TITLE

NAME

A dissertation submitted to the University of Bristol
in accordance with the requirements for award of
the degree of Doctor of Philosophy in
the Faculty of Science

School of Mathematics

September 2013

NUMBER words

Abstract

Acknowledgments

Author's declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

Signed: Date:

Table of contents

List of tables

List of figures

0.1 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 **?** for the UK treasury, indicating the seriousness of the market players requirements for new spectrum. The recent 4g spectrum auction raised £2.3 billion **?**- with initial networks being rolled out by the end of 2013.

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

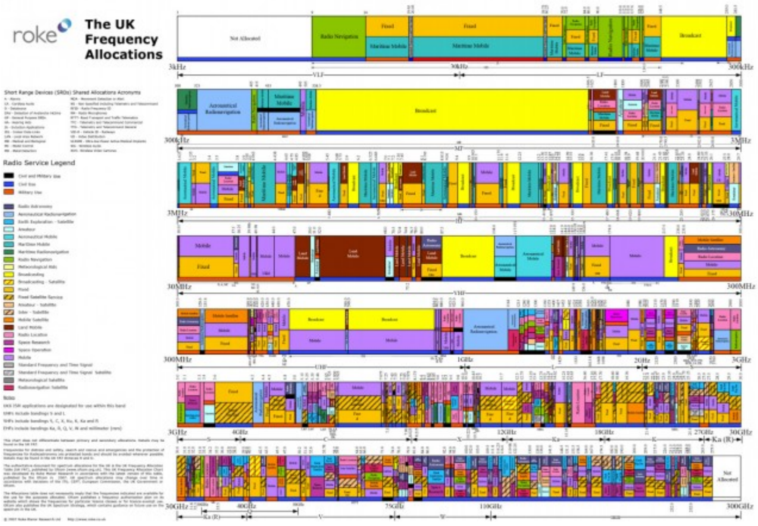


Figure 0.1.1: A digram of current Spectral allocation **?**

bar underneath the blocks indicates Government use). Compare this to the actual utilisation of spectrum: much of it is not used at all. Figure 0.1.2 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.

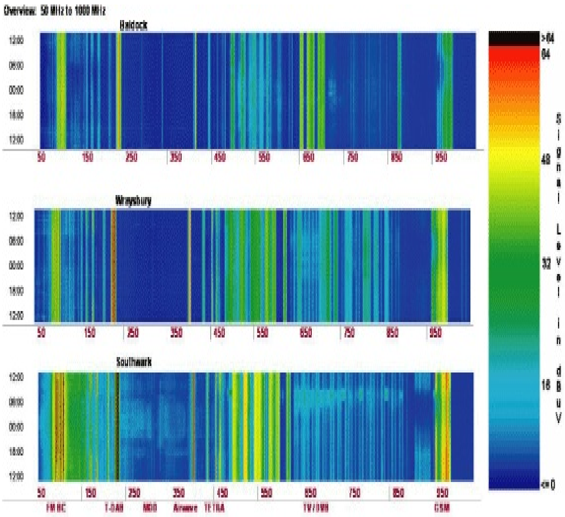


Figure 0.1.2: A snapshot of frequency utilisation in various areas: many frequencies are not used at all, whilst there is significant activity on others **?**

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 **?** and Donoho **?**, 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 possible to take an image of a patient's heart within a single breath, as well as dynamic imaging of the heart (**?** 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.

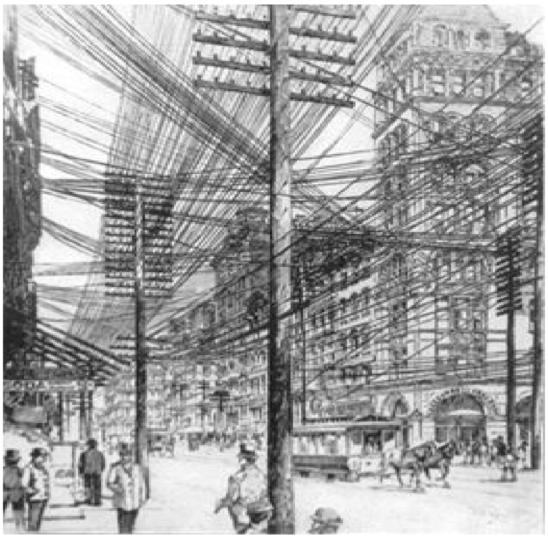


Figure 0.1.3*:* A picture of early 20th century New York: Bandwidth has always been an issue

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 coexistence 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 structure of the report is as follows: first a few 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) are discussed and the limitations of these are highlighted for the problem of sensing spectrum for Cognitive Radios. Then (more appropriate) wideband sensing techniques are discussed.

Compressive Sensing, and Group Testing are then introduced and it is argued that they offer appropriate solutions to the sensing problems for cognitive radios. Sampling schemes married to these reconstruction algorithms are discussed.

Finally the results and simulations performed over the time-frame of the project are discussed, with reference to further work.

0.2 Narrowband Spectrum Sensing

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

$$H_0 : y\left[n\right] = w\left[n\right], \mathrm{n} = 1 \ldots N \tag{0.2.0.1}$$

$$H_1 : y\left[n\right] = x\left[n\right] + w\left[n\right], \mathrm{n} = 1 \ldots N \tag{0.2.0.2}$$

Where x is the primary users signal (in the case of this thesis these signals are from TV broadcasts), having a specific structure stemming from modern coding and modulation techniques, w is additive white Gaussian noise and y is the received signal.

Spectrum detection methods rely on assumptions about the signal. The more detailed, and and accurate, the assumptions, the better the sensing performance. However, this entails a trade off: signals which do not meet the assumptions may not be detected by these schemes, even though they are a valid transmission. More complex assumptions also require a complex implementations, and as such are more expensive to deploy.

To decide whether the observations $\mathbf{y}$ were generated under H_0 or H_1 is accomplished by forming a test statistic $\Gamma\left(y\right)$ 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:

$$\Gamma\left(\mathbf{y}\right) = \frac{p\left(\mathbf{y} \mid H_0\right)}{p\left(\mathbf{y} \mid H_1\right)} \tag{0.2.0.3}$$

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

$$P_D = Pr\left(\Gamma\left(\mathbf{y}\right) > \lambda \mid H_1\right) \tag{0.2.0.4}$$

and the probability of false alarm

$$P_{FA} = Pr\left(\Gamma\left(\mathbf{y}\right) > \lambda \mid H_0\right) \tag{0.2.0.5}$$

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:

$$M = \sum_{n=0}^{N} |y[n]|^2 \tag{0.2.0.6}$$

As we are interested only in the power of the signal and the signal has an unknown form, it can be modelled as a $N\left(0, \sigma_s^2\right)$ random variable. Because of this assumption we can derive expressions for the metric, the detection probability and the false alarm probability:

$$M = \begin{cases} \frac{\sigma_w^2}{2} & H_0 \\ \frac{\sigma_w^2+\sigma_s^2}{2} & H_1 \end{cases} \tag{0.2.0.7}$$

$$P_D = 1 - \Gamma\left(1, \frac{\lambda}{1+\frac{\sigma_s^2}{\sigma_w^2}}\right) \tag{0.2.0.8}$$

$$P_{FA} = 1 - \Gamma\left(1, \frac{\lambda}{\sigma_w^2}\right) \tag{0.2.0.9}$$

Where $\Gamma(1, x)$ is the incomplete gamma function:

$$\Gamma(s, x) = \int_x^{\infty} t^{s-1} e^{-t} dt \tag{0.2.0.10}$$

From these equations we see that the performance of energy detection based sensing degrades at low SNRs. See **?**[figure 3] for curves quantifying the performance.

Despite the simplicity of implementation, and the paucity of assumptions energy detectors make about the signal, they suffer from several practical drawbacks.

The threshold used to make the decision is based on the noise variance σ_w^2, any error in the noise power estimation can cause significant performance loss. Various algorithms have been proposed to estimate this variance adaptively. However, the presence of any interfering signals in-band leads to significant false alarm rates: as the detector will be detecting incorrect signals.

Energy detectors perform poorly under fading conditions, as in this case it's not clear how to set the detection threshold in the presence of channel notches.

Energy detectors are unable to distinguish between modlated signals, interference and noise. As such, it can't make use of adaptive signal processing techniques such as interference cancellation to improve performance.

Further this type of detector is not efficient at detecting spread spectrum signals, as the signal power is spread over many frequencies.

Cyclostationary Feature Detection

Signals used in practical communication systems contain distinctive features that can be exploited for detection, thus it is possible to achieve a detection performance which substantially surpasses the energy detector. This is in contrast to the predictions of information theory where maximum entropy signals will be statistically white and Gaussian, and the energy detector is the optimal detector in AWGN.

Known signal features can be exploited to estimate unknown parameters such as noise power, carrier frequency offset etc. 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.

Cyclostationary Feature Detection 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 Power Spectral Density (or autocorrelation sequence) for detecting signals present in a given spectrum. Cyclic correlations are able to differentiate noise from primary users' signals - noise will have no periodic correlation, in contrast to PU signals.

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

$$\mathbb{E}x\left(t\right)=\mathbb{E}x\left(t+mT\right) \tag{0.2.0.11}$$

$$\mathbb{R}\left(t,\tau\right)=\mathbb{R}\left(t+mT,\tau\right)=\mathbb{E}\left[x\left(t\right)\right.$$

Conjecture 0.2.1. $x\ (t_{+}\tau)$ (0.2.0.12)

where t is time, τ is the autocorrelation lag, $x\left(t\right)$ 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:

$$\mathbb{R}\left(t,\tau\right)=\sum_{\alpha}r\left(\alpha,\tau\right)e^{2\pi j\alpha t} \tag{0.2.0.13}$$

with Fourier coefficients:

$$r\left(\alpha,\tau\right)=\frac{1}{T}\int_{T}x\left(t+\frac{\tau}{2}\right)$$

Conjecture 0.2.2. $x\left(t+\frac{\tau}{2}\right)e^{-2\pi j\alpha t}dt$(0.2.0.14)where α is the cyclic frequency

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

$$S\left(f\right)=\int_{-\infty}^{\infty}r\left(\alpha,\tau\right)e^{-2\pi jf\tau}d\tau \tag{0.2.0.15}$$

For a fixed lag τ, 0.2 can be re-written as:

$$R_{xx}\left(t,\tau\right)=R_{xx}\left(\tau\right)+\sum_{\alpha}r\left(\alpha,\tau\right)e^{2\pi j\alpha t} \tag{0.2.0.16}$$

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, (0.2, 0.2), the continuous portion of the signal exists, but the cyclostationary portion only exists under 0.2 when $\alpha\neq0$. Thus we only need to test for the presence of a cyclo-statrionary component.

To this end re-write the hypotheses as:

$$H_0 : y\left[n\right] = S_w^{\alpha}\left[n\right], \mathrm{n} = 1 \ldots N \tag{0.2.0.17}$$

$$H_1 : y\left[n\right] = S_x^{\alpha}\left[n\right] + S_w^{\alpha}\left[n\right], \mathrm{n} = 1 \ldots N \tag{0.2.0.18}$$

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

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

Conjecture 0.2.3. S_x^{α}(0.2.0.19)

we can formulate the cyclo-stationary detector as:

$$d = \begin{cases} 0 & \chi < \lambda \\ 1 & \chi \geq \lambda \end{cases} \tag{0.2.0.20}$$

where λ is some pre-determined threshold **?**.

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.

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, $r\left(t\right)$ defined over $0 \leq t \leq T$ the corresponding matched filter is $h\left(t\right) = r\left(T - t\right)$.

A test statistic can be formed by sampling the output of the filter every nT seconds and choosing 0.2 if the statistic is below some threshold and 0.2 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.

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.

0.3 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.

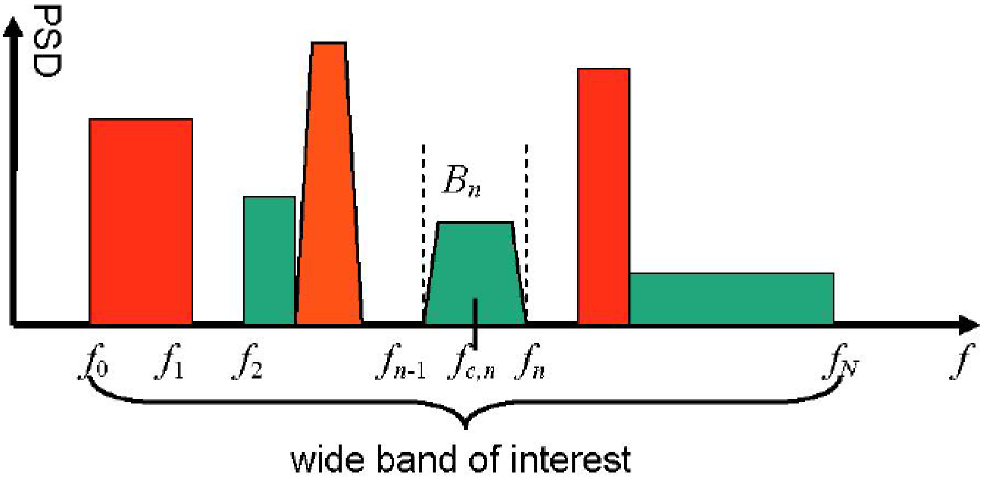


Figure 0.3.4: A digram of the Spectrum Sensing model **?**

0.3.1 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.

$$x\left(nT\right) = \left(t - nT\right) x\left(t\right) \tag{0.3.1.21}$$

where

$$\left(t - nT\right) = \sum_{k=-\infty}^{\infty} \delta\left(t - kT\right) \tag{0.3.1.22}$$

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

$$\hat{X}_s\left(f\right) = \sum_{k=-\infty}^{\infty} x\left(t - kT\right) \tag{0.3.1.23}$$

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:

$$x\left(t\right) = \sum_{n=-\infty}^{\infty} x\left(nT\right) sinc\left(\frac{t_n T}{T}\right) \tag{0.3.1.24}$$

0.3.2 Compressed Sensing

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.

$$y_k = \langle f,\phi_k \rangle \ \ k = 1 \dots m \tag{0.3.2.25}$$

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

$$f(t) = \sum_{i=1}^{n} x_i \psi_i(t) \tag{0.3.2.26}$$

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 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 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 ? 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:

$$m \geq C\mu^2(\phi,\psi) S \log\left(\frac{n}{\delta}\right) \tag{0.3.2.27}$$

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) **?**:

$$(1-\delta)\,||x||_{l_2}^2 \leq ||Ax||_{l_2}^2 \leq (1+\delta)\,||x||_{l_2}^2 \tag{0.3.2.28}$$

We are also in a position to evaluate the meaning of the constant μ in 0.20. We are considering sampling within orthonormal systems (for example, Time and Frequency):

$$A * A = nI \tag{0.3.2.29}$$

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) **?**.

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:

$$\mu = max_{k,j}|\langle\phi_k, \psi_j\rangle| \tag{0.3.2.30}$$

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:

$$min\|\tilde{x}\|_{l_0} \text{ subject to } y_k = langle\phi_k,\psi x^* \;\forall k \in M \subset [1\ldots n] \tag{0.3.2.31}$$

where

$$||s||_0 = |s| \tag{0.3.2.32}$$

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 **?**. 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 **?** and Greedy approaches such as Orthogonal Matching Pursuit **?**.

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

$$min\|\tilde{x}\|_{l_1} \text{ subject to } y_k = \langle \phi_k, \psi x^* \rangle \ \forall k \in M \subset [1 \ldots n] \qquad (0.3.2.33)$$

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

0.3.3 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.

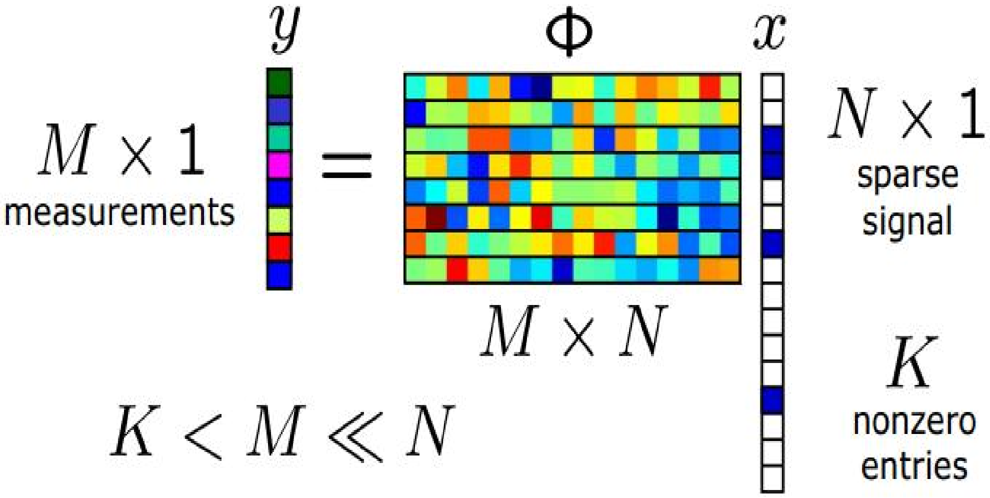


Figure 0.3.5: A visualisation of the Compressive Sensing problem as an under-determined system

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 0.3.6 shows this.

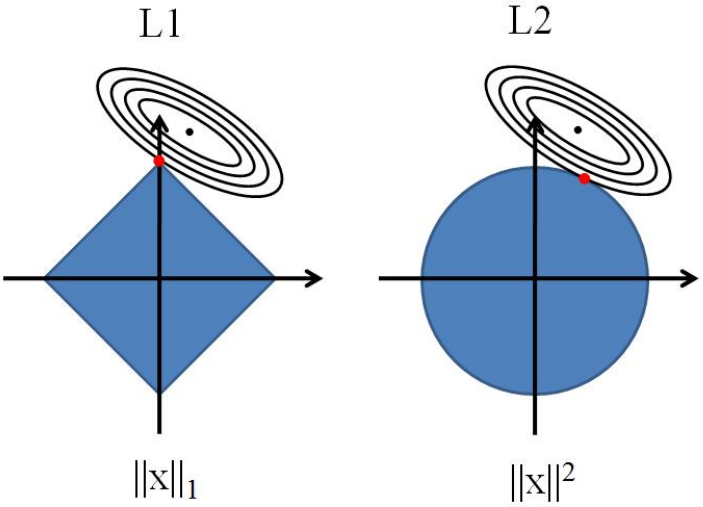


Figure 0.3.6: Solutions to the Compressive Sensing optimisation problem intersect the l_1 norm the points where all components (but one) of the vector are zero (i.e. it is sparsity promoting) **?**

Bayesian Compressive Sensing

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

$$\mathbf{g} = \Phi\mathbf{w} \tag{0.3.3.34}$$

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:

$$\mathbf{g} = \Phi\mathbf{w} + \mathbf{n} \tag{0.3.3.35}$$

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 0.3.3 we have a Gaussian likelihood model:

$$p\left(\mathbf{g} \mid \mathbf{w},\sigma^2\right) = (2\pi\sigma^2)^{-K/2} \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{g} - \Phi\mathbf{w}\|_2^2\right) \quad (0.3.3.36)$$

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:

$$p\left(w \mid \lambda\right) = \left(\frac{\lambda}{2}\right)^N exp{-}\lambda \sum_{i=1}^{N} |w_i| \quad (0.3.3.37)$$

Note that the solution the convex optimisation problem **??** 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 0.3.7 **?**).

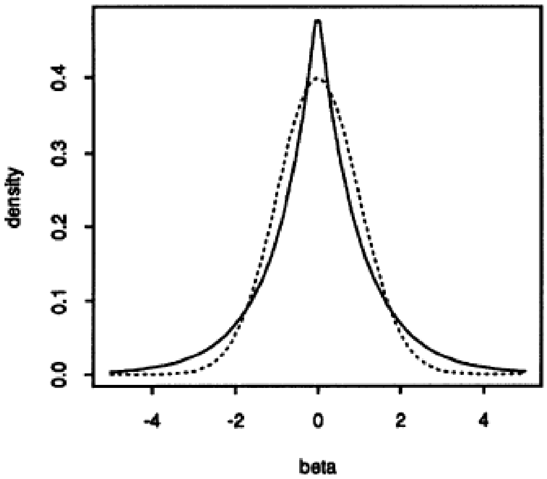


Figure 0.3.7: The Laplace (l_1-norm, bold line) and Normal (l_2-norm, dotted line) densities. Note that the Laplace density is sparsity promoting as it penalises solutions away from zero more than the Gaussian density. **?**

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:

$$p\left(w \mid a\right) = \prod_{i=1}^{N} \mathbb{N}\left(w_i \mid 0, \alpha_i^-1\right) \quad (0.3.3.38)$$

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

$$p\left(\alpha \mid a, b\right) = \prod_{i=1}^{N} \Gamma\left(\alpha_i \mid a, b\right) \tag{0.3.3.39}$$

The overall prior is found by marginalising over the hyperparameters:

$$p\left(w \mid a, b\right) = \prod_{i=1}^{N} \int_0^{\infty} \mathbb{N}\left(w_i \mid 0, \alpha_i^{-}1\right) \Gamma\left(\alpha_i \mid a, b\right) \tag{0.3.3.40}$$

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 0.3.10. This model, and other CS models which not necessarily have closed form solutions, can be solved via belief-propagation **?**

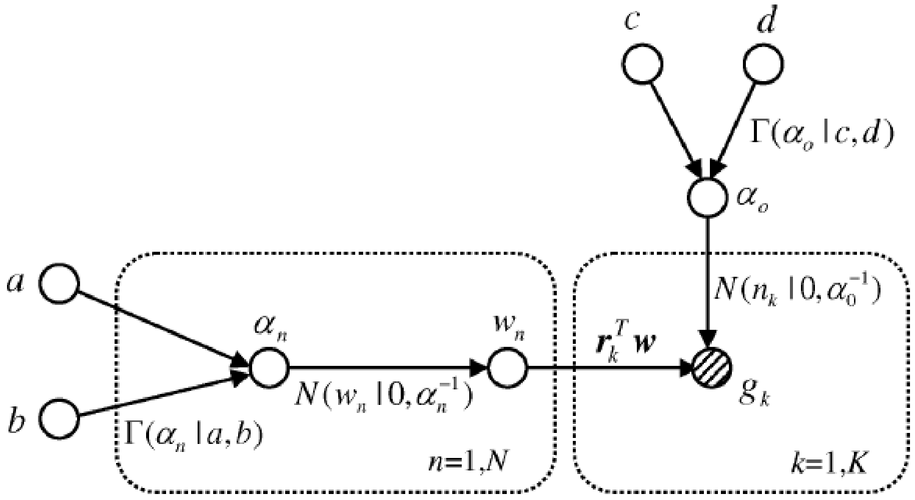

c
d
$\Gamma(\alpha_o \mid c, d)$
α_o
a
b
α_n
w_n
$N(n_k \mid 0, \alpha_0^{-1})$
$r_k^T w$
g_k
$N(w_n \mid 0, \alpha_n^{-1})$
$\Gamma(\alpha_n \mid a, b)$
n=1,N
k=1,K

Figure 0.3.8: The hierarchical model for the Bayesian CS formulation **?**

0.3.4 **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 **?**, 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:

$$A\left(S\right) = 1 \sum_i x_i \geq 1 \tag{0.3.4.41}$$

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 powre 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

$$K \lceil \log_2 N \rceil \leq K \log_2 N + K \tag{0.3.4.42}$$

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:

$$y = \mathbf{A}x \tag{0.3.4.43}$$

$y \in \{0,1\}^K, x \in \{0,1\}^N$, 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.

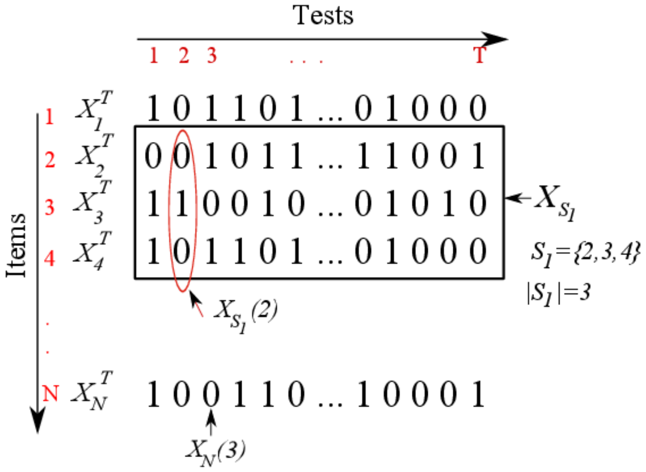

Tests
1 2 3 . . . T
Items
1 X_1^T 1 0 1 1 0 1 ... 0 1 0 0 0
2 X_2^T 0 0 1 0 1 1 ... 1 1 0 0 1
3 X_3^T 1 1 0 0 1 0 ... 0 1 0 1 0
4 X_4^T 1 0 1 1 0 1 ... 0 1 0 0 0
N X_N^T 1 0 0 1 1 0 ... 1 0 0 0 1
X_{S_1}
$S_1=\{2,3,4\}$
$|S_1|=3$
$X_{S_1}(2)$
$X_N(3)$

Figure 0.3.9: The Group Testing model: multiplication with a short, fat matrix **?**

See **?** 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.

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 **?** 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:

$$P\left(Success\right) \leq \frac{T}{\log_2 \binom{N}{K}} \tag{0.3.4.44}$$

However, a tighter upper bound has recently been found **?** at:

$$P\left(Success\right) \leq \frac{2^T}{\binom{N}{K}} \tag{0.3.4.45}$$

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:

$$T = \log_2 \binom{N}{K} + K \tag{0.3.4.46}$$

tests. The Combinatorial Orthogonal Matching Pursuit algorithm, considered in **?**, is guaranteed to recover the defective set with probability $N^{-\delta}$ in

$$T = \left(\left(1 + \delta\right) e\right) K \ln N \tag{0.3.4.47}$$

tests. For all N and K) we have:

$$K \log_2 \frac{N}{K} \leq \log_2 \binom{N}{K} \leq K \log_2 \frac{Ne}{K} \tag{0.3.4.48}$$

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:

$$T = \beta K \log_2 N + K (\log_2 e + 1) \tag{0.3.4.49}$$

tests, whilst the COMP algorithm succeeds with:

$$1.88 (1 + \delta) K \log_2 N \tag{0.3.4.50}$$

tests. It's worthwhile to contrast these results, to gain some insight into the problem. 0.3.4 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 **?**. 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 **?** 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, **?** considers the noisy Group Testing problem and the reconstruction of the defective set via belief propagation whilst **?** gives explicit theorems on conditions for the recovery of the defective set for the case of the binary symmetric channel. **?** takes this further and finds the capacity of Group Testing for a number of cases.

0.3.5 Sub-Nyquist Sampling techniques

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

Wideband Modulated Converter

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

It works by mixing the incoming analogue signal $x\left(t\right)$ with a mixing function $p_i\left(t\right)$ aliasing the spectrum. $x\left(t\right)$ 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:

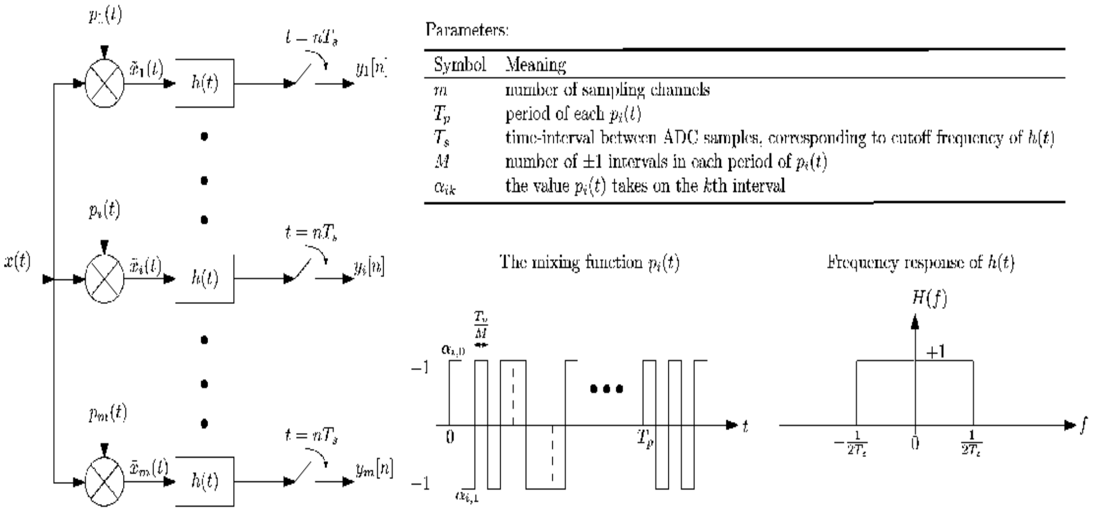
$$p_i\left(t\right) = \sum_{l=-\infty}^{\infty} c_{il} expjlt\frac{2\pi}{T_p} \tag{0.3.5.51}$$

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:

$$\begin{aligned} X_i\left(f\right) = && \int_{-\infty}^{\infty} x\left(t\right) p_i\left(t\right) dt && (0.3.5.52) \\ = && \sum_{l=-\infty}^{\infty} c_{il} X\left(f - lf_p\right) && (0.3.5.53) \end{aligned}$$

(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 $X\left(f\right)$, with at most $\lceil f_N YQ/f_p \rceil$ terms since $X\left(f\right)$ 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:



Parameters:

Symbol	Meaning
m	number of sampling channels
T_p	period of each $p_i(t)$
T_s	time-interval between ADC samples, corresponding to cutoff frequency of $h(t)$
M	number of ±1 intervals in each period of $p_i(t)$
α_{ik}	the value $p_i(t)$ takes on the kth interval

Figure 0.3.10: The operation of the Modulated Wideband Converter **?**

$$Y_i\left(e^{j2\pi f T_s}\right) = \sum_{l=-L_0}^{+L_0} \tag{0.3.5.54}$$

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 $X(f)$ over $[-f_2/2, f_s/2]$ - at most $\lceil f_N YQ/f_p \rceil$ if we choose $f_s = f_p$. These relations can be written in matrix form as:

$$\mathbf{y} = \_\mathbf{A}\mathbf{x} \tag{0.3.5.55}$$

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 $x(t)$.

0.4 Results and Simulations

To compare the efficacy of Group Testing and Compressive Sensing, Hwang's algorithm and the algorithm presented in **?** 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 0.4.11 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

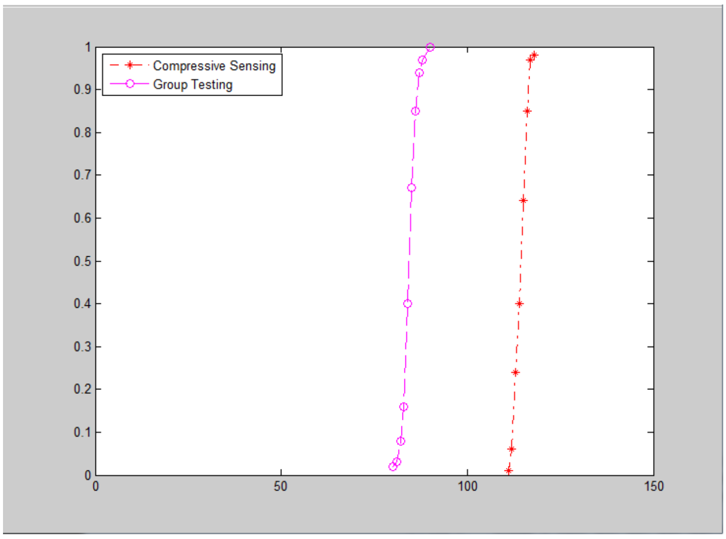


Figure 0.4.11: Group Testing vs Compressive Sensing

performance increase is substantial.

0.5 Further Work and Project Plan

Both the Compressive Sensing and Group Testing problems have been well studied for the case where there is no prior belief over where the non-zero components of the signal are (or equivalently, which items constitute the defective set). Despite some work being done solving the CS problem in the Bayesian paradigm, the prior distributions used are sparsifying (for example a Laplace prior), and not over the placement of the non-zero components.

AS such, there is some 'low hanging fruit' to be picked: would using extra information about the likelihood of an item being defective improve the performance of recovery suitably defined? For example, would a Group Testing procedure require fewer tests if there was reason to believe that some subset of items was more likely to be defective. In the coin weighing example discussed above, the input set may contain a coin which is a different colour than the others - by testing this 'suspicious' coin in the first weighing we could find the counterfeit in a single test (otherwise we would find the counterfeit in at most five tests - one for the suspicious coin, and four following the procedure outlined in the Group Testing section).

This suggests that using prior information in could significantly reduce the number of tests required to recover the defective set, however if initial tests come back negative the performance of these algorithms could be worsened by using a non-iid distribution.

To try and characterise this numerical simulations and probabilistic analyses are required. The goal is to design algorithms which have better average case performance than current GT algorithms by incorporating prior beliefs, whilst at the same time do no worse than current performance guarantees. In the example above we'd seek an algorithm that could find the counterfeit coin in a single weighing (by testing the suspicious coin initially), yet could guarantee to find the counterfeit in at most four weighings of the pan balance.

For the 80 coins, 1 counterfeit discussed above this is probably not possible: adding an extra test which fails before proceeding with the standard approach will always be worse than simply assuming all coins are equally likely to be fake and proceeding accordingly. Do many cases like this exist, and what characterises them? Will testing suspicious items initially always give a hit in performance should those tests fail?

A few directions into answering these questions are sketched below:

1. Implicit in the discussion of Hwang's algorithm above is that each item is defective with the same probability $\frac{K}{N}$. Using prior knowledge over the input set we could test groups of size 2^{β}, where β is the empirical expectation of the input set. I.e. modify Hwang's algorithm to test groups in size roughly equal to the (empirical) expectation of the distribution over the input. **Timeline:** the rest of October 2013. **Risk:** this is a low risk activity and should take at most a fortnight - the Author has already implemented Hwang's algorithm, so all that needs to be done is to calculate the empiracle mean of the input distribution, and characterise the algorithm in a few cases.

2. Re-formulate the problem in terms of source coding as per **?** and represent the items in the input set as a Huffman tree. **Timeline:** October-December 2013. **Risk:** medium risk. The Group Testing algorithm will take place over the tree with th groups to test being the branches of the Huffman tree. A bit of the code is ready, however it's not clear how to fit it all together to realise a full solution

3. It has usually been assumed that defective items are independent. Is it possible to relax this assumption, and if so what are the consequences of this for Group Testing algorithms? For example, what happens if defectives occur in clumps? This is a high risk activity, and should take over 6 months to begin to characterise. **Timeline:** October 2013-March 2014. **Risk:** medium risk. Defectives are currently assumed to be independent from each other in the classical Group Testing literature. Motivated by Spectrum Sensing for Cognitive Radios, where available frequencies are not necessarily independent between measurements, we will investigate Group Testing algorithms where the defectives are not independently distributed. Examples include inputs with defective which are pairwise independent, input sets in which the defectives are negatively dependent, and input sets where there is some (random or deterministic) process affecting the distribution of defectives.

4. Some work has been done considering Group Testing as the reverse of Coding **?**, **?**, but there is still much to be done. How close is the analogy? Coding theory has a wealth of noise cases - could these be imported into the Group Testing literature, and how useful will they be? **Timeline:** October 2013 - September 2014. **Risk:** medium to high risk. A few cases have been explicitly solved (the 'binary-symmetric' and 'erasure' models). Does this analogy extend further, and can techniques from Coding help us in Group Testing. Can we formulate a more general 'Shannon Theory' type theory for Group Testing?

5. To make these models suitable for engineering applications, noisy cases must be considered. **Timeline:** January - September 2013. **Risk:** High risk: whereas Group Testing has

previously been utilised in Medium Access Control protocol design, it has not been applied to spectrum sensing as of yet. The exact mechanism that can robustly sense spectral opportunities, and make decisions needs careful consideration. Example problems include how should we sample spectra using Group Testing - will a series of wideband pulses over the TVWS band be sufficient, or are is there some other possibility? Once we have these samples, how do then infer spectral opportunities? The work of previous sections (non-iid input distributions, 2D input sets, defectives which are not independent in some way etc) will inform this part of the research.

6. There currently exist no two-dimensional models of Group Testing in the literature. This is particularly relevant for the problem faced by Cognitive Radios: the radios could represent the measurement point on a geographic grid, and the 'groups' they need to infer are sets of available frequencies. **Timeline:** January-April 2014. **Risk:** High risk: there currently is no work in the literature on 2D group testing. Conjectured methods of solving this problem include factor graph methods from Coding Theory. It's not clear that this is an appropriate method, however.

Table 1: Things Which Do/Don't Work

	Haar Wavelets	db10 Wavelets	Splines	Differences
Centralised	Y	Y	Y	Y
Distributed	Y	N	Y	Y
Noise	N(0,1)	N(0,1)	N(0,1)	N(0,1)
Regulariser	$\lambda \lvert\lvert x \rvert\rvert_1$	$\lambda \lvert\lvert x \rvert\rvert_1$	$\lambda \lvert\lvert x \rvert\rvert_1$	$\lambda \lvert\lvert x \rvert\rvert_1$
Lambda	$\sqrt{2\log(M)}$ / $\sqrt{40\log(M)}$	$\sqrt{2\log(M)}$	$\sqrt{2\log(M)}$ $2.5\sqrt{2\log(M)}$	$\sqrt{2\log(M)}$ / $10\sqrt{2\log(}$
Step Size	0.5	$\frac{1}{max\lvert eig(A^T A)\rvert}$	$\frac{1}{max\lvert eig(A^T A)\rvert}$	$\frac{1}{max\lvert eig(A^T A)\rvert}$

0.6 Summary

- Extending the DADMM algorithm to composite functions.
- Providing new, clearer, proofs of the edge decomposition of the Lagrangian
- Extending the Modulated Wideband Converter to model signals composed of rectangles.
- De-noising OFCOM spectrum data.

0.7 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 **?**.

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 **?** 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 **?**), 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) **?** 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 **?**, and **?** 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.

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 **?**) 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 **?**) and can be applied to other composite optimisation problems.

We also give new proofs of ideas found in **?**.

The structure of the report is as follows: in section 0.16 we introduce the sensing model, in section 0.17 we describe the distributed reconstruction algorithm **?**, and finally in section

0.18 we show some results of the reconstruction quality of this model.

0.8 ADMM

Given a set of measurements of the form

$$y = Ax + n \tag{0.8.0.56}$$

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:

$$L = \frac{1}{2} \left|\left|Ax - y\right|\right|_2^2 + \lambda \left|\left|x\right|\right|_1 \tag{0.8.0.57}$$

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

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

$$\begin{aligned} &\arg\min_x f(x) + g(z) \\ &\quad \text{s.t } Ux + Vz = c \end{aligned} \tag{0.8.0.58}$$

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 **?**, with the following iterations:

$$x^{k+1} := \arg\min_x L_\rho \left(x, z^k, \eta^k\right) \tag{0.8.0.59}$$

$$z^{k+1} := \arg\min_z L_\rho \left(x^{k+1}, z, \eta^k\right) \tag{0.8.0.60}$$

$$\eta^{k+1} := \eta^k + \rho \left(Ux^{k+1} + Vz^{k+1} - c\right) \tag{0.8.0.61}$$

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 (0.8.0.57). We can write (0.8.0.57) in constrained form as:

$$\frac{1}{2}\|Ax-b\|_2^2+\lambda\|z\|_1 \tag{0.8.0.62}$$

$$\text{s.t } z = x \tag{0.8.0.63}$$

i.e this is of the form (0.8.0.58) with $f(x)=\|Ax-y\|_2^2$, $g(z)=\lambda\|z\|_1$, $U=I$, $V=-I$, and $c=0$.

The associated (augmented) Lagrangian is:

$$L_\rho=\frac{1}{2}\|Ax-b\|_2^2+\lambda\|z\|_1+\eta(x-z)+\frac{\rho}{2}\|x-z\|^2 \tag{0.8.0.64}$$

The ADMM iterations for LASSO, which can be found by alternately differentiating (0.8.0.64) with respect to x,z and η, are (in closed form):

$$x^{k+1}:=\left(A^TA+\rho I\right)^{-1}\left(A^Ty+\rho\left(z^k-\eta^k/\rho\right)\right) \tag{0.8.0.65}$$

$$z^{k+1}:=S_{\lambda/\rho}\left(x^{k+1}+\eta^k/\rho\right) \tag{0.8.0.66}$$

$$\eta^{k+1}:=\eta^k+\rho\left(x^{k+1}-z^{k+1}\right) \tag{0.8.0.67}$$

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 (0.8.0.64) with respect to x and z as follows:

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

as

$$\frac{\partial}{\partial x}\,||F(x)||_2^2=2\left(\frac{\partial}{\partial x}F(x)\right)F(x) \tag{0.8.0.68}$$

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 (0.8.0.68) to zero and collecting like terms:

$$\left(A^T A + \rho I\right) x = A^T y + \rho z - \eta \tag{0.8.0.69}$$

so we find the optimal x is:

$$x = \left(A^T A + \rho I\right)^{-1} \left(A^T y + \rho \left(z - \eta / \rho\right)\right) \tag{0.8.0.70}$$

note that this estimator is a weighted average of the ordinary least squares estimate ($A^T y$) 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 $z > 0$

$$\frac{\partial L}{\partial z} = \lambda + \rho(x - z) - \eta \tag{0.8.0.71}$$

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 $z < 0$:

$$\frac{\partial L}{\partial z} = -\lambda + \rho(x - z) \tag{0.8.0.72}$$

setting (0.8.0.72) 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.

$$\mid x + \frac{\eta}{\rho} \mid \leq \frac{\lambda}{\rho} \tag{0.8.0.73}$$

combining (0.8.0.72), (0.8.0.71), (0.8.0.73) together we find the optimal z is:

$$z = \text{sign}(x + \frac{\eta}{\rho}) \max\left(\mid x + \frac{\eta}{\rho} \mid - \frac{\lambda}{\rho}, 0\right) \tag{0.8.0.74}$$

Together (0.8.0.70), (0.8.0.74) and the third step of (0.8.0.67) 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:

$$y = x + w \tag{0.8.0.75}$$

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

$$\hat{x} = \arg\max_x \mathbb{P}_{x|y}(x|y) \tag{0.8.0.76}$$

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

$$\hat{x} = \arg\max_x \left[\log \mathbb{P}_w(y-x) + \log \mathbb{P}(x)\right] \tag{0.8.0.77}$$

The term $\mathbb{P}_n(y-x)$ 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:

$$\mathbb{P}(x) = \frac{1}{\sqrt{2}\sigma} \exp - \frac{\sqrt{2}}{\sigma}|x| \tag{0.8.0.78}$$

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

Taking the maximum of 0.8.0.77 we obtain:

$$\frac{y-\hat{x}}{\sigma_n^2} - \frac{\sqrt{2}}{\sigma} sign(x) = 0 \tag{0.8.0.79}$$

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

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

or

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

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

0.8.1 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 $\mathcal{H}$ is defined as:

Definition 0.8.1 (Proximity Operator)**.**

$$\mathrm{Prox}_f\left(y\right) := \underset{y\in\mathcal{H}}{\arg\min} f\left(y\right) + \frac{1}{2}\left|\left|y-x\right|\right|^2 \tag{0.8.1.80}$$

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 $\mathrm{Prox}\left(\circ\right)$ 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 0.8.2 (Moreau Envelope)**.**

$$\mathrm{M}_f\left(y\right) := \min_{y\in\mathcal{H}} f\left(y\right) + \frac{1}{2}\left|\left|y-x\right|\right|^2 \tag{0.8.1.81}$$

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 $\mathcal{H}$-valued.

Properties

Theorem 0.8.3 (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 0.8.4 (Moreau '65)**.** *Let $f\in\Gamma$. Then for all $z\in\mathcal{H}$*

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

Theorem 0.8.5 (Moreau '65)**.** *The Moreau envelope is (Frechet) differentiable, with*

$$\nabla\mathrm{M}_f = Id - \mathrm{Prox}_f = \mathrm{Prox}_{f^*} \tag{0.8.1.82}$$

Theorem 0.8.6 (Moreau '65)**.** $\mathrm{Prox}_f : \left(\mathcal{H}, \left|\left|\circ\right|\right|\right) \leftarrow \left(\mathcal{H}, \left|\left|\circ\right|\right|\right)$ *is 1-Lipchitz continuous.*

Motivation

We are solving problems of the following form:

$$\min_{x\in\mathcal{H}} f(x) + g(z) \tag{0.8.1.83}$$
$$\text{s.t } x - z = 0 \tag{0.8.1.84}$$

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

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

and then performing the following iterative minimisation:

$$x^{k+1} := \arg\min_x L_\rho\left(x, z^k, \eta^k\right) \tag{0.8.1.85}$$
$$z^{k+1} := \arg\min_z L_\rho\left(x^{k+1}, z, \eta^k\right) \tag{0.8.1.86}$$
$$\eta^{k+1} := \eta^k + \rho\left(x^{k+1} + z^{k+1}\right) \tag{0.8.1.87}$$

i.e.

$$x^{k+1} := \arg\min_x \left(f(x) + \eta^{kT}x + \frac{\rho}{2}\left|\left|x - z^k\right|\right|^2\right) \tag{0.8.1.88}$$
$$z^{k+1} := \arg\min_z \left(g(z) - \eta^{kT}z + \frac{\rho}{2}\left|\left|x^{k+1} - z\right|\right|^2\right) \tag{0.8.1.89}$$
$$\eta^{k+1} := \eta^k + \rho\left(x^{k+1} + z^{k+1}\right) \tag{0.8.1.90}$$

pulling the linear terms into the quadratic ones we get:

$$x^{k+1} := \arg\min_x \left(f(x) + \frac{\rho}{2}\left|\left|x - z^k + (1/\rho)\,\eta^k\right|\right|^2\right) \tag{0.8.1.91}$$
$$z^{k+1} := \arg\min_z \left(g(z) + \frac{\rho}{2}\left|\left|x^{k+1} - z - (1/\rho)\,\eta^k\right|\right|^2\right) \tag{0.8.1.92}$$
$$\eta^{k+1} := \eta^k + \rho\left(x^{k+1} + z^{k+1}\right) \tag{0.8.1.93}$$

i.e.

$$x^{k+1} := \text{Prox}_f \left(z^k - u^k \right) \tag{0.8.1.94}$$

$$z^{k+1} := \text{Prox}_f \left(x^{k+1} + u^k \right) \tag{0.8.1.95}$$

$$u^{k+1} := u^k + \left(x^{k+1} + z^{k+1} \right) \tag{0.8.1.96}$$

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 0.8.7 (Indicator)**.** *From the definition*

$$\text{Prox}_I (x) := \arg\min_y I_C(y) + \frac{1}{2} \left|\left| y - x \right|\right|^2 \tag{0.8.1.97}$$

$$= \arg\min_{y \in C} \frac{1}{2} \left|\left| y - x \right|\right|^2 \tag{0.8.1.98}$$

$$= P_C (x) \tag{0.8.1.99}$$

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

Example 0.8.8 (l_2 norm)**.** *For $f(y) = \frac{\mu}{2} \left|\left| y \right|\right|^2$ the* Prox *operator is:*

$$\text{Prox}_f (x) := \arg\min_y \frac{\mu}{2} \left|\left| y \right|\right|^2 + \frac{1}{2} \left|\left| y - x \right|\right|^2 \tag{0.8.1.100}$$

$$= \frac{1}{1+\mu} x \tag{0.8.1.101}$$

Example 0.8.9 (l_1 norm)**.** $f = \|x\|_1$

$$\text{Prox}_f (x) := \text{sign}(x_i) \left(|x_i| - \gamma \right)^+ = S_\gamma (x)_i \tag{0.8.1.102}$$

Example 0.8.10 (Elastic Net)**.** *Consider*

$$f(x) = \lambda \|x\|_1 + \mu \left|\left| x \right|\right| \tag{0.8.1.103}$$

$$\text{Prox}_f\,(x) := \frac{\lambda}{1+\mu} S_\gamma\,(x)_i \tag{0.8.1.104}$$

Example 0.8.11 (Fused Lasso)**.** *Consider*

$$f(x) = \|x\|_1 + \sum_{i=1}^{d-1} (x_i - x_{i-1}) \tag{0.8.1.105}$$

i.e the sum of the l_1 and TV norms

$$\text{Prox}_f\,(x) := \text{Prox}_{l_1} \circ \text{Prox}_{TV} = S_\gamma\,(\text{Prox}_{TV})_i \tag{0.8.1.106}$$

Example 0.8.12 (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:

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

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:

$$L_\rho\,(x,y) = \sum_i^n \left(f_i\,(x_i) + y_i^T\,(x_i - z) + \frac{\rho}{2}\|x_i - z\|_2^2 \right) \tag{0.8.1.107}$$

The ADMM iterations for this Lagrangian are:

$$x_i^{k+1} := \arg\min x_i \left(f_i\,(x_i) + y_i^{kT}\,(x_i - z) + \frac{\rho}{2}\|x_i - z\|_2^2 \right) \tag{0.8.1.108}$$

$$z^{k+1} := \frac{1}{n} \sum_i^n \left(x_i^{k+1} + (1\rho)\, y_i^k \right) \tag{0.8.1.109}$$

$$y_i^{k+1} := y_i^k + \rho \left(x_i^{k+1} - z^{k+1} \right) \tag{0.8.1.110}$$

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:

$$x_i^{k+1} := \arg\min x_i \left(f_i\left(x_i\right) + y_i^{kT}\left(x_i - \bar{x}^k\right) + \frac{\rho}{2}\|x_i - \bar{x}^k\|_2^2 \right) \tag{0.8.1.111}$$

$$y_i^{k+1} := y_i^k + \rho\left(x_i^{k+1} - z^{k+1}\right) \tag{0.8.1.112}$$

where

$$\bar{x}^k = \frac{1}{n}\sum_i^n x_i^k \tag{0.8.1.113}$$

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.

0.8.2 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.

0.8.3 Acceleration

0.9 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 0.16):

$$\mathbf{y}_p = \mathbf{A}_p\mathbf{x} + \mathbf{n}_p \tag{0.9.0.114}$$

where $\mathbf{A}_p$ is the p^{th} row of the sensing matrix from (0.16.0.159), and the system (0.16.0.159) 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 $C = \{1 \ldots c\}$, 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, 0.16.0.159), 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:

$$\begin{aligned} \arg\min_{\bar{x}} \sum_{c=1}^{C} \sum_{j \in c} f\left(x_j\right) + \frac{\lambda}{J} g\left(x_j\right) \\ \text{and } x_i = x_j \text{ if } \{i, j\} \in E \\ \text{and } x_i = z_i \ \forall i \in \{1, \ldots, C\} \end{aligned} \tag{0.9.0.115}$$

with a particular special case being:

$$\begin{aligned} \arg\min_{\bar{x}} \sum_{c=1}^{C} \sum_{j \in c} \|A_j x_j - y_j\|_2^2 + \frac{\lambda}{J} \|z\|_1 \\ \text{and } x_i = x_j \text{ if } \{i, j\} \in E \\ \text{and } x_i = z_i \ \forall i \in \{1, \ldots, C\} \end{aligned} \tag{0.9.0.116}$$

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

$$\bar{x} = \sum_{c=1}^{C} w_c \otimes x_c = \left[x_{c(1)}^T, \ldots, x_{c(J)}^T\right]^T \tag{0.9.0.117}$$

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 (0.9.12) and (0.9.13) 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

$$\sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_c = 0 \tag{0.9.0.118}$$

note that $\left(B^T \otimes I_n\right) \in \mathbb{R}^{nE \times nJ}$. Together (0.17.0.162) and (0.17.0.164), suggests that the problem (0.17.0.163) can be re-written as:

$$\begin{aligned} \arg\min_{\bar{x}} \sum_{c=1}^{C} \sum_{j \in C_c} f\left(x_j\right) + \frac{\lambda}{J} g\left(z_j\right) \\ \text{s.t. } \sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_c = 0 \\ \text{and } \bar{x}_c - \bar{z}_c = 0 \end{aligned} \tag{0.9.0.119}$$

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

The global Augmented Lagrangian **?** for the problem (0.17.0.165) can be written down as:

$$\begin{aligned} L_\rho = \sum_{c=1}^{C} (\sum_{j \in c} & f\left(x_j\right) + \frac{\lambda}{J} g\left(z_j\right) + \\ & + \theta^T\left(\bar{x}_j - \bar{z}_j\right) + \frac{\rho}{2}\left|\left|\bar{x}_j - \bar{z}_j\right|\right|_2^2) + \\ & + \eta^T \left(B_c^T \otimes I_n\right) \bar{x}_c + \frac{\rho}{2} \left|\left| \sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_c \right|\right|_2^2 \end{aligned} \tag{0.9.0.120}$$

This is, superficially, similar to the Augmented Lagrangian for the Lasso problem **?**[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 0.17.0.166 can be posed as a constrained optimisation problem at each node, we introduce the following variable (so that the the final term of 0.17.0.166 is separable across edges of the graph):

Definition 3.

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

where we have used the definition (0.17.0.162) *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* $B_c = w_c^T B$*.*

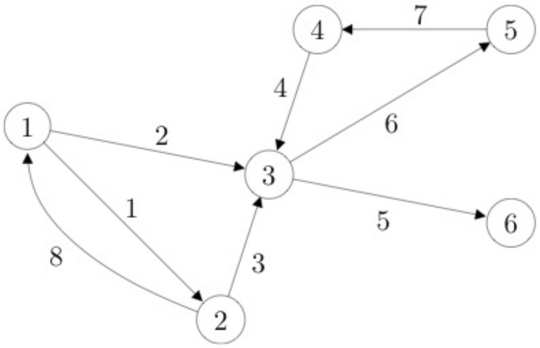
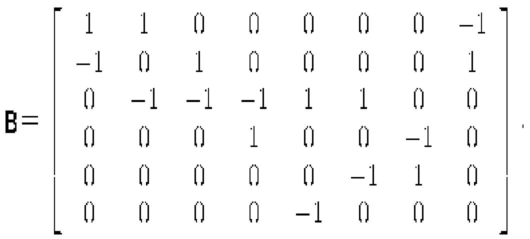


Figure 0.9.12*:* An example of a network



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

Figure 0.9.13*:* The incidence matrix associated with Figure (0.9.12)

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 (0.17.0.166), can be decomposed across edges, using the following lemma:

Lemma 0.9.1 (Edge Decomposition)**.**

$$\left|\left| \sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_c \right|\right|^2 = \sum_{j \in C_1} \left(D_j \, ||x_j||_2^2 - \sum_{k \in N_j} x_j^T x^k \right) \tag{0.9.0.121}$$

and

$$\eta^T \sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_1 = \sum_{l \in C_c} \sum_{m \in N_l} sign\left(m-l\right) \eta_{ml}^T x_l \tag{0.9.0.122}$$

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.

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

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:

$$\eta = \sum_{c=1}^{C} w_c \otimes \eta_c \tag{0.9.0.123}$$

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 (0.17.0.166) as:

$$\begin{aligned} L_\rho = \sum_{c=1}^{C} \sum_{j \in C_c} \left(f\left(x_j\right) + \beta g\left(z_j\right)\right) + \nu^T x_j \\ \theta\left(x_j - z_j\right) + \frac{\rho}{2} D_i \left|\left|x_j\right|\right|^2 + \frac{\rho}{2} \|x_j - z_j\|^2 \end{aligned} \tag{0.9.0.124}$$

where we have defined:

$$\nu_i = \left(\sum_{k \in \mathcal{N}_i} sign\left(k-i\right) \eta_{\{i,k\}} - \rho x_k \right) \tag{0.9.0.125}$$

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

Then by differentiating (0.17.0.170) with respect to x_j and z_j we can find closed forms for the updates as:

Theorem 1.

$$x_j^{k+1} := \left(A_j^T A_j + (\rho D_J + 1) I\right)^{-1} \left(A_j^T y_j + z^k - \nu^{kT}\right) \tag{0.9.0.126}$$

$$z_j^{k+1} := S_{\beta/\rho}\left(x_j^{k+1}\right) \tag{0.9.0.127}$$

$$\theta_j^{k+1} := \theta_j^k + \rho\left(x^{k+1} - z^{k+1}\right) \tag{0.9.0.128}$$

$$\eta_j^{k+1} := \eta_j^k + \rho\left(\sum_{m \in N_j} z_m^k - z_j^k\right) \tag{0.9.0.129}$$

This algorithm can be thought of as follows: each node performs an iteration of (non multiblock) 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.

0.10 **Compressive Sensing Architechtures**

0.10.1 **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 **?**. The calculations which follow are taken from **?** as well.

The nodes individually take measurements (as in **?**) 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.

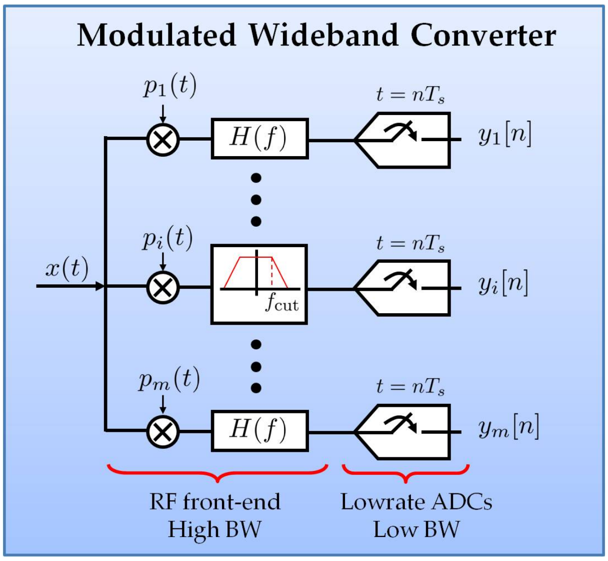


Figure 0.10.14*:* Mse vs SNR for the sensing model, with AWGN only, showing the performance of distributed and centralised solvers

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:

$$p_i\left(t\right) = \sum_{l=-\infty}^{\infty} c_{il} \exp\left(jlt\frac{2\pi}{T_p}\right) \tag{0.10.1.130}$$

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 $X\left(\dot{}\right)$):

$$\begin{aligned} X_i\left(f\right) = & \int_{-\infty}^{\infty} x\left(t\right) p_i\left(t\right) dt \\ = & \sum_{l=-\infty}^{\infty} c_{il} X\left(f - lf_p\right) \end{aligned} \tag{0.10.1.131}$$

(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 $X\left(f\right)$, with at most $\lceil f_N YQ/f_p \rceil$ terms since $X\left(f\right)$ 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:

$$Y_i\left(e^{j2\pi f T_s}\right) = \sum_{l=-L_0}^{+L_0} c_{il} X\left(f - l f_p\right) \tag{0.10.1.132}$$

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:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w} \tag{0.10.1.133}$$

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

i.e. **A** can be written:

$$\mathbf{A}^{m\times L} = \mathbf{S}^{m\times L}\mathbf{F}^{L\times L}\mathbf{D}^{L\times L}\mathbf{H}^{L\times L} \tag{0.10.1.134}$$

The system 0.16.0.159 can then be solved (in the sense of finding the sparse vector **x** by convex optimisation via minimising the objective function:

$$\frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{x}\|_1 \tag{0.10.1.135}$$

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

0.10.2 Random Demodulator

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

$$x(t) = \sum_{n=1}^{N} \alpha_n \psi_n(t) \tag{0.10.2.136}$$

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 $\mathcal{M}$ with a traditional ADC.

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

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

$$y\left[m\right] = \int_{-\infty}^{\infty} x\left(\tau\right) p_c\left(\tau\right) h\left(t-\tau\right) |_{t=m\mathcal{M}}\, d\tau \tag{0.10.2.137}$$

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

$$y\left[m\right] = \sum_{n=1}^{N} \alpha_n \int_{-\infty}^{\infty} \psi_n\left(t\right) p_c\left(\tau\right) h\left(m\mathcal{M}-\tau\right) d\tau \tag{0.10.2.138}$$

we see that the output can be written as:

$$y = Ax \tag{0.10.2.139}$$

with

$$A_{m,n} = \int_{-\infty}^{\infty} \psi_n\left(t\right) p_c\left(\tau\right) h\left(m\mathcal{M}-\tau\right) d\tau \tag{0.10.2.140}$$

0.11 Joint Space-Frequency Model

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

$$\phi_s = \sum_b \beta_{bs} \psi_b\left(f\right) \tag{0.11.0.141}$$

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:

$$\phi_r = \sum_s g_{sr}\phi_s = \sum_s g_{sr} \sum_b \beta_{bs}\psi_b\left(f\right) \tag{0.11.0.142}$$

where

$$g_{sr} = \exp\left(-||x_r - x_s||_2^{\alpha}\right) \tag{0.11.0.143}$$

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:

$$y_r = \left(e_r^T \bigotimes I_{n_b}\right) y \tag{0.11.0.144}$$

with,

$$y = \left(\tilde{G} \bigotimes I_{n_b}\right) \phi \tag{0.11.0.145}$$

Now, we have

$$\phi = e_s \bigotimes \phi_s \tag{0.11.0.146}$$

so,

$$y = \left(\tilde{G} \bigotimes I_{n_b}\right) \left(e_s \bigotimes \phi_s\right) \tag{0.11.0.147}$$

finally we have,

$$y_r = \left(e_r^T \bigotimes I_{n_b}\right) \left[\left(\tilde{G} \bigotimes I_{n_b}\right) \left(e_s \bigotimes \phi_s\right)\right] \tag{0.11.0.148}$$

$\beta_{bs} \in \mathbb{R}^{1 \times n_b}$, $g_{sr} \in \mathbb{R}^{n_r \times n_s}$ 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.

0.12 **Results**

The model described in section (0.16), equation (0.16.0.159) 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 (0.12.16).

The MSE was calculated as follows:

$$\frac{\left|\left|Z^k - Z*\right|\right|}{||Z*||} \tag{0.12.0.149}$$

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 (0.17), 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 **?**.

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.

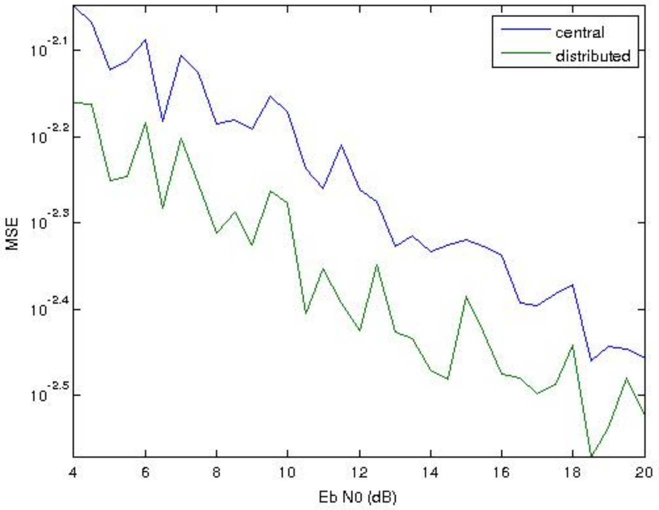


Figure 0.12.15: Mse vs SNR for the sensing model, with AWGN only, showing the performance of distributed and centralised solvers

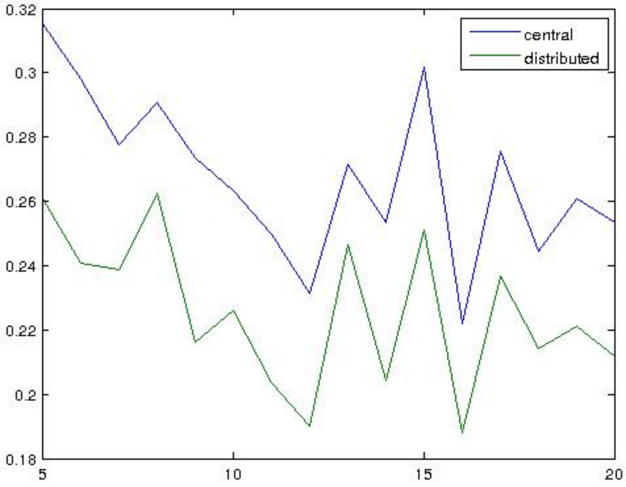


Figure 0.12.16: Mse vs SNR for the sensing model, showing the performance of distributed and centralised solvers

0.13 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).

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 **?**. Specifically, we seek to dampen the ringing seen in Figure 0.12.23

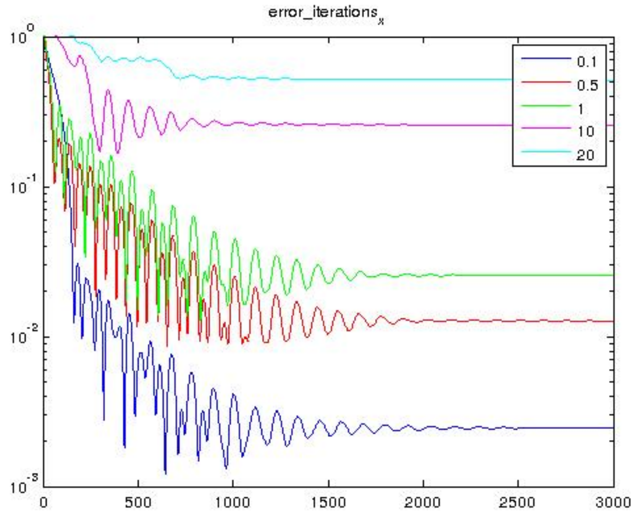


Figure 0.12.17: The progress of the distributed solver as a function of the number of iterations, with different values of the regression parameter λ

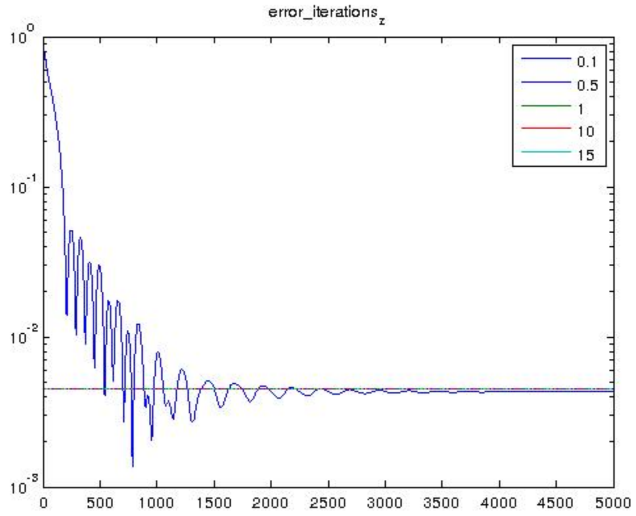


Figure 0.12.18: The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$

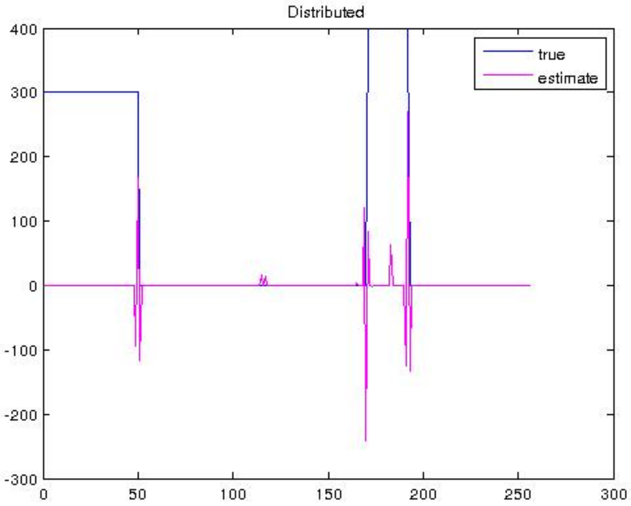


Figure 0.12.19*:* The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$

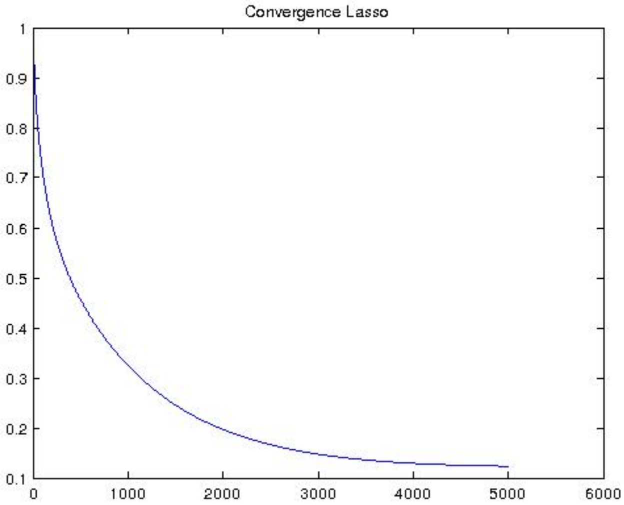


Figure 0.12.20*:* The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$

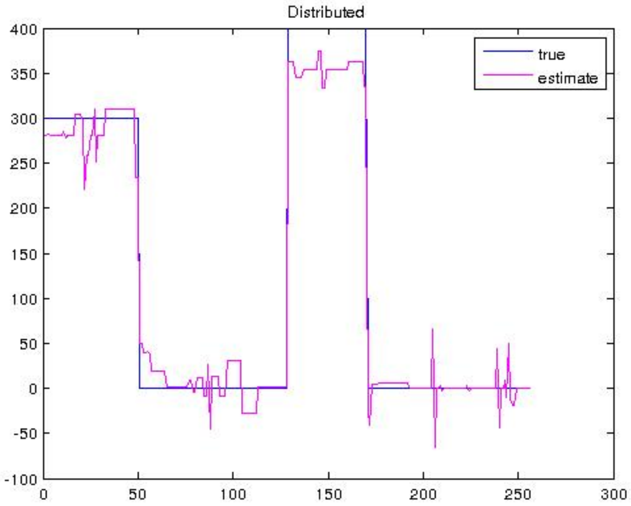


Figure 0.12.21*:* The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$

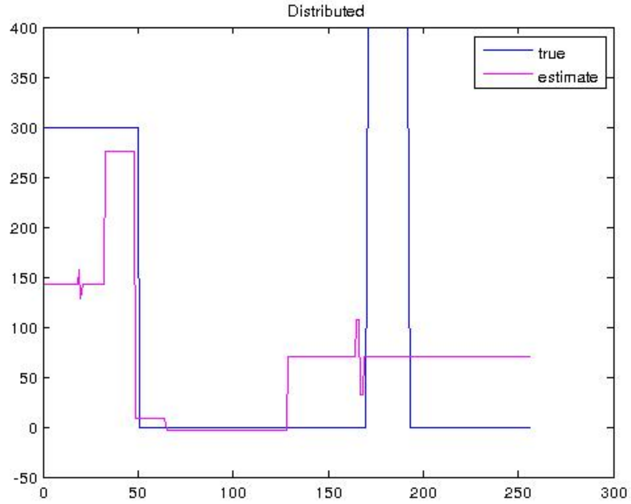


Figure 0.12.22*:* The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$

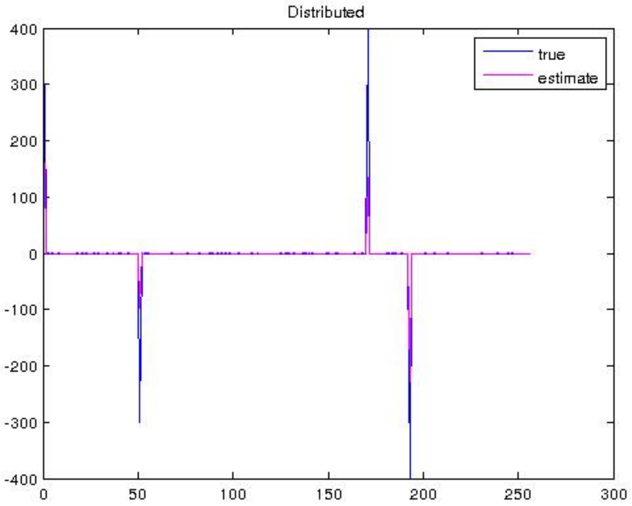


Figure 0.12.23*:* The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$

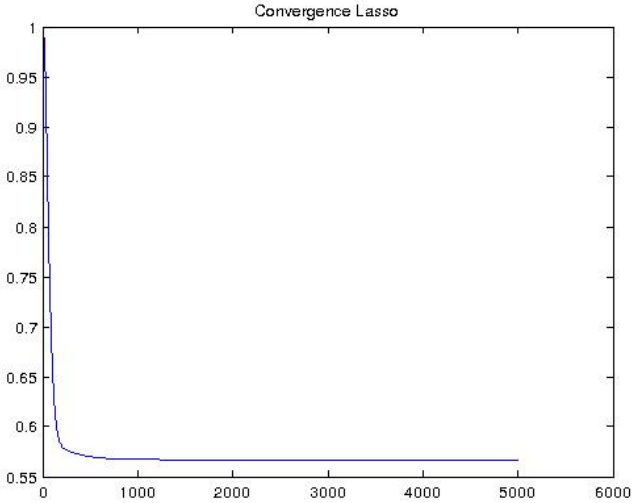


Figure 0.12.24*:* The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$

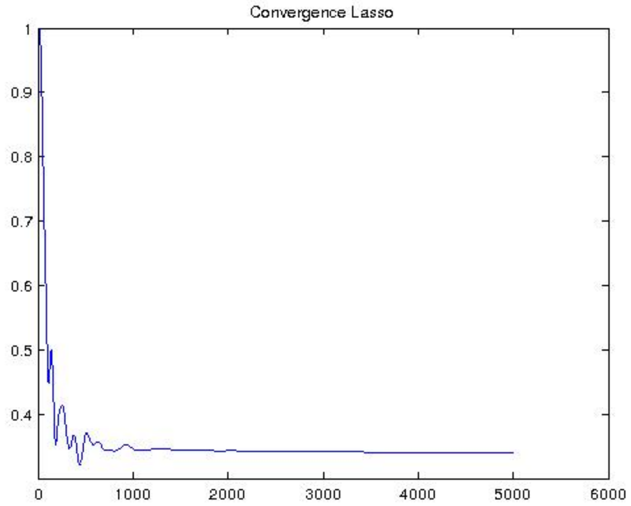


Figure 0.12.25*:* The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$

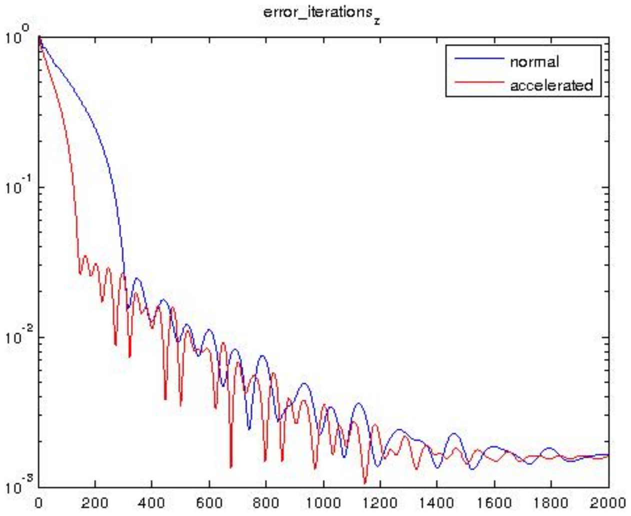


Figure 0.12.26*:* The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$

0.14 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 **?** 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) **?** 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 **?**, and **?** 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 **?**. 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 **?**.

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 **?** - 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 **?**.

The structure of the paper is as follows: in section 0.16 we introduce the sensing model, in section 0.17 we describe the distributed reconstruction algorithm **?**, and finally in section 0.18 we show some results of the reconstruction quality of this model.

0.15 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:

$$l_i(x) = \begin{cases} 1 & \text{if } x \leq i \\ 0 & \text{otherwise} \end{cases} \tag{0.15.0.150}$$

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

The basis (0.22.2.221) can be expressed as a matrix in $\mathbb{R}^{n \times n}$ as:

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 & 0\dots0 \\ 1 & 1 & 0 & 0 & 0\dots0 \\ 1 & 1 & 1 & 0 & 0\dots0 \\ \dots & & & & \\ 1 & 1 & 1 & 1 & 1\dots1 \end{pmatrix} \tag{0.15.0.151}$$

By direct computation, this inverse of L is:

$$D = \begin{pmatrix} 1 & 0 & 0 & 0 & 0\dots0 \\ -1 & 1 & 0 & 0 & 0\dots0 \\ 0 & -1 & 0 & 0 & 0\dots0 \\ \dots & & & & \\ 0 & 0 & 0 & 0\dots-1 & 1 \end{pmatrix} \tag{0.15.0.152}$$

We model our PSD signal g as a linear combination of the basis functions (0.22.2.221):

$$g\left(x\right) = \sum_i a_i l_i\left(x\right) = L^T a \tag{0.15.0.153}$$

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 0.15.1**.**

$$D^T g = a \tag{0.15.0.154}$$

Proof.

$$\begin{aligned} D^T g &= D^T L^T a && (0.15.0.155) \\ &= \left(LD\right)^T a && (0.15.0.156) \\ &= a && (0.15.0.157) \end{aligned}$$

as $LD = I$.

□

0.16 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:

$$y_j = A_j H_j g + w_j \tag{0.16.0.158}$$

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 (0.18), this corresponds to the concatenated system:

$$y = AHg + w \tag{0.16.0.159}$$

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

The system 0.16.0.159 can then be solved (in the sense of finding the sparse vector a (0.22.2.221) by convex optimisation via minimising the objective function:

$$\hat{a} = \arg\min_a \frac{1}{2} \|AHL^T a - y\|_2^2 + \lambda \|a\|_1 \tag{0.16.0.160}$$

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

0.17 Constrained Optimisation on Graphs

We model the network of sensors as an undirected graph $G = (V, E)$, where $V = \{1 \ldots J\}$ 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 $C = \{1 \ldots c\}$, 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 0.16):

$$\mathbf{y}_j = \mathbf{M}_j \mathbf{x} + \mathbf{n}_j \tag{0.17.0.161}$$

where $\mathbf{M}_j = \left(\mathbf{AHL^T}\right)_j$ is the p^{th} row of the sensing matrix from (0.16.0.159).

To find the $\mathbf{x}$ we are seeking (the solution to (0.16.0.160), 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 4. *We define vectors x_c which represent the subset of nodes with colour c, where $c = 1, \ldots, C$, and write the vector of length nJ:*

$$\bar{x} = \sum_{c=1}^{C} w_c \otimes x_c = \left[x_{c(1)}^T, \ldots, x_{c(J)}^T\right]^T \tag{0.17.0.162}$$

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:

$$\begin{aligned} \arg\min_{\bar{x}} \sum_{c=1}^{C} \sum_{j \in c} \|M_j x_j - y_j\|_2^2 + \frac{\lambda}{J}\|z\|_1 \\ \text{and } x_i = x_j \text{ if } \{i, j\} \in E \\ \text{and } x_i - z_i = \ \forall i \in \{1, \ldots, C\} \end{aligned} \tag{0.17.0.163}$$

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 (0.17.0.162) the constraint $x_i = x_j$ if $\{i, j\} \in E$ can now be written:

$$\sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_c = 0 \tag{0.17.0.164}$$

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

Together (0.17.0.162) and (0.17.0.164), suggests that the problem (0.17.0.163) can be rewritten as:

$$\begin{aligned} \arg\min_{\bar{x}} \sum_{c=1}^{C} \sum_{j \in C_c} \|M_j x_j - y_j\|_2^2 + \beta\|z_j\|_1 \\ \text{s.t. } \sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_c = 0 \\ \text{and } \bar{x}_c - \bar{z}_c = 0 \end{aligned} \tag{0.17.0.165}$$

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

The global Augmented Lagrangian **?** for the problem (0.17.0.165) can be written down as:

$$
\begin{aligned}
L_\rho = \sum_{c=1}^{C} \Bigg(\sum_{j \in c} \|M_j x_j - y_j\|_2^2 + \beta \|z_j\|_1 + \\
+ \theta^T \left(\bar{x}_j - \bar{z}_j\right) + \frac{\rho}{2} \left|\left|\bar{x}_j - \bar{z}_j\right|\right|_2^2 \Bigg) + \\
+ \eta^T \left(B_c^T \otimes I_n\right) \bar{x}_c + \frac{\rho}{2} \left|\left| \sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_c \right|\right|_2^2
\end{aligned}
\qquad (0.17.0.166)
$$

This is, superficially, similar to the Augmented Lagrangian for the Lasso problem **?**[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 0.17.0.166 can be posed as a constrained optimisation problem at each node, we introduce the following variable:

Definition 5 (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 (0.17.0.162) *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 (0.17.0.166), can be decomposed across edges, using the following lemma:

Lemma 0.17.1 (Edge Decomposition)**.**

$$
\left|\left| \sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_c \right|\right|^2 = \sum_{j \in C_c} \left(D_j \left|\left|x_j\right|\right|_2^2 - \sum_{k \in \mathcal{N}_j} x_j^T x^k \right) \qquad (0.17.0.167)
$$

and

$$
\eta^T \sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_1 = \sum_{l \in C_c} \sum_{m \in \mathcal{N}_l} \operatorname{sign}(m - l)\, \eta_{ml}^T x_l \qquad (0.17.0.168)
$$

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* $x_i = x_j$.

[1] DADMMy_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**
end
$c = 1, \dots, C$ $x^{k+1} \leftarrow Q\left(w_j + z^k - \theta^{kT} - \nu^{kT}\right)$ $z^{k+1} \leftarrow S_{\beta/\rho}\left(x_j^{k+1}\right)$ $\theta^{k+1} \leftarrow \theta_j^k + \rho\left(x^{k+1} - z^{k+1}\right)$
Each node transmits x^{k+1} in $\mathcal{N}_j$ and calculates $\nu_j^{k+1} \leftarrow \nu_j^k + \rho\left(\sum_{m \in N_j} z_m^k - z_j^k\right)$ **return** z^{k+1}

Figure 0.17.27: The algorithm at Node 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 $B_{c_1}B_{c_2}^T = w_{c_1}^T BB^T w_{c_2}$, the trace of $B_{c_1}B_{c_1}^T$ is simply the sum of the degrees of nodes with colour 1. 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:

$$\eta = \sum_{c=1}^{C} w_c \otimes \eta_c \tag{0.17.0.169}$$

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 - $\sum_{l \in C_c}\sum_{m \in N_l} \operatorname{sign}(m - l)$. □

Adding together this with the lemma, lets us write (0.17.0.166) as:

$$L_\rho = \sum_{c=1}^{C} \Bigg(\sum_{j \in C_c} \|M_j x_j - y_j\|_2^2 + \beta \|z_j\|_1 + \nu^T x_j$$

$$+\theta \left(x_j - z_j\right) + \frac{\rho}{2} D_i \left|\left|x_j\right|\right|^2 + \frac{\rho}{2} \left|\left|x_j - z_j\right|\right|^2 \Bigg) \tag{0.17.0.170}$$

where we have defined:

$$\nu_i = \left(\sum_{k \in \mathcal{N}_i} sign\left(k - i\right) \eta_{\{i,k\}} - \rho x_k \right) \tag{0.17.0.171}$$

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

Then by differentiating (0.17.0.170) with respect to x_j and z_j we can find closed forms for the updates as:

$$x_j^{k+1} := \left(M_j^T M_j + (\rho D_J + 1) I\right)^{-1} \left(M_j^T y_j + z^k - \theta^{kT} - \nu^{kT}\right) \tag{0.17.0.172}$$

$$z_j^{k+1} := S_{\beta/\rho}\left(x_j^{k+1}\right) \tag{0.17.0.173}$$

$$\theta_j^{k+1} := \theta_j^k + \rho\left(x^{k+1} - z^{k+1}\right) \tag{0.17.0.174}$$

$$\nu_j^{k+1} := \nu_j^k + \rho \left(\sum_{m \in \mathcal{N}_j} z_m^k - z_j^k \right) \tag{0.17.0.175}$$

Where we have defined

Definition 6 (Soft-thresholding)**.**

$$S_\tau\left(y\right) := \text{sign}(y)\text{max}(y - |\tau|, 0) \tag{0.17.0.176}$$

Remark 0.17.2. *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.

0.18 **Results**

The model described in section (0.16), equation (0.16.0.159) 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 (0.18.28) (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:

$$\frac{\left|\left|L^t z^k - g^*\right|\right|}{||g^*||} \tag{0.18.0.177}$$

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

$$\frac{||g^*||}{||w||} \tag{0.18.0.178}$$

and averaged over the 500 repetitions. The results are shown in figure (0.18.29). Following **?**, for each repetition we chose

$$\lambda = \sqrt{2\sigma_n^2 \log n} \tag{0.18.0.179}$$

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 **?**; this is due to the differing sparsity models: **?** 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 (0.18.28) (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 reconstructi ons.

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 (0.18.30), 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 **?**, 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 **?** and **?**.

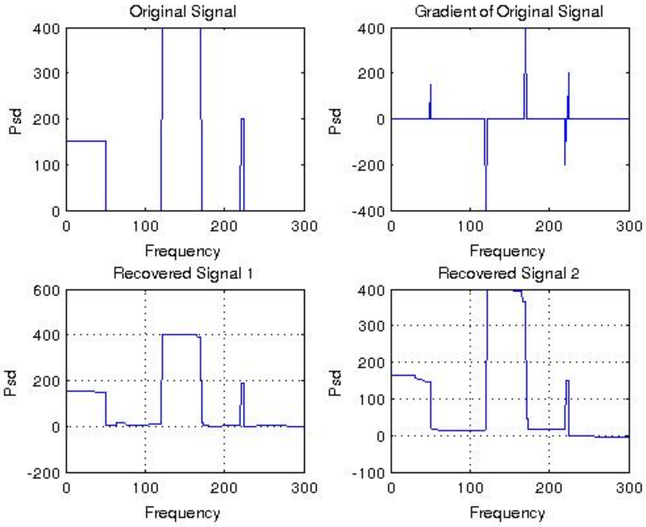


Figure 0.18.28*:* Left to right: (a) The original signal. (b) The gradient (0.15.1) of the original signal. (c) Recovery using DADMM, 1000 iterations, $\sigma_n^2 = 5$. (d) Recovery using DADMM, 1000 iterations, $\sigma_n^2 = 20$

0.19 **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 **?** and **?**)

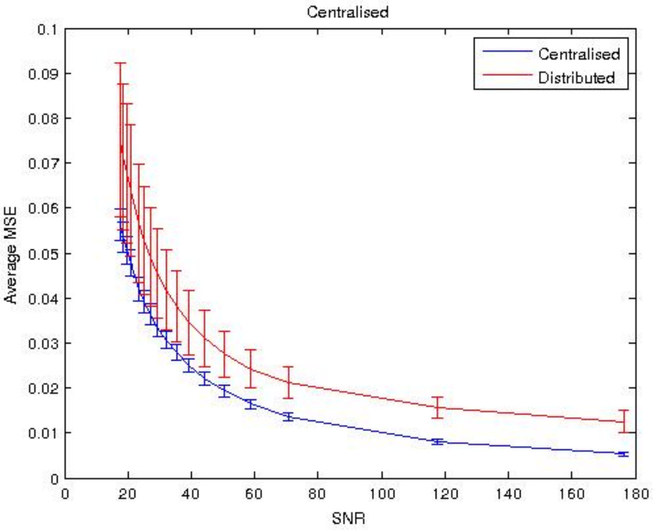


Figure 0.18.29: MSE vs SNR for the sensing model showing the performance of distributed and centralised solvers. The performance of DADMM is consistently within 10^{-2} of ADMM, and within the error bars of ADMM at low SNRs. The variance of estimates produced by DADMM is larger than ADMM, due to nodes 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.

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).

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 **?**, 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.

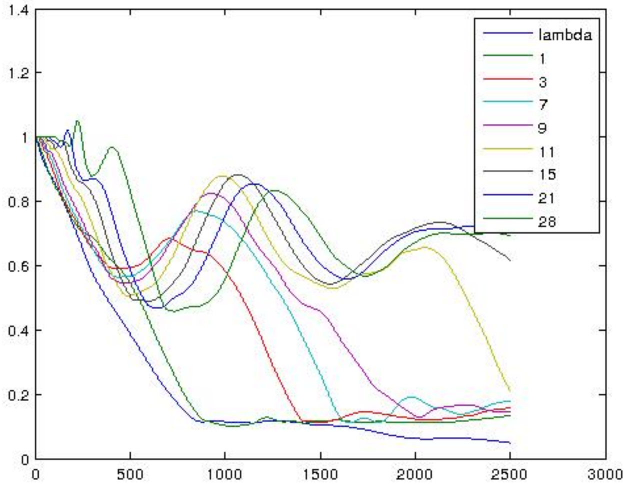


Figure 0.18.30*:* The progress of the distributed solver as a function of the number of iterations, with different values of the regression parameter λ. For a fixed λ there is a single unique optimal solution, with higher λ favouring sparser solutions. The convergence of DADMM is slowed by smaller λ. This is intuitive: solutions with fewer non-zero components should be identified in fewer iterations.

0.20 **Introduction**

The recent work of Candes and Tao **?** and Donoho **?** 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.

$$x\left(nT\right) = \text{Ш}\left(t - nT\right) x\left(t\right) \tag{0.20.0.180}$$

where

$$\text{Ш}\left(t - nT\right) = \sum_{k=-\infty}^{\infty} \delta\left(t - kT\right) \tag{0.20.0.181}$$

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

$$\hat{X}_s\left(f\right) = \sum_{k=-\infty}^{\infty} x\left(t - kT\right) \tag{0.20.0.182}$$

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:

$$x\left(t\right) = \sum_{n=-\infty}^{\infty} x\left(nT\right) sinc\left(\frac{t_n T}{T}\right) \tag{0.20.0.183}$$

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 $w \sim N\left(0, 1\right) \in \mathbb{R}^m$ 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 0.20.1 (RIP)**.** *We say that a matrix A satisifes the RIP of order δ if $\exists$ a $\delta \in (0,1)$ such that:*

$$(1-\delta)\,||x||_2^2 \leq ||Ax||_2^2 \leq (1+\delta)\,||x||_2^2 \tag{0.20.0.184}$$

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

Definition 0.20.2 (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:*

$$\mu(A) = \max_{1\leq i,j\leq n, i\neq j} \frac{|\langle a_i^T, a_j\rangle|}{||a_i||_2\,||a_j||_2} \tag{0.20.0.185}$$

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

Theorem ? 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:

$$m \geq C\mu^2(A) S \log\left(\frac{n}{\delta}\right) \tag{0.20.0.186}$$

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 **?**, that the minimiser of the program:

$$\arg\min_x \frac{1}{2}\,||y - Ax||_2^2 + \lambda\,||x||_1 \tag{0.20.0.187}$$

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 **?** and **?** provide an introductory answer for the cases of filtering, detection, classification and estimation.

The structure of this document is as follows: sections (0.21), is a literature review of relevant material from compressed sensing, Wishart matrices, and maximum likelihood estimation of uncompressed signals in noise. Section (0.22) gives an overview of the problem of estimating a signal from a known set of basis functions.

0.21 Preliminaries

0.21.1 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:

$$y = Ax \tag{0.21.1.188}$$

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 0.21.1.

$$\Sigma_s = \{x \in \mathbb{R}^n : \operatorname{supp}(x) \leq s\} \tag{0.21.1.189}$$

where supp (x) *is the set of indices on which x is non-zero.*

Definition 0.21.2 (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* $x \in \Sigma_s$*:*

$$(1-\delta)\,||x||_2^2 \leq ||Ax||_2^2 \leq (1+\delta)\,||x||_2^2 \tag{0.21.1.190}$$

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

Remark 0.21.3 (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* $x \,/\!= x_2$, $x_1, x_2 \in \Sigma_s$, *which is equivalent to* $||A(x_1 - x_2)||_2^2 > 0$.

This is guaranteed as long as A satisfies the RIP of order 2*s with constant* δ *- as the vector* $x_1 - x_2$ *will hve at most* 2*s non-zero entries, and so will be distinguishable after multiplication with A. To complete the argument take* $x = x_1 - x_2$ *in definition* (0.21.2)*, guaranteeing* $||A(x_1 - x_2)||_2^2 > 0$*, and requiring the RIP order of A to be* 2*s.*

Remark 0.21.4 (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 0.21.5 (δ-stable embedding). *We say that a mapping is a δ-stable embedding of $U, V \subset \mathbb{R}^n$ if*

$$(1-\delta)\,||u-v||_2^2 \le ||Au-Av||_2^2 \le (1+\delta)\,||u-v||_2^2 \qquad (0.21.1.191)$$

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

Remark 0.21.6. *Note that a matrix A, satisfying the RIP of order $2s$ is a δ-stable embedding of Σ_s, Σ_s.*

Remark 0.21.7. *Definition* 0.21.5 has a simple interpretation: the matrix A must approximately preserve Euclidean distances between all points in the signal model Σ_s.

0.21.2 Random Matrix Constructions

To construct matrices satisfying definition 0.21.5, given m, n we generate A by A_{ij} being i.i.d random variables from distributions with the following conditions **?**

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

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

Random variables A_{ij} satisfying conditions (1) and (2) satisfy the following concentration inequality **?**, **?**[Lemma 6.1]:

Condition 3 (sub-Gaussian).

$$\mathbb{P}\left(\mid ||Ax||_2^2 - ||x||_2^2 \mid \ge \varepsilon\,||x||_2^2\right) \le 2e^{-cM\varepsilon^2} \qquad (0.21.2.192)$$

Then in **?** the following theorem is proved:

Theorem 0.21.8. *Suppose that m, n and $0 < \delta < 1$ are given. If the probability distribution generating A satisfies condition* (0.21.2.192)*, then there exist constants c_1, c_2 depending only on δ such that the RIP* (0.21.2) *holds for A with the prescribed δ and any $s \le \frac{c_1 n}{\log n/s}$ with probability $\ge 1 - 2e^{-c_2 n}$*

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

0.21.3 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.

$$X_i = \left(x_1^{(i)}, \ldots, x_p^{(i)}\right) \sim N(0, H) \qquad (0.21.3.193)$$

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

Definition 0.21.9 (Wishart Matrix)**.** *Let*

$$W = \sum_{j=1}^{r} X_j X_j^T = XX^T \tag{0.21.3.194}$$

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

$$W_r\left(H, p\right) \tag{0.21.3.195}$$

where p is the number of degrees of freedom.

Remark 0.21.10**.** *This distribution is a generalisation of the Chi-squared distribution: let $p = H = 1$.*

Theorem 0.21.11 (Expected Value)**.**

$$\mathbb{E}\left(W\right) = rH \tag{0.21.3.196}$$

Proof.

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

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

Remark 0.21.12**.** *The matrix $M = A^T A$, where A is constructed by the methods from section* 0.21.2, 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 **?**:

$$p\left(\lambda_1, \ldots, \lambda_r\right) = c_r \prod_{i=1}^{r} e^{-\lambda_i} \prod_{i<j} \left(\lambda_i - \lambda_j\right)^2 \tag{0.21.3.197}$$

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

0.21.4 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.

$$y = s + n \tag{0.21.4.198}$$

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 $\{\phi_i\}_{i=1}^n$:
$$s = \sum_{i=1}^n \alpha_i \phi_i \tag{0.21.4.199}$$
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:

$$f(y \mid s) = \left(\frac{1}{\sqrt{2\pi}}\right)^n \exp\left(-\frac{(y - s(\Theta))^T (y - s(\Theta))}{2}\right) \tag{0.21.4.200}$$

Maximising this is equivalent to maximising:

$$\ln f(\Theta) = -||y||_2^2 + 2\langle y, s(\Theta)\rangle - ||s||_2^2 \tag{0.21.4.201}$$

Since the terms $||y||_2^2$ and $||s||_2^2$ do not change, we can write the ML estimate of Θ as:

$$\hat{\Theta} = \arg\max_{\Theta} \langle y, s(\Theta)\rangle \tag{0.21.4.202}$$

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:

$$\langle y, \phi_i\rangle = \sum_{j=1}^n \alpha_j \phi_j^T \phi_i + n^T \phi_i \tag{0.21.4.203}$$

$$= \alpha_i + \varepsilon_i \tag{0.21.4.204}$$

where $\epsilon_i = \langle n^T, \phi_i \rangle$ is some small error. Thus the maximum likelihood estimate of s is:

$$\hat{s} = \sum_{i=1}^{n} \hat{\alpha}_i \phi_i \tag{0.21.4.205}$$

where

$$\hat{\alpha} = \langle y, \phi_i \rangle \tag{0.21.4.206}$$

0.22 Compressive Estimation

In this section, we develop some intuition into constructing estimators for the signal s directly on the compressive measurements:

$$y = A(s+n) \tag{0.22.0.207}$$

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):

$$f(y \mid s) = \left(\frac{1}{\sqrt{2\pi}} \right)^m \exp\left(-\frac{(y - As)^T (y - As)}{2} \right) \tag{0.22.0.208}$$

Taking the logarithm and expanding, we find

$$\ln f = -y^T y - s^T A^T A s + 2\langle y, As \rangle \tag{0.22.0.209}$$

which is equal to:

$$lnf = -||y||_2^2 - ||As||_2^2 + 2\langle y, As \rangle \tag{0.22.0.210}$$

The first term of (0.22.0.210) is constant, for the same reasons as in section (0.22). The term

$$||As||_2^2 = \langle As, As \rangle \tag{0.22.0.211}$$

can be written as

$$\langle A^T As, s \rangle \tag{0.22.0.212}$$

We will replace this with its expectation $\mathbb{E}\left(\langle A^T As, s \rangle\right)$

$$\begin{aligned}\mathbb{E}\Big(\langle A^TAs,s\rangle\Big) &= \mathbb{E}\sum_{i=1}^{n}(A^TAs)_i^T s_i \\ &= \sum_{i=1}^{n}\mathbb{E}(A^TAs)_i s_i \\ &= \sum_{i=1}^{n}(\frac{1}{m}e_i s_i)_i^T s_i \\ &= \frac{1}{m}\langle s,s\rangle\end{aligned}$$

because

$$\mathbb{E}A^TA = \frac{1}{m}I \tag{0.22.0.213}$$

as it is a Wishart matrix (see section0.21).
So we can further approximate (0.22.0.210):

$$\ln f = -\,||y||_2^2 + \frac{1}{m}\,||s||_2^2 + 2\langle y, As\rangle \tag{0.22.0.214}$$

The only, non-constant part of (0.22.0.214) is the third term and so we define the estimator:

$$\hat{s} = \arg\max_{\Theta}\langle y, As\,(\Theta)\rangle \tag{0.22.0.215}$$

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:

$$\hat{s} = \sum_{i=1}^{n} m\langle y, A\phi_i\rangle\phi_i \tag{0.22.0.216}$$

Consider the case where $y = As$ (no noise). Then

$$y^TA\phi_j = \sum_i \alpha_i\phi_i^TA^TA\phi_j$$

So

$$y^TA\phi_j = \sum_i \alpha_i\phi_i^TA^TA\phi_j \sim \frac{\alpha_i}{m}\delta_{ij}$$

giving

$$\hat{\alpha_i} = m\left(y^T A \phi_j\right) \tag{0.22.0.217}$$

Remark 0.22.1. *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* (0.21.2.192)*).*

0.22.1 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

$$s = \alpha_i \delta_i \tag{0.22.1.218}$$

with

$$a_i \sim \mathcal{N}(100, 1) \tag{0.22.1.219}$$

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:

$$\hat{\alpha}_j = m\langle (A\alpha_i\delta_i), A\delta_j \rangle \tag{0.22.1.220}$$

We plot the $\hat{\alpha_j}$ below, figure 0.22.31, (red circles), with the original signal (in blue, continuous line). Note how the maximum of the $\hat{\alpha_j}$, coincides with the true signal.

0.22.2 Estimating a single rectangle

We show how to estimate the signal, composed of a single rectangle (0.22.32) expanded in the following basis

$$f_i(x) = \begin{cases} 1 & \text{if } x \le i \\ 0 & \text{otherwise} \end{cases} \tag{0.22.2.221}$$

We model our signal g as a linear combination of the basis functions (0.22.2.221):

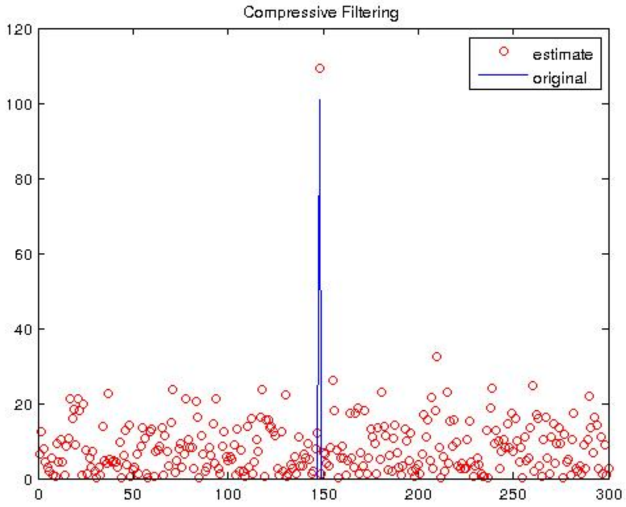


Figure 0.22.31

$$g\left(x\right)=\sum_{i}a_{i}f_{i} \tag{0.22.2.222}$$

To find the a_i, we correlate (take the inner product of) the signal against the basis (0.22.2.221).

Definition 0.22.2.

$$\begin{aligned}h_{j} &= \langle g,f_{j}\rangle && (0.22.2.223)\\ &= \sum_{j}g\left(x\right)f_{j}\left(x\right) && (0.22.2.224)\\ &= \sum_{j}a_{i}f_{i}\left(x\right)f_{j}\left(x\right) && (0.22.2.225)\\ &= a_{i}\langle f_{i},f_{j}\rangle && (0.22.2.226)\\ &\left(=\sum_{x=1}^{j}g\left(x\right)\right) && (0.22.2.227)\end{aligned}$$

As per the previous sections, we take compressive measurements:

$$y=Ag \tag{0.22.2.228}$$

where $A\in\mathbb{R}^{m\times n}$, $A_{ij}\sim\mathcal{N}\left(0,1/m\right)$, and then compute

$$\langle y,Af_{i}\rangle=a_{j}f_{j}^{T}A^{t}Af_{j}\sim\frac{a_{j}}{m}\langle f_{j},f_{i}\rangle \tag{0.22.2.229}$$

for the set of basis vectors $f_1\ldots f_n$ i.e. the estimator from the previous section, corresponding to this set of basis functions (0.22.0.215).

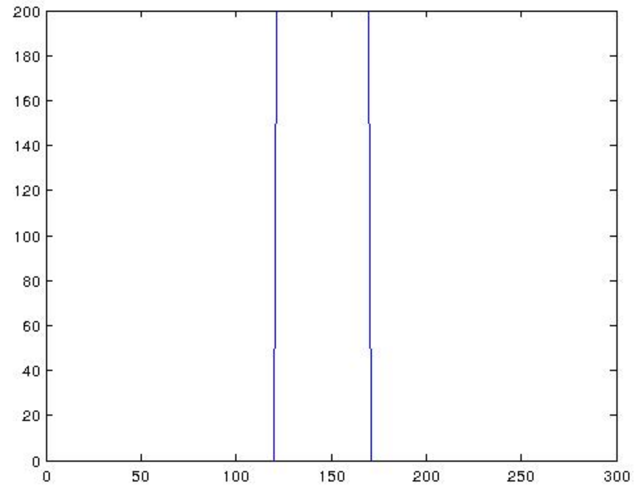


Figure 0.22.32

We then form the vector

$$\hat{h} = m \sum_i \langle y, Af_i \rangle f_j \sim h \tag{0.22.2.230}$$

An example can be seen in figure 0.22.33, for a matrix $A \in \mathbb{R}^{200 \times 300}$.

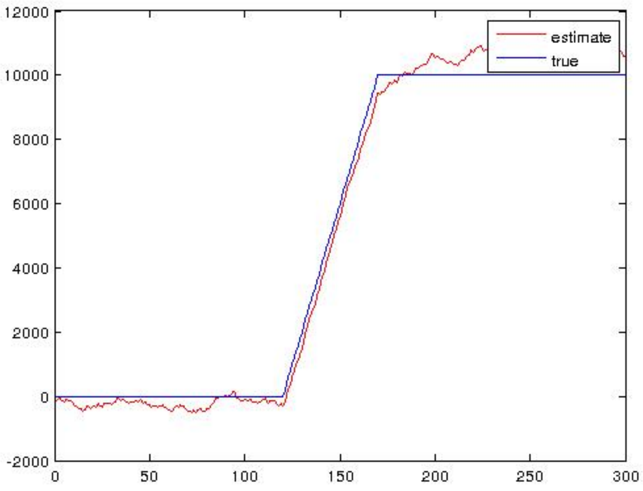


Figure 0.22.33

0.22.3 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.

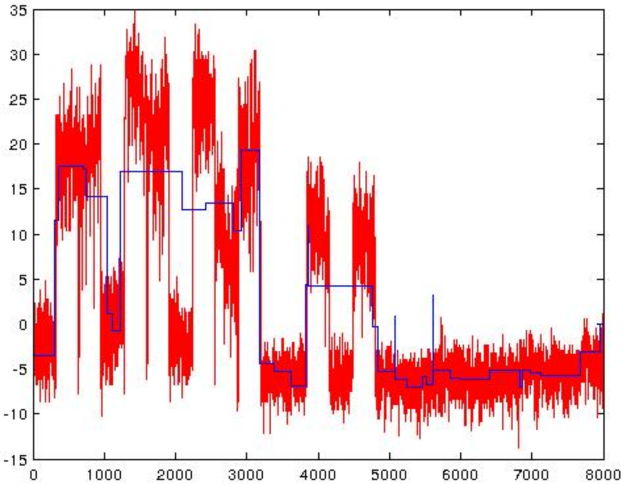


Figure 0.22.34

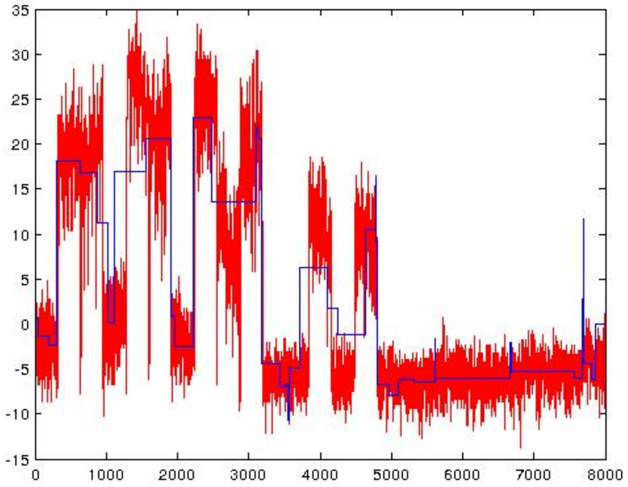


Figure 0.22.35

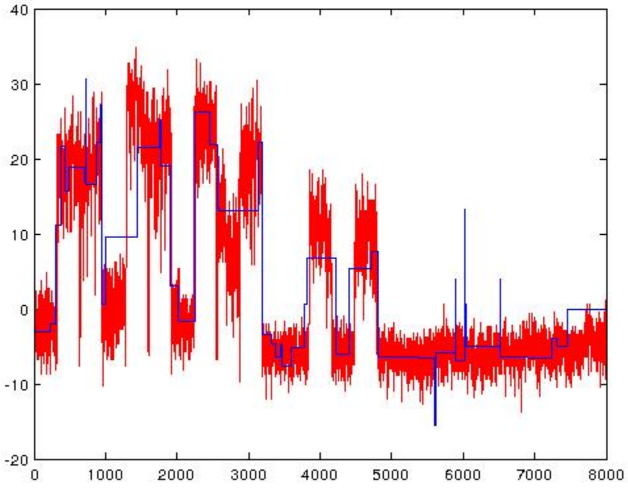


Figure 0.22.36

0.23 Introduction and notation

0.23.1 The Probabilistic group testing problem

Group testing is a sparse inference problem, first introduced by Dorfman **?** in the context of testing for rare diseases. Given a large population of items $\mathcal{P}$, indexed by $\{1, \ldots N\}$, 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 $\mathcal{X}_1, \ldots, \mathcal{X}_T$, where each $\mathcal{X}_i \subseteq \mathcal{P}$. We represent the interesting ('defective') items by a random vector $\mathbf{U} = (U_1, \ldots, U_N)$, 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 $\mathcal{X}_i$, and the outcome y_i is 'positive' ($y_i = 1$) if and only if any item in $\mathcal{X}_i$ is defective. In other words, $y_i = \mathbb{I}\left(\sum_{j \in \mathcal{X}_i} U_j\right)$, 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 $\mathcal{X}_i$ based on a knowledge of sets $\mathcal{X}_1, \ldots, \mathcal{X}_{i-1}$ and outcomes $y_1, \ldots, y_{i-1}$. The group testing problem requires us to infer $\mathbf{U}$ with high probability given a low number of tests T.

Since Dorfman's paper **?**, there has been considerable work on the question of how to design the sets $\mathcal{X}_i$ in order to minimise the number of tests T required. In this context, we briefly mention so-called combinatorial designs (see **??** for a summary, with **?** 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 $\mathcal{X}_i$ are chosen at random, either using constructions such as independent Bernoulli designs **???** or more sophisticated random designs based on LDPC codes **?**.

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. **?**, 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 **?**, 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 **?** or more recent research **?** involving screening for genetic conditions, values of p_i might summarise prior information based on a risk profile or family history.

0.23.2 Group testing capacity

It is possible to characterize performance tradeoffs in group testing from an information-theoretic point of view – see for example **????**. These papers have focussed on group testing as a channel coding problem, with **??** explicitly calculating the mutual information. The paper **?** 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 0.23.1. *Consider a sequence of group testing problems where the ith 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* $\epsilon > 0$*:*

(a) any sequence of algorithms with

$$\liminf_{i\to\infty} \frac{H(\mathbf{U}^{(i)})}{T(i)} \geq C + \epsilon, \qquad (0.23.2.231)$$

has success probability $\mathbb{P}(\mathrm{suc})$ *bounded away from 1,*

(b) and there exists a sequence of algorithms with

$$\liminf_{i\to\infty} \frac{H(\mathbf{U}^{(i)})}{T(i)} \geq C - \epsilon \qquad (0.23.2.232)$$

with success probability $\mathbb{P}(\mathrm{suc}) \to 1$.

Remark 0.23.2. *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 (***?***, Eq. (1) and (2)). In the Probabilistic case (as in* **?***) 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 0.23.3. *If for* $\liminf_{i\to\infty} \frac{H(\mathbf{U}^{(i)})}{T(i)} \geq C + \epsilon$*, the success probability* $\mathbb{P}(\mathrm{suc}) \to 0$ *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.*

0.23.3 Main results

The principal contribution of (**?**, Theorem 1.2) was the following result:

Theorem 0.23.4 (**?**). *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 (**?**, Theorem 3.1) the authors proved a new upper bound on success probability

$$\mathbb{P}(\mathrm{suc}) \leq \frac{2^T}{\binom{N}{K}}, \qquad (0.23.3.233)$$

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) **?**, **?** 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 **?**, 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 0.25.9, stated and proved in Section 0.25.5 below, which implies the following Probabilistic group testing version of Theorem 0.23.4.

Corollary 0.23.5. *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 0.25.9 using complementary bounds on both sides. First in Section 0.24.1 we recall a universal upper bound on success probability, Theorem 0.24.1, taken from **?**, which implies a weak converse. In **?**, Li et al. introduce the Laminar Algorithm for Probabilistic group testing. In Section 0.24.3 we propose a refined version of this Laminar Algorithm, based on Hwang's HGBSA **?**, which is analysed in Section 0.25.5, and shown to imply performance close to optimal in the sense of capacity.

0.24 Algorithms and existing results

0.24.1 Upper bounds on success probability

Firstly (**?**, Theorem 1) can be restated to give the following upper bound on success probability:

Theorem 0.24.1. *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 0.23.1, 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 0.24.1, any such algorithms have $\mathbb{P}(\text{suc}) \leq 1/(1 + \epsilon)$, meaning that we have established that a weak converse holds.

Remark 0.24.2. *It remains an open and interesting problem to prove an equivalent of* (0.23.3.233) *as in (**?**, Theorem 3.1). 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.*

0.24.2 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 $\mathcal{P}$ into search sets $\mathcal{S}$. First, all the items in a search set $\mathcal{S}$ are tested together, using a test set $\mathcal{X}_1 = \mathcal{S}$. If the result is negative ($y_1 = 0$), we can be certain that $\mathcal{S}$ contains no defectives. However, if the result is positive ($y_1 = 1$), $\mathcal{S}$ 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 $\mathcal{S}$ in two, and test the 'left-hand' set, say $\mathcal{X}_2$. If $y_2 = 1$, then we know that $\mathcal{X}_2$ contains at least one defective. If $y_2 = 0$, then $\mathcal{X}_2$ contains no defective, so we can deduce that $\mathcal{S} \setminus \mathcal{X}_2$ 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 $\mathcal{X}_i$ is of size 1, and we have isolated a single defective item.

However this strategy may not find every defective item in $\mathcal{S}$. 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 **?** 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 (**?**, Theorem 2):

Theorem 0.24.3**.** *The expected number of tests required by the Laminar Algorithm* **?** *is* $\leq 2H(\mathbf{U}) + 2\mu$*. Under a technical condition (referred to as non-skewedness), the success probability can be bounded by* $\mathbb{P}(\mathrm{suc}) \geq 1 - \epsilon$ *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 **?**) $T = 2e(3+2)H(\mathbf{U}) = 10eH(\mathbf{U})$ 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.

0.24.3 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 0.25.7 (in terms of expected number of tests) and Theorem 0.25.9 (in terms of error probabilities). The key ideas are:

Algorithm 1: Algorithm for the non-iid group testing problem

```
Data: A Set $S$ of $|S| = n$ items, $\mu$ 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 $\theta$
Result: The 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
```

(a) To partition the population $\mathcal{P}$ into search sets $\mathcal{S}$ containing items which have similar probabilities, expressed through the Bounded Ratio Condition 4. This is discussed in Section 0.25.1, and optimised in the proof of Proposition 0.25.7.

(b) The way in which we deal with sets $\mathcal{S}$ which contain more than one defective, as discussed in Remark 0.25.2 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.

(c) 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 **?** and discussed in Section 0.25.2, with a new bound given in Lemma 0.25.4.

(d) Careful analysis in Section 0.25.4 of the properties of search sets $\mathcal{S}$ gives Proposition 0.25.7, which shows that the expected number of tests required can be expressed as $H(\mathbf{U})$ plus an error term. In Section 0.25.5, we give an analysis of the error probability using Bernstein's inequality, Theorem 0.25.8, allowing us to prove Theorem 0.25.9.

0.24.4 **Wider context: sparse inference problems**

Recent work **??** 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 $\{a, \dots, z\}$.

Often, once a signal has been acquired it will be compressed. However, the compressive sensing paradigm introduced by **??** 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 **??**).

0.25 Analysis and new bounds

0.25.1 Searching a set of bounded ratio

Recall that we have a population $\mathcal{P}$ of items to test, each with associated probability of defectiveness p_i. The strategy of the proof is to partition $\mathcal{P}$ into search sets $\mathcal{S}_1, \ldots, \mathcal{S}_G$, each of which contains items which have comparable values of p_i.

Condition 4 (Bounded Ratio Condition)**.** *Given $C \geq 1$, say that a set $\mathcal{S}$ satisfies the Bounded Ratio Condition with constant C if*

$$\max_{i,j \in \mathcal{S}} \frac{p_j}{p_i} \leq C. \tag{0.25.1.234}$$

(For example clearly if $p_i \equiv p$, any set $\mathcal{S}$ satisfies the condition for any $C \geq 1$).

Lemma 0.25.1**.** *Consider a set $\mathcal{S}$ satisfying the Bounded Ratio Condition with constant C and write $P_{\mathcal{S}} = \sum_{j \in \mathcal{S}} p_j$. In a Shannon–Fano tree for the probability distribution $\overline{p}_i := p_i / P_{\mathcal{S}}$, each item has length $\ell_i^{(\mathcal{S})}$ bounded by*

$$\ell_i^{(\mathcal{S})} \leq \ell_{\max}^{(\mathcal{S})} := \frac{h(\mathcal{S})}{P_{\mathcal{S}}} + \log_2 C + \log_2 P_{\mathcal{S}} + 1, \tag{0.25.1.235}$$

where we write $h(\mathcal{S}) := -\sum_{j \in \mathcal{S}} p_j \log_2 p_j$.

Under the Bounded Ratio Condition, for any i and j, we know that by taking logs of (0.25.1.234)

$$-\log_2 p_i \leq -\log_2 p_j + \log_2 C.$$

Multiplying by p_j and summing over all $j \in \mathcal{S}$, we obtain that

$$-P_{\mathcal{S}} \log_2 p_i \leq h(\mathcal{S}) + P_{\mathcal{S}} \log_2 C. \tag{0.25.1.236}$$

Now, the Shannon–Fano length of the ith item is

$$\begin{aligned} \ell_i^{(\mathcal{S})} = \lceil -\log_2 \overline{p}_i \rceil &\leq -\log_2 p_i + \log_2 P_{\mathcal{S}} + 1 \qquad (0.25.1.237) \\ &\leq \left(\frac{h(\mathcal{S})}{P_{\mathcal{S}}} + \log_2 C \right) + \log_2 P_{\mathcal{S}} + 1. \end{aligned}$$

and the result follows by (0.25.1.236).

Next we describe our search strategy:

Remark 0.25.2. *Our version of the algorithm will find every defective in a set $\mathcal{S}$. We start as before by testing every item in $\mathcal{S}$ 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 $\mathcal{S} \setminus \{d_1\}$ 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 $\mathcal{S} \setminus \{d_1, d_2\}$ 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 0.25.3. *Consider a search set $\mathcal{S}$ satisfying the Bounded Ratio Condition and write $P_{\mathcal{S}} = \sum_{j \in \mathcal{S}} p_j$. If (independently) item i is defective with probability p_i, we can recover all defective items in the set using $T_{\mathcal{S}}$ tests, where $\mathbb{E} T_{\mathcal{S}} \leq T_{\rm bd}(\mathcal{S})$ for*

$$T_{\rm bd}(\mathcal{S}) := h(\mathcal{S}) + P_{\mathcal{S}} \log_2 C + P_{\mathcal{S}} \log_2 P_{\mathcal{S}} + P_{\mathcal{S}} + 1. \qquad (0.25.1.238)$$

Using the algorithm of Remark 0.25.2, laid out on the Shannon-Fano tree constructed in Lemma 0.25.1, 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 $\ell_{\max}^{(\mathcal{S})}$ given in (0.25.1.235).

Recall that we write U_i for the indicator of the event that the ith item is defective, $U_{\mathcal{S}} = \sum_{i \in \mathcal{S}} U_i$ and $l_i^{(\mathcal{S})}$ for the length of the word in the Shannon Fano tree. As discussed in Remark 0.25.2 this search procedure will take

$$\begin{aligned} T_{\mathcal{S}} &= 1 + \sum_{i \in \mathcal{S}} U_i \ell_i^{(\mathcal{S})} \\ &= \sum_{i \in \mathcal{S}} p_i \ell_i^{(\mathcal{S})} + 1 + \sum_{i \in S} \ell_i^{(\mathcal{S})} (U_i - p_i) \\ &\leq \sum_{i \in \mathcal{S}} p_i \ell_{\max}^{(\mathcal{S})} + 1 + \sum_{i \in \mathcal{S}} V_i^{(\mathcal{S})} \\ &= P_{\mathcal{S}} \ell_{\max}^{(\mathcal{S})} + 1 + \sum_{i \in S} V_i^{(\mathcal{S})} \\ &\leq T_{\rm bd}(\mathcal{S}) + \sum_{i \in S} V_i^{(\mathcal{S})} \quad \text{tests.} \qquad (0.25.1.239) \end{aligned}$$

Here we write $V_i^{(\mathcal{S})} = \ell_i^{(\mathcal{S})}(U_i - p_i)$, which has expectation zero, and (0.25.1.239) follows using the expression for $\ell_{\max}^{(\mathcal{S})}$ given in Lemma 0.25.1.

0.25.2 Discarding low probability items

As in **?**, we use a probability threshold θ, and write $\mathcal{P}^*$ for the population having removed items with $p_i \leq \theta$. If an item lies in $\mathcal{P} \setminus \mathcal{P}^*$ 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

$\mathcal{P} \setminus \mathcal{P}^*$ is defective. By the union bound, this truncation operation contributes a total of $\mathbb{P}(\mathcal{P} \setminus \mathcal{P}^* \text{ contains a defective}) \leq \rho := \sum_{i=1}^n p_i \mathbb{I}(p_i \leq \theta)$ to the error probability.

Lemma 0.25.4. *Choosing $\theta(P_e)$ such that*

$$-\log_2 \theta(P_e) = \min\left(\log_2\left(\frac{2n}{P_e}\right), \frac{2H(\mathbf{U})}{P_e}\right) \tag{0.25.2.240}$$

ensures that

$$\mathbb{P}(\ \mathcal{P} \setminus \mathcal{P}^* \textit{ contains a defective}) \leq P_e/2. \tag{0.25.2.241}$$

The approach of **?** is essentially to bound $\mathbb{I}(p_i \leq \theta) \leq \theta/p_i$ so that $\rho = \sum_{i=1}^n p_i \mathbb{I}(p_i \leq \theta) \leq \sum_{i=1}^n p_i(\theta/p_i) = n\theta$. 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: $\mathbb{I}(p_i \leq \theta) \leq (-\log_2 p_i)/(-\log_2 \theta)$, so that

$$\rho = \sum_{i=1}^n p_i \mathbb{I}(p_i \leq \theta) \leq \sum_{i=1}^n p_i \left(\frac{-\log_2 p_i}{-\log_2 \theta}\right) \leq \frac{H(\mathbf{U})}{-\log_2 \theta},$$

so we deduce the result.

0.25.3 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 $1/(2C^B) \leq \theta < 1/(2C^{B-1})$, meaning that $(B-1)\log_2 C < -\log_2(2\theta)$, so

$$B \leq \frac{-\log_2(2\theta)}{\log_2 C} + 1. \tag{0.25.3.242}$$

We split the items in each bin into search sets $\mathcal{S}_i$, motivated by the following definition:

Definition 0.25.5. *A set of items $\mathcal{S}$ is said to be full if $P_{\mathcal{S}} = \sum_{i \in \mathcal{S}} p_i \geq \frac{1}{2}$.*

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 0.25.6. *This splitting procedure will divide $\mathcal{P}^*$ into search sets $\mathcal{S}_1, \ldots, \mathcal{S}_G$, where the total number of sets is*

$$G \leq 2\mu + B \leq 2\mu + \left(\frac{-\log_2(2\theta)}{\log_2 C} + 1 \right).$$

Each set $\mathcal{S}_j$ satisfies the Bounded Ratio Condition and has total probability $P_j := P_{\mathcal{S}_j} \leq 1$.

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 4 with constant C, and have probability satisfying $P_j \leq 1$. For each of bins $1, \ldots, B$:

(a) 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.

(b) For each resulting set $\mathcal{S}_j$, the total probability $P_j \leq 1$ (since just before we add the final item, $\mathcal{S}_j$ is not full, so at that stage has total probability $\leq 1/2$, and each element in bins $1, \ldots, B$ has probability $\leq 1/2$).

(c) Since each set $\mathcal{S}_j$ 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

$$\mu = \sum_{i \in \mathcal{P}} p_i \geq \sum_{i \in \mathcal{P}^*} p_i = \sum_{j=1}^{G} P_j \geq \sum_{j:\ \mathcal{S}_j \text{ full}} P_j \geq \left| \mathcal{S}_j \text{ full} \right| \frac{1}{2}. \tag{0.25.3.243}$$

Since, as discussed in point 6a) above, the total number of sets is bounded by the number of full sets plus B, the result follows using Equation (0.25.3.242).

0.25.4 Bounding the expected number of tests

We allow the algorithm to work until all defectives in $\mathcal{P}^*$ are found, and write T for the (random) number of tests this takes.

Proposition 0.25.7. *Given a population $\mathcal{P}$ where (independently) item i is defective with probability p_i, we recover all defective items in $\mathcal{P}^*$ in T tests with $\mathbb{E}T \leq T_{\mathrm{bd}}$, where*

$$T_{\mathrm{bd}} := (H(\mathbf{U}) + 3\mu + 1) + 2\sqrt{\mu\left(-\log_2(2\theta)\right)}. \tag{0.25.4.244}$$

Given a value of C, Proposition 0.25.6 shows that our splitting procedure divides $\mathcal{P}^*$ into G sets $\mathcal{S}_1, \ldots, \mathcal{S}_G$, such that each set $\mathcal{S}_j$ satisfies the Bounded Ratio Condition with constant C and has total probability $P_j \leq 1$. Using the notation of Lemma 0.25.3, $T = \sum_{j=1}^{G} T_{\mathcal{S}_j}$, where $\mathbb{E}T_{\mathcal{S}_j} \leq T_{\mathrm{bd}}(\mathcal{S}_j)$.

Adding this bound over the different sets, since $P_j \leq 1$ means that $P_j \log_2 P_j \leq 0$, we obtain

$$\begin{aligned}
&\sum_{j=1}^{G} T_{\text{bd}}(\mathcal{S}_j) \\
&\leq \sum_{j=1}^{G} \left(h(\mathcal{S}_j) + P_j(\log_2 C + 1) + 1\right) \\
&= \sum_{j \in \mathcal{P}^*} -p_j \log_2 p_j + \mu(\log_2 C + 1) + G \\
&\leq \sum_{j \in \mathcal{P}^*} h(p_j) + 3\mu + 1 + \left(\frac{-\log_2(2\theta)}{\log_2 C} + \mu \log_2 C\right) \\
&\leq (H(\mathbf{U}) + 3\mu + 1) + \left(\frac{-\log_2(2\theta)}{\log_2 C} + \mu \log_2 C\right). \qquad (0.25.4.245)
\end{aligned}$$

This follows by the bound on G in Proposition 0.25.6, 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 (0.25.4.245). Differentiation shows that the optimal C satisfies $\log_2 C = \sqrt{-\log_2(2\theta)/\mu}$, meaning that the bracketed term

$$\frac{-\log_2(2\theta)}{\log_2 C} + \mu \log_2 C = 2\sqrt{\mu\left(-\log_2(2\theta)\right)},$$

and the result follows.

0.25.5 **Controlling the error probabilities**

Although Section 0.25.4 proves that $\mathbb{E}T \leq T_{\text{bd}}$, to bound the capacity, we need to prove that with high probability T is not significantly larger than T_{bd}. This can be done using Bernstein's inequality (see for example Theorem 2.8 of **?**):

Theorem 0.25.8 (Bernstein). *For zero-mean random variables V_i which are uniformly bounded by $|V_i| \leq M$, if we write $L := \sum_{j=1}^{n} \mathbb{E}V_j^2$ then*

$$\mathbb{P}\left(\sum_{j=1}^{n} V_j \geq t\right) \leq \exp\left(-\frac{t^2}{4L}\right), \textit{ for any } 0 \leq t \leq \tfrac{L}{M}. \qquad (0.25.5.246)$$

We deduce the following result:

Theorem 0.25.9. *Write $L = \sum_{j \in \mathcal{P}^*} l_j^2 p_j(1 - p_j)$, $M = -\log_2 \theta + 1$ and $\psi = (L/(4M^2))^{-1/3}$. Define*

$$T_{\text{nec}} = T_{\text{bd}} + \psi H(\mathbf{U}), \qquad (0.25.5.247)$$

where T_{bd} is given in (0.25.4.244).

(a) *If we terminate our group testing algorithm after T_{nec} tests, the success probability*

$$\mathbb{P}(\text{suc}) \geq 1 - \frac{1}{2}\sqrt{\frac{\mu}{H(\mathbf{U})}} - \exp\left(-\left(\frac{L}{4M^2}\right)^{1/3}\right). \qquad (0.25.5.248)$$

(b) *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)* $\liminf H(\mathbf{U})/T_{\text{nec}} \geq 1/(1+\epsilon)$ *for any ϵ and (b)* $\mathbb{P}(\text{suc}) \to 1$, *so the capacity* $C = 1$.

We first prove the success probability bound (0.25.5.248). Recall that our algorithm searches the reduced population set $\mathcal{P}^*$ for defectives. This gives two error events – either there are defective items in the set $\mathcal{P} \setminus \mathcal{P}^*$, or the algorithm does not find all the defectives in $\mathcal{P}^*$ using T_{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 0.25.4) the first event has probability $\leq P_e/2$, contributing $\frac{1}{2}\sqrt{\mu/H(\mathbf{U})}$ to (0.25.5.248).

Our analysis of the second error event is based on the random term from Equation (0.25.1.239), which we previously averaged over but now wish to bound. There will be an error if $T_{\text{nec}} \leq T$, or (rearranging) if

$$\psi H \leq T - T_{\text{bd}} \leq \sum_{j=1}^{G}\left(T_{\mathcal{S}_j} - T_{\text{bd}}(\mathcal{S}_j)\right) = \sum_{i \in \mathcal{P}^*} V_i.$$

For brevity, for $i \in \mathcal{S}$, we write $V_i = V_i^{(\mathcal{S})} = \ell_i^{(\mathcal{S})}(U_i - p_i)$ and $\ell_i = \ell_i^{(\mathcal{S})}$, where V_i has expectation zero.

We have discarded elements with probability below θ, as given by (0.25.2.240), and by design all the sets $\mathcal{S}$ have total probability $P_{\mathcal{S}} \leq 1$. Using (0.25.1.237) we know that the V_i are bounded by

$$|V_i| \leq \ell_i \leq -\log_2 p_i + \log_2 P_{\mathcal{S}} + 1 \leq -\log_2 \theta + 1. \qquad (0.25.5.249)$$

Hence, the conditions of Bernstein's inequality, Theorem 0.25.8, are satisfied. Observe that since all $l_j \leq M$,

$$\frac{L}{HM} = \frac{\sum_{j \in \mathcal{P}^*} l_j^2 p_j (1-p_j)}{HM} \leq \frac{\sum_{j \in \mathcal{P}^*} l_j p_j}{H} \leq 1.$$

Hence Theorem 0.25.8 gives that

$$\begin{aligned}
\mathbb{P}\left(\sum_{j \in \mathcal{P}^*} V_j \geq \psi H\right) &\leq \mathbb{P}\left(\sum_{j \in \mathcal{P}^*} V_j \geq \psi L/M\right) \\
&\leq \exp\left(-\frac{L\psi^2}{4M^2}\right) \\
&= \exp\left(-\left(\frac{L}{4M^2}\right)^{1/3}\right).
\end{aligned}$$

Using the union bound, the probability bound (0.25.5.248) follows.

We next consider the capacity bound of 6b). Since $-\log_2 \theta \leq 2H/P_e$, using (0.25.4.244) and (0.25.5.247)

$$\begin{aligned}\frac{T_{\text{nec}}}{H} &= \frac{T_{\text{bd}}}{H} + \psi \\ &= 1 + 3\frac{\mu}{H} + \frac{1}{H} + 2\sqrt{\frac{\mu}{HP_e}} + \psi \\ &= 1 + 3\frac{\mu}{H} + \frac{1}{H} + 2\left(\frac{\mu}{H}\right)^{1/4} + \psi, \end{aligned} \tag{0.25.5.250}$$

which in our regime of interest is $\leq 1 + \epsilon$ in the limit.

[Proof of Corollary 0.23.5] 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.

0.26 **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 (0.24.1).

Figures (0.26.38) and (0.26.39) show that the performance is relatively insensitive to the cutoff 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).

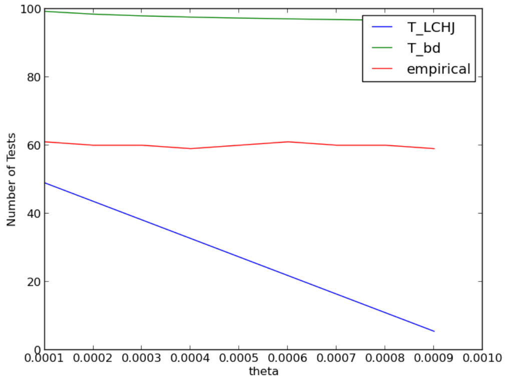


Figure 0.26.37*:* Theoretical lower and upper bounds and empirical Test frequencies as functions of θ

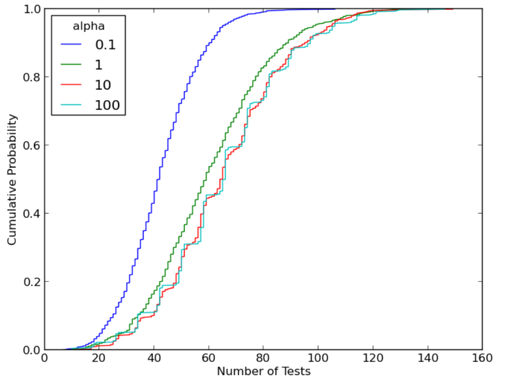


Figure 0.26.38*:* Cumulative distribution curves of the modified Hwang algorithm with fixed $\theta = 0.0001$ and α varying

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.

0.27 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 **?**, 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 (0.25.4.244) which has leading term $H(\mathbf{U})$, and not on tightening bounds on the coefficient of μ in (0.25.4.244). For completeness, we mention that this coefficient can be reduced from 3, under a simple further condition:

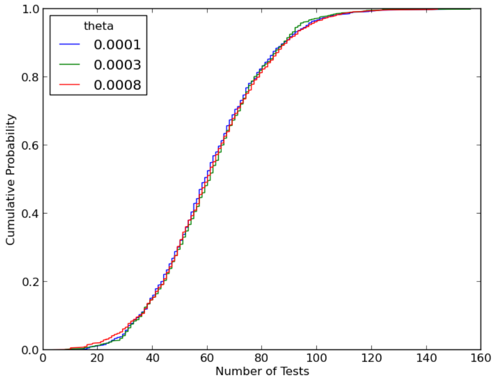


Figure 0.26.39*:* Cumulative distribution curves for fixed $\alpha = 1$ and varying θ

Remark 0.27.1**.** *For some $c \leq 1/2$, we assume that all the $p_i \leq c$, and we alter the definition of 'fullness' to assume that a set is full if it has total probability less than α. In this case, the term $P_{\mathcal{S}} \log_2 P_{\mathcal{S}}$ in* (0.25.1.238) *becomes $P_{\mathcal{S}} \log_2(\alpha + c)$, the bound in* (0.25.3.243) *becomes μ/α, and since $((1-p)\log_2(1-p))/p$ is decreasing in p, we can add a term $(1-c)\log_2(1-c)$ to* (0.25.4.245)*. 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 0.24.1, 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.