University of Bristol

Doctoral Thesis

Compressive Wideband Spectrum Sensing for Cognitive Radios

Author:
Thomas Kealy

Supervisor:
Dr Oliver Johnson,
Dr Robert Pietchocki

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

in the

Communications, Statistics
Communications CDT

October 1, 2015

Declaration of Authorship

I, Thomas Kealy, declare that this thesis titled, "Compressive Wideband Spectrum Sensing for Cognitive Radios" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

"Thanks to my solid academic training, today I can write hundreds of words on virtually any topic without possessing a shred of information, which is how I got a good job in journalism."

Dave Barry

UNIVERSITY OF BRISTOL

Abstract

Faculty Name
Communications CDT

Doctor of Philosophy

Compressive Wideband Spectrum Sensing for Cognitive Radios

by Thomas Kealy

The Thesis Abstract is written here (and usually kept to just this page). The page is kept centered vertically so can expand into the blank space above the title too...

Acknowledgements

The acknowledgements and the people to thank go here, don't forget to include your project advisor...

Contents

List of Figures

List of Tables

List of Abbreviations

LAH	**L**ist **A**bbreviations **H**ere
WSF	**W**hat (it) **S**tands **F**or

For/Dedicated to/To my...

Chapter 1

The Probabilistic group testing problem

Group testing is a sparse inference problem, first introduced by Dorfman [?] in the context of testing for rare diseases. Given a large population of items $\mathcal{P}$, indexed by $\{1, \dots N\}$, 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, $y_i = \mathbb{I}\left(\sum_{j \in \mathcal{X}_i} U_j \geq 1\right)$, 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 [?], there has been considerable work on the question of how to design the sets $\mathcal{X}_i$ in order to minimise the number of tests T required. In this context, we briefly mention so-called combinatorial designs (see [?, ?] for a summary, with [?] giving invaluable references to an extensive body of Russian work in the 1970s and 1980s). Such designs typically aim to ensure that set-theoretic properties known as disjunctness and separability occur. In contrast, for simplicity of analysis, as well as performance of optimal order, it is possible to consider random designs. Here sets $\mathcal{X}_i$ are chosen at random, either using constructions such as independent Bernoulli designs [?, ?, ?] or more sophisticated random designs based on LDPC codes [?].

Much previous work has focussed on the Combinatorial group testing problem, where there are a fixed number of defectives K, and the defectivity vector $\mathbf{U}$ is chosen uniformly among all binary vectors of weight K. In contrast, in this paper we study a Probabilistic group testing problem as formulated for example in the work of Li et al. [?], in that we suppose each item is defective independently with probability p_i, or equivalently take U_i to be independent Bernoulli(p_i).

This Probabilistic framework, including non-uniform priors, is natural for many applications of group testing. For example, see [?], the cognitive radio problem can be formulated in terms of a population of communication bands in frequency spectra with some (unknown) occupied bands you must not utilise. Here, the values of p_i may be chosen based on some database of past spectrum measurements or other prior information. Similarly, as in Dorfman's original work [?] or more recent research [?] involving screening for genetic conditions, values of p_i might summarise prior information based on a risk profile or family history.

1.0.1 Group testing capacity

It is possible to characterize performance tradeoffs in group testing from an information-theoretic point of view – see for example [?, ?, ?, ?]. These papers have focussed on group testing as a channel coding problem, with [?, ?] explicitly calculating the mutual information. The paper [?] defined the capacity of a Combinatorial group testing procedure, which characterizes the number of bits of information about the defective set which we can learn per test. We give a more general definition here, which covers both the Combinatorial and Probabilistic cases.

Definition 1.0.1. *Consider a sequence of group testing problems where the ith problem has defectivity vector $\mathbf{U}^{(i)}$, and consider algorithms which are given $T(i)$ tests. We refer to a constant C as the (weak) group testing capacity if for any $\epsilon > 0$:*

1. *any sequence of algorithms with*

$$\liminf_{i\to\infty} \frac{H(\mathbf{U}^{(i)})}{T(i)} \geq C + \epsilon, \tag{1.0.1.1}$$

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

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

$$\liminf_{i\to\infty} \frac{H(\mathbf{U}^{(i)})}{T(i)} \geq C - \epsilon \tag{1.0.1.2}$$

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

Remark 1.0.2. *In the Combinatorial case of K defective items with all defective sets equally likely, $H(\mathbf{U}) = \log_2 \binom{N}{K}$, which is the term found in the denominator in [?, Eq. (1) and (2)]. In the Probabilistic case (as in [?]) we know $H(\mathbf{U}) = -\sum_{i=1}^{N} h(p_i)$ where $h(t) = -t\log_2 t - (1-t)\log_2(1-t)$ is the binary entropy function.*

Remark 1.0.3. *If for $\liminf_{i\to\infty} \frac{H(\mathbf{U}^{(i)})}{T(i)} \geq C + \epsilon$, the success probability $\mathbb{P}(\text{suc}) \to 0$ we say that C is the strong group testing capacity, following standard terminology in information theory. Such a result is referred to as a strong converse.*

1.0.2 Main results

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

Theorem 1.0.4 ([?])**.** *The strong capacity of the adaptive noiseless Combinatorial group testing problem is $C = 1$, in any regime such that $K/N \to 0$.*

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

$$\mathbb{P}(\text{suc}) \leq \frac{2^T}{\binom{N}{K}}, \tag{1.0.2.3}$$

which implied a strong converse ($C \leq 1$). This was complemented by showing that, in the Combinatorial case, an algorithm based on Hwang's Generalized Binary Splitting Algorithm (HGBSA) [?], [?] is essentially optimal in the required sense, showing that $C = 1$ is achievable.

It may be useful to characterize the Probabilistic group testing problem in terms of the effective sparsity $\mu^{(N)} := \sum_{i=1}^{N} p_i$. In particular, if the p_i are (close to) identical, we would expect performance similar to that in the Combinatorial case with $K = \mu^{(N)}$ defectives. As in [**?**], we focus on asymptotically sparse cases, where $\mu^{(N)}/N \to 0$. In contrast, Wadayama [**?**] considered a model where p_i are identical and fixed. The main result of the present paper is Theorem 1.2.9, stated and proved in Section 1.2.5 below, which implies the following Probabilistic group testing version of Theorem 1.0.4.

Corollary 1.0.5. *In the case where $p_i \equiv p$, the weak capacity of the adaptive noiseless Probabilistic group testing problem is $C = 1$, in any regime such that $\mu^{(N)}/N \to 0$ and $\mu^{(N)} \to \infty$.*

Again we prove our main result Theorem 1.2.9 using complementary bounds on both sides. First in Section 1.1.1 we recall a universal upper bound on success probability, Theorem 1.1.1, taken from [**?**], which implies a weak converse. In [**?**], Li et al. introduce the Laminar Algorithm for Probabilistic group testing. In Section 1.1.3 we propose a refined version of this Laminar Algorithm, based on Hwang's HGBSA [**?**], which is analysed in Section 1.2.5, and shown to imply performance close to optimal in the sense of capacity.

Bounds of [[**?**]] (see Theorem 1.1.3 below) which shows that $10eH(\mathbf{U})$ tests are required to guarantee convergence to zero success probability. Our calculation is to improve this to $H(\mathbf{U})$ plus an error term , which is optimal up to the size of the error term.

1.1 Algorithms and existing results

1.1.1 Upper bounds on success probability

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

Theorem 1.1.1. *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 1.0.1, 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 1.1.1, any such algorithms have $\mathbb{P}(\text{suc}) \leq 1/(1 + \epsilon)$, meaning that we have established that a weak converse holds.

Remark 1.1.2. *It remains an open and interesting problem to prove an equivalent of (1.0.2.3) as in [**?**, Theorem 3.1]. That is we hope to find an upper bound on success probability in a form which implies a strong converse, and hence that the strong capacity of Probabilistic group testing is equal to 1.*

1.1.2 Binary search algorithms

The main contribution of this work is to describe and analyse algorithms that will find the defective items. In brief, we can think of Hwang's HGBSA algorithm as dividing the population $\mathcal{P}$ into search sets $\mathcal{S}$. First, all the items in a search set $\mathcal{S}$ are tested

together, using a test set $\mathcal{X}_1 = \mathcal{S}$. If the result is negative ($y_1 = 0$), we can be certain that $\mathcal{S}$ contains no defectives. However, if the result is positive ($y_1 = 1$), $\mathcal{S}$ must contain at least one defective.

If $y_i = 1$, we can be guaranteed to find at least one defective, using the following binary search strategy. We split the set $\mathcal{S}$ in two, and test the 'left-hand' set, say $\mathcal{X}_2$. If $y_2 = 1$, then we know that $\mathcal{X}_2$ contains at least one defective. If $y_2 = 0$, then $\mathcal{X}_2$ contains no defective, so we can deduce that $\mathcal{S} \setminus \mathcal{X}_2$ contains at least one defective. By repeated use of this strategy, we are guaranteed to find a succession of nested sets which contain at least one defective, until $\mathcal{X}_i$ is of size 1, and we have isolated a single defective item.

However this strategy may not find every defective item in $\mathcal{S}$. To be specific, it is possible that at some stage both the left-hand and right-hand sets contain a defective. The Laminar Algorithm of [**?**] essentially deals with this by testing both sets. However, we believe that this is inefficient, since typically both sets will not contain a defective. Nonetheless, the Laminar Algorithm satisfies the following performance guarantees proved in [**?**, Theorem 2]:

Theorem 1.1.3. *The expected number of tests required by the Laminar Algorithm [**?**] is* $\leq 2H(\mathbf{U}) + 2\mu$. *Under a technical condition (referred to as non-skewedness), the success probability can be bounded by* $\mathbb{P}(\text{suc}) \geq 1 - \epsilon$ *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 [**?**]) $T = 2e(3+2)H(\mathbf{U}) = 10eH(\mathbf{U})$ 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.

1.1.3 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 $p_i \in \left[1/2C^r, 1/2C^{r-1}\right)$ in bin r.

Sort the items in each bin into sets s.t. the (normalised) probability of each set is less than 1/2.

Test each set in turn

if *The test is positive* **then**

  Arrange the items in the set on a Shannon-Fano/Huffman Tree and search the set for all the defectives it contains

end

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 1.2.7 (in terms of expected number of tests) and Theorem 1.2.9 (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 1.2.1, and optimised in the proof of Proposition 1.2.7.

2. The way in which we deal with sets $\mathcal{S}$ which contain more than one defective, as discussed in Remark 1.2.2 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 [?] and discussed in Section 1.2.2, with a new bound given in Lemma 1.2.4.

4. Careful analysis in Section 1.2.4 of the properties of search sets $\mathcal{S}$ gives Proposition 1.2.7, which shows that the expected number of tests required can be expressed as $H(\mathbf{U})$ plus an error term. In Section 1.2.5, we give an analysis of the error probability using Bernstein's inequality, Theorem 1.2.8, allowing us to prove Theorem 1.2.9.

1.1.4 Wider context: sparse inference problems

Recent work [?, ?] has shown that many arguments and bounds hold in a common framework of sparse inference which includes group testing and compressive sensing.

Digital communications, audio, images, and text are examples of data sources we can compress. We can do this, because these data sources are sparse: they have fewer degrees of freedom than the space they are defined upon. For example, images have a well known expansion in either the Fourier or Wavelet bases. The text of an English document will only be comprised of words from the English dictionary, and not all the possible strings from the space of strings made up from the characters $\{a, \ldots, z\}$.

Often, once a signal has been acquired it will be compressed. However, the compressive sensing paradigm introduced by [?, ?] shows that this isn't necessary. In those papers it was shown that a 'compressed' representation of a signal could be obtained from random linear projections of the signal and some other basis (for example White Gaussian Noise). The question remains, given this representation how do we recover the original signal? For real signals, a simple linear programme suffices. Much of the work in this area has been couched in terms of the sparsity of the signal and the various bases the signal can be represented in (see for example [?, ?]).

1.2 Analysis and new bounds

1.2.1 Searching a set of bounded ratio

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

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

$$\max_{i,j \in \mathcal{S}} \frac{p_j}{p_i} \leq C. \tag{1.2.1.4}$$

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

Lemma 1.2.1. *Consider a set $\mathcal{S}$ satisfying the Bounded Ratio Condition with constant C and write $P_{\mathcal{S}} = \sum_{j\in\mathcal{S}} p_j$. In a Shannon–Fano tree for the probability distribution $\overline{p}_i := p_i/P_{\mathcal{S}}$, each item has length $\ell_i^{(\mathcal{S})}$ bounded by*

$$\ell_i^{(\mathcal{S})} \le \ell_{\max}^{(\mathcal{S})} := \frac{h(\mathcal{S})}{P_{\mathcal{S}}} + \log_2 C + \log_2 P_{\mathcal{S}} + 1, \tag{1.2.1.5}$$

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 (1.2.1.4)

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

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

$$-P_{\mathcal{S}} \log_2 p_i \le h(\mathcal{S}) + P_{\mathcal{S}} \log_2 C. \tag{1.2.1.6}$$

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

$$\begin{aligned} \ell_i^{(\mathcal{S})} &= \lceil -\log_2 \overline{p}_i \rceil && (1.2.1.7)\\ &\le -\log_2 p_i + \log_2 P_{\mathcal{S}} + 1 && (1.2.1.8)\\ &\le \left(\frac{h(\mathcal{S})}{P_{\mathcal{S}}} + \log_2 C\right) + \log_2 P_{\mathcal{S}} + 1. \end{aligned}$$

and the result follows by (1.2.1.6). □

Next we describe our search strategy:

Remark 1.2.2. *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 section II-B 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 $\le \ell$ 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 $\le \ell U + 1$ tests.

Lemma 1.2.3. *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 $\mathbb{E}T_{\mathcal{S}} \le T_{\mathrm{bd}}(\mathcal{S})$ for*

$$T_{\mathrm{bd}}(\mathcal{S}) := h(\mathcal{S}) + P_{\mathcal{S}} \log_2 C + P_{\mathcal{S}} \log_2 P_{\mathcal{S}} + P_{\mathcal{S}} + 1. \tag{1.2.1.9}$$

Proof. Using the algorithm of Remark 1.2.2, laid out on the Shannon-Fano tree constructed in Lemma 1.2.1, 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 (1.2.1.5).

Recall that we write U_i for the indicator of the event that the ith item is defective, we will write $U_{\mathcal{S}} = \sum_{i\in\mathcal{S}} U_i$ for the total number of defectives in $\mathcal{S}$, and $l_i^{(\mathcal{S})}$ for the

length of the word in the Shannon Fano tree. As discussed in Remark 1.2.2 this search procedure will take

$$\begin{aligned} T_{\mathcal{S}} &= 1 + \sum_{i \in \mathcal{S}} U_i \ell_i^{(\mathcal{S})} \\ &= \sum_{i \in \mathcal{S}} p_i \ell_i^{(\mathcal{S})} + 1 + \sum_{i \in S} \ell_i^{(\mathcal{S})} (U_i - p_i) \\ &\leq \sum_{i \in \mathcal{S}} p_i \ell_{\max}^{(\mathcal{S})} + 1 + \sum_{i \in \mathcal{S}} V_i^{(\mathcal{S})} \\ &= P_{\mathcal{S}} \ell_{\max}^{(\mathcal{S})} + 1 + \sum_{i \in S} V_i^{(\mathcal{S})} \\ &\leq T_{\mathrm{bd}}(\mathcal{S}) + \sum_{i \in S} V_i^{(\mathcal{S})} \quad \text{tests.} \end{aligned} \tag{1.2.1.10}$$

Here we write $V_i^{(\mathcal{S})} = \ell_i^{(\mathcal{S})}(U_i - p_i)$, which has expectation zero, and (1.2.1.10) follows using the expression for $\ell_{\max}^{(\mathcal{S})}$ given in Lemma 1.2.1. □

1.2.2 Discarding low probability items

As in [?], we use a probability threshold θ, and write $\mathcal{P}^*$ for the population having removed items with $p_i \leq \theta$. If an item lies in $\mathcal{P} \setminus \mathcal{P}^*$ we do not test it, and simply mark it as non-defective. This truncation operation gives an error if and only if some item in $\mathcal{P} \setminus \mathcal{P}^*$ is defective. By the union bound, this truncation operation contributes a total of $\mathbb{P}(\mathcal{P} \setminus \mathcal{P}^* \text{ contains a defective}) \leq \rho := \sum_{i=1}^n p_i \mathbb{I}(p_i \leq \theta)$ to the error probability.

Lemma 1.2.4. *Choosing $\theta(P_e)$ such that*

$$-\log_2 \theta(P_e) = \min\left(\log_2\left(\frac{2n}{P_e}\right), \frac{2H(\mathbf{U})}{P_e}\right) \tag{1.2.2.11}$$

ensures that

$$\mathbb{P}(\ \mathcal{P} \setminus \mathcal{P}^* \textit{ contains a defective}) \leq P_e/2. \tag{1.2.2.12}$$

Proof. The approach of [?] is essentially to bound $\mathbb{I}(p_i \leq \theta) \leq \theta/p_i$ so that $\rho = \sum_{i=1}^n p_i \mathbb{I}(p_i \leq \theta) \leq \sum_{i=1}^n p_i(\theta/p_i) = n\theta$. Hence, choosing a threshold of $\theta = P_e/(2n)$ guarantees the required bound on ρ.

We combine this with another bound, constructed using a different function: $\mathbb{I}(p_i \leq \theta) \leq (-\log_2 p_i)/(-\log_2 \theta)$, so that

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

so we deduce the result. □

1.2.3 Searching the entire set

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

The probability threshold θ means that there will be a finite number of such bins, with the index B of the last bin defined by the fact that $1/(2C^B) \leq \theta < 1/(2C^{B-1})$, meaning that $(B-1)\log_2 C < -\log_2(2\theta)$, so

$$B \leq \frac{-\log_2(2\theta)}{\log_2 C} + 1. \tag{1.2.3.13}$$

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

Definition 1.2.5. *A set of items $\mathcal{S}$ is said to be full if $P_{\mathcal{S}} = \sum_{i\in\mathcal{S}} p_i \geq \frac{1}{2}$.*

Our splitting procedure is as follows: we create a list of possible sets $\mathcal{S}_1, \mathcal{S}_2, \ldots$. For i increasing from 0 to B, we place items from bin i into sets $\mathcal{S}_{b_i+1}, \ldots, \mathcal{S}_{b_{i+1}}$, for some b_i, where $b_0 = 0$. Taking the items from bin i, while $\mathcal{S}_{b_i+1}$ is not full (has total probability $< \frac{1}{2}$) we will place items into it. Once enough items have been added to fill $\mathcal{S}_{b_i+1}$, we will proceed in the same way to fill $\mathcal{S}_{b_i+2}$, and so on until all the items in bin i have been assigned to sets $\mathcal{S}_{b_i+1}, \ldots, \mathcal{S}_{b_{i+1}}$, where $\mathcal{S}_{b_{i+1}}$ may remain not full.

Proposition 1.2.6. *This splitting procedure will divide $\mathcal{P}^*$ into search sets $\mathcal{S}_1, \ldots, \mathcal{S}_G$, where the total number of sets is*

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

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

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 $1, \ldots, B$:

1. For each bin i, it is possible that the last set $\mathcal{S}_{b_{i+1}}$ will not be full, but every other set corresponding to that bin will be full. Hence, there are no more than B sets which are not full.

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

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

$$\mu = \sum_{i\in\mathcal{P}} p_i \tag{1.2.3.14}$$

$$\geq \sum_{i\in\mathcal{P}^*} p_i = \sum_{j=1}^{G} P_j \tag{1.2.3.15}$$

$$\geq \sum_{j:\ \mathcal{S}_j \text{ full}} P_j \geq |\mathcal{S}_j \text{ full}|\,\frac{1}{2}. \tag{1.2.3.16}$$

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 (1.2.3.13). □

1.2.4 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 1.2.7. *Given a population $\mathcal{P}$ where (independently) item i is defective with probability p_i, we recover all defective items in $\mathcal{P}^*$ in T tests with $\mathbb{E}T \leq T_{\mathrm{bd}}$, where*

$$T_{\mathrm{bd}} := (H(\mathbf{U}) + 3\mu + 1) + 2\sqrt{\mu\,(-\log_2(2\theta))}. \tag{1.2.4.17}$$

Proof. Given a value of C, Proposition 1.2.6 shows that our splitting procedure divides $\mathcal{P}^*$ into G sets $\mathcal{S}_1, \ldots, \mathcal{S}_G$, such that each set $\mathcal{S}_j$ satisfies the Bounded Ratio Condition with constant C and has total probability $P_j \leq 1$. Using the notation of Lemma 1.2.3, $T = \sum_{j=1}^{G} T_{\mathcal{S}_j}$, where $\mathbb{E}T_{\mathcal{S}_j} \leq T_{\mathrm{bd}}(\mathcal{S}_j)$.

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

$$\begin{aligned}
&\sum_{j=1}^{G} T_{\mathrm{bd}}(\mathcal{S}_j) \\
&\leq \sum_{j=1}^{G} (h(\mathcal{S}_j) + P_j(\log_2 C + 1) + 1) \\
&= \sum_{j\in\mathcal{P}^*} -p_j \log_2 p_j + \mu(\log_2 C + 1) + G \\
&\leq \sum_{j\in\mathcal{P}^*} h(p_j) + 3\mu + 1 + \left(\frac{-\log_2(2\theta)}{\log_2 C} + \mu \log_2 C\right) \\
&\leq (H(\mathbf{U}) + 3\mu + 1) && (1.2.4.18) \\
&+ \left(\frac{-\log_2(2\theta)}{\log_2 C} + \mu \log_2 C\right). && (1.2.4.19)
\end{aligned}$$

This follows by the bound on G in Proposition 1.2.6, 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 (1.2.4.19). 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. □

1.2.5 Controlling the error probabilities

Although Section 1.2.4 proves that $\mathbb{E}T \leq T_{\mathrm{bd}}$, to bound the capacity, we need to prove that with high probability T is not significantly larger than T_{bd}. This can be done using Bernstein's inequality (see for example Theorem 2.8 of [**?**]):

Theorem 1.2.8 (Bernstein). *For zero-mean random variables V_i which are uniformly bounded by $|V_i| \le M$, if we write $L := \sum_{j=1}^{n} \mathbb{E}V_j^2$ then*

$$\mathbb{P}\left(\sum_{j=1}^{n} V_j \ge t\right) \le \exp\left(-\frac{t^2}{4L}\right), \text{ for any } 0 \le t \le \tfrac{L}{M}. \tag{1.2.5.20}$$

We deduce the following result:

Theorem 1.2.9. *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*

$$T_{\mathrm{nec}} = T_{\mathrm{bd}} + \psi H(\mathbf{U}), \tag{1.2.5.21}$$

where T_{bd} is given in (1.2.4.17).

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

$$\mathbb{P}(\mathrm{suc}) \ge 1 - \frac{1}{2}\sqrt{\frac{\mu}{H(\mathbf{U})}} - \exp\left(-\left(\frac{L}{4M^2}\right)^{1/3}\right). \tag{1.2.5.22}$$

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

Proof. We first prove the success probability bound (1.2.5.22). 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 $\mathcal{P}\backslash\mathcal{P}^*$, 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 1.2.4) the first event has probability $\le P_e/2$, contributing $\frac{1}{2}\sqrt{\mu/H(\mathbf{U})}$ to (1.2.5.22).

Our analysis of the second error event is based on the random term from Equation (1.2.1.10), which we previously averaged over but now wish to bound. There will be an error if $T_{\mathrm{nec}} \le T$, or (rearranging) if

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

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

We have discarded elements with probability below θ, as given by (1.2.2.11), and by design all the sets $\mathcal{S}$ have total probability $P_{\mathcal{S}} \le 1$. Using (1.2.1.8) we know that the V_i are bounded by

$$|V_i| \le \ell_i \le -\log_2 p_i + \log_2 P_{\mathcal{S}} + 1 \le -\log_2\theta + 1. \tag{1.2.5.23}$$

Hence, the conditions of Bernstein's inequality, Theorem 1.2.8, are satisfied. Observe that since all $l_j \le M$,

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

Hence Theorem 1.2.8 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 (1.2.5.22) follows.

We next consider the capacity bound of 2). Since $-\log_2\theta \leq 2H/P_e$, using (1.2.4.17) and (1.2.5.21)

$$\begin{aligned}\frac{T_{\text{nec}}}{H} &= \frac{T_{\text{bd}}}{H} + \psi\\ &= 1 + 3\frac{\mu}{H} + \frac{1}{H} + 2\sqrt{\frac{\mu}{HP_e}} + \psi\\ &= 1 + 3\frac{\mu}{H} + \frac{1}{H} + 2\left(\frac{\mu}{H}\right)^{1/4} + \psi, \qquad (1.2.5.24)\end{aligned}$$

which in our regime of interest is $\leq 1+\epsilon$ in the limit, since $\Phi = \left(L/\left(4M^2\right)\right) \to 0$ by assumption. □

Proof of Corollary 1.0.5. In the case where all p are identical, $\mu = Np$, $H = Np(-\log p)$, so $\mu/H = 1/(-\log p) \to 0$. Similarly, $L = Np(-\log_2 p)^2$ and $M = (-\log_2 p)$ so that $L/M^2 = Np \to \infty$ as required. □

1.3 Results

The performance of 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 an output distribution whose uniformity can be controlled via the parameter α, 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 simulated the algorithm 1000 times. We plot the empirical distribution of tests, varying θ as well as the uniformity/concentration of the probability distribution (via the parameter α of the Dirichlet distribution). We also plot (in figure (1.3.1), the theoretical lower and upper bounds on the number of Tests required for successful recovery alongside the empirical number tests (all as a function of θ).

Note that the Upper bound is not optimal and there still is some room for improvement. Note also that the lower bound degrades with θ_i. The lower bound (T_{LCHJ}) was generated according to Theorem (1.1.1).

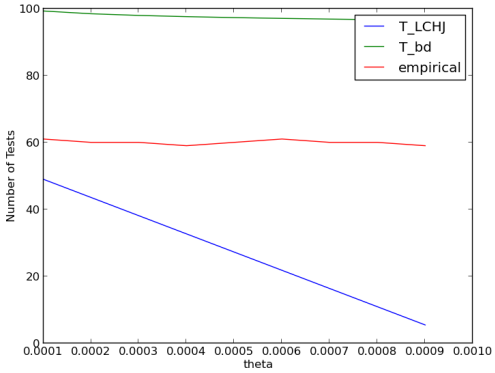

FIGURE 1.3.1: Theoretical lower and upper bounds and empirical Test frequencies as functions of θ

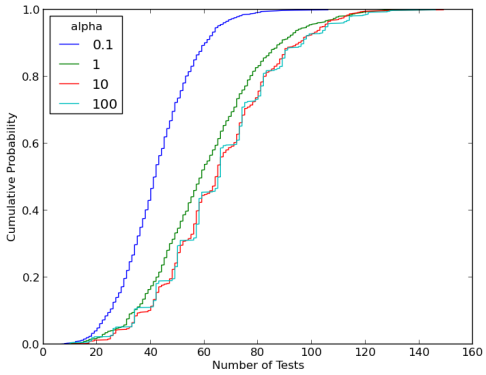

FIGURE 1.3.2: Cumulative distribution curves of the modified Hwang algorithm with fixed $\theta = 0.0001$ and α varying

Figures (1.3.2) and (1.3.3) 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.

1.4 Discussion

We have introduced and analysed an algorithm for Probabilistic group testing which uses 'just over' $H(\mathbf{U})$ tests to recover all the defectives with high probability. Combined with a weak converse taken from [**?**], this allows us to deduce that the weak capacity of Probabilistic group testing is $C = 1$. These results are illustrated by simulation.

For simplicity, this work has concentrated on establishing a bound T_{bd} in (1.2.4.17) which has leading term $H(\mathbf{U})$, and not on tightening bounds on the coefficient of μ in (1.2.4.17). For completeness, we mention that this coefficient can be reduced from 3, under a simple further condition:

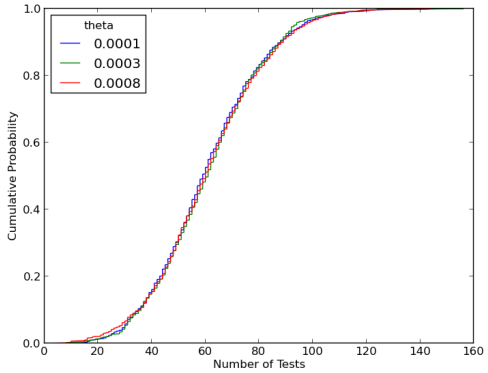

FIGURE 1.3.3: Cumulative distribution curves for fixed $\alpha = 1$ and varying θ

Remark 1.4.1. *For some $c \leq 1/2$, we assume that all the $p_i \leq c$, and we alter the definition of 'fullness' to assume that a set is full if it has total probability less than α. In this case, the term $P_{\mathcal{S}} \log_2 P_{\mathcal{S}}$ in (1.2.1.9) becomes $P_{\mathcal{S}} \log_2(\alpha + c)$, the bound in (1.2.3.14) 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 (1.2.4.19). 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 1.1.1, in order prove a strong converse, and hence confirm that the strong capacity is also equal to 1.

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

Acknowledgments

This work was supported by the Engineering and Physical Sciences Research Council [grant number `EP/I028153/1`]; Ofcom; and the University of Bristol. The authors would particularly like to thank Gary Clemo of Ofcom for useful discussions.

Chapter 2

Constrained Optimisation on Graphs

2.1 Introduction

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

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

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

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

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

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

The realisation of any Cognitive Radio standard (such as IEEE 802.22 [**?**]), requires the co-existence of primary (e.g. TV users) and secondary (everybody else who wants to use TVWS spectrum) users of the frequency spectrum to ensure proper interference mitigation and appropriate network behaviour. We note, that whereas TVWS bands are an initial step towards dynamic spectrum access, the principles and approaches we describe are applicable to other frequency bands - in particular it makes ultra-wideband spectrum sensing possible.

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

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

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

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

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

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

The structure of the report is as follows: in section 3 we introduce the sensing model, in section 2.3 we describe the distributed reconstruction algorithm [?], and finally in section 3.2 we show some results of the reconstruction quality of this model.

2.2 ADMM

Given a set of measurements of the form

$$y = Ax + n \tag{2.2.0.1}$$

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

$$L = \frac{1}{2} ||Ax - y||_2^2 + \lambda ||x||_1 \tag{2.2.0.2}$$

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

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

$$\begin{aligned} &\arg\min_x f(x) + g(z) \\ &\text{s.t } Ux + Vz = c \end{aligned} \tag{2.2.0.3}$$

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

ADMM consists of iteratively minimising the augmented Lagrangian

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

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

$$x^{k+1} := \arg\min_x L_\rho\left(x, z^k, \eta^k\right) \tag{2.2.0.4}$$

$$z^{k+1} := \arg\min_z L_\rho\left(x^{k+1}, z, \eta^k\right) \tag{2.2.0.5}$$

$$\eta^{k+1} := \eta^k + \rho\left(Ux^{k+1} + Vz^{k+1} - c\right) \tag{2.2.0.6}$$

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

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

ADMM can be formulated as an iterative MAP estimation procedure for the problem (2.2.0.2). We can write (2.2.0.2) in constrained form as:

$$\frac{1}{2}\|Ax - b\|_2^2 + \lambda\|z\|_1 \tag{2.2.0.7}$$

$$\text{s.t } z = x \tag{2.2.0.8}$$

i.e this is of the form (2.2.0.3) with $f(x) = \|Ax - y\|_2^2$, $g(z) = \lambda\|z\|_1$, $U = I$, $V = -I$, and $c = 0$.

The associated (augmented) Lagrangian is:

$$L_\rho = \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|z\|_1 + \eta(x - z) + \frac{\rho}{2}\|x - z\|^2 \tag{2.2.0.9}$$

The ADMM iterations for LASSO, which can be found by alternately differentiating (2.2.0.9) with respect to x,z and η, are (in closed form):

$$x^{k+1} := \left(A^TA + \rho I\right)^{-1}\left(A^Ty + \rho\left(z^k - \eta^k/\rho\right)\right) \tag{2.2.0.10}$$

$$z^{k+1} := S_{\lambda/\rho}\left(x^{k+1} + \eta^k/\rho\right) \tag{2.2.0.11}$$

$$\eta^{k+1} := \eta^k + \rho\left(x^{k+1} - z^{k+1}\right) \tag{2.2.0.12}$$

where $S_{\lambda/\rho}(\circ)$ is the soft thresholding operator: $S_\gamma(x)_i = \text{sign}(x_i)(|x_i| - \gamma)^+$.
These can be found differentiating (2.2.0.9) with respect to x and z as follows:

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

as

$$\frac{\partial}{\partial x}||F(x)||_2^2 = 2\left(\frac{\partial}{\partial x}F(x)\right)F(x) \tag{2.2.0.13}$$

by the chain rule, and $\partial/\partial x(Ax) = -A^T$ (see the Matrix Cookbook) as differentiation exchanges a linear operator with its adjoint.
Setting (2.2.0.13) to zero and collecting like terms:

$$\left(A^TA + \rho I\right)x = A^Ty + \rho z - \eta \tag{2.2.0.14}$$

so we find the optimal x is:

$$x = \left(A^TA + \rho I\right)^{-1}\left(A^Ty + \rho\left(z - \eta/\rho\right)\right) \tag{2.2.0.15}$$

note that this estimator is a weighted average of the ordinary least squares estimate (A^Ty) and a Gaussian prior. This is to be expected, as the minimisation problem w.r.t x is an l_2-regularised MAP problem.
for $z > 0$

$$\frac{\partial L}{\partial z} = \lambda + \rho(x - z) - \eta \tag{2.2.0.16}$$

from which we obtain:

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

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

$$\frac{\partial L}{\partial z} = -\lambda + \rho(x - z) \tag{2.2.0.17}$$

setting (2.2.0.17) to zero we obtain:

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

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

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

i.e.

$$|\, x + \frac{\eta}{\rho}\,| \leq \frac{\lambda}{\rho} \tag{2.2.0.18}$$

combining (2.2.0.17), (2.2.0.16), (2.2.0.18) together we find the optimal z is:

$$z = \text{sign}(x + \frac{\eta}{\rho}) \max\left(\mid x + \frac{\eta}{\rho} \mid -\frac{\lambda}{\rho}, 0 \right) \tag{2.2.0.19}$$

Together (2.2.0.15), (2.2.0.19) and the third step of (2.2.0.12) constitute the steps of the ADMM algorithm.

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

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

$$y = x + w \tag{2.2.0.20}$$

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

$$\hat{x} = \arg\max_x \mathbb{P}_{x|y}(x|y) \tag{2.2.0.21}$$

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

$$\hat{x} = \arg\max_x \left[\log \mathbb{P}_w(y - x) + \log \mathbb{P}(x) \right] \tag{2.2.0.22}$$

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

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

$$\mathbb{P}(x) = \frac{1}{\sqrt{2}\sigma} \exp -\frac{\sqrt{2}}{\sigma}|x| \tag{2.2.0.23}$$

Note that $f(x) = \log \mathbb{P}_x(x) \ -\frac{\sqrt{2}}{\sigma}|x|$, and so by differentiating $f'(x) = -\frac{\sqrt{2}}{\sigma}\text{sign}(x)$ Taking the maximum of 2.2.0.22 we obtain:

$$\frac{y - \hat{x}}{\sigma_n^2} - \frac{\sqrt{2}}{\sigma} sign(x) = 0 \tag{2.2.0.24}$$

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

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

or

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

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

2.2.1 The Proximity Operator

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

Definition 2.2.1 (Proximity Operator)**.**

$$\mathrm{Prox}_f(y) := \arg\min_{y \in \mathcal{H}} f(y) + \frac{1}{2}||y - x||^2 \tag{2.2.1.25}$$

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

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

The corresponding Moreau envelope is defined as

Definition 2.2.2 (Moreau Envelope)**.**

$$\mathrm{M}_f(y) := \min_{y \in \mathcal{H}} f(y) + \frac{1}{2}||y - x||^2 \tag{2.2.1.26}$$

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

Properties

Theorem 2.2.3 (Moreau '65)**.** *Let $f \in \Gamma$ and f^* be its Fenchel conjugate. Then the following are equivalent:*

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

Theorem 2.2.4 (Moreau '65)**.** *Let $f \in \Gamma$. Then for all $z \in \mathcal{H}$*

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

Theorem 2.2.5 (Moreau '65)**.** *The Moreau envelope is (Frechet) differentiable, with*

$$\nabla \mathrm{M}_f = Id - \mathrm{Prox}_f = \mathrm{Prox}_{f^*} \tag{2.2.1.27}$$

Theorem 2.2.6 (Moreau '65)**.** $\mathrm{Prox}_f : (\mathcal{H}, ||\circ||) \leftarrow (\mathcal{H}, ||\circ||)$ *is 1-Lipchitz continuous.*

Motivation

We are solving problems of the following form:

$$\min_{x \in \mathcal{H}} f(x) + g(z) \tag{2.2.1.28}$$

$$\text{s.t } x - z = 0 \tag{2.2.1.29}$$

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

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

and then performing the following iterative minimisation:

$$x^{k+1} := \arg\min_x L_\rho\left(x, z^k, \eta^k\right) \tag{2.2.1.30}$$
$$z^{k+1} := \arg\min_z L_\rho\left(x^{k+1}, z, \eta^k\right) \tag{2.2.1.31}$$
$$\eta^{k+1} := \eta^k + \rho\left(x^{k+1} + z^{k+1}\right) \tag{2.2.1.32}$$

i.e.

$$x^{k+1} := \arg\min_x \left(f(x) + \eta^{kT}x + \frac{\rho}{2}\left|\left|x - z^k\right|\right|^2\right) \tag{2.2.1.33}$$
$$z^{k+1} := \arg\min_z \left(g(z) - \eta^{kT}z + \frac{\rho}{2}\left|\left|x^{k+1} - z\right|\right|^2\right) \tag{2.2.1.34}$$
$$\eta^{k+1} := \eta^k + \rho\left(x^{k+1} + z^{k+1}\right) \tag{2.2.1.35}$$

pulling the linear terms into the quadratic ones we get:

$$x^{k+1} := \arg\min_x \left(f(x) + \frac{\rho}{2}\left|\left|x - z^k + (1/\rho)\eta^k\right|\right|^2\right) \tag{2.2.1.36}$$
$$z^{k+1} := \arg\min_z \left(g(z) + \frac{\rho}{2}\left|\left|x^{k+1} - z - (1/\rho)\eta^k\right|\right|^2\right) \tag{2.2.1.37}$$
$$\eta^{k+1} := \eta^k + \rho\left(x^{k+1} + z^{k+1}\right) \tag{2.2.1.38}$$

i.e.

$$x^{k+1} := \mathrm{Prox}_f\left(z^k - u^k\right) \tag{2.2.1.39}$$
$$z^{k+1} := \mathrm{Prox}_f\left(x^{k+1} + u^k\right) \tag{2.2.1.40}$$
$$u^{k+1} := u^k + \left(x^{k+1} + z^{k+1}\right) \tag{2.2.1.41}$$

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

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

Examples

Example 2.2.7 (Indicator)**.** *From the definition*

$$\text{Prox}_I(x) := \arg\min_y I_C(y) + \frac{1}{2}||y-x||^2 \tag{2.2.1.42}$$

$$= \arg\min_{y\in C} \frac{1}{2}||y-x||^2 \tag{2.2.1.43}$$

$$= P_C(x) \tag{2.2.1.44}$$

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

Example 2.2.8 (l_2 norm). *For $f(y) = \frac{\mu}{2}||y||^2$ the* Prox *operator is:*

$$\text{Prox}_f(x) := \arg\min_y \frac{\mu}{2}||y||^2 + \frac{1}{2}||y-x||^2 \tag{2.2.1.45}$$

$$= \frac{1}{1+\mu}x \tag{2.2.1.46}$$

Example 2.2.9 (l_1 norm). $f = \|x\|_1$

$$\text{Prox}_f(x) := \text{sign}(x_i)(|x_i| - \gamma)^+ = S_\gamma(x)_i \tag{2.2.1.47}$$

Example 2.2.10 (Elastic Net). *Consider*

$$f(x) = \lambda\|x\|_1 + \mu||x|| \tag{2.2.1.48}$$

$$\text{Prox}_f(x) := \frac{\lambda}{1+\mu}S_\gamma(x)_i \tag{2.2.1.49}$$

Example 2.2.11 (Fused Lasso). *Consider*

$$f(x) = \|x\|_1 + \sum_{i=1}^{d-1}(x_i - x_{i-1}) \tag{2.2.1.50}$$

i.e the sum of the l_1 and TV norms

$$\text{Prox}_f(x) := \text{Prox}_{l_1} \circ \text{Prox}_{TV} = S_\gamma(\text{Prox}_{TV})_i \tag{2.2.1.51}$$

Example 2.2.12 (Consensus). *Suppose we want to solve a problem such as:*

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

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

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

$$\textit{subject to} \quad x_i - z = 0$$

where x_i are local variables (for example local to each node in a spectrum sensing) and $x_i - z = 0$ are the consensus constraints. Consensus and regularisation can be achieved

by adding a regularisation term $g(z)$ - for example $g(z) = \lambda||x||_1$ corresponds to the LASSO, and the f_i would be $f_i = ||A_i x_i - b||_2^2$.

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

$$L_\rho(x, y) = \sum_i^n \left(f_i(x_i) + y_i^T (x_i - z) + \frac{\rho}{2} \|x_i - z\|_2^2 \right) \tag{2.2.1.52}$$

The ADMM iterations for this Lagrangian are:

$$x_i^{k+1} := \arg\min x_i \left(f_i(x_i) + y_i^{kT} (x_i - z) + \frac{\rho}{2} \|x_i - z\|_2^2 \right) \tag{2.2.1.53}$$

$$z^{k+1} := \frac{1}{n} \sum_i^n \left(x_i^{k+1} + (1\rho) y_i^k \right) \tag{2.2.1.54}$$

$$y_i^{k+1} := y_i^k + \rho \left(x_i^{k+1} - z^{k+1} \right) \tag{2.2.1.55}$$

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

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

$$x_i^{k+1} := \arg\min x_i \left(f_i(x_i) + y_i^{kT} \left(x_i - \bar{x}^k \right) + \frac{\rho}{2} \|x_i - \bar{x}^k\|_2^2 \right) \tag{2.2.1.56}$$

$$y_i^{k+1} := y_i^k + \rho \left(x_i^{k+1} - z^{k+1} \right) \tag{2.2.1.57}$$

where

$$\bar{x}^k = \frac{1}{n} \sum_i^n x_i^k \tag{2.2.1.58}$$

This algorithm can be summarised as follows: in each iteration

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

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

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

2.2.2 Statistical Interpretation

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

2.2.3 Acceleration

2.3 Constrained Optimisation on Graphs

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

Individually nodes make the following measurements (as discussed in section 3):

$$\mathbf{y}_p = \mathbf{A}_p \mathbf{x} + \mathbf{n}_p \tag{2.3.0.59}$$

where $\mathbf{A}_p$ is the p^{th} row of the sensing matrix from (3.0.1.4), and the system (3.0.1.4) is formed by concatenating the individual nodes' measurements together.

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

To find the $\mathbf{x}$ we are seeking (the solution to the linear system, 3.0.1.4), to each node we give a copy of $\mathbf{x}, \mathbf{x}_p$ and we constrain the copies to be identical across all edges in the network. Each node, thus has a separate optimisation to solve, subject to the constraint that it is consistent with its neighbours.

The problem then is to solve:

$$\begin{aligned} \arg\min_{\bar{x}} \sum_{c=1}^{C} \sum_{j \in c} f(x_j) + \frac{\lambda}{J} g(x_j) \\ \text{and } x_i = x_j \text{ if } \{i, j\} \in E \\ \text{and } x_i = z_i \ \forall i \in \{1, \dots, C\} \end{aligned} \tag{2.3.0.60}$$

with a particular special case being:

$$\begin{aligned} \arg\min_{\bar{x}} \sum_{c=1}^{C} \sum_{j \in c} \|A_j x_j - y_j\|_2^2 + \frac{\lambda}{J} \|z\|_1 \\ \text{and } x_i = x_j \text{ if } \{i, j\} \in E \\ \text{and } x_i = z_i \ \forall i \in \{1, \dots, C\} \end{aligned} \tag{2.3.0.61}$$

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

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

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

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

$$\bar{x} = \sum_{c=1}^{C} w_c \otimes x_c = \left[x_{c(1)}^T, \dots, x_{c(J)}^T \right]^T \tag{2.3.0.62}$$

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

These constraints can be written more compactly by introducing the node-arc incidence matrix B: a V by E matrix where each column is associated with an edge $(i, j) \in E$ and has 1 and -1 in the ith and jth entry respectively. Figures (2.3.1) and (2.3.2) show examples of a network and it's associated incidence matrix.

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

$$\sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_c = 0 \tag{2.3.0.63}$$

note that $\left(B^T \otimes I_n\right) \in \mathbb{R}^{nE \times nJ}$. Together (2.3.0.62) and (2.3.0.63), suggests that the problem (2.3.0.61) can be re-written as:

$$\begin{aligned} \arg\min_{\bar{x}} & \sum_{c=1}^{C} \sum_{j \in C_c} f\left(x_j\right) + \frac{\lambda}{J} g\left(z_j\right) \\ \text{s.t.} & \sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_c = 0 \\ & \text{and } \bar{x}_c - \bar{z}_c = 0 \end{aligned} \tag{2.3.0.64}$$

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

The global Augmented Lagrangian [?] for the problem (2.3.0.64) can be written down as:

$$\begin{aligned} L_\rho = \sum_{c=1}^{C} (\sum_{j \in c} & f\left(x_j\right) + \frac{\lambda}{J} g\left(z_j\right) + \\ & + \theta^T \left(\bar{x}_j - \bar{z}_j\right) + \frac{\rho}{2} \left|\left|\bar{x}_j - \bar{z}_j\right|\right|_2^2) + \\ & + \eta^T \left(B_c^T \otimes I_n\right) \bar{x}_c + \frac{\rho}{2} \left|\left| \sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_c \right|\right|_2^2 \end{aligned} \tag{2.3.0.65}$$

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

To make it possible that 2.3.0.65 can be posed as a constrained optimisation problem at each node, we introduce the following variable (so that the the final term of 2.3.0.65 is separable across edges of the graph):

Definition 2.

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

where we have used the definition (2.3.0.62) in the second line, and the property of Kronecker products $(A \otimes C)(B \otimes D) = (AB \otimes CD)$ between the second and third lines, and we write $B_c = w_c^T B$.

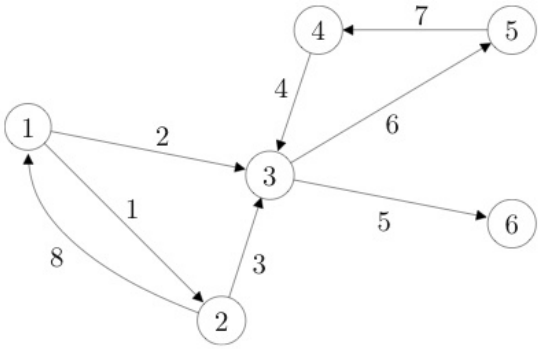

FIGURE 2.3.1: An example of a network

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

FIGURE 2.3.2: The incidence matrix associated with Figure (2.3.1)

The terms $\| \sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_c \|^2$ and $\eta^T \left(B_c^T \otimes I_n\right) \bar{x}_c$ of (2.3.0.65), can be decomposed across edges, using the following lemma:

Lemma 2.3.1 (Edge Decomposition)**.**

$$\left|\left| \sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_c \right|\right|^2 = \sum_{j \in C_1} \left(D_j \left|\left|x_j\right|\right|_2^2 - \sum_{k \in N_j} x_j^T x^k \right) \tag{2.3.0.66}$$

and

$$\eta^T \sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_1 = \sum_{l \in C_c} \sum_{m \in N_l} sign\,(m-l)\, \eta_{ml}^T x_l \tag{2.3.0.67}$$

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

Proof.

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

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

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

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

$$\eta = \sum_{c=1}^{C} w_c \otimes \eta_c \tag{2.3.0.68}$$

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

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

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

Adding together this with the lemma, lets us write (2.3.0.65) as:

$$\begin{aligned} L_\rho = \sum_{c=1}^{C} \sum_{j \in C_c} &\left(f\left(x_j\right) + \beta g\left(z_j\right)\right) + \nu^T x_j \\ &\theta\left(x_j - z_j\right) + \frac{\rho}{2} D_i \left|\left|x_j\right|\right|^2 + \frac{\rho}{2} \|x_j - z_j\|^2 \end{aligned} \tag{2.3.0.69}$$

where we have defined:

$$\nu_i = \left(\sum_{k \in \mathcal{N}_i} sign\left(k - i\right) \eta_{\{i,k\}} - \rho x_k \right) \tag{2.3.0.70}$$

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

Then by differentiating (2.3.0.69) with respect to x_j and z_j we can find closed forms for the updates as:

Theorem 1.

$$x_j^{k+1} := \left(A_j^T A_j + (\rho D_J + 1)I\right)^{-1} \left(A_j^T y_j + z^k - \nu^{kT}\right) \tag{2.3.0.71}$$

$$z_j^{k+1} := S_{\beta/\rho}\left(x_j^{k+1}\right) \tag{2.3.0.72}$$

$$\theta_j^{k+1} := \theta_j^k + \rho\left(x^{k+1} - z^{k+1}\right) \tag{2.3.0.73}$$

$$\eta_j^{k+1} := \eta_j^k + \rho\left(\sum_{m \in N_j} z_m^k - z_j^k\right) \tag{2.3.0.74}$$

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

Chapter 3

Compressive Sensing Architechtures

3.0.1 Modulated Wideband Converter

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

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

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

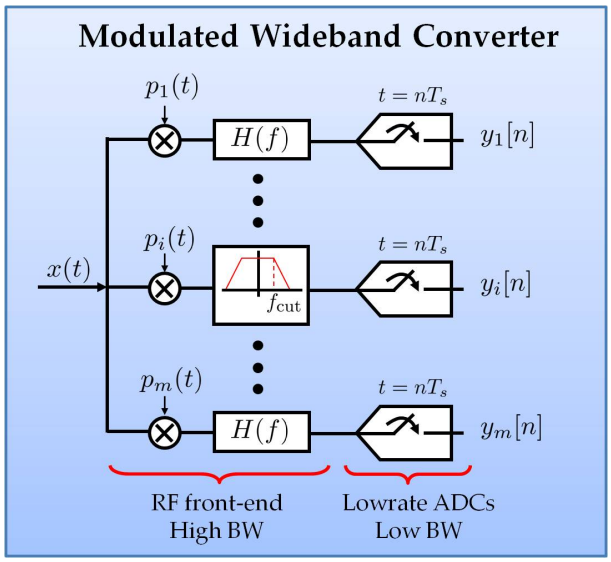

FIGURE 3.0.1: Mse vs SNR for the sensing model, with AWGN only, showing the performance of distributed and centralised solvers

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

$$p_i(t) = \sum_{l=-\infty}^{\infty} c_{il} \exp\left(jlt\frac{2\pi}{T_p}\right) \tag{3.0.1.1}$$

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

$$\begin{aligned} X_i(f) &= \int_{-\infty}^{\infty} x(t)\, p_i(t)\, dt \\ &= \sum_{l=-\infty}^{\infty} c_{il} X(f - l f_p) \end{aligned} \tag{3.0.1.2}$$

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

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

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

$$Y_i\left(e^{j2\pi f T_s}\right) = \sum_{l=-L_0}^{+L_0} c_{il} X(f - l f_p) \tag{3.0.1.3}$$

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

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w} \tag{3.0.1.4}$$

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

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

$$\mathbf{A}^{m \times L} = \mathbf{S}^{m \times L} \mathbf{F}^{L \times L} \mathbf{D}^{L \times L} \mathbf{H}^{L \times L} \tag{3.0.1.5}$$

The system 3.0.1.4 can then be solved (in the sense of finding the sparse vector $\mathbf{x}$ by convex optimisation via minimising the objective function:

$$\frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{x}\|_1 \tag{3.0.1.6}$$

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

3.0.2 Random Demodulator

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

$$x(t) = \sum_{n=1}^{N} \alpha_n \psi_n(t) \tag{3.0.2.7}$$

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

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

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

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

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

$$y[m] = \int_{-\infty}^{\infty} x(\tau)\, p_c(\tau)\, h(t-\tau)\, |_{t=m\mathcal{M}}\, d\tau \tag{3.0.2.8}$$

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

$$y[m] = \sum_{n=1}^{N} \alpha_n \int_{-\infty}^{\infty} \psi_n(t)\, p_c(\tau)\, h(m\mathcal{M}-\tau)\, d\tau \tag{3.0.2.9}$$

we see that the output can be written as:

$$y = Ax \tag{3.0.2.10}$$

with

$$A_{m,n} = \int_{-\infty}^{\infty} \psi_n(t)\, p_c(\tau)\, h(m\mathcal{M}-\tau)\, d\tau \tag{3.0.2.11}$$

3.1 Joint Space-Frequency Model

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

$$\phi_s = \sum_b \beta_{bs} \psi_b(f) \tag{3.1.0.12}$$

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

Given this, the psd at the *rth* receiver is:

$$\phi_r = \sum_s g_{sr}\phi_s = \sum_s g_{sr} \sum_b \beta_{bs}\psi_b(f) \tag{3.1.0.13}$$

where

$$g_{sr} = \exp\left(-||x_r - x_s||_2^{\alpha}\right) \tag{3.1.0.14}$$

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

This model can be summarised using Kronecker products as follows:

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

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

$$y_r = \left(e_r^T \bigotimes I_{n_b}\right) y \tag{3.1.0.15}$$

with,

$$y = \left(\tilde{G} \bigotimes I_{n_b}\right) \phi \tag{3.1.0.16}$$

Now, we have

$$\phi = e_s \bigotimes \phi_s \tag{3.1.0.17}$$

so,

$$y = \left(\tilde{G} \bigotimes I_{n_b}\right) \left(e_s \bigotimes \phi_s\right) \tag{3.1.0.18}$$

finally we have,

$$y_r = \left(e_r^T \bigotimes I_{n_b}\right) \left[\left(\tilde{G} \bigotimes I_{n_b}\right) \left(e_s \bigotimes \phi_s\right)\right] \tag{3.1.0.19}$$

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

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

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

3.2 Results

The model described in section (3), equation (3.0.1.4) was simulated, with a wideband signal of 201 channels and a network of 50 nodes (i.e. the signal will be sampled at a 1/4 of rate predicted by Nyquist theory). The mixing patterns were generated from iid Gaussian sources (i.e the matrix S had each entry drawn from an iid Gaussian source). Monte Carlo simulations were performed at SNR values ranging from 5 to 20, and the expected Mean Squared Error (MSE) of solutions of a centralised solver (spgl1) and a distributed solver (ADMM) were calculated over 10 simulations per SNR value. The results can be seen in fig (3.2.3).

The MSE was calculated as follows:

$$\frac{||Z^k - Z*||}{||Z*||} \tag{3.2.0.20}$$

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

These results indicate that for both centralised and distributed solvers, adding noise to the system results in a degrading of performance. Interestingly note, that the distributed solver seems to (slightly) outperform the centralised solver at all SNRs. This is counter-intuitive, as it would be expected that centralised solvers knowing *all* the

available information would outperform distributed solutions. We conjecture that the updates described in section (2.3), take into account differences in noise across the network. The distributed averaging steps, which form the new prior for each node, then penalise updates from relatively more noisy observations. This corroborates observations from [**?**].

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

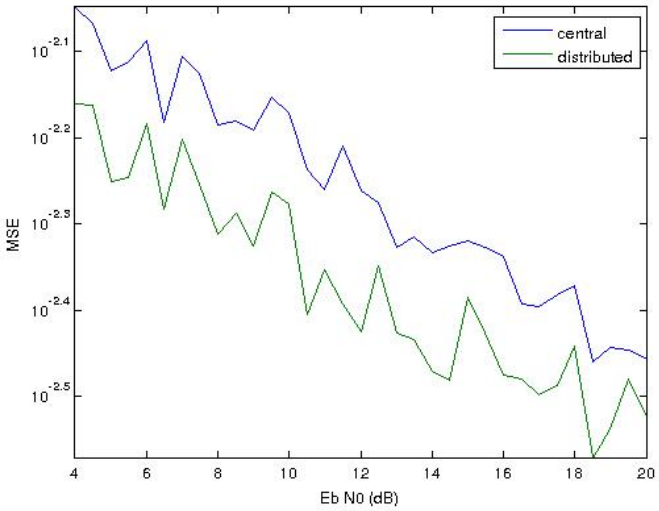

FIGURE 3.2.2: Mse vs SNR for the sensing model, with AWGN only, showing the performance of distributed and centralised solvers

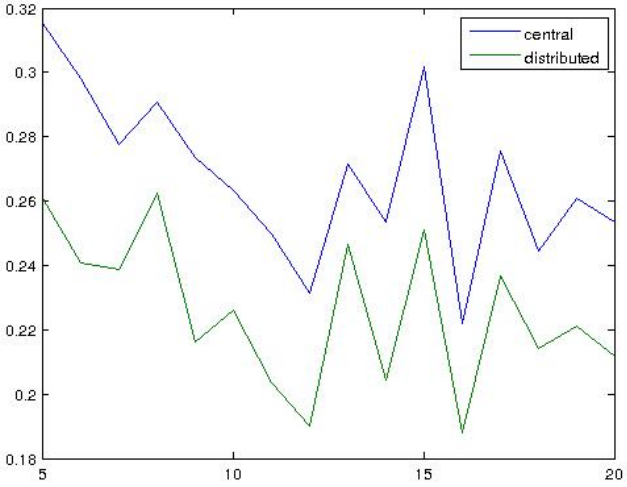

FIGURE 3.2.3: Mse vs SNR for the sensing model, showing the performance of distributed and centralised solvers

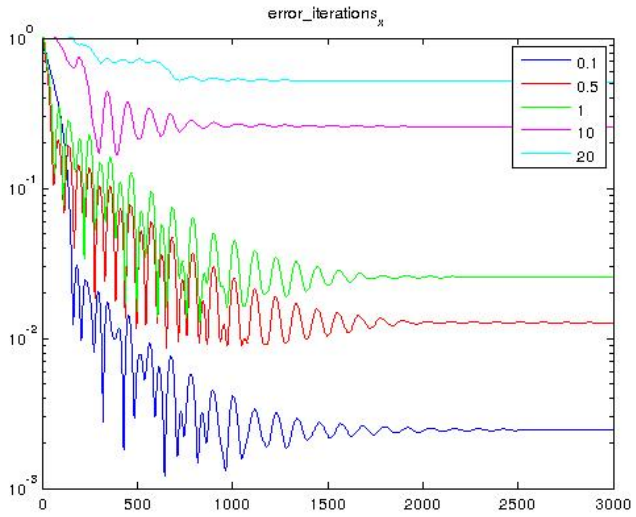

FIGURE 3.2.4: The progress of the distributed solver as a function of the number of iterations, with different values of the regression parameter λ

3.3 Conclusions

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

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

We plan to work on larger, more detailed, models for the frequency spectra and to accelerate the convergence via Nesterov type methods to smooth the convergence of the distributed algorithm [**?**]. Specifically, we seek to dampen the ringing seen in Figure 3.2.10

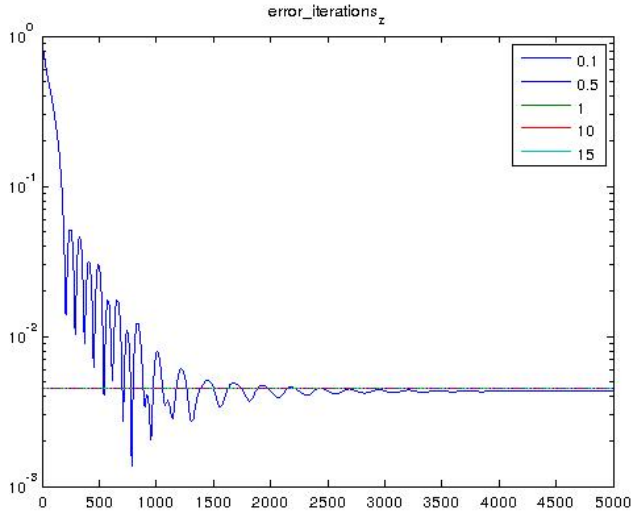

FIGURE 3.2.5: The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$

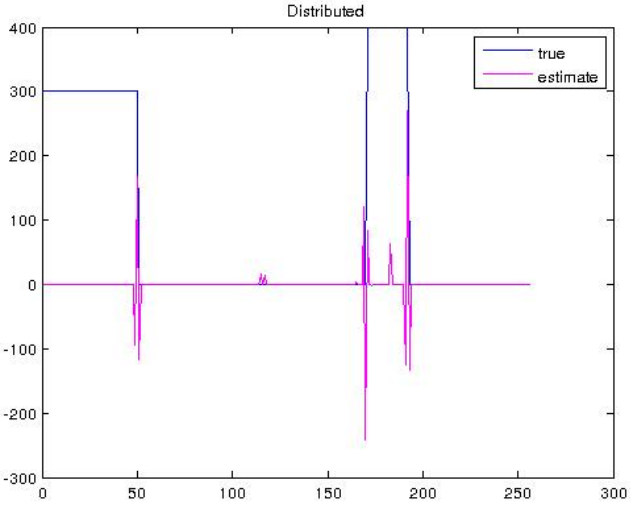

FIGURE 3.2.6: The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$

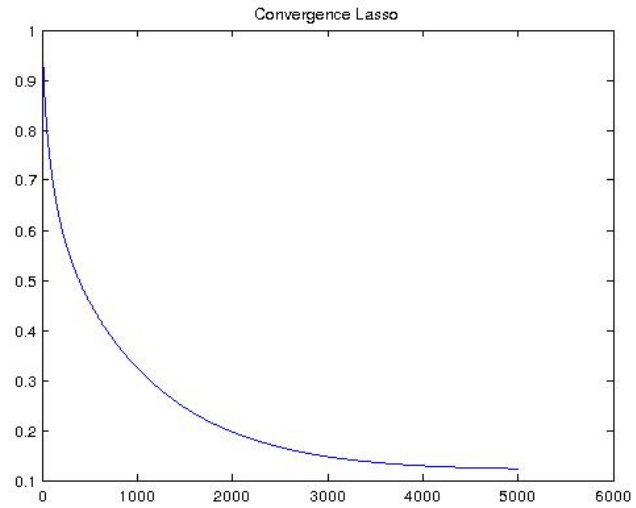

FIGURE 3.2.7: The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$

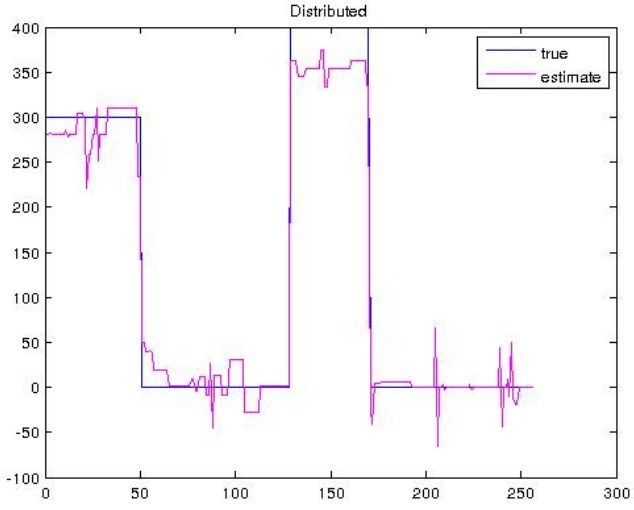

FIGURE 3.2.8: The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$

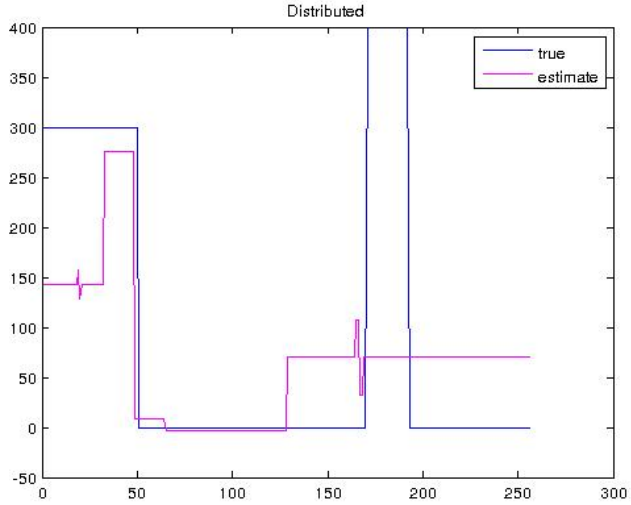

FIGURE 3.2.9: The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$

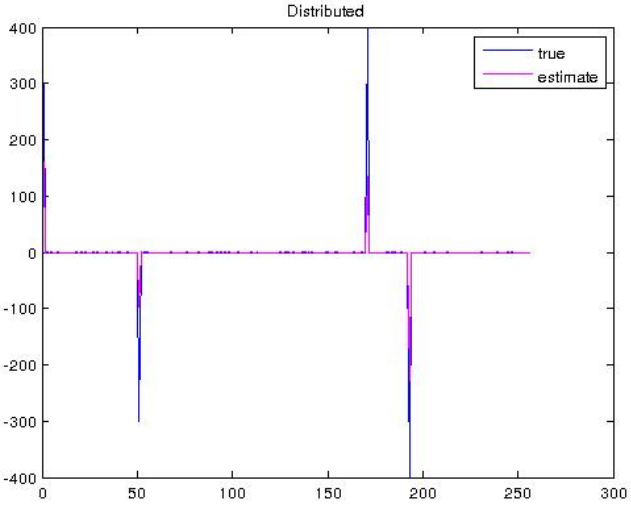

FIGURE 3.2.10: The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$

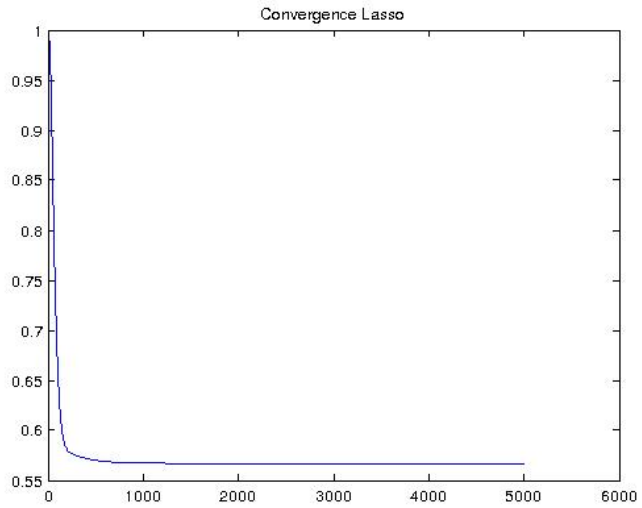

FIGURE 3.2.11: The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$

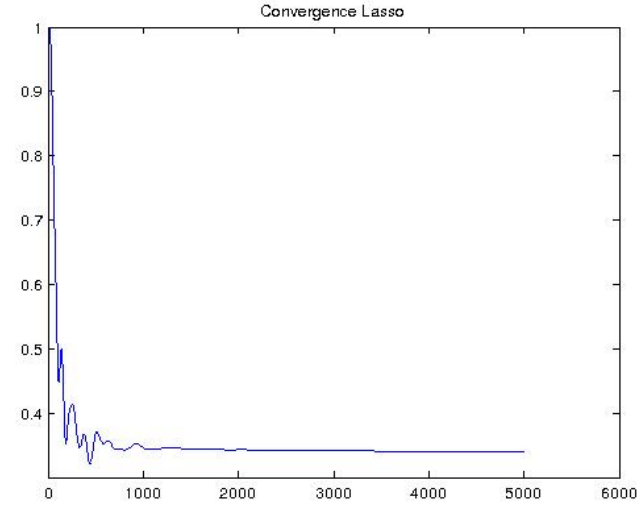

FIGURE 3.2.12: The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$

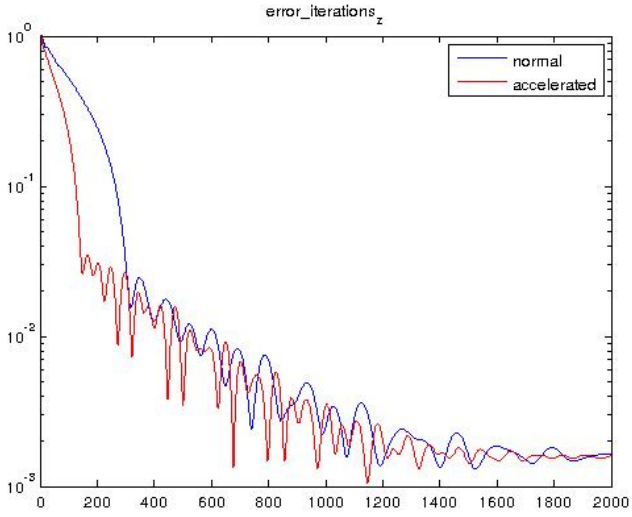

FIGURE 3.2.13: The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$

Chapter 4

Rectangular basis

4.1 Preliminaries

Consider the basis defined by the function:

$$f_i(x) = \begin{cases} 1 & \text{if } x \leq i \\ 0 & \text{otherwise} \end{cases} \tag{4.1.0.1}$$

That is, f_i is a left-hand step function. We define the inner product between two vectors as follows:

Definition 4.1.1.

$$\langle x, y \rangle = x^T y = \sum_i x_i y_i \tag{4.1.0.2}$$

where x_i, y_i are the components of the vectors x, y in the i^{th} direction with respect to some basis vectors e_i.

We can define a matrix representation for the set of basis vectors f_i, by taking all inner products between all pairs basis vectors:

Definition 4.1.2.

$$F_{n,ij} = \langle f_i, f_j \rangle \tag{4.1.0.3}$$

This matrix has the representation:

$$F_{ij} = min(i, j) \tag{4.1.0.4}$$

An example of such a matrix is:

$$F_n = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 & 2 \\ 1 & 2 & 3 & 3 & 3 \\ 1 & 2 & 3 & 4 & 4 \\ 1 & 2 & 3 & 4 & 5 \end{pmatrix} \tag{4.1.0.5}$$

This matrix is invertible.

Theorem 4.1.3.

$$det(F_n) = 1 \tag{4.1.0.6}$$

Proof. Consider the matrix F^n. Subtract the $n-1$th column from the nth. We obtain a matrix with 0 on the final column except the entry $F^n_(n, n) = 1$. Since the top $(n-1) \times (n-1)$ is F^{n-1} we find that

$$det(F_n) = 1 \times det(F_{n-1}) \tag{4.1.0.7}$$

By recursion and $det(F_1) = 1$ we have $det(F_n) = 1$.

□

This matrix can be factorised as $F = LL^T$ where

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \dots 0 \\ 1 & 1 & 0 & 0 & 0 \dots 0 \\ 1 & 1 & 1 & 0 & 0 \dots 0 \\ \dots & & & & \\ 1 & 1 & 1 & 1 & 1 \dots 1 \end{pmatrix} \tag{4.1.0.8}$$

From this it follows that

$$F^{-1} = \begin{pmatrix} 2 & -1 & 0 & 0 & 0 \dots 0 \\ -1 & 2 & -1 & 0 & 0 \dots 0 \\ 0 & -1 & 2 & -1 & 0 \dots 0 \\ \dots & & & & \\ 0 & 0 & 0 & 0 \dots -1 & 1 \end{pmatrix} \tag{4.1.0.9}$$

Another matrix we will use a lot of is:

$$D = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \dots 0 \\ -1 & 1 & 0 & 0 & 0 \dots 0 \\ 0 & -1 & 0 & 0 & 0 \dots 0 \\ \dots & & & & \\ 0 & 0 & 0 & 0 \dots -1 & 1 \end{pmatrix} \tag{4.1.0.10}$$

and by direct computation:

$$DL = I \tag{4.1.0.11}$$

4.2 Spectrum Sensing

We model our PSD signal g as a linear combination of the basis functions (4.1.0.1):

$$g(x) = \sum_i a_i f_i \tag{4.2.0.12}$$

To find the a_i, we correlate (take the inner product of) the signal against the basis (4.1.0.1).

Definition 4.2.1.

$$\begin{aligned} h_j &= \langle g, f_j \rangle && (4.2.0.13) \\ &= \sum_j g(x) f_j(x) && (4.2.0.14) \\ &= \sum_j a_i f_i(x) f_j(x) && (4.2.0.15) \\ &= a_i \langle f_i, f_j \rangle && (4.2.0.16) \\ &\left(= \sum_{x=1}^{j} g(x) \right) && (4.2.0.17) \end{aligned}$$

In matrix language $h = Fa^T$. This is the inner product between the signal g and the basis functions f_i.

Definition 4.2.2.

$$d_i(x) = \begin{cases} 1 & \text{if } x = i \\ -1 & \text{if } x = i-1 \\ 0 & \text{otherwise} \end{cases} \tag{4.2.0.18}$$

i.e. d_i is the i^{th} row of the matrix D.

Theorem 4.2.3.

$$D^T g = a \tag{4.2.0.19}$$

Proof. From the definition of g (4.2.0.12)

$$D^T g = D^T \sum_i a_i f_i \tag{4.2.0.20}$$

$$= \sum_i a_i d_j^T f_i \tag{4.2.0.21}$$

$$= a\langle d_j, f_i \rangle \tag{4.2.0.22}$$

$$= a \tag{4.2.0.23}$$

as

$$\langle d_j, f_i \rangle == \begin{cases} 1 & \text{if } i \leq j \\ 0 & \text{otherwise} \end{cases} \tag{4.2.0.24}$$

□

This means

$$g = \left(D^T\right)^{-1} a = \left(D^{-1}\right)^T a \tag{4.2.0.25}$$

so using (4.1.0.11),

$$g = L^T a \tag{4.2.0.26}$$

To recover $\hat{a}$, we minimise

$$||h - Fa||_2^2 \tag{4.2.0.27}$$

in the noiseless case, and

$$||h_\varepsilon - Fa||_2^2 + \lambda ||a||_1 \tag{4.2.0.28}$$

in the noisy case, where

$$(h_\varepsilon)_j = \langle (g + \varepsilon), f_j \rangle \tag{4.2.0.29}$$

and $\varepsilon \sim \mathcal{N}(0, \sigma^2\ I)$.
From $\hat{a}$ we can recover $\hat{g}$ from the following relation:

$$\hat{g} = L^T \hat{a} \tag{4.2.0.30}$$

4.3 Results

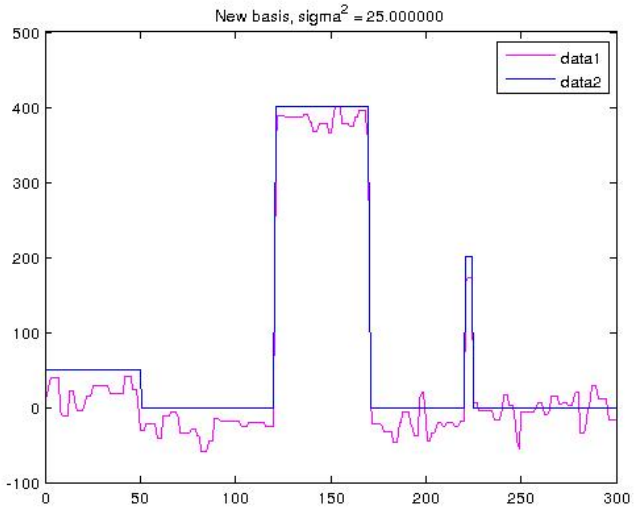

FIGURE 4.3.1

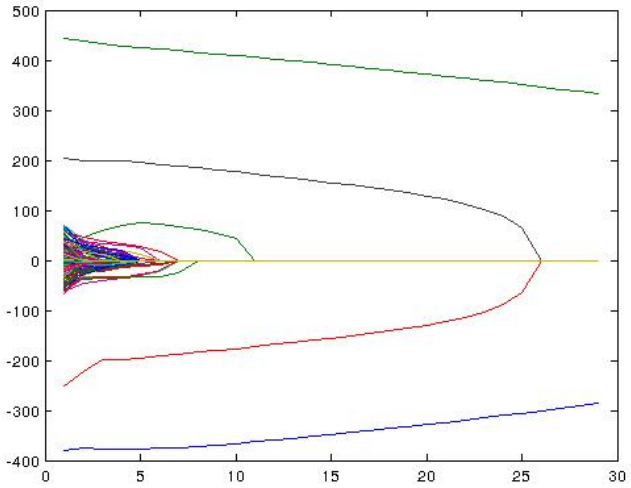

FIGURE 4.3.2

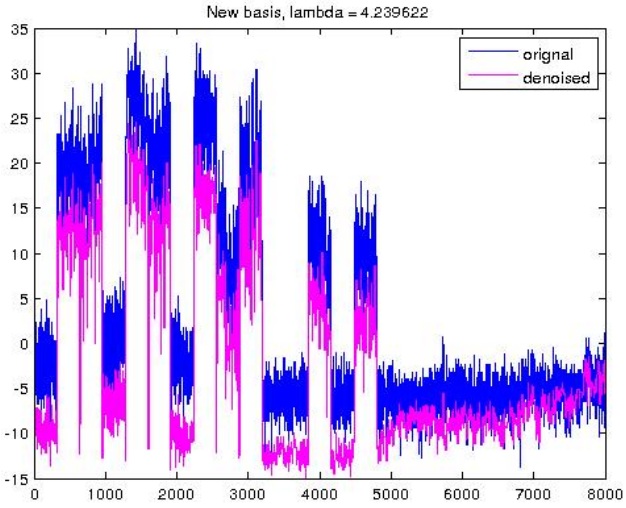

FIGURE 4.3.3

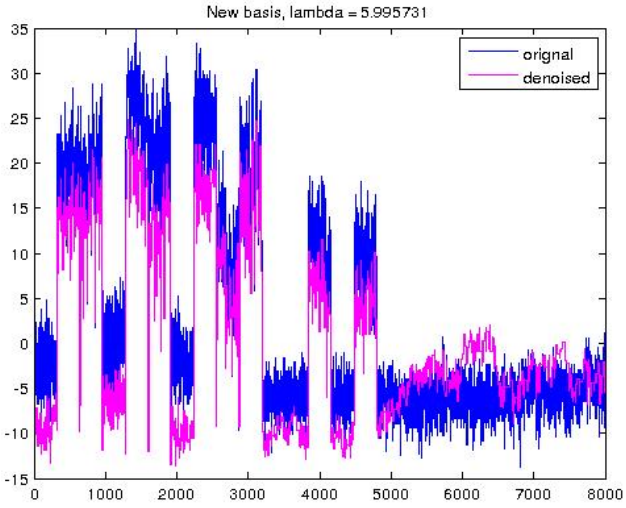

FIGURE 4.3.4

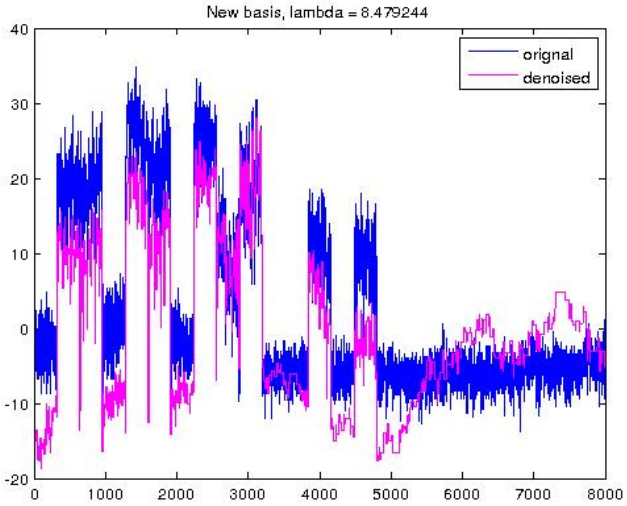

FIGURE 4.3.5

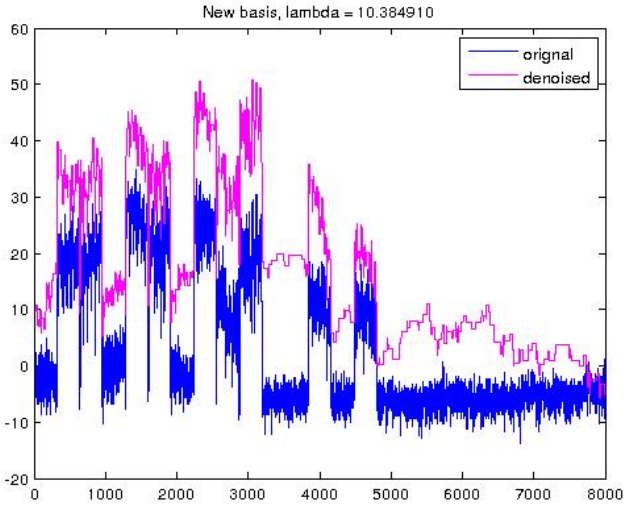

FIGURE 4.3.6

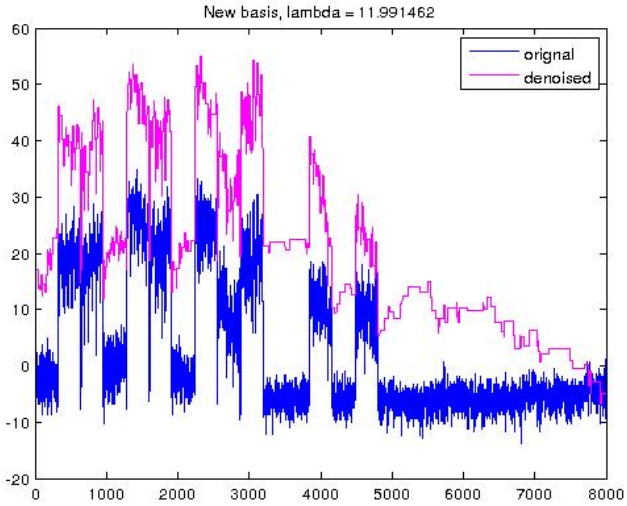
New basis, lambda = 11.991462
orignal
denoised
60
50
40
30
20
10
0
-10
-20
0
1000
2000
3000
4000
5000
6000
7000
8000

FIGURE 4.3.7

Appendix A

Appendix Title Here

Write your Appendix content here.