm bd}$ is given in (8.3.4.24)*.*

1. *If we terminate our group testing algorithm after $T_{\rm nec}$ tests, the success probability*

$$\mathbb{P}(\text{suc}) \ge 1 - \frac{1}{2}\sqrt{\frac{\mu}{H(\mathbf{U})}} - \exp\left(-\left(\frac{L}{4M^2}\right)^{1/3}\right). \tag{8.3.5.28}$$

2. *Hence in any regime where $\mu \to \infty$ with $\mu/H(\mathbf{U}) \to 0$ and $L/M^2 \to \infty$, our group testing algorithm has (a) $\liminf H(\mathbf{U})/T_{\rm nec} \ge 1/(1+\epsilon)$ for any ϵ and (b) $\mathbb{P}(\text{suc}) \to 1$, so the capacity $C = 1$.*

Proof. We first prove the success probability bound (8.3.5.28). Recall that our algorithm searches the reduced population set $\mathscr{P}^*$ for defectives. This gives two error events – either there are defective items in the set $\mathscr{P}\setminus\mathscr{P}^*$, or the algorithm does not find all the defectives in $\mathscr{P}^*$ using $T_{\rm nec}$ tests. We consider them separately, and control the probability of either happening using the union bound.

Writing $H = H(\mathbf{U})$ for brevity and choosing $P_e = \sqrt{\mu/H}$ ensures that (by Lemma 8.3.4) the first event has probability $\le P_e/2$, contributing $\frac{1}{2}\sqrt{\mu/H(\mathbf{U})}$ to (8.3.5.28).

Our analysis of the second error event is based on the random term from Equation (8.3.1.19), which we previously averaged over but now wish to bound. There will be an error if $T_{\rm nec} \le T$, or (rearranging) if

$$\psi H \le T - T_{\rm bd} \le \sum_{j=1}^G \left(T_{\mathscr{S}_j} - T_{\rm bd}(\mathscr{S}_j)\right) = \sum_{i\in\mathscr{P}^*} V_i.$$

For brevity, for $i \in \mathscr{S}$, we write $V_i = V_i^{(\mathscr{S})} = \ell_i^{(\mathscr{S})}(U_i - p_i)$ and $\ell_i = \ell_i^{(\mathscr{S})}$, where V_i has expectation zero.

We have discarded elements with probability below θ, as given by (8.3.2.20), and by design all the sets $\mathscr{S}$ have total probability $P_{\mathscr{S}} \le 1$. Using (8.3.1.17) we know that the V_i are bounded by

$$|V_i| \le \ell_i \le -\log_2 p_i + \log_2 P_{\mathscr{S}} + 1 \le -\log_2\theta + 1. \tag{8.3.5.29}$$

Hence, the conditions of Bernstein's inequality, Theorem 8.3.8, are satisfied. Observe that since all $l_j \le M$,

$$\frac{L}{HM} = \frac{\sum_{j\in\mathscr{P}^*} l_j^2 p_j(1-p_j)}{HM} \le \frac{\sum_{j\in\mathscr{P}^*} l_j p_j}{H} \le 1.$$

Hence Theorem 8.3.8 gives that

$$\begin{aligned}\mathbb{P}\left(\sum_{j\in\mathscr{D}^*} V_j \geq \psi H\right) &\leq \mathbb{P}\left(\sum_{j\in\mathscr{D}^*} V_j \geq \psi L/M\right)\\ &\leq \exp\left(-\frac{L\psi^2}{4M^2}\right)\\ &= \exp\left(-\left(\frac{L}{4M^2}\right)^{1/3}\right).\end{aligned}$$

Using the union bound, the probability bound (8.3.5.28) follows.

We next consider the capacity bound of 2). Since $-\log_2\theta \leq 2H/P_e$, using (8.3.4.24) and (8.3.5.27)

$$\begin{aligned}\frac{T_{\text{nec}}}{H} &= \frac{T_{\text{bd}}}{H}+\psi\\ &= 1+3\frac{\mu}{H}+\frac{1}{H}+2\sqrt{\frac{\mu}{HP_e}}+\psi\\ &= 1+3\frac{\mu}{H}+\frac{1}{H}+2\left(\frac{\mu}{H}\right)^{1/4}+\psi, \qquad (8.3.5.30)\end{aligned}$$

which in our regime of interest is $\leq 1+\epsilon$ in the limit. □

Proof of Corollary 8.1.5. In the case where all p are identical, $\mu = Np$, $H = Np(-\log p)$, so $\mu/H = 1/(-\log p) \to 0$. Similarly, $L = Np(-\log_2 p)^2$ and $M = (-\log_2 p)$ so that $L/M^2 = Np \to \infty$ as required. □

8.4 Results

The performance of the Algorithm 1 (in terms of the sample complexity) was analysed by simulating 500 items, with a mean number of defectives equal to 8. I.e. $N = 500$ and $\mu^{(N)} = 8$.

The probability distribution $\mathbf{p}$ was generated by a Dirichlet distribution with parameter α. This produces output which can be made more or less uniform, as opposed to simply choosing a set of random numbers and normalise by the sum. Consider the case of two random numbers, (x, y), distributed uniformly on the square $[0,1]^2$. Normalising by the sum $(x+y)$ projects the point (x, y) onto the line $x+y=1$ and so favours points closer to $(0.5, 0.5)$ than the endpoints. The Dirichlet distribution avoids this by generating points directly on the simplex.

We then chose values of the cutoff parameter θ from 0.0001 to 0.01, and for each θ_i simulated the algorithm 1000 times. We plot the empirical distribution of tests, varying theta as well as the uniformity/concentration of the probability distribution (via the parameter α of the Dirichlet distribution). We also plot, the theoretical lower and upper bounds on the number of Tests required for successful recovery alongside the empirical number tests (all as a function of θ).

Note that the Upper bound is not optimal and there still is some room for improvement. Note also that the lower bound degrades with θ_i. The lower bound (T_{LCHJ}) was generated according to theorem (8.2.1).

Figures (8.4) and (8.5) show that the performance is relatively insensitive to the cut-off θ, and more sensitive to the uniformity (or otherwise) of the probability distribution $\mathbf{p}$. Heuristically, this is for because distributions which are highly concentrated on a few items algorithms can make substantial savings on the testing budget by testing those highly likely items first (which is captured in the bin structure of the above algorithm).

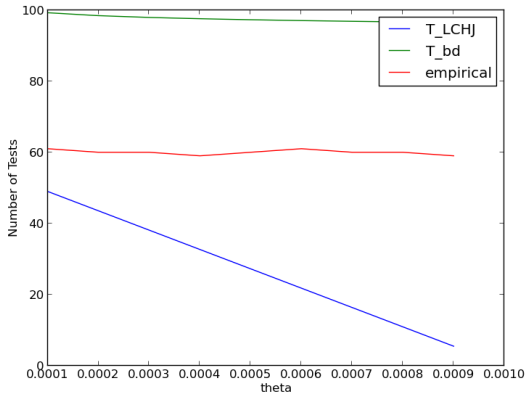

Figure 8.3: Theoretical lower and upper bounds and empirical Test frequencies as functions of θ

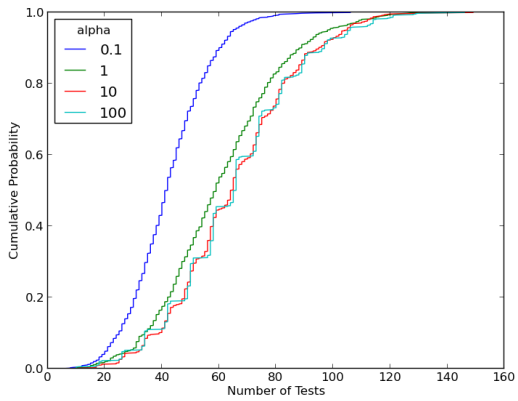

Figure 8.4: Cumulative distribution curves of the modified Hwang algorithm with fixed $\theta = 0.0001$ and α varying

The insensitivity to the cutoff θ is due to items below θ being overwhelmingly unlikely to be defective - which for small θ means that few items (relative to the size of the problem) get discarded.

8.5 Discussion

We have introduced and analysed an algorithm for Probabilistic group testing which uses 'just over' $H(\mathbf{U})$ tests to recover all the defectives with high probability. Combined with a weak converse taken from [**li5**], this allows us to deduce that the weak capacity of Probabilistic group testing is $C = 1$. These results are illustrated by simulation.

For simplicity, this work has concentrated on establishing a bound T_{bd} in (8.3.4.24) which has leading term $H(\mathbf{U})$, and not on tightening bounds on the coefficient of μ in (8.3.4.24). For completeness, we mention that this coefficient can be reduced from 3, under a simple further condition:

Remark 8.5.1. *For some $c \leq 1/2$, we assume that all the $p_i \leq c$, and we alter the definition of*

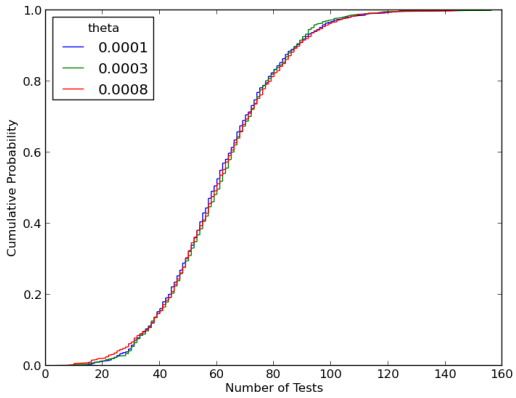

Figure 8.5: Cumulative distribution curves for fixed $\alpha = 1$ and varying θ

'fullness' to assume that a set is full if it has total probability less than α*. In this case, the term* $P_{\mathscr{S}} \log_2 P_{\mathscr{S}}$ *in* (8.3.1.18) *becomes* $P_{\mathscr{S}} \log_2(\alpha + c)$*, the bound in* (8.3.3.23) *becomes* μ/α*, and since* $\big((1-p)\log_2(1-p)\big)/p$ *is decreasing in* p*, we can add a term* $(1-c)\log_2(1-c)$ *to* (8.3.4.25)*. Overall, the coefficient of* μ *becomes* $f(a,c) := \log_2(\alpha + c) + 1 + 1/\alpha + (1-c)\log_2(1-c)$*, which we can optimize over* α*. For example, if* $c = 1/4$*, taking* $\alpha = 0.88824$*, we obtain* $f(a,c) = 2.00135$*.*

It remains of interest to tighten the upper bound of Theorem 8.2.1, in order prove a strong converse, and hence confirm that the strong capacity is also equal to 1.

In future work, we hope to explore more realistic models of defectivity, such as those where the defectivity of U_i are not necessarily independent, for example by imposing a Markov neighbourhood structure.

Acknowledgments

This work was supported by the Engineering and Physical Sciences Research Council [grant number EP/I028153/1]; Ofcom; and the University of Bristol. The authors would particularly like to thank Gary Clemo of Ofcom for useful discussions.

thesis