

Group Testing and Spectrum Sensing

Tom Kealy

September 9, 2014

Introduction

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 bar underneath the blocks indicates Government use). Compare this to the actual utilisation of

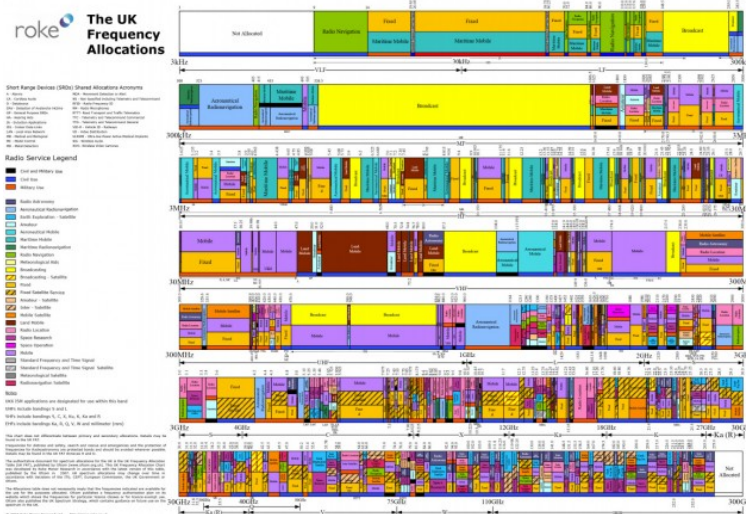


Figure 1.1: A diagram of current Spectral allocation [22]

spectrum: much of it is not used at all. Figure 1.2 shows a snapshot of frequency utilisation in three diverse locations in the UK over the radio spectrum, note that many frequencies are not utilised (coloured blue) whilst others have significant activity (coloured yellow). Note that the plot for Southwark (central London) is barely different from Braddock - a rural area.

How do we then go about solving this issue - how can we obtain the most significant bits of information from our sensing mechanism, whilst obviating the need to compress the data once we are done? How do we dynamically assign spectrum? The work of Candes, Tao [8] and Donoho [11], 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 its 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 ([13] 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 be 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 still providing good bandwidth for data transmission. These TV frequencies are being opened up

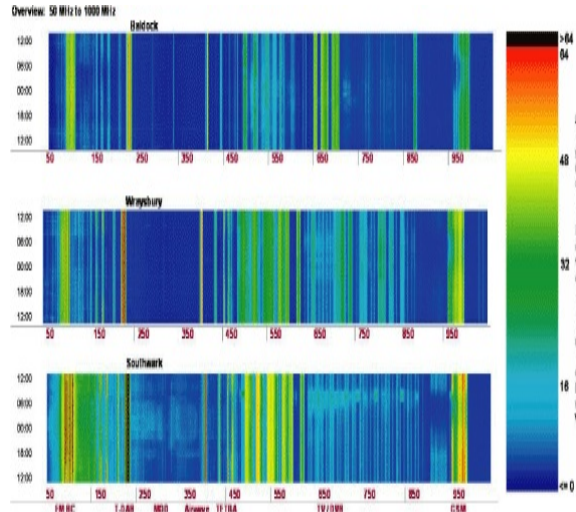


Figure 1.2: A snapshot of frequency utilisation in various areas: many frequencies are not used at all, whilst there is significant activity on others [5]

to civilian and commercial users: spectral holes will be able to be exploited opportunistically by devices, so long as they don't interfere with the reception of TV. Historically, this is the single largest gift of new spectrum, and because there is no requirement for licensing this spectrum is free.

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

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

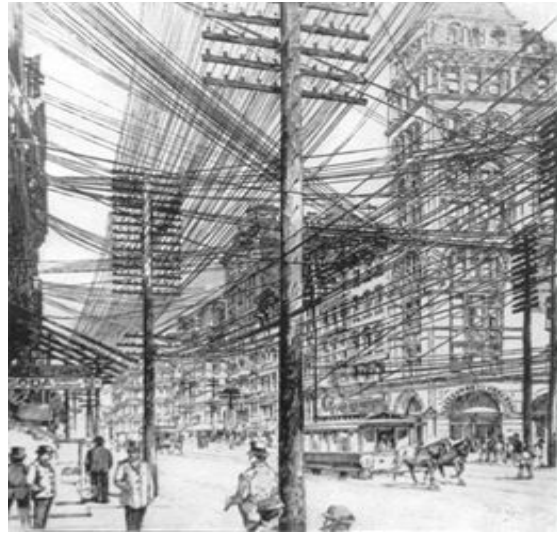


Figure 1.3: A picture of early 20th century New York: Bandwidth has always been an issue

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

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

Spectrum sensing should involve:

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

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

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

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

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

The focus of Chapter 3 is again compressive sensing, this time in a distributed setting - given a connected network of nodes, how can we organise sensing to reconstruct a wideband signal? The chapter begins by discussing various signal models in the literature, and justifies the use of a single model. Then the sensing model is presented as a multinode extension of the Modulated Wideband Converter ([20]). There follows an extended discussion of constrained convex optimisation, and an introduction to the Alternating Direction Method of Multipliers. Finally, how to sense the requisite signals and use a distributed setup to solve the system is explained. Some tentative results of simulations are also presented.

2 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:

$$H_0 : y[n] = w[n], n = 1 \dots N \quad (2.0.1)$$

$$H_1 : y[n] = x[n] + w[n], n = 1 \dots N \quad (2.0.2)$$

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

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

$$\Gamma(\mathbf{y}) = \frac{p(\mathbf{y} | H_0)}{p(\mathbf{y} | H_1)} \quad (2.0.3)$$

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

$$P_D = Pr(\Gamma(\mathbf{y}) > \lambda | H_1) \quad (2.0.4)$$

and the probability of false alarm

$$P_{FA} = Pr(\Gamma(\mathbf{y}) > \lambda | H_0) \quad (2.0.5)$$

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.

2.0.1 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:

$$M = \sum_{n=0}^N |y[n]|^2 \quad (2.0.6)$$

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

$$M = \begin{cases} \frac{\sigma_w^2}{2} & H_0 \\ \frac{\sigma_w^2 + \sigma_s^2}{2} & H_1 \end{cases} \quad (2.0.7)$$

$$P_D = 1 - \Gamma\left(1, \frac{\lambda}{1 + \frac{\sigma_s^2}{\sigma_w^2}}\right) \quad (2.0.8)$$

$$P_F = 1 - \Gamma\left(1, \frac{\lambda}{\sigma_w^2}\right) \quad (2.0.9)$$

Where $\Gamma(1, x)$ is the incomplete gamma function. From these equations it's clear to see that the performance of energy detection based sensing faces challenges at low SNR values. See (REF) figure 3 for curves quantifying the performance. Also energy detectors perform poorly under extreme fading conditions as they are unable to distinguish primary users and noise. Further this type of detector is not efficient at detecting spread spectrum signals.

Similarly because 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. See [27] for more details.

2.0.2 Cyclostationary Feature Detection

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

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

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

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

$$\mathbb{E}(t) = \mathbb{E}(t + mT) = \mathbb{E}[x(t)] \quad (2.0.10)$$

$$\mathbb{R}(t, \tau) = \mathbb{R}(t + mT, \tau) = \mathbb{E}[x(t)x(t + \tau)] \quad (2.0.11)$$

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

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

$$\mathbb{R}(t, \tau) = \sum_{\alpha} r(\alpha, \tau) e^{2\pi j \alpha t} \quad (2.0.12)$$

with Fourier coefficients:

$$r(\alpha, \tau) = \frac{1}{T} \int_T x\left(t + \frac{\tau}{2}\right) x\left(t + \frac{\tau}{2}\right) e^{-2\pi j \alpha t} dt \quad (2.0.13)$$

where α is the cyclic frequency

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

$$S(f) = \int_{-\infty}^{\infty} r(\alpha, \tau) e^{-2\pi j f \tau} d\tau \quad (2.0.14)$$

For a fixed lag τ , 2.0.2 can be re-written as:

$$R_{xx}(t, \tau) = R_{xx}(\tau) + \sum_{\alpha} r(\alpha, \tau) e^{2\pi j \alpha t} \quad (2.0.15)$$

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

Under both hypotheses, (2, 2), the continuous portion of the signal exists, but the cyclo-stationary portion only exists under 2 when $\alpha \neq 0$. Thus we only need to test for the presence of a cyclo-stationary component.

To this end re-write the hypotheses as:

$$H_0 : y[n] = S_w^\alpha[n], n = 1 \dots N \quad (2.0.16)$$

$$H_1 : y[n] = S_x^\alpha[n] + S_w^\alpha[n], n = 1 \dots N \quad (2.0.17)$$

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

$$\chi = \sum_{\alpha \neq 0} \sum_n S_x^\alpha \bar{S}_x^\alpha \quad (2.0.18)$$

we can formulate the cyclo-stationary detector as:

$$d = \begin{cases} 0 & \chi < \lambda \\ 1 & \chi \geq \lambda \end{cases} \quad (2.0.19)$$

where λ is some pre-determined threshold [17].

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

A test statistic can be formed by sampling the output of the filter every nT seconds and choosing 1 if the statistic is below some threshold and 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.

2.0.4 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.

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.

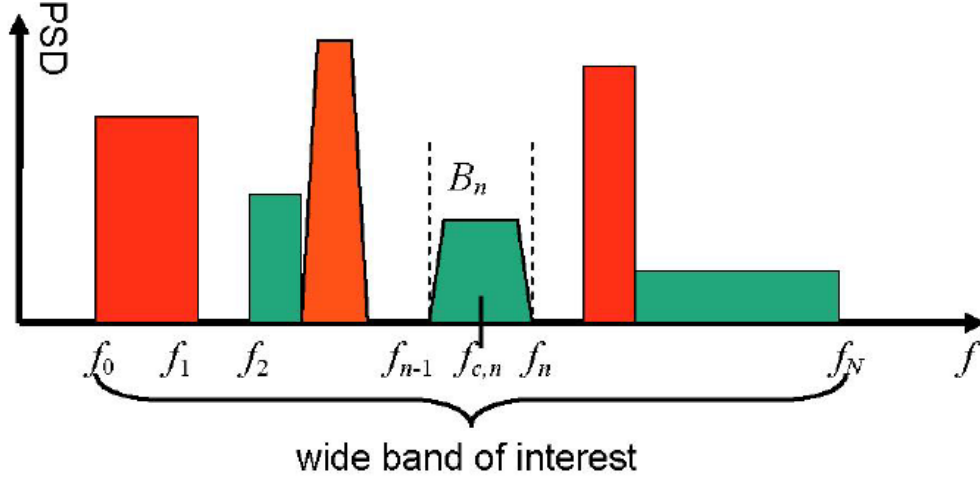


Figure 3.4: A diagram of the Spectrum Sensing model [23]

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 .

$$x(nT) = \text{III}(t - nT) x(t) \quad (3.1.20)$$

where

$$\text{III}(t - nT) = \sum_{k=-\infty}^{\infty} \delta(t - kT) \quad (3.1.21)$$

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

$$\hat{X}_s(f) = \sum_{k=-\infty}^{\infty} x(t - kT) \quad (3.1.22)$$

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

$$x(t) = \sum_{n=-\infty}^{\infty} x(nT) \text{sinc}\left(\frac{t_n T}{T}\right) \quad (3.1.23)$$

3.2 Compressed Sensing

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

Compressed Sensing considers situations where the signal is *undersampled* i.e. situations in which the number of samples is much smaller than the dimension of the signal (or the number of samples required by classical sampling theory). This is equivalent to a system of linear equations which is under-determined. That is, this is a method of measuring the informative parts of a signal directly without acquiring unessential information at the same time (i.e. the parts of the signal that would be discarded in traditional compression applications). The questions then are how can we acquire these measurements in the first place, and how to 'decompress' them once they are obtained [11].

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

1. A sine wave at frequency ω is defined as a single spike in the frequency domain yet has an infinite support in the time domain
2. An image will have values for every pixel, yet the wavelet decomposition of the image will typically only have a few non-zero coefficients

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

$$y_k = \langle f, \phi_k \rangle \quad k = 1 \dots m \quad (3.2.24)$$

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

$$f(t) = \sum_{i=1}^n x_i \psi_i(t) \quad (3.2.25)$$

where the x_i are the coefficient sequence of f .

An example of a practical Compressive Sensing system is the single-pixel camera at Rice University [16]. Typical camera devices obtain pixel samples by exposing a bank of photon detectors (one for each pixel) to the incident light field. This data is then processed into an image.

The single pixel camera takes pictures by first directing the incoming light field onto an array of tiny mirrors (one for each pixel). Each mirror can be either be oriented towards a single photon detector, or oriented away from the detector. In this setup, a measurement is taken as the sum of all the incident light beams. Afterwards, the mirrors are flipped to a new random configuration, and another measurement is taken. This process is repeated, until enough information has been collected to reconstruct the image. Figure 3.5 shows the operation of the single pixel camera.

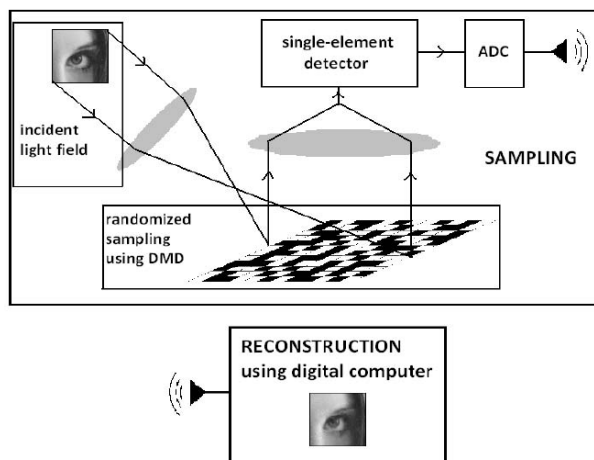


Figure 3.5: The operation of the single pixel camera at Rice University [?]

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

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

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

Theorem [8] Fix $f \in \mathbb{R}^n$ with a sparse coefficient basis, x_i in ψ . Then a reconstruction from m random measurements in ϕ is possible with probability $1 - \delta$ if:

$$m \geq C\mu^2(\phi, \psi)S \log\left(\frac{n}{\delta}\right) \quad (3.2.26)$$

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

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

$$\min \|\tilde{x}\|_{l_1} \text{ subject to } y_k = \langle \phi_k, \psi x^* \rangle \quad \forall k \in M \subset [1 \dots n] \quad (3.2.27)$$

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

Thus the role of l_1 minimisation is to decompress the data. There are many ways to perform this operation: some popular methods are basis pursuit [10] and Greedy approaches such as Orthogonal Matching Pursuit [25].

It may be remarked that the l_0 norm:

$$\|l_0\| = \{\#i, x_i \neq 0\} \quad (3.2.28)$$

is a more appropriate functional to minimise. In fact under this norm, $m \geq 2k$ measurements will suffice to uniquely determine all k -sparse signals. However, this norm is not convex and so minimising it is an NP-hard problem. As we are seeking sparse solutions the l_1 -norm will suffice [12]. 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.

3.2.1 Incoherence, isometries and all that

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

This can be summed up in the following inequality (Restricted Isometry Property) [7]:

Definition 3.1 (Restricted Isometry Property).

$$(1 - \delta) \|x\|_{l_2}^2 \leq \|Ax\|_{l_2}^2 \leq (1 + \delta) \|x\|_{l_2}^2 \quad (3.2.29)$$

We are also in a position to evaluate the meaning of the constant μ in 3.2. We are considering sampling within orthonormal systems (for example, Time and Frequency):

$$A * A = nI \quad (3.2.30)$$

so that each row or column has l_2 norm equal to \sqrt{n} . 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) [6].

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:

$$\mu = \max_{k,j} |\langle \phi_k, \psi_j \rangle| \quad (3.2.31)$$

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

3.2.2 Short, Fat matrices

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

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

However, such an 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 3.6 shows this.

3.2.3 Bayesian Compressive Sensing

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

$$\mathbf{g} = \Phi \mathbf{w} \quad (3.2.32)$$

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

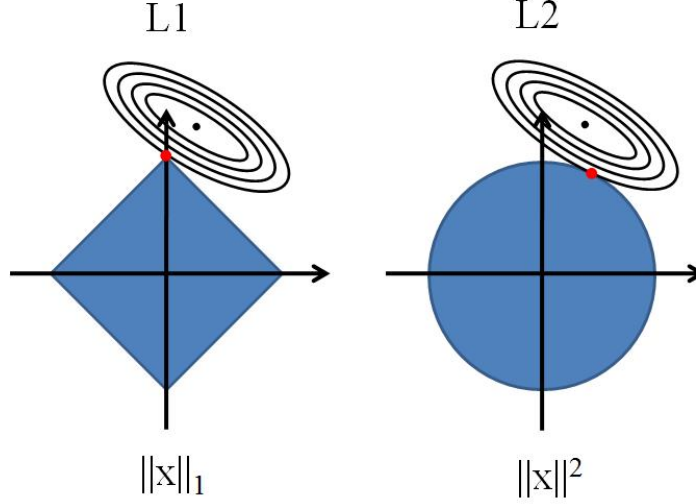


Figure 3.6: Solutions to the Compressive Sensing optimisation problem intersect the l_1 norm the points where all components (but one) of the vector are zero (i.e. it is sparsity promoting) [24]

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

$$\mathbf{g} = \Phi \mathbf{w} + \mathbf{n} \quad (3.2.33)$$

Where \mathbf{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 3.2.3 we have a Gaussian likelihood model:

$$p(\mathbf{g} | \mathbf{w}, \sigma^2) = (2\pi\sigma^2)^{-K/2} \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{g} - \Phi \mathbf{w}\|_2^2\right) \quad (3.2.34)$$

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

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

$$p(w | \lambda) = \left(\frac{\lambda}{2}\right)^N \exp -\lambda \sum_{i=1}^N |w_i| \quad (3.2.35)$$

Note that the solution the convex optimisation problem 3.2 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 3.7 [24]).

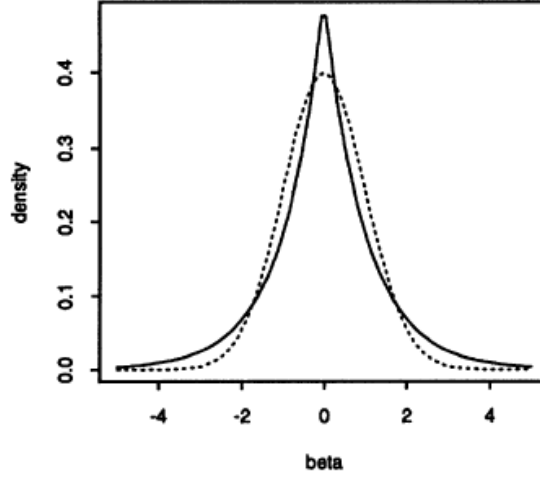


Figure 3.7: The Laplace (l_1 -norm, bold line) and Normal (l_2 -norm, dotted line) densities. Note that the Laplace density is sparsity promoting as it penalises solutions away from zero more than the Gaussian density. [24]

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

$$p(w | a) = \prod_{i=1}^N \mathbb{N}(w_i | 0, \alpha_i^{-1}) \quad (3.2.36)$$

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

$$p(\alpha | a, b) = \prod_{i=1}^N \Gamma(\alpha_i | a, b) \quad (3.2.37)$$

The overall prior is found by marginalising over the hyperparameters:

$$p(w | a, b) = \prod_{i=1}^N \int_0^\infty \mathbb{N}(w_i | 0, \alpha_i^{-1}) \Gamma(\alpha_i | a, b) \quad (3.2.38)$$

This integral can be done analytically and is a Student-t distribution. Choosing the parameters a, b appropriately we can make the Student-t distribution peak strongly around $w_i = 0$ i.e. sparsifying. This process can be repeated for the noise variance σ^2 . The hierarchical model for this process is shown in 1.9. This model, and other CS models which not necessarily have closed form solutions, can be solved via belief-propagation [4]

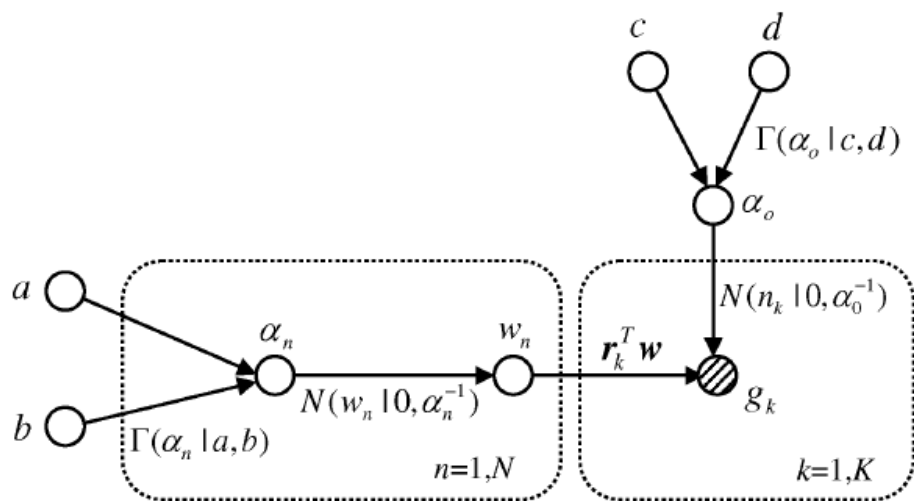


Figure 3.8: The hierarchical model for the Bayesian CS formulation [19]

Group Testing

1 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 [14], 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 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 that 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:

$$A(S) = 1 \sum_i x_i \geq 1 \tag{1.0.39}$$

Note that the addition is the binary-or in the above summation. The goal of Group Testing is to minimise the number of tests required to reconstruct the defective set.

Algorithms

An initial algorithm to consider is a simple binary search of the set to be tested. That is, given a set of size $N = 2^r$ i.e a power of 2, we can recover a single defective in $\lceil \log_2 N \rceil$ tests.

To do this, create a new set of size $S = 2^{\lceil \log_2 n \rceil}$ which is guaranteed to contain a defective. Label the items with integers, and test the items in the sets $1, 2, \dots, S/2$ and $S/2 + 1, \dots, S$ 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

$$K \lceil \log_2 N \rceil \leq K \log_2 N + K \quad (1.0.40)$$

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:

$$y = \mathbf{A}x \quad (1.0.41)$$

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

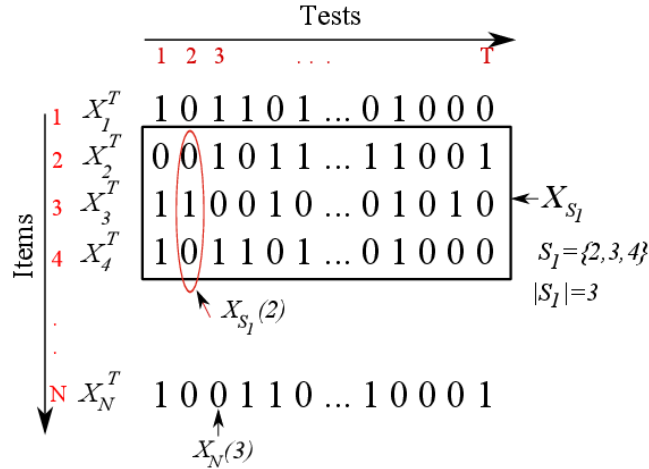


Figure 1.9: The Group Testing model: multiplication with a short, fat matrix [?]

See (REF HWANGDU) 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 explicit combinatorial algorithms which return the codeword with certainty, towards probabilistic 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 [18] 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 group 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:

$$P(\text{Success}) \leq \frac{T}{\log_2 \binom{N}{K}} \quad (1.0.42)$$

However, a tighter upper bound has recently been found [1] at:

$$P(\text{Success}) \leq \frac{2^T}{\binom{N}{K}} \quad (1.0.43)$$

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:

$$T = \log_2 \binom{N}{K} + K \quad (1.0.44)$$

tests. The Combinatorial Orthogonal Matching Pursuit algorithm, considered in [9], is guaranteed to recover the defective set with probability $N^{-\delta}$ in

$$T = ((1 + \delta)e) K \ln N \quad (1.0.45)$$

tests. For all N and K) we have:

$$K \log_2 \frac{N}{K} \leq \log_2 \binom{N}{K} \leq K \log_2 \frac{Ne}{K} \quad (1.0.46)$$

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:

$$T = \beta K \log_2 N + K (\log_2 e + 1) \quad (1.0.47)$$

tests, whilst the COMP algorithm succeeds with:

$$1.88 (1 + \delta) K \log_2 N \quad (1.0.48)$$

tests. It's worthwhile to contrast these results, to gain some insight into the problem. 1 suggests that for very sparse problems (β tending towards 1) that Hwang's adaptive algorithm will outperform a simmlar 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 [3]. 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.

1.0.4 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 [7] 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, [21] considers the noisy Group Testing problem and the reconstruction of the defective set via belief propagation whilst [26] gives explicit theorems on conditions for the recovery of the defective set for the case of the binary symmetric channel. [3] takes this further and finds the capacity of Group Testing for a number of cases.

1.1 The Probabilistic group testing problem

Group testing is a sparse inference problem, first introduced by Dorfman [28] in the context of testing for rare diseases. Given a large population of items \mathcal{P} , indexed by $\{1, \dots, N\}$, where some small fraction of the items are interesting in some way, how can we find the interesting items efficiently?

We perform a sequence of T pooled tests defined by test sets $\mathcal{X}_1, \dots, \mathcal{X}_T$, where each $\mathcal{X}_i \subseteq \mathcal{P}$. We represent the interesting (‘defective’) items by a random vector $\mathbf{U} = (U_1, \dots, U_N)$, where U_i is the indicator of the event that item i is defective. For each test i , we jointly test all the items in \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 $\mathcal{X}_1, \dots, \mathcal{X}_{i-1}$ and outcomes y_1, \dots, y_{i-1} . The group testing problem requires us to infer \mathbf{U} with high probability given a low number of tests T .

Since Dorfman’s paper [28], 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 [29, 30] for a summary, with [30] 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 [31, 32, 33] or more sophisticated random designs based on LDPC codes [34].

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. [35], 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 [36], 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 [28] or more recent research [37] involving screening for genetic conditions, values of p_i might summarise prior information based on a risk profile or family history.

1.2 Group testing capacity

It is possible to characterize performance tradeoffs in group testing from an information-theoretic point of view – see for example [31, 38, 33, 39]. These papers have focussed on group testing as a channel coding problem, with [31, 39] explicitly calculating the mutual information. The paper [38] 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 1.1. *Consider a sequence of group testing problems where the i th problem has defectivity vector $\mathbf{U}^{(i)}$, and consider algorithms which are given $T(i)$ tests. We refer to a constant C as the (weak) group testing capacity if for any $\epsilon > 0$:*

1. *any sequence of algorithms with*

$$\liminf_{i \rightarrow \infty} \frac{H(\mathbf{U}^{(i)})}{T(i)} \geq C + \epsilon, \quad (1.2.49)$$

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

2. *and there exists a sequence of algorithms with*

$$\liminf_{i \rightarrow \infty} \frac{H(\mathbf{U}^{(i)})}{T(i)} \geq C - \epsilon \quad (1.2.50)$$

with success probability $\mathbb{P}(\text{suc}) \rightarrow 1$.

Remark 1.2. *In the Combinatorial case of K defective items with all defective sets equally likely, $H(\mathbf{U}) = \log_2 \binom{N}{K}$, which is the term found in the denominator in [38, Eq. (1) and (2)]. In the Probabilistic case (as in [35]) 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 1.3. *If for $\liminf_{i \rightarrow \infty} \frac{H(\mathbf{U}^{(i)})}{T(i)} \geq C + \epsilon$, the success probability $\mathbb{P}(\text{suc}) \rightarrow 0$ 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.*

1.3 Main results

The principal contribution of [38, Theorem 1.2] was the following result:

Theorem 1.4 ([38]). *The strong capacity of the adaptive noiseless Combinatorial group testing problem is $C = 1$, in any regime such that $K/N \rightarrow 0$.*

This argument came in two parts. First, in [38, Theorem 3.1] the authors proved a new upper bound on success probability

$$\mathbb{P}(\text{suc}) \leq \frac{2^T}{\binom{N}{K}}, \quad (1.3.51)$$

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) [40], [29] 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 [38], we focus on asymptotically sparse cases, where $\mu^{(N)}/N \rightarrow 0$ (in contrast, Wadayama [34] considered a model where p_i are identical and fixed). The main result of the present paper is Theorem 3.9, stated and proved in Section 3.5 below, which implies the following Probabilistic group testing version of Theorem 1.4.

Corollary 1.5. *In the case where $p_i \equiv p$, the weak capacity of the adaptive noiseless Probabilistic group testing problem is $C = 1$, in any regime such that $\mu^{(N)}/N \rightarrow 0$ and $\mu^{(N)} \rightarrow \infty$.*

Again we prove our main result Theorem 3.9 using complementary bounds on both sides. First in Section 2.1 we recall a universal upper bound on success probability, Theorem 2.1, taken from [35], which implies a weak converse. In [35], Li et al. introduce the Laminar Algorithm for Probabilistic group testing. In Section 2.3 we propose a refined version of this Laminar Algorithm, based on Hwang’s HGBSA [40], which is analysed in Section 3.5, and shown to imply performance close to optimal in the sense of capacity.

2 Algorithms and existing results

2.1 Upper bounds on success probability

Firstly [35, Theorem 1] can be restated to give the following upper bound on success probability:

Theorem 2.1. *Any Probabilistic group testing algorithm using T tests with noiseless measurements has success probability satisfying*

$$\mathbb{P}(\text{suc}) \leq \frac{T}{H(\mathbf{U})}.$$

Rephrased in terms of Definition 1.1, this tells us that the weak capacity of noiseless Probabilistic group testing is ≤ 1 . The logic is as follows; if the capacity were $1 + 2\epsilon$ for some $\epsilon > 0$, then there would exist a sequence of algorithms with $H(\mathbf{U}^{(i)})/T(i) \geq 1 + \epsilon$ with success probability tending to 1. However, by Theorem 2.1, any such algorithms have $\mathbb{P}(\text{suc}) \leq 1/(1 + \epsilon)$, meaning that we have established that a weak converse holds.

Remark 2.2. *It remains an open and interesting problem to prove an equivalent of (1.3.51) as in [38, 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.*

2.2 Binary search algorithms

The main contribution of this work is to describe and analyse algorithms that will find the defective items. In brief, we can think of Hwang’s HGBSA algorithm as dividing the population \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 [35] 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 [35, Theorem 2]:

Theorem 2.3. *The expected number of tests required by the Laminar Algorithm [35] is $\leq 2H(\mathbf{U}) + 2\mu$. Under a technical condition (referred to as non-skewedness), the success probability can be bounded by $\mathbb{P}(\text{suc}) \geq 1 - \epsilon$ using $T = (1 + \delta)(2^{\Gamma + \log_2 3} + 2)H(\mathbf{U})$ tests, where Γ is defined implicitly in terms of ϵ , and $\delta \geq 2e - 1$.*

Ignoring the Γ term, and assuming the non-skewedness condition holds, this implies that (using the methods of [35]) $T = 2e(3+2)H(\mathbf{U}) = 10eH(\mathbf{U})$ 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.

2.3 Summary of our contribution

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

Data: A Set S of $|S| = n$ items, μ of which are actually defective in expectation, a probability vector $\mathbf{p}^{(n)}$ describing each item’s independent probability of being defective, and a cutoff θ

Result: The set of defective items

Discard items with $p_i \leq \theta$

Sort the remaining items into B bins, collecting items together with $p_i \in [1/2C^r, 1/2C^{r-1})$ 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

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 3.7 (in terms of expected number of tests) and Theorem 3.9 (in terms of error probabilities). The key ideas are:

1. To partition the population \mathcal{P} into search sets \mathcal{S} containing items which have similar probabilities, expressed through the Bounded Ratio Condition 1. This is discussed in Section 3.1, and optimised in the proof of Proposition 3.7.
2. The way in which we deal with sets \mathcal{S} which contain more than one defective, as discussed in Remark 3.2 below. Essentially we do not backtrack after each test by testing both left- and right-hand sets, but only backtrack after each defective is found.
3. To discard items which have probability below a certain threshold, since with high probability none of them will be defective. This is an idea introduced in [35] and discussed in Section 3.2, with a new bound given in Lemma 3.4.
4. Careful analysis in Section 3.4 of the properties of search sets \mathcal{S} gives Proposition 3.7, which shows that the expected number of tests required can be expressed as $H(\mathbf{U})$ plus an error term. In Section 3.5, we give an analysis of the error probability using Bernstein's inequality, Theorem 3.8, allowing us to prove Theorem 3.9.

2.4 Wider context: sparse inference problems

Recent work [41, 39] 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 [42, 43] 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 [42, 43]).

3 Analysis and new bounds

3.1 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 $\mathcal{S}_1, \dots, \mathcal{S}_G$, each of which contains items which have comparable values of p_i .

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

$$\max_{i,j \in \mathcal{S}} \frac{p_j}{p_i} \leq C. \quad (3.1.52)$$

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

Lemma 3.1. *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 $\bar{p}_i := p_i/P_{\mathcal{S}}$, each item has length $\ell_i^{(\mathcal{S})}$ bounded by*

$$\ell_i^{(\mathcal{S})} \leq \ell_{\max}^{(\mathcal{S})} := \frac{h(\mathcal{S})}{P_{\mathcal{S}}} + \log_2 C + \log_2 P_{\mathcal{S}} + 1, \quad (3.1.53)$$

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

Proof. Under the Bounded Ratio Condition, for any i and j , we know that by taking logs of (3.1.52)

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

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

$$-P_{\mathcal{S}} \log_2 p_i \leq h(\mathcal{S}) + P_{\mathcal{S}} \log_2 C. \quad (3.1.54)$$

Now, the Shannon–Fano length of the i th item is

$$\begin{aligned} \ell_i^{(\mathcal{S})} = \lceil -\log_2 \bar{p}_i \rceil &\leq -\log_2 p_i + \log_2 P_{\mathcal{S}} + 1 \\ &\leq \left(\frac{h(\mathcal{S})}{P_{\mathcal{S}}} + \log_2 C \right) + \log_2 P_{\mathcal{S}} + 1. \end{aligned} \quad (3.1.55)$$

and the result follows by (3.1.54). \square

Next we describe our search strategy:

Remark 3.2. *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 3.3. *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 $\mathbb{E}T_{\mathcal{S}} \leq T_{\text{bd}}(\mathcal{S})$ for*

$$T_{\text{bd}}(\mathcal{S}) := h(\mathcal{S}) + P_{\mathcal{S}} \log_2 C + P_{\mathcal{S}} \log_2 P_{\mathcal{S}} + P_{\mathcal{S}} + 1. \quad (3.1.56)$$

Proof. Using the algorithm of Remark 3.2, laid out on the Shannon-Fano tree constructed in Lemma 3.1, we are guaranteed to find every defective. The number of tests to find one defective thus corresponds to the depth of the tree, which is bounded by $\ell_{\max}^{(\mathcal{S})}$ given in (3.1.53).

Recall that we write U_i for the indicator of the event that the i th 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 3.2 this search procedure will take

$$\begin{aligned}
T_{\mathcal{S}} &= 1 + \sum_{i \in \mathcal{S}} U_i \ell_i^{(\mathcal{S})} \\
&= \sum_{i \in \mathcal{S}} p_i \ell_i^{(\mathcal{S})} + 1 + \sum_{i \in \mathcal{S}} \ell_i^{(\mathcal{S})} (U_i - p_i) \\
&\leq \sum_{i \in \mathcal{S}} p_i \ell_{\max}^{(\mathcal{S})} + 1 + \sum_{i \in \mathcal{S}} V_i^{(\mathcal{S})} \\
&= P_{\mathcal{S}} \ell_{\max}^{(\mathcal{S})} + 1 + \sum_{i \in \mathcal{S}} V_i^{(\mathcal{S})} \\
&\leq T_{\text{bd}}(\mathcal{S}) + \sum_{i \in \mathcal{S}} V_i^{(\mathcal{S})} \quad \text{tests.}
\end{aligned} \tag{3.1.57}$$

Here we write $V_i^{(\mathcal{S})} = \ell_i^{(\mathcal{S})} (U_i - p_i)$, which has expectation zero, and (3.1.57) follows using the expression for $\ell_{\max}^{(\mathcal{S})}$ given in Lemma 3.1. \square

3.2 Discarding low probability items

As in [35], 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 3.4. *Choosing $\theta(P_e)$ such that*

$$-\log_2 \theta(P_e) = \min \left(\log_2 \left(\frac{2n}{P_e} \right), \frac{2H(\mathbf{U})}{P_e} \right) \tag{3.2.58}$$

ensures that

$$\mathbb{P}(\mathcal{P} \setminus \mathcal{P}^* \text{ contains a defective}) \leq P_e/2. \tag{3.2.59}$$

Proof. The approach of [35] 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. \square

3.3 Searching the entire set

Having discarded items with p_i below this probability threshold θ and given bounding ratio C , we create a series of bins. We collect together items with probabilities $p \in [1/2, 1]$ in bin 0, $p \in [1/(2C), 1/2]$ in bin 1, items with probabilities $p \in [1/(2C^2), 1/(2C)]$ in bin 2, \dots , 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

$$B \leq \frac{-\log_2(2\theta)}{\log_2 C} + 1. \quad (3.3.60)$$

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

Definition 3.5. *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 $\mathcal{S}_1, \mathcal{S}_2, \dots$. For i increasing from 0 to B , we place items from bin i into sets $\mathcal{S}_{b_i+1}, \dots, \mathcal{S}_{b_{i+1}}$, 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 $\mathcal{S}_{b_i+1}, \dots, \mathcal{S}_{b_{i+1}}$, where $\mathcal{S}_{b_{i+1}}$ may remain not full.

Proposition 3.6. *This splitting procedure will divide \mathcal{P}^* into search sets $\mathcal{S}_1, \dots, \mathcal{S}_G$, 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 $P_j := P_{\mathcal{S}_j} \leq 1$.

Proof. First, note that the items from bin 0 each lie in a set \mathcal{S} on their own. These sets will be full, trivially satisfy the Bounded Ratio Condition 1 with constant C , and have probability satisfying $P_j \leq 1$. For each of bins $1, \dots, B$:

1. For each bin i , it is possible that the last set $\mathcal{S}_{b_{i+1}}$ will not be full, but every other set corresponding to that bin will be full. Hence, there are no more than B sets which are not full.
2. For each resulting set \mathcal{S}_j , the total probability $P_j \leq 1$ (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 $1, \dots, B$ has probability $\leq 1/2$).
3. 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

$$\mu = \sum_{i \in \mathcal{P}} p_i \geq \sum_{i \in \mathcal{P}^*} p_i = \sum_{j=1}^G P_j \geq \sum_{j: \mathcal{S}_j \text{ full}} P_j \geq |\mathcal{S}_j \text{ full}| \frac{1}{2}. \quad (3.3.61)$$

Since, as discussed in point 1) above, the total number of sets is bounded by the number of full sets plus B , the result follows using Equation (3.3.60). \square

3.4 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 3.7. *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 $\mathbb{E}T \leq T_{\text{bd}}$, where*

$$T_{\text{bd}} := (H(\mathbf{U}) + 3\mu + 1) + 2\sqrt{\mu(-\log_2(2\theta))}. \quad (3.4.62)$$

Proof. Given a value of C , Proposition 3.6 shows that our splitting procedure divides \mathcal{P}^* into G sets $\mathcal{S}_1, \dots, \mathcal{S}_G$, 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 3.3, $T = \sum_{j=1}^G T_{\mathcal{S}_j}$, where $\mathbb{E}T_{\mathcal{S}_j} \leq T_{\text{bd}}(\mathcal{S}_j)$.

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

$$\begin{aligned} \sum_{j=1}^G T_{\text{bd}}(\mathcal{S}_j) &\leq \sum_{j=1}^G (h(\mathcal{S}_j) + P_j(\log_2 C + 1) + 1) \\ &= \sum_{j \in \mathcal{P}^*} -p_j \log_2 p_j + \mu(\log_2 C + 1) + G \\ &\leq \sum_{j \in \mathcal{P}^*} h(p_j) + 3\mu + 1 + \left(\frac{-\log_2(2\theta)}{\log_2 C} + \mu \log_2 C \right) \\ &\leq (H(\mathbf{U}) + 3\mu + 1) + \left(\frac{-\log_2(2\theta)}{\log_2 C} + \mu \log_2 C \right). \end{aligned} \quad (3.4.63)$$

This follows by the bound on G in Proposition 3.6, as well as the fact that $0 \leq p_j \leq 1$ means that for any i , $-p_j \log_2 p_j = (1 - p_j) \log_2(1 - p_j) + h(p_j) \leq h(p_j)$.

Finally, we choose $C > 1$ to optimize the second bracketed term in Equation (3.4.63). Differentiation shows that the optimal C satisfies $\log_2 C = \sqrt{-\log_2(2\theta)/\mu}$, meaning that the bracketed term

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

and the result follows. \square

3.5 Controlling the error probabilities

Although Section 3.4 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 [44]):

Theorem 3.8 (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*

$$\mathbb{P}\left(\sum_{j=1}^n V_j \geq t\right) \leq \exp\left(-\frac{t^2}{4L}\right), \text{ for any } 0 \leq t \leq \frac{L}{M}. \quad (3.5.64)$$

We deduce the following result:

Theorem 3.9. 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

$$T_{\text{nec}} = T_{\text{bd}} + \psi H(\mathbf{U}), \quad (3.5.65)$$

where T_{bd} is given in (3.4.62).

1. If we terminate our group testing algorithm after T_{nec} tests, the success probability

$$\mathbb{P}(\text{suc}) \geq 1 - \frac{1}{2} \sqrt{\frac{\mu}{H(\mathbf{U})}} - \exp \left(- \left(\frac{L}{4M^2} \right)^{1/3} \right). \quad (3.5.66)$$

2. Hence in any regime where $\mu \rightarrow \infty$ with $\mu/H(\mathbf{U}) \rightarrow 0$ and $L/M^2 \rightarrow \infty$, our group testing algorithm has (a) $\liminf H(\mathbf{U})/T_{\text{nec}} \geq 1/(1 + \epsilon)$ for any ϵ and (b) $\mathbb{P}(\text{suc}) \rightarrow 1$, so the capacity $C = 1$.

Proof. We first prove the success probability bound (3.5.66). 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 3.4) the first event has probability $\leq P_e/2$, contributing $\frac{1}{2} \sqrt{\mu/H(\mathbf{U})}$ to (3.5.66).

Our analysis of the second error event is based on the random term from Equation (3.1.57), which we previously averaged over but now wish to bound. There will be an error if $T_{\text{nec}} \leq T$, or (rearranging) if

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

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 (3.2.58), and by design all the sets \mathcal{S} have total probability $P_{\mathcal{S}} \leq 1$. Using (3.1.55) we know that the V_i are bounded by

$$|V_i| \leq \ell_i \leq -\log_2 p_i + \log_2 P_{\mathcal{S}} + 1 \leq -\log_2 \theta + 1. \quad (3.5.67)$$

Hence, the conditions of Bernstein's inequality, Theorem 3.8, 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 3.8 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 (3.5.66) follows.

We next consider the capacity bound of 2). Since $-\log_2 \theta \leq 2H/P_e$, using (3.4.62) and (3.5.65)

$$\begin{aligned} \frac{T_{\text{nec}}}{H} &= \frac{T_{\text{bd}}}{H} + \psi \\ &= 1 + 3\frac{\mu}{H} + \frac{1}{H} + 2\sqrt{\frac{\mu}{HP_e}} + \psi \\ &= 1 + 3\frac{\mu}{H} + \frac{1}{H} + 2\left(\frac{\mu}{H}\right)^{1/4} + \psi, \end{aligned} \tag{3.5.68}$$

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

Proof of Corollary 1.5. In the case where all p are identical, $\mu = Np$, $H = Np(-\log p)$, so $\mu/H = 1/(-\log p) \rightarrow 0$. Similarly, $L = Np(-\log_2 p)^2$ and $M = (-\log_2 p)$ so that $L/M^2 = Np \rightarrow \infty$ as required. \square

3.6 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).

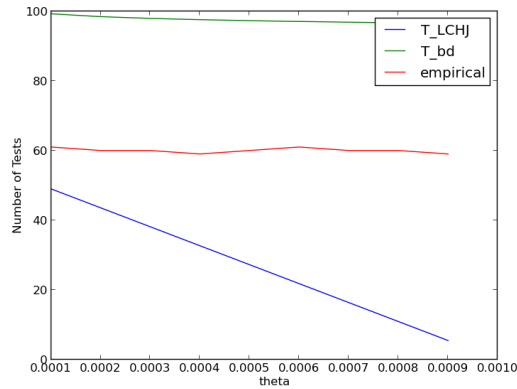


Figure 3.10: Theoretical lower and upper bounds and empirical Test frequencies as functions of θ

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 (2.1).

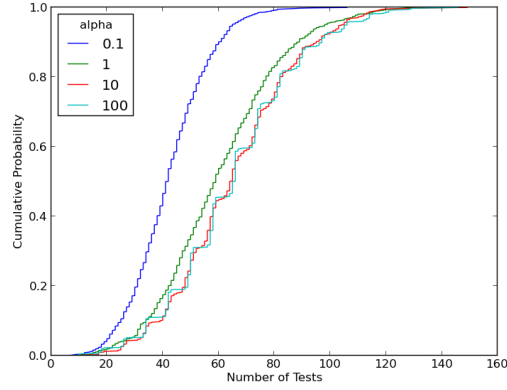


Figure 3.11: Cumulative distribution curves of the modified Hwang algorithm with fixed $\theta = 0.0001$ and α varying

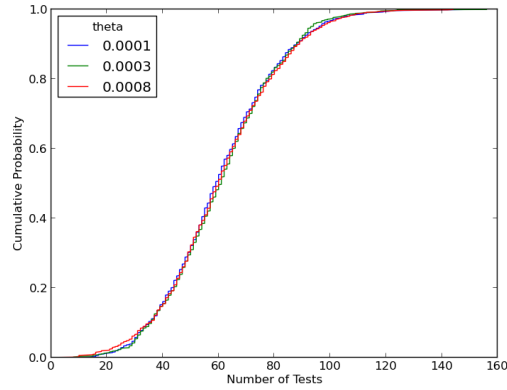


Figure 3.12: Cumulative distribution curves for fixed $\alpha = 1$ and varying θ

Figures (3.11) and (3.12) show that the performance is relatively insensitive to the cut-off θ , and more sensitive to the uniformity (or otherwise) of the probability distribution \mathbf{p} . Heuristically, this is for because distributions which are highly concentrated on a few items algorithms can make substantial savings on the testing budget by testing those highly likely items first (which is captured in the bin structure of the above algorithm).

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.

3.7 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 [35], 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 (3.4.62) which has leading term $H(\mathbf{U})$, and not on tightening bounds on the coefficient of μ in (3.4.62). For completeness, we mention that this coefficient can be reduced from 3, under a simple further condition:

Remark 3.10. *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 $P_S \log_2 P_S$ in (3.1.56) becomes $P_S \log_2(\alpha + c)$, the bound in (3.3.61) 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 (3.4.63). 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 2.1, in order prove a strong converse, and hence confirm that the strong capacity is also equal to 1.

In future work, we hope to explore more realistic models of defectivity, such as those where the defectivity of U_i are not necessarily independent, for example by imposing a Markov neighbourhood structure.

Optimisation on Graphs

1 Distributed Compressed Sensing

Suppose you are given the task of sensing the spectrum in any given ambient radio environment. At your dispose you have a set of sensors, which have a configurable sampling rate, which you may place anywhere. The only constraint is that no single node is capable of sensing the entire spectrum entirely; however, you are free to decide whether nodes may arrive at consensus (if one can be proven to exist) either in an entirely distributed fashion, or in a coordinated fashion with some node being regarded as centralised, and which does the majority of the data processing.

As we are constrained by the sensing budgets of each node, the data we collect will necessarily be undersampled. However, as discussed in the first chapter, we are capable of designing sensing schemes that are capable of reconstructing signals from undersampled measurements. The trick is designing one which can be distributed over the network, so that we may glimpse the radio environment.

We do all of this, so that we can make decisions about available transmission opportunities in the spectra. As discussed in Chapter 1 there is a considerable amount of new spectrum being released immanently, and it is being released with the intent that it will be unlicensed. This means that all frequencies in this band (the TV white-spaces) will be available for transmission, without users having to pay for the right to do so. It is hoped that by actively sensing spectral holes, there won't be as many saturated frequencies and that the system will balance itself out. Of course, new users must not interfere with already existing users.

To achieve our aim of sensing with a set of nodes, we drop our nodes into some geographical area and let them communicate. It's not necessary for every node to talk to every other node.

Individually the nodes sense a sparse signal, at rates far below the Nyquist rate for that signal, and then collaboratively reconstruct what was sensed. The nodes may all sense the entire signal, or they may sense disjoint portions. In addition, they may sense a signal which is jointly sparse: all the nodes sense a common set of signal components, as well as a set of signal components unique to a particular node. For clarity let $y_j \in \mathbb{R}^{m_j}$ be the measurement vector obtained by the j_{th} node, then the measurements can be described by the following linear system:

$$\mathbf{y}_j = \mathbf{A}_j \mathbf{x}_j + \mathbf{w}_j \quad (1.0.69)$$

where $A_j \in \mathbb{R}^{m_j \times n}$ is a measurement matrix satisfying the restricted isometry property (3.1), chosen so that the signal $x_j \in \mathbb{R}^n$ is suitably sparse (for example, A could be the product of time and frequency representation matrices). \mathbf{w}_j is additive white Gaussian noise.

The matrix is constructed by concatenating the measurements from each sensor, row-wise. That is, we stack the measurements from each node on top of each other.

1.1 Signal Models

The core ideas of Distributed Compressed Sensing are to distribute the sensing load among many nodes to reduce the rate required at each node, and to exploit correlations between the data sensed at each node so as to improve the quality of the signals reconstructed at each node.

The signals at each node are modelled either as having an individual and common part, or as consisting entirely of a common signal. That is,

$$x_j = x_{common} + x_{private,j} \quad (1.1.70)$$

where x_j is an s -sparse signal, and x_{common} and $x_{private}$ are s_c -sparse and s_{p_j} -sparse respectively ($s_j = s_c + s_p$) and are assumed to have disjoint support sets.

1.1.1 Common Signal Model

This model assumes that all nodes sense the same signal. For example, in wireless telecommunications, all nodes within the same cell would sense a common frequency spectrum. The spectra of distant cells would be independent from each other. Another example could be using sensor networks to monitor seismic activity.

I.e we have

$$x_j = x_{common} \quad \forall j \in J \quad (1.1.71)$$

note that the sensors may all have different sampling parameters and different noise properties (i.e different SNRs depending on the environment).

Currently, in this project the common signal model is all that is being considered.

1.1.2 Mixed Signal Model

This is a natural extension of the previous model, allowing for each node to sense individual signal components in addition to the joint component shared by all nodes. The model is,

$$x_j = x_{common} + x_{private,j} \quad (1.1.72)$$

we recover the previous model by setting $x_{private} = 0$. This is a slightly more realistic model for spectrum sensing, as it's reasonable to assume that over a cell the signal might not be constant. For example, the joint component might be the signal from a television base-station, whilst the individual signal components may be multipath reflections of other secondary users.

1.1.3 Other models

There are a variety of other models to consider, but all are some variety of the common plus private paradigm discussed above. For example all nodes could share a common support set for a signal x , however each node may only sense a part of the signal with the full signal being contained in the union of the measurements from all nodes.

This model can be extended to the common plus private scenario with each node sensing a portion of the joint component in addition to an individual signal component.

It's important to distinguish between distributed models, and distributed solvers. In this sections we have previously described several models in which the sensing is distributed in some way. Now, we proceed to describe several distributed solvers as well as preliminary results of experiments comparing distributed solvers to centralised solvers.

2 Sensing Model

We are considering a radio environment with a single primary user (PU) and a network of $J=50$ nodes collaboratively trying to sense and reconstruct the PU signal, either in a fully distributed manner (by local message passing), or by transmitting measurements to a fusion centre which then solves the linear system.

2.1 Setup

We are trying 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 $[0, 1] \times [0, 1]$. They individually take measurements in the following way: 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.

This process is repeated in parallel at each node (unrelated to N_{sig} so that each band in x appears in baseband. The mixing functions - which are independent for each node - are required to be periodic, with period T_p . Since p_i is periodic it has Fourier expansion:

$$p_i(t) = \sum_{l=-\infty}^{\infty} c_{il} \exp\left(jlt \frac{2\pi}{T_p}\right) \quad (2.1.73)$$

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 $x p_i$, with Fourier transform:

$$X_i(f) = \int_{-\infty}^{\infty} x(t) p_i(t) dt \quad (2.1.74)$$

$$= \sum_{l=-\infty}^{\infty} c_{il} X(f - l f_p) \quad (2.1.75)$$

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

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:

$$Y_i(e^{j2\pi f T_s}) = \sum_{l=-L_0}^{+L_0} \quad (2.1.76)$$

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 $\lceil f_N Y Q / f_p \rceil$ if we choose $f_s = f_p$. These relations can be written in matrix form as:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w} \quad (2.1.77)$$

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

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

$$\mathbf{A}^{m \times L} = \mathbf{S}^{m \times L} \mathbf{F}^{L \times L} \mathbf{D}^{L \times L} \mathbf{H}^{L \times L} \quad (2.1.78)$$

The measurements \mathbf{y} are transmitted to a Fusion Centre via a control channel. The system can then be solved by linear programming.

3 Constrained Convex Optimisation

We now have a slight diversion into constrained convex optimisation, the denouement of which will not be apparent until later. This section follows (REF), and only contains enough to make the optimisation algorithm on graphs

Say, for example, that you are taking a walk up a mountain and would like to calculate the highest point along your path. You may then be tempted to solve a problem which looked like this:

$$\begin{aligned} & \underset{x}{\text{minimize}} && f(x) \\ & \text{subject to} && Ax = b \end{aligned}$$

where f is a function in \mathbb{R} , $A \in \mathbb{R}^{n \times m}$ is a matrix, and $x \in X$, X is some convex set.

This is a problem of constrained optimisation, which can be solved by adding a Lagrange multiplier like term to the unconstrained objective (which coincides with the constrained optimisation problem):

$$L(x, y) = f(x) + y^T (Ax - b) \quad (3.0.79)$$

where y is the Lagrange multiplier, and L is referred to as the Lagrangian. The components of y can be interpreted as the rate of change of the objective function as a function of the constraint variable. For example, if we are required to find the maximum height along a path on the mountain, the Lagrange multiplier would represent the rate of change in height along that specific path.

We can find the solution of the original problem via the duality theorem of Fenchel:

$$g(y) = \inf_x L(x, y) = -f^*(-A^T y) - b^T y \quad (3.0.80)$$

where f^* is the convex conjugate (Legendre-Fenchel) transform of f . The problem is now to maximise g .

This maxima can be expressed in closed form as:

$$x^* = \underset{x}{\text{minimize}} L(x, y^*) \quad (3.0.81)$$

where y^* is the soln of the dual problem.

Together 3.0.80 and 3.0.81 suggest a iterative algorithm, which proceeds iteratively by finding a minima of the objective function (up to a convex transformation) and then updating the Lagrange multiplier until some convergence criterion is met:

$$x^{k+1} := \underset{x}{\text{minimize}} L(x, y^*) \quad (3.0.82)$$

$$y^{k+1} := y^k + \alpha^k (Ax^{k+1} - b) \quad (3.0.83)$$

This is a form of gradient ascent, with α as the step size and k is the iteration counter. Assuming that g is differentiable, the algorithm is monotone (i.e. $g(y^{k+1}) > g(y^k)$).

This algorithm is easily extended to the case where the objective function is linearly separable. That is scenarios where we can split $f(x)$ into a sum of functions $f_i(x_i)$, each over a disjoint domain $x_i \in X_i$ (in our hill walking example, we can split our chosen path into a series of continuous intervals which lay next to each other). Extending the algorithm is simple: all that needs to be done is to perform the minimisation step in (3.0.83) in each of the functions f_i , and then jointly update the multiplier.

We have assumed the differentiability g throughout, and it is this smoothness condition which allows the previous algorithm to terminate on a solution of the original constrained problem.

For problems with g not differentiable, the augmented Lagrangian is introduced as:

$$L_\rho(x, y) = f(x) + y^T (Ax - b) + \frac{\rho}{2} \|Ax - b\|_2^2 \quad (3.0.84)$$

where ρ is a penalty parameter. That is we're no longer seeking exact solutions, but instead seek solutions close in mean square error.

The associated dual function is $g_\rho(y) = \inf_x L_\rho$. By adding the penalty term, g_ρ is no differentiable, and the $\rho \rightarrow 0$ regime corresponds to the solution of the original problem.

The algorithm in this case can be found by first minimising over x and then evaluating the resulting equality constraint residual, i.e.:

$$x^{k+1} := \arg \min_x L_\rho \quad (3.0.85)$$

$$y^{k+1} := y^k + \rho (Ax^{k+1} - b) \quad (3.0.86)$$

This algorithm is known as the method of multipliers for solving the problem (3). Note that even if f is separable, the augmented Lagrangian may not be.

3.1 ADMM

The alternating direction method of multipliers (ADMM), is an algorithm intended to blend the decomposability of the gradient ascent algorithm 3.0.83, with the more general method of multipliers 3.0.86, i.e. when the objective function is decomposable but the dual function is not differentiable. The algorithm solves problems of the form

$$\min_x f(x) + g(z) \quad (3.1.87)$$

$$\text{s.t } Ax + Bz = c \quad (3.1.88)$$

where f and g are assumed to be convex function with range in \mathbb{R} , $A \in \mathbb{R}^{p \times n}$ and $B \in \mathbb{R}^{p \times m}$ are matrices (not assumed to have full rank), and $c \in \mathbb{R}^p$. We begin by illustrating a few examples, relevant to the type of problems encountered in signal processing.

Example 3.1. *The Basis Pursuit problem*

$$\begin{aligned} & \underset{x}{\text{minimize}} \quad \|x\|_1 \\ & \text{subject to} \quad Ax = b \end{aligned}$$

can be written in the ADMM form (3.1.88):

$$\begin{aligned} & \underset{x}{\text{minimize}} \quad f(x) + \|z\|_1 \\ & \text{subject to} \quad x - z = 0 \end{aligned}$$

where f is the indicator function of $\{x \in \mathbb{R}^n : Ax = b\}$.

Example 3.2. *Similarly the LASSO:*

$$\underset{x}{\text{minimize}} \quad \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|x\|_1 \quad (3.1.89)$$

$\lambda > 0$, can be solved by the ADMM with $f = \frac{1}{2} \|Ax - b\|_2^2$ and $g = \lambda \|x\|_1$.

The solution of (3.1.88) is:

$$x^* = \inf \{f(x) + g(z) : Ax + Bz = c\} \quad (3.1.90)$$

As previously we form the Lagrangian:

$$L_\rho(x, y) = f(x) + g(z) + y^T (Ax + Bz - c) + \frac{\rho}{2} \|Ax + Bz - c\|_2^2 \quad (3.1.91)$$

ADMM consists of the iterations:

$$x^{k+1} := \arg \min_x L_\rho(x, z^k, y^k) \quad (3.1.92)$$

$$z^{k+1} := \arg \min_z L_\rho(x^{k+1}, z, y^k) \quad (3.1.93)$$

$$y^{k+1} := y^k + \rho (Ax^{k+1} + Bz^{k+1} - c) \quad (3.1.94)$$

Note that this algorithm is similar to both the gradient ascent and method of multipliers from the last section: it has an x minimisation step, an z minimisation step and a dual variable update. I.e. the Lagrangian is updated simultaneously with respect to the two primal variables. Separating minimisation into two steps (as opposed to joint minimisation) is what allows for decomposition if f or g (or both) are separable.

4 Constrained Optimisation on Graphs

We model the network 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 nodes can communicate. The number of nodes node i can communicate is written \mathcal{N}_i and the degree of node i is $D_i = |\mathcal{N}_i|$.

We assume that a proper (or approximate) colouring of the graph is available: that is each node is assigned a number from a set $C = \{1 \dots c\}$, and no node shares a colour with any neighbour.

Given the measurement model from the section (SECTION), we can represent the linear system by concatenating the measurements of each node into a single system. Where, each row of the matrix represents the measurements of a single node. I.e we are trying to solve the following problem:

$$\min \|x\|_1 \text{ subject to } A_p x = b_p \quad (4.0.95)$$

The problem is coupled at each node by the variable x . To ease this issue, at each node create a local copy, denoted x_j (so there will be J copies of this variable).

Then, for consistency, we constrain the problem, so that each copy is identical: $x_1 = x_2 = \dots x_J$. Equivalently, given the graphical structure of the problem, we require that $x_i = x_j$ for each edge of the network.

So now we are solving the problem

$$\min \frac{1}{J} \sum_j \|x_j\|_1 \text{ subject to } A_p x = b_p \text{ and } x_i = x_j \{i, j\} \in E \quad (4.0.96)$$

This is exactly the kind of problem that can be attacked by the Alternating Direction Method of Multipliers, described previously. We present a short example for bipartite graphs, which can easily be generalised to graphs with many colours.

4.1 Example: bipartite graphs

We are here considering graphs which are connected, and 2-colourable, that is nodes 1 to c have colour 1 (red say), and nodes $c + 1$ to J have colour 2 (blue). We will demonstrate that we can decompose problem (4.0.96) into c problems which can be executed in parallel by minimising x at the 1st set of nodes, and $J - c$ problems which also can be executed in parallel by minimising at the second set of nodes.

Firstly, for compactness, we introduced the arc-incidence matrix of the graph B . This is the $J \times E$ matrix where each column corresponds to an edge $\{i, j\} \in E$, with the i th and j th entries equal to 1 and -1 respectively. With this notation, we can rewrite (4.0.96) as:

$$\min \frac{1}{J} \sum_j \|x_j\|_1 \text{ subject to } A_p x = b_p \text{ and } (B^T \circ I) \bar{x} = 0 \quad (4.1.97)$$

where $\bar{x} = (x_1, x_2, \dots x_J)$, and \circ is the Kronecker product.

In this form, the Lagrangian of the problem (4.1.97) can be written in the following way:

$$L(\bar{x}_1, \bar{x}_2; \lambda) = \frac{1}{J} \sum_{j \in C_1} \|x_j\|_1 + \frac{1}{J} \sum_{j \in C_2} \|x_j\|_1 + \quad (4.1.98)$$

$$\phi_1(\bar{x}_1, \lambda) + \phi_2(\bar{x}_2, \lambda) + \quad (4.1.99)$$

$$\rho \bar{x}_1^T (B_1 B_2^T \circ I_n) \bar{x}_2 \quad (4.1.100)$$

where

$$\phi_i(\bar{x}_i, \lambda) = \lambda^T (B_i^T \circ I_n) \bar{x}_i + \frac{\rho}{2} \|(B_i^T \circ I_n) \bar{x}_i\|^2 \quad (4.1.101)$$

$$= ((B_i^T \circ I_n) \lambda)^T \bar{x}_i + \frac{\rho}{2} \bar{x}_i^T (B_1 B_2^T \circ I_n) \bar{x}_i \quad (4.1.102)$$

Since, no nodes within C_i are neighbours $B_i B_i^T$ is a diagonal matrix (with the degree of node i as the i th entry). So, we can finally re-write

$$\phi_i(\bar{x}_i, \lambda) = \sum_{j \in C_i} \left(\left(\sum_{k \in \mathcal{N}_i} \text{sign}(k-i) \lambda_{\{i,k\}} \right)^T x_i + \frac{\rho}{2} D_i \|x_i\|^2 \right) \quad (4.1.103)$$

where the dual variable λ has been decomposed edge-wise so that $\lambda_{\{i,j\}} = \lambda_{\{j,i\}}$ and is associated with edge $\{i,j\}$ and the constraint $x_i = x_j$.

5 Results and Simulations

50 nodes were dropped into the square $[0, 1] \times [0, 1]$, and they make measurements according to the sensing model in section 2. A random 'spike-train' signal was generated: where the spikes were placed uniformly at random in a length 201 vector (representing 201 available channels in a power spectral density of a wideband signal).

For a series of SNR values, the system (2.1.78) was simulated, the mean squared error (mse):

$$\mathbb{E} \left(\frac{\|\hat{x} - x\|}{\|x\|} \right) \quad (5.0.104)$$

was calculated for both a centralised (SPG-L1) and decentralised solver (D-ADMM). In the equation above (5.0.104) \hat{x} refers to the estimator of the signal x , found as the solution to the basis pursuit denoise problem:

$$\underset{x}{\text{minimize}} \quad \frac{1}{2} \|\hat{x} - Ax\|_2^2 + \lambda \|x\|_1$$

The following diagram shows the results of this experiment:

This is clearly incorrect: for example it implies that by adding noise you can improve a communication system. The same system was simulated without a graph, and the following result was produced:

It remains an ongoing task find the discrepancy in the simulation.

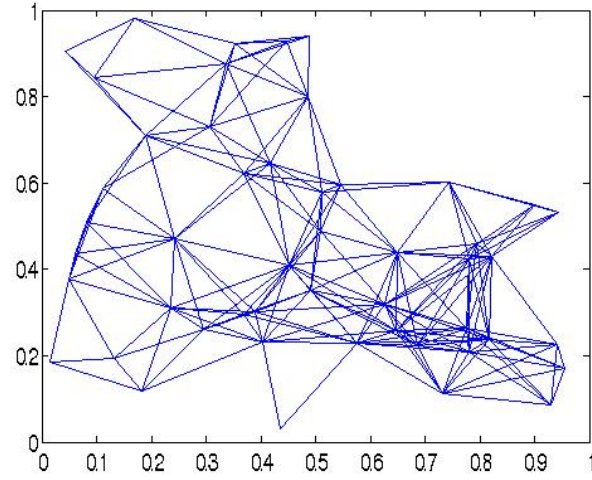


Figure 5.13: An example configuration of nodes

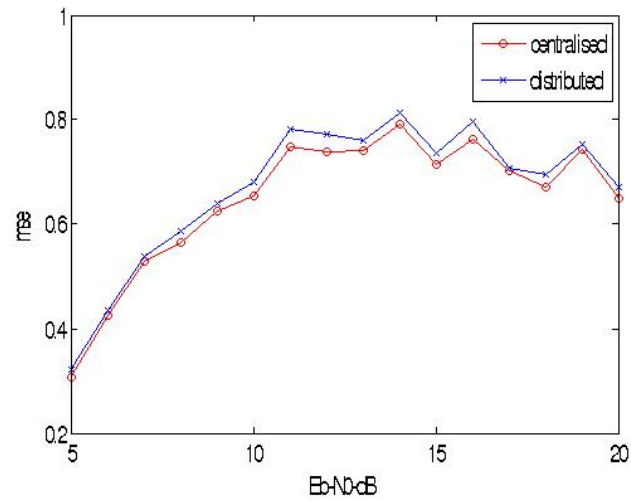


Figure 5.14: Mse vs SNR for the sensing model, for distributed and centralised solvers

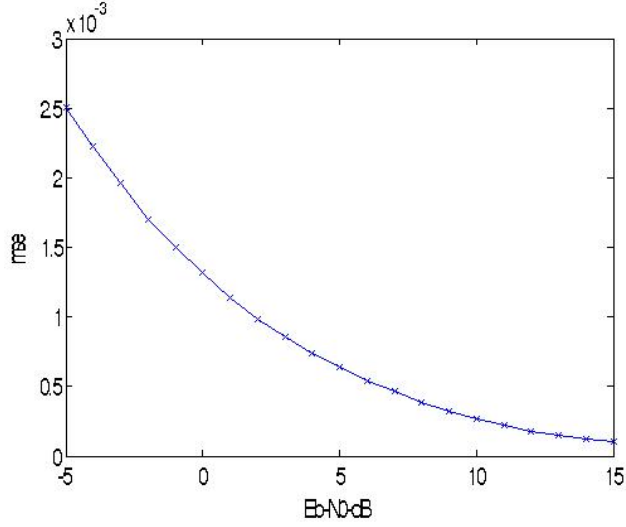


Figure 5.15: Mse vs SNR for the sensing model, for distributed and centralised solvers

6 Further Work and Project Plan

Initially, the simulations will be extended to included explicit spatial models. For example, place several simulated base-stations in the square $[0, 1] \times [0, 1]$, at position vectors $\{\mathbf{x}_s\}_{s=1}^{N_s}$.

Given some technical assumptions ([49]) - uncorrelated channel taps, and independent, stationary, and memoryless sources -, the spectral density can be written as follows:

$$\Phi_s = (f) = \sum_{\nu=1}^{N_b} \theta_{s\nu} b_{\nu}(f). \quad (6.0.105)$$

where b_{ν} are some suitably chosen bases (for example 2D Gaussians, Splines or rectangle functions), and $\theta_{i\nu}$ are expansion coefficients to be determined. The PSD at the receivers can be determined as:

$$\Phi_r = \sum_{s=1}^{N_s} \gamma_{sr} \Phi_s(f) + \sigma_r^2 \quad (6.0.106)$$

$$= \sum_{s=1}^{N_s} \gamma_{sr} \sum_{\nu=1}^{N_b} \theta_{s\nu} b_{\nu}(f) + \sigma_r^2 \quad (6.0.107)$$

$$= \mathbf{b}_r^T(f) \theta + \sigma_r^2 \quad (6.0.108)$$

This is exactly the kind of convex optimisation problem that can be attacked by the methods described in previous sections (by finding the minimum θ and σ pairs which satisfy the measurements.

The end result of this series of simulations should be a 'heat map' showing the locations of the PUs (with heat representing signal strength in decibels). This is a medium risk project - even though the experiments have been done before, setting up and solving the spatial model is quite involved. This work should take me to January at the latest.

Currently all the data being used to verify the correctness of the algorithms are synthetic. This is a expedient situation, as currently we do not possess real data. (It also makes debugging the system much easier). By using real data we can get a much better idea of how the system performs, as well as an idea of the real challenges such systems face.

Investigating the performance on real data would be a medium to high risk endeavour. It depends entirely on the data. Simply providing a noisy PSD would present challenges in terms of the computational load, and a slight re-thinking of the model may be required (this is an unlikely scenario). Detailed spatio-frequency data of central London would require a complete rethink of the approach, as the data would be far too rich for the model provided. Obtaining, cleaning and running the data on the model would take until Christmas, but it could be several months afterwards until definite conclusions could be drawn.

Current distributed optimisation algorithms (such as D-ADMM) require a colouring of the graph (which acts like a TDMA protocol) to co-ordinate message passing. However, as nodes join the graph, a recolouring is required (else the colouring will end up as n colours for n nodes). As colouring a graph is an NP-hard operation, this situation is undesirable. Investigating colouring schemes which may be sub-optimal but nevertheless can add nodes in an ad-hoc manner, but which still preserve the colouring would be one way to avoid the previous situation.

An alternative approach would be to investigate the use of well established graphical algorithms such as belief-propagation (this may require some thinking as to exactly how the coding problem translates to the convex optimisation scenario). Or, see if other graph parameters (or other multiple access schemes) will suffice to 'localise' the algorithm to make the system more flexible.

Move away from MWC-type measurements at the nodes (e.g. try random demodulators, naive sensing etc)

On the Group Testing front, explicit dependence between items is not currently modelled in the literature. You would expect, however, that defective items would be clustered together (you can imagine that soldiers returning from the same part of the world would have similar incidences of syphilis for example). Also, the source coding models used (Huffman/Shannon-Fano coding) are notoriously inflexible: they require full knowledge of the probabilities 'up-front', and are relatively computationally intensive. It would be desirable to change this.

One approach to both these problems, would be to investigate the use of the 'succinct' data structure ([50]), 'Succincter' developed by Mihai Patrescu as a more flexible source coding model (section 4 of ([50]) describes how to implement arithmetic coding using this structure). How to perform 'pilot' tests, when the items have dependencies is still unclear. This is a high risk piece of work, as it requires some new ideas about how to test items. Previously, the independence of defectives has been leveraged to design testing strategies. It's not as straightforwardly clear what to do when items aren't defective, let alone optimise such a procedure.

We also have a Group Testing algorithm which is a more direct generalisation of Hwang's original algorithm but we have run into difficulties analysing the 'pilot test' part. This is an ongoing project, and will only be worked on as ideas present themselves.

Finally, we have ideas about general strong converses in Group Testing (and sparse inference more generally) that we are exploring. This work is highly ill-defined at present, and as such presents a high-risk. However, it is likely to prove fruitful and can be worked on in conjunction with other

projects.

References

- [1] M. Aldridge, L. Baldassini, and O. Johnson. Group testing algorithms : bounds and simulations. 2013.
- [2] A. Aldroubi, H. Wang, and K. Zarringhalam. Sequential adaptive compressed sampling via huffman codes. pages 1–19.
- [3] L. Baldassini. The Capacity of Adaptive Group Testing. pages 2676–2680, 2013.
- [4] D. Baron. Bayesian compressive sensing via belief propagation. *Signal Processing, IEEE ...*, 58(1):269–280, 2010.
- [5] E. Burbidge and P. Levine. Cognitive Radio Technology A Study for Ofcom – Summary Report. (1):1–24, 2007.
- [6] E. Candès and J. Romberg. Sparsity and incoherence in compressive sampling. *Inverse Problems*, 23(3):969–985, June 2007.
- [7] E. Candès and T. Tao. Decoding by Linear Programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.
- [8] E. J. Candès, J. Romberg, and T. Tao. Robust Uncertainty Principles : Exact Signal Frequency Information. 52(2):489–509, 2006.
- [9] C. L. Chan, P. H. Che, S. Jaggi, and V. Saligrama. Non-adaptive probabilistic group testing with noisy measurements: Near-optimal bounds with efficient algorithms. In *2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1832–1839. Ieee, Sept. 2011.
- [10] S. S. Chen, D. L. Donoho, and M. a. Saunders. Atomic Decomposition by Basis Pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, Jan. 1998.
- [11] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.
- [12] D. L. Donoho. For Most Large Underdetermined Systems of Linear Equations Is Also the Sparsest Solution. *Communications on Pure and Applied Mathematics*, 59:797–829, 2006.
- [13] D. L. Donoho, J. M. Santos, and J. M. Pauly. Compressed Sensing MRI. (March 2008):72–82.
- [14] R. Dorfman. The Detection of Defective Members of Large Populations. *The Annals of Mathematical Statistics*, 14(4):436–440, 1943.
- [15] D. Du and F. K. Hwang. *Combinatorial group testing and its applications*. Series on Applied Mathematics. World Scientific, Singapore, New Jersey, London, 2000.
- [16] M. Duarte, M. Davenport, D. Takhar, J. Laska, K. Kelly, and R. Baraniuk. Single-Pixel Imaging via Compressive Sampling. *IEEE Signal Processing Magazine*, 25(2):83–91, Mar. 2008.

- [17] M. Ghozzi, F. Marx, M. Dohler, and J. Palicot. Cyclostationarity-Based Test for Detection of Vacant Frequency Bands, 2006.
- [18] F. K. Hwang. Method for Detecting All Defective by Members Group in a Population. *Journal of the American Statistical Association*, 67(339):605–608, 1972.
- [19] S. Ji, Y. Xue, and L. Carin. Bayesian Compressive Sensing. *IEEE Transactions on Signal Processing*, 56(6):2346–2356, June 2008.
- [20] M. Mishali and Y. Eldar. From theory to practice: Sub-Nyquist sampling of sparse wideband analog signals. *Selected Topics in Signal Processing*, . . . , 4(2):375–391, 2010.
- [21] D. Sejdinovic and O. Johnson. Note on noisy group testing: Asymptotic bounds and belief propagation reconstruction. In *2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 998–1003. Ieee, Sept. 2010.
- [22] O. C. o. U. S. Strategy. UNITED KINGDOM FREQUENCY ALLOCATION TABLE. (17), 2013.
- [23] Z. Tian and G. B. Giannakis. A Wavelet Approach to Wideband Spectrum Sensing for Cognitive Radio Networks. In *Cognitive Radio Oriented Wireless Networks and Communications, 2006. 1st International Conference on*.
- [24] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B* (. . . , 58(1):267–288, 1996.
- [25] J. A. Tropp and A. C. Gilbert. Via Orthogonal Matching Pursuit. 53(12):4655–4666, 2007.
- [26] T. Wadayama. Non-Adaptive Group Testing based on Sparse Pooling Graphs. <http://arxiv.org/pdf/1301.7519.pdf>, 2013:1–19, 2013.
- [27] T. Yucek and H. Arslan. A survey of spectrum sensing algorithms for cognitive radio applications. *IEEE Communications Surveys Tutorials*, 11(1):116–130, 2009.
- [28] R. Dorfman, “The detection of defective members of large populations,” *The Annals of Mathematical Statistics*, pp. 436–440, 1943.
- [29] D. Du and F. Hwang, *Combinatorial Group Testing and Its Applications*, ser. Series on Applied Mathematics. World Scientific, 1993.
- [30] M. Malyutov, “Search for sparse active inputs: a review,” in *Information Theory, Combinatorics and Search Theory*, ser. Lecture notes in Computer Science. London: Springer, 2013, vol. 7777, pp. 609–647.
- [31] G. Atia and V. Saligrama, “Boolean compressed sensing and noisy group testing,” *IEEE Trans. Inform. Theory*, vol. 58, no. 3, pp. 1880 –1901, March 2012.
- [32] D. Sejdinovic and O. T. Johnson, “Note on noisy group testing: Asymptotic bounds and belief propagation reconstruction,” in *Proceedings of the 48th Annual Allerton Conference on Communication, Control and Computing*, 2010, pp. 998–1003.

- [33] M. P. Aldridge, L. Baldassini, and O. T. Johnson, "Group testing algorithms: bounds and simulations," *IEEE Trans. Inform. Theory*, vol. 60, no. 6, pp. 3671–3687, 2014.
- [34] T. Wadayama, "An analysis on non-adaptive group testing based on sparse pooling graphs," in *2013 IEEE International Symposium on Information Theory*, 2013, pp. 2681–2685.
- [35] T. Li, C. L. Chan, W. Huang, T. Kaced, and S. Jaggi, "Group testing with prior statistics," 2014, see [arxiv:1401.3667](https://arxiv.org/abs/1401.3667).
- [36] G. Atia, S. Aeron, E. Ermiş, and V. Saligrama, "On throughput maximization and interference avoidance in cognitive radios," in *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE*. IEEE, 2008, pp. 963–967.
- [37] N. Shental, A. Amir, and O. Zuk, "Identification of rare alleles and their carriers using compressed sequencing," *Nucleic acids research*, vol. 38, no. 19, pp. e179–e179, 2010.
- [38] L. Baldassini, O. T. Johnson, and M. P. Aldridge, "The capacity of adaptive group testing," in *2013 IEEE International Symposium on Information Theory, Istanbul Turkey, July 2013*, 2013, pp. 2676–2680.
- [39] V. Tan and G. Atia, "Strong impossibility results for sparse signal processing," *IEEE Signal Processing Letters*, vol. 21, no. 3, pp. 260–264, March 2014.
- [40] F. K. Hwang, "A method for detecting all defective members in a population by group testing," *Journal of the American Statistical Association*, vol. 67, no. 339, pp. 605–608, 1972.
- [41] C. Aksoylar, G. Atia, and V. Saligrama, "Sparse signal processing with linear and non-linear observations: A unified Shannon theoretic approach," in *2013 IEEE Information Theory Workshop (ITW)*, Sept 2013, pp. 1–5.
- [42] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 489–509, 2006.
- [43] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.
- [44] V. V. Petrov, *Limit Theorems of Probability Theory: Sequences of Independent Random Variables*. Oxford: The Clarendon Press, 1995.
- [45] J. Mota, J. Xavier, P. Aguiar, M. Puschel, "A unified algorithmic approach to distributed optimisation." 2014.
- [46] J. Mota, J. Xavier, P. Aguiar, M. Puschel, "Distributed Compressed Sensing Algorithms: Completing the Puzzle." 2014.
- [47] J. Bazerque, G. Mateos, G. Giannakis, "Distributed Lasso for Network Linear Regression." 2014.

- [48] H. Zhang, Z. Zhang, Y. Chau Distributed Compressed Wideband Sensing in Cognitive Radio Sensor Networks. 2014.
- [49] J. Bazerque, G. Giannakis Distributed Spectrum Sensing for Cognitive Radio Networks by Exploiting Sparsity. 2014.
- [50] M. Patrescu Succincter. 2008.