On fundamental computational barriers in the mathematics of information

Alexander James Bastounis

Churchill College, Cambridge

This dissertation is submitted for the degree of Doctor of Philosophy
June, 2018

Declaration

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the Preface and specified in the text. It is not substantially the same as any that I have submitted, or, is being concurrently submitted for a degree or diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. I further state that no substantial part of my dissertation has already been submitted, or, is being concurrently submitted for any such degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text.

Acknowledgements

I think every PhD student knows before starting that writing a PhD thesis is going to be difficult. What they may not (and certainly I did not) appreciate is the scale of the difficulty that they are likely to struggle against whilst writing a thesis. I believe that a PhD student needs a couple of things to have a chance - firstly, they need supportive friends and family who will be patient when things don't go well and secondly they need an advisor who can help suggest interesting problems that are tractable in the duration of the thesis. I was extremely lucky - I had both.

I had my family - of course, without my parents Janice and Basil Bastounis this thesis would not have been possible (I wouldn't be born) but my parents provided so much more than that. Now that I have sadly left childhood the scale of the sacrifices they have made to provide me with the best opportunities in life is apparent to me. The support, advice and kindness they offered me was invaluable and I am eternally grateful to them. I am also lucky enough to have an intelligent and funny brother Nicholas who never stopped making me laugh and encouraging me even when things looked very tough.

I had my school teachers, particularly those who taught me A-level maths and computer science and encouraged me to go to university and study my passions. I had my school friends, who would always make it a pleasure to visit my home town of Beckenham and who reassured me with their kind words that I had the capacity to finish. Going for a casual drink with them is something that I hope I get to do for a very long time.

I had my friends from Warwick both in and outside of the maths department - even though they are now distributed in a variety of places across the world I was able to stay in touch with many of them and I still value the chances I get to spend time with them. In particular, I should mention (soon to be) Dr Matthew Lee at Imperial College London for going through some of the basics of machine learning with me.

In addition to things that I had before I started, there were new people that I met who have since become lifelong friends. My colleagues from the CCA were never shy to help me when I felt that I was struggling and I definitely enjoyed working with many of them directly in the first year of the CDT program. I met an enormous number of very likeable and interesting people during my time at Churchill College and will leave with many great memories that I will take with me for the rest of my life. I would like to specifically mention Dr Steve Marsh and Dr Clara Tang for helping me get through this particularly strenuous hurdle of writing the thesis, a process that I have found to be especially testing and which I suspect I would not have completed without them.

I would also like to specifically mention Dr Florian Theil who taught me at Warwick University. He encouraged me from day one to go further with mathematics and advised me in making the transition between school mathematics and the beginning of graduate studies. Academia is a small world and I hope that one day our paths shall cross again.

I would like to thank my thesis examiners for their helpful suggestions in improving this thesis. I feel that their contributions have undoubtedly improved the exposition since the initial submission.

Finally, and most importantly, I would like to thank my supervisor Dr Anders Hansen. Anders was excellent at suggesting seemingly small problems that would become far larger results with hard work. His enthusiasm for mathematics was contagious - even when things weren't working I would usually come out of conversations feeling a renewed passion and a determination to attack the problem that I was facing in a new way. He put in an enormous amount of time and effort at almost any hour of the day to discuss whatever results I had been able to produce and to help me place them into a wider context. I can only hope this thesis is something that reflects the investment he put into me - certainly, without him there, I suspect I would have left the program long ago.

Even though I have mentioned only a few people specifically by name, it should be clear that there are an enormous number of people who helped make this possible. They know who they are, and should this thesis pass I will certainly celebrate with each of them.

Abstract

This thesis is about computational theory in the setting of the mathematics of information. The first goal is to demonstrate that many commonly considered problems in optimisation theory cannot be solved with an algorithm if the input data is only known up to an arbitrarily small error (modelling the fact that most real numbers are not expressible to infinite precision with a floating point based computational device). This includes computing the minimisers to basis pursuit, linear programming, lasso and image deblurring as well as finding an optimal neural network given training data. These results are somewhat paradoxical given the success that existing algorithms exhibit when tackling these problems with real world datasets and a substantial portion of this thesis is dedicated to explaining the apparent disparity, particularly in the context of compressed sensing. To do so requires the introduction of a variety of new concepts, including that of a *breakdown epsilon*, which may have broader applicability to computational problems outside of the ones central to this thesis. We conclude with a discussion on future research directions opened up by this work.

Contents

Chapter 1

Introduction

It is impossible to overstate the importance of data processing techniques in the modern world. To give a few key examples, *linear programming* allows us to optimise a desired quantity given certain constraints; *compressed sensing* has changed the effectiveness and speed of various scanning devices; *statistical estimation* allows us to find relevant features of large datasets; *image processing* has improved the quality of imaging devices from smartphone cameras to special effects used in films, and *machine learning* continues to find new uses everyday, ranging through topics as vast as self-driving cars to automated music production. The underlying method used in each of these fields is to solve (using a computational device) a specific mathematical problem dependent on the dataset and the application under consideration. Because of the now ubiquitous nature of these problems and their use in potentially life threatening situations, it is crucial to understand when this approach will see success and conversely, when it will fail.

The bulk of this thesis is dedicated to providing results on this question from a computational standpoint. Namely, we consider the *existence* of algorithms to solve the mathematical problem associated to each of the techniques listed above. Similar questions on the limitations of computation have been considered by a variety of authors. In particular, Hilbert's question [70] on the existence of algorithms for decision problems led to Turing's work [120], widely considered to be a key paper in the development of modern computer science. A different example taken from scientific computing (and thus more closely related to the work in this thesis) is Smale's question on the existence of purely iterative generally convergent algorithms for polynomial root finding. This was answered in the negative by McMullen [90, 91]. In related work that highlights the subtlety of such issues, Doyle and McMullen showed a positive result that the quintic could be solved by purely iterative generally convergent algorithms by using several limits [51].

With the strength of the observed results when the techniques of the opening paragraph are applied to realistic datasets, it would be natural to conjecture that algorithms will exist to solve these problems. Specifically, we ask for an algorithm that computes in finite time the solution to any of these problems up to a prescribed output accuracy. To model the idea that computers cannot store all real numbers

without infinite memory, we assume that the algorithm cannot access the exact input dataset. Instead, it can ask for arbitrarily precise approximations to the dataset, where the precision used is determined at runtime.

Somewhat surprisingly, we show that there is no such algorithm for any of the problems listed above. This begs the following natural question:

Q1: Why are these problems mathematically impossible and yet, in many circumstances, solved on a day to day basis by data scientists?

As an initial answer to this question, one might suggest the following:

A1: The class of inputs for which the algorithm must give an answer with error smaller than the prescribed output accuracy is too large to represent realistic datasets.

This opens up the following question, central to the thesis

Q2: Which criteria on the input datasets allow us to create algorithms that can solve the problem to any given prescribed output accuracy?

It transpires that this question is very subtle. Insisting that inputs are *well conditioned* can have various effects on the computability of the problem. As a brief demonstration of this phenomena, one of the results of this thesis is that there are input sets which are very poorly conditioned yet computable. In some sense, the example we provide is not pathological in the sense that the inputs with high condition number are in fact very natural for the problem under consideration.

Actually, whilst it will be possible to create algorithms for some of the problems above for realistic datasets, for some of the problems above even with restrictions it will remain impossible. We thus provide the following answer:

A2: Even if it is impossible to create algorithms to solve some of the above problems to a given *arbitrary* prescribed accuracy ϵ when sensible restrictions are placed on the inputs, it will be possible to create algorithms that can solve the problems to a *tolerable* output accuracy.

For example, we may not be able to create an algorithm that can solve the problem with thirty decimal places, but it may be possible to solve the problem with an answer that is correct in its first fifteen decimal places. To quantify exactly what is meant by 'tolerable', we introduce the new concept of a *breakdown epsilon*. Essentially, the breakdown epsilon represents the best obtainable precision for a given class of inputs. A small non-zero breakdown epsilon can therefore indicate that although a problem is non-computable in the sense that it cannot be computed to arbitrary accuracy, we can still obtain a tolerable approximation to the true solution.

As a final major point of discussion, let us assume that we pay a time penalty to get more accuracy on the input set. For example, if the true input contains π or $\sqrt{2}$ then computing a better approximation to the true input requires time. We will find that in some circumstances and with some restrictions on the input set, it will be possible

to create an algorithm that can compute the solution to any desired output accuracy. Even so, it may be the case that there is a collection of valid inputs and a fixed output accuracy for which the algorithm will require increasingly accurate measurements to compute the approximate solutions. In this case, we are in a situation where we are guaranteed that the algorithm will eventually terminate but we have no idea when this will occur. In particular, there will be inputs that take longer than fifty billion years to process. Since this is obviously undesirable, we shall also try to find criteria that guarantee that the required input accuracy is bounded uniformly across all valid inputs.

The main theme of this thesis is to examine various criteria that lead to either existence or non-existence results of algorithms that either solve the problems exactly or give answers that have a small breakdown epsilon. There are of course a myriad of different conditions that could be considered when attempting to answer these questions. Ideally, we want to work with realistic criteria that are observed with the types of dataset that are used in the real world. However, finding such criteria is in itself a very large and difficult problem (this is particularly the case for deep learning [46, 52]) and it is out of the scope of this thesis to show results for every possible sensible criteria for each problem. We thus choose compressed sensing as a case study to examine in detail. In this field, we highlight some issues with existing approaches and introduce a collection of new criteria with which we prove positive results from the perspectives of both computation and recovery.

In summary, once we have established the basic non-computability results and generated a sensible input set criteria for compressed sensing algorithms, our task will be to examine the following questions:

Q3: For any $\epsilon > 0$, is there an algorithm that executes in finite time and computes an approximate solution no further than ϵ away from a true solution? The algorithm can choose the input accuracy $\hat{\epsilon}$ to be as small as desired (as a function of ϵ and the input) to produce the output.

Q4: Can we get a positive answer to the previous question if, for a given $\epsilon > 0$, $\hat{\epsilon}$ is uniformly bounded across all possible inputs?

Q5: Suppose that the answer to the two previous questions is negative. Can we replace the requirement that the algorithm has to exist for each $\epsilon > 0$ with one that insists that the algorithm must only work for a 'small' $\epsilon > 0$ and get a positive answer?

1.1 Outline of the thesis

This thesis is split into six chapters. In the remainder of this one, we will precisely define *linear programming*, *compressed sensing* (basis pursuit and basis pursuit denoising), *statistical estimation* (lasso), *neural networks* and *image processing*. In Chapter 2, we will define various models of computation (in particular, the *Turing machine* and *BSS* models) and generalise them to the concept of a *general algorithm*. We shall

add additional generality that allows for random algorithms by defining a *probablistic general algorithm*. Afterwards, we discuss the idea of *breakdown epsilons* that allow us to further understand the degree of non-computability that a mathematical problem may have. We then state a key and novel result (proven in Chapter 5) that will prove crucial throughout the remainder of the thesis. Chapter 2 concludes with a discussion on the various forms of condition number and their definitions when adapted to the problems defined in this introduction.

Chapter 3 introduces some new criteria that can be applied to the compressed sensing problem. Specifically, we shall introduce the new concept of the RIP and nullspace properties in levels and justify them as better explanations for the observed success of the methods of compressed sensing. These concepts will also be used later as criteria for computability.

Chapter 4 contains the main results of the thesis. Firstly, we discuss the negative results in the settings mentioned above. This will open up a (large) number of new questions about the computability of the mathematics of information. We shall provide partial answers to these questions for compressed sensing and statistical estimation. Here, we will make use of the results of Chapter 3 to provide guarantees using the nullspace property in levels. The result of this analysis is that we give an explanation not only for why the method of using basis pursuit works well, but also for why algorithms are able to execute the method despite the non-computability results.

In the context of neural networks Chapter 4 also provides a collection of results to supplement the non-computability result for training a neural network. These results have a somewhat perplexing interpretation. Roughly speaking, there are uncountably many problems for which training the neural network (assuming that there is an oracle which can do so) will lead to a neural network that is in some sense unstable. However there will always exist a stable neural network that can solve the problem. The results of this section also open up questions on the very notion of success for a neural network as well as casting doubts on the technique of cross validation. Significantly more detail and precise formulations of the results for neural networks are given in Section 4.4.

Although the proofs of Chapter 3 are provided within that chapter (and thus the methodological issues for compressed sensing are mostly self contained in Chapter 3), we provide the proofs for the computational results in Chapter 5. This includes all the proofs from Chapter 4 as well as the key result stated in Chapter 2.

Finally, we conclude the thesis in Chapter 6. A brief summary of the entire thesis is provided, along with a discussion on a wide variety of possible future directions and open questions that arise from this thesis.

1.2 On novelty

In this thesis, unless otherwise stated a theorem, lemma, proposition or corollary is original work. In particular, all the non-computability results, the positive results for basis pursuit and lasso, the results on neural networks (with the exception of the universal approximation theorem stated in the introduction) and all the results on the RIP in levels and nullspace property in levels in Chapter 3 are new to this thesis. Note

however that there is some discussion in [24] on how imprecise initial data can affect the output of linear programming.

Additionally, this thesis introduces several new concepts. The various breakdown epsilons defined in Sections 2.6 and 2.8 are important new concepts first defined here. The idea of a randomised general algorithm used to demonstrate non-computability defined in Section 2.7 is also new to this thesis. The RIP and nullspace properties in levels defined in Chapter 3 and used to explain the methodological and algorithmic success of compressed sensing are novel definitions.

In general, everything that comes after (but not including) Section 2.6 as well as the breakdown epsilons defined in Section 2.6 is novel work unless otherwise stated with the exception of the well known definitions of condition numbers restated in 2.11. The introduction is novel in exposition only and the ideas of an SCI hierarchy [13], inexact input (and specifically Δ_1 information) [13], a Turing machine [120] and a BSS machine [17] are well known to the community.

It should however be noted that the majority of Chapter 3 has appeared in my own work published in SIIMS [9]. This work, which was written with some guidance from Anders Hansen, was recently featured on the front cover of *SIAM News* [10]. Furthermore, the results on non-computability and the corresponding definitions are the result of a close collaboration between myself and Anders Hansen with additional insight provided by Verner Vlacic.

1.3 Condition

We begin this section with an example. We solve the following problem using MATLAB R2016b [88] (an example of a linear programming problem, see Section 1.4):

$$\arg\min_{x\in\mathbb{R}^N} c^T x \text{ such that } Ax = y, \quad x \geq 0$$

with

$$A = \begin{pmatrix} 1 & \epsilon \\ 1 & 0 \end{pmatrix}, \quad y = \begin{pmatrix} 1+\epsilon \\ 1. \end{pmatrix}$$

It is easy to see that the correct solution to this problem is the vector $(1,1)^T$. The matlab code for $\epsilon = 10^{-16}$ is as follows:

```
Large = 1e16; Epsilon = 1/Large;
y = [(Epsilon+1) 1].';
A = [1 Epsilon; 1 0];
c = [1 1];
options = optimoptions('linprog','Algorithm','dual-simplex');
options.MaxIterations = 1e100;
options.OptimalityTolerance = 1e-10;
AlglinprogAns = linprog(c,[],[],A,y,[0 0],[],options)
```

With those commands, MATLAB prints the following output messages:

```
Optimal solution found.


AlglinprogAns =

1.0000
0
```

In mathematical notation, MATLAB produces the vector $(1,0)^T$ as a solution. Importantly, this vector is not optimal - in fact, it isn't even feasible. It is easy to see that the true solution to this linear programming problem is instead $\hat{x}$, where $\hat{x} = (1,1)^T$. Actually, $\hat{x}$ is the unique solution to $Ax = y$. If we enter the matlab command $A\backslash y$ (which solves for x the *linear systems problem* $Ax = y$), we get the following output.

```
Warning: Matrix is close to singular or badly scaled.
Results may be inaccurate. RCOND = 1.000000e-16.

ans =

1
0
```

Once again, MATLAB is unable to solve the problem and returns the incorrect vector $(1,0)^T$. However, here we observe a subtle but important difference: there is a warning printed by MATLAB about *condition*. Essentially, MATLAB has informed us that it is unable to solve the problem because the input set is somehow badly suited to the algorithm being used. In the linear programming case no such warning is printed. It transpires that the condition of linear programming is a far more complicated problem than that of linear systems. The importance of condition is highlighted by the recent book by Bürgisser and Cucker [24], who argue that condition is very important for understanding when algorithms will work and when they will fail. We shall adopt this philosophy in this thesis, giving a detailed analysis on how a variety of different notions of condition affect the computability of a problem.

Remark 1.3.1. Note that although the algorithm chosen for linear programming is the 'dual-simplex' algorithm, it is possible to use the same set of commands (with potentially different values of the variable 'Large') to get a similar result for the other algorithms supported by MATLAB R2016b.

Remark 1.3.2. The tolerance parameter 10^{-10} is the minimal permitted tolerance for the dual-simplex algorithm.

1.4 Linear programming

In many scenarios, we might seek to maximise/minimise some parameter (e.g. profit for a goods manufacturing facility) but we might be constrained by some linear factors

(e.g. stock available to use in constructing goods). A highly celebrated technique for solving such issues is *linear programming*. This is the collection of problems that can be represented in the following way

$$\arg\max_{x\in\mathbb{R}^N} c^T x \text{ subject to } Ax = y, \quad x \geq 0$$

with $c \in \mathbb{R}^N$, $A \in \mathbb{R}^{m\times N}$ and $y \in \mathbb{R}^m$ and the inequality $x \geq 0$ is evaluated element-wise. For example, if the goods manufactured are chemicals, x might be a vector that represents the (non-negative) amount of each chemical that will be manufactured, c might represent the profit that is obtained by selling each chemical and the restriction that $Ax = y$ is set so that there is a limited amount of each element with which to manufacture new chemicals. It is important to note the following:

- The equality constraint can be replaced by an inequality constraint i.e. $Ax \leq y$ where the relation $\leq$ is defined element-wise.
- Sometimes a different problem is considered, also termed 'linear programming'. This is the following:

  $$\max_{x\in\mathbb{R}^N} c^T x \text{ subject to } Ax = y, \quad x \geq 0$$

  Here, we are not interested in the value of x that maximises the relation but the value of the objective function $c^T x$ at such a maximum (this is exactly the difference between argmax and max). Although there may be circumstances where this is useful, it is often crucial to know x itself (for example, in the above example of a chemical production plant it is far more important to know how to obtain the best possible profit than simply knowing what the maximal profit is).
- It is not guaranteed that such a problem has a solution. Since this is uninteresting from the perspective of knowing whether or not we can solve linear programming on a computer, we focus mainly on the case where the problems are indeed feasible.
- At this point there may be some confusion: there is a well known algorithm for computing the minimisers of a linear program, namely the *simplex* algorithm [43]. However, the simplex algorithm works on exact information. This may not be possible if it is impossible to obtain exact information on c, A or b or if any of these parameters are somehow noisy. Our work will focus on this case.
- Sometimes the problem is done over something other than the real numbers like the integers or rational numbers. This however brings about its own difficulties - if the problem is done over the integers then we are left with *integer programming* which is a famous NP-complete problem [75]. If the problem is done over the rational numbers then using floating point arithmetic can lead to similar imprecision issues.

As a passing remark let us briefly consider the additional problem of semidefinite programming, which has received attention from a number of mathematicians

[55, 56, 122]. We note that as linear programming is a special case of semidefinite programming, any non-computability result on linear programming immediately implies a corresponding non-computability result for semidefinite programming.

1.5 Compressed sensing

We shall begin by discussing the general ideas of compressed sensing [18, 29, 32, 41, 47, 50, 53, 58, 60, 60, 69, 105] as it is used in linear inverse problems. Consider the problem of recovering information $x \in \mathbb{C}^n$ from a scanning device, represented by an invertible matrix $M \in \mathbb{C}^{n\times n}$, given observed measurements $b := Mx$. In general, we require knowledge of every element of b to be able to accurately recover x without additional structure. Indeed, let $\Omega = \{\alpha_1, \alpha_2, \ldots, \alpha_m\}$ with $1 \leq \alpha_1 < \alpha_2 < \alpha_3 < \cdots < \alpha_m \leq n$ and define the projection map $P_\Omega : \mathbb{C}^n \to \mathbb{C}^m$ so that $P_\Omega(x_1, x_2, \ldots, x_n) := (x_{\alpha_1}, x_{\alpha_2}, \ldots, x_{\alpha_m})$. If m is strictly less than n then for a given y there are at least two distinct vectors $x_1 \in \mathbb{C}^n$ and $x_2 \in \mathbb{C}^n$ with $P_\Omega b = P_\Omega M x_1 = P_\Omega M x_2$, so that knowledge of $P_\Omega b$ will not allow us to distinguish between multiple candidates for x. Ideally though we would like to be able to take $m \ll n$ to reduce either the computational or financial costs associated with using the scanning device M.

So far, we have not assumed any additional structure on x. However, let us consider the case where the vector x consists mostly of zeros. More precisely, we make the following definition:

Definition 1.5.1 (Sparsity)**.** *A vector $x \in \mathbb{C}^n$ is said to be* s-sparse *for some natural number s if $|\mathrm{supp}(x)| \leq s$, where* $\mathrm{supp}(x)$ *denotes the support of x.*

If x is sufficiently sparse, it is natural to expect that looking for the solution $\hat{x}$ to $P_\Omega M\hat{x} = P_\Omega b$ with the least number of zeros would be an effective way of recovering x. However, this problem is non-convex. Instead, the key to CS is the fact that, under certain conditions, any minimiser of either the *Basis Pursuit (BP)* problem (sometimes known as *Basis Pursuit with ℓ^1 regularisation*)

$$z \in \arg\min \|\widehat{x}\|_1 \text{ such that } y = U\widehat{x} \tag{BP}$$

or the *Basis Pursuit Denoising (BPDN)* problem (defined for fixed $\delta > 0$ and again, sometimes known as *Basis Pursuit Denoising with ℓ^1 regularisation*)

$$z \in \arg\min \|\widehat{x}\|_1 \text{ such that } \|U\widehat{x} - (y + \eta)\|_2 \leq \delta. \tag{BPDN}$$

(where $U := P_\Omega M$ is a matrix in $\mathbb{C}^{m\times n}$, $\eta \in \mathbb{C}^m$ is a noise term with $\|\eta\|_2 \leq \delta$ and $y = P_\Omega b = Ux$) gives a good approximation to x, provided that x has some form of sparsity.

There are two natural questions that arise from the statement of these problems

- Under which circumstances on x, U and δ does the solution to either (BP) or (BPDN) give a good approximation to x?
- Are (BP) or (BPDN) computable?

Establishing criteria for the first of these questions is non-trivial. For example, if U is a subsampled identity matrix then there will always be x which are sparse but cannot be recovered. We shall further discuss this question in Chapter 3.

The second question is one of the primary focuses of this thesis. We will see that if we permit arbitrary y and U, neither (BP) nor (BPDN) can be computed. This will be discussed in detail in section 4.2.

To give a reference for the types of U that are typically chosen, sometimes we insist that $M = AW^{-1}$ for some measurement operator A and some sparsifying operator W. In particular, frequent choices include the case where A is a fourier matrix and W is a a wavelet transform.

A different collection of commonly studied choices for U include the *Hadamard* and *Bernoulli* random matrices [60]. Since these matrices will be important later, we shall give a precise definition:

Definition 1.5.2 (Hadamard random matrices)**.** *The* Hadamard *random matrices of size $m \times 2^k$ is the matrix $P^m H_k$ where H_k is defined by the recurrence relation*

$$H_j = H_1 \otimes H_{j-1}, \quad H_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad H_0 = 1, \quad j \in \mathbb{N}$$

and P^m is a random selection of m rows followed by scaling by a factor of $1/\sqrt{m}$.

Definition 1.5.3 (Bernoulli random matrices)**.** *The* Bernoulli *random matrices B of size $m \times N$ is a matrix with independent identically distributed entries such that $\mathbb{P}(B_{i,j} = 1) = \mathbb{P}(B_{i,j} = -1) = 1/2$.*

Let us conclude this section with a simple example taken from [10]. We considered an original image of a pomegranate which used 2048^2 pixels. The aim is to capture this image using a simulated MRI scanner. First, we considered the approach of simply reading the central 256^2 fourier frequencies and then performing a direct matrix inversion on that data. This is the *linear recovery* method. Next, we compared to an approach using compressed sensing - we took 256^2 measurements in a level based structure (see Chapter 3 and the first sampling pattern in 3.2 for a similar example). We then attempted to compute the minimiser of the basis pursuit procedure (BP) with $U = P_\Omega M$. Ω was defined to be the sampling pattern and the matrix M was given by a composition of the 2D fourier transform and an inverse curvelet transform. The results are presented in Figure 1.1.

(a) (b) (c)

Figure 1.1: An example using compressed sensing. (a) Original image with small synthetic detail added, shown in the red region. (b) Zoomed linear recovery from the central 256^2 (around the 0 Fourier frequency) Fourier measurements. (c) Zoomed compressed sensing with a multilevel random sampling pattern with subsampling using $m = 256^2$ measurements.

It is clear that with the same number of samples, compressed sensing has shown a vast improvement in image quality. This may be very important if instead of a pomegranate, the object being scanned is a human organ. In that scenario, small artefacts may cause a misdiagnosis of a patient's medical issues.

1.6 Statistical estimation

Statistical estimation is the task of finding out which features (and how strongly) can be used to predict observations. Specifically, we are given m observations of a dependent variable y and, for each y, corresponding values of N features $A_{i,j}$ for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, N$. We wish to establish which of the N features are important and to quantify this to make future predictions based on knowledge of the intensity of each feature. The simplest initial idea for doing this is to solve the equation $Ax = y$ for x. However, there is no reason to suspect that A is invertible. To expand on this simple initial idea, one could try to solve a least squares problem, namely find $\operatorname{argmin}_{x\in\mathbb{R}^N} \|Ax - y\|_2^2$. Unfortunately, this may lead to infinitely many solutions. In particular, this can occur if we only have a small number of observations relative to the number of features, i.e. $m < N$.

If however we know a-priori that few features are likely to be relevant (even if we do not know which ones are the relevant ones), we can instead try to solve some regularised version of least squares. *Constrained LASSO* [66, 115] does exactly this: for a fixed $t > 0$, we solve $\operatorname{argmin}_{x\in\mathbb{R}^N} \|Ax - y\|_2^2$ such that $\|x\|_1 \leq \tau$. By forcing the ℓ^1 norm to be small, we can avoid overfitting noisy data (which may happen if we instead solve the least squares problem) and penalise solutions that have many non-zero features (whilst still maintaining the desirable convex property of the problem).

A related problem is that of *Unconstrained LASSO*. This is in some sense the dual of constrained lasso and can be written as $\operatorname{argmin}_{x\in\mathbb{R}^N} \|Ax - y\|_2^2 + \lambda\|x\|_1$ where $\lambda \geq 0$

is fixed. In practice, both problems are studied. A consequence of convex analysis is that if for some t, the vector x solves constrained lasso with the observations y and predictors A then there is a λ (depending on x) so that x (with the same observations and predictors) also solves unconstrained lasso and vice-versa. The two problems are therefore very similar.

Despite the apparent similarity, the problems are not identical in the sense that if S is the set of all solutions to a constrained lasso problem then it may not be the case that there is a lambda for which S is exactly the solution set to unconstrained lasso. Indeed, consider the following constrained lasso problem:

$$A = \begin{pmatrix} 1 & 1/2 & 1/3 \end{pmatrix}, \quad y = 1, \quad \tau = 2.$$

It is clear that the vectors $x^1 = (1,0,0)^T, x^2 = (0,2,0)^T$ are both solutions (since both have ℓ^1 norm bounded by 2 and $Ax^1, Ax^2 = y$) but the vector $x^3 = (0,0,3)^T$ is not a solution. For unconstrained lasso however, if $\lambda > 0$ and x^1 is a solution then $\|Ax^2 - y\|^2 + \lambda\|x^2\|_1 > \|Ax^1 - y\|_2^2 + \lambda\|x^1\|_1$ and so x^2 is not a solution. If instead $\lambda = 0$ then x^1, x^2 and x^3 are all solutions. Thus there is no lambda for which x^1, x^2 are solutions and x^3 isn't. Because of this observation we shall find that the two problems have different computational properties. We shall elaborate on this in Section 4.3.

1.7 Neural networks

Suppose that we are given a collection of images and we wish to determine automatically whether or not a given image contains a dog. Such a problem is a specific example of a more general classification problem. We assume that there is an intrinsic but not known function $f : M \to \{0,1\}$ that classifies our inputs which exist in a space $M \subseteq \mathbb{R}^d$ (in this case we might take $M = [0,1]^d$ to be the values of the d pixels of the image and $0, 1$ represent 'not dog' and 'dog' respectively). Our task is to find from a fixed finite collection of samples $S \subseteq M$ and knowledge of $f(x)$ for each $x \in S$ an approximation to f on the entirety of M. Of course, we should not expect any approach to work for all possible f and M, but we can still see reasonable success at this problem by using a *neural network*.

1.7.1 Perceptrons

Before explaining neural networks, let us consider a more basic concept: a perceptron [92]. The basic idea is that we can make choices based on weighing up different inputs. The perceptron takes input x, applies weights to it, and then applies a simple *activation function* to determine whether $f(x) = 0$ or $f(x) = 1$. Mathematically, a perceptron works in the following way: there are $w \in \mathbb{R}^{1\times d}$, $b \in \mathbb{R}$ and $g : \mathbb{R} \to \{0,1\}$ such that for a given x, the output of the perceptron is $g(w^T x + b)$. The activation function here is taken to be $g(x) = 1$ if $x > 0$ and $g(x) = 0$ if $x \leq 0$. w represents the weights of the network and b is a bias that either pushes the perceptron towards or away from 0 or 1. This is demonstrated in Figure 1.2.

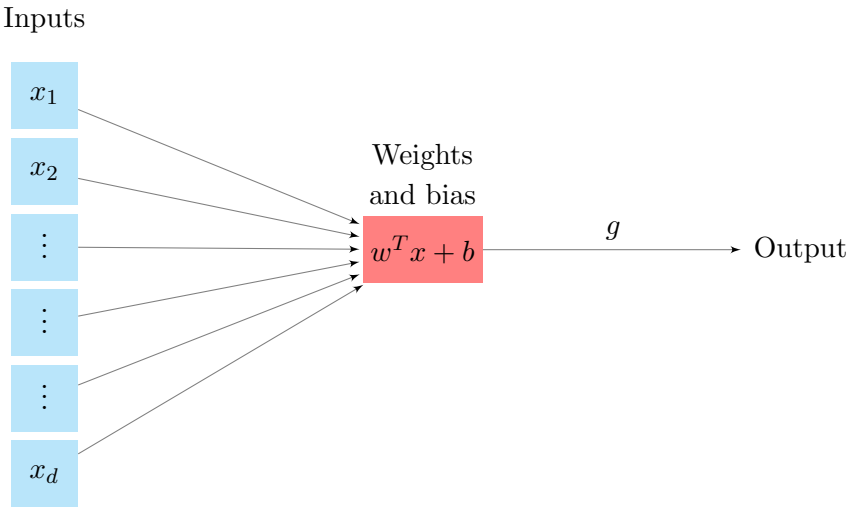


Figure 1.2: A simple perceptron model.

We then 'train' the perceptron by attempting to find suitable w and b so that $g(x) \approx f(x)$ whenever $x \in S$. This model will see success with some patterns f. However, let us consider the exclusive or pattern i.e. $M = \{(0,0),(1,0),(0,1),(1,1)\}$ and

$$f(a,b) = \begin{cases} 1 \text{ if exactly one of } a, b = 1 \\ 0. \text{ otherwise} \end{cases}$$

It is easy to see that no perceptron can exactly capture f, even though f is in some sense relatively simple (such a result was demonstrated in [92]). Indeed to capture f exactly, we require $g(w_1+b) = 1$, $g(w_2+b) = 1$ and $g(b) = 0$. Thus $w_1+w_2+b \geq w_1+w_2+2b \geq (w_1+b)+(w_2+b) > 0$. But this immediately implies that $g(w_1+w_2+b) = 1$ and thus $g \neq f$. We therefore take a different approach.

1.7.2 Feed forward neural networks

To move from perceptrons to neural networks (see [82] and references within), we make two changes. Firstly, we allow for a different class of activation functions than the simple function g that we described earlier. Instead, we deal with a fixed $\rho : \mathbb{R} \to \mathbb{R}$. This has computational consequences: the g defined above is clearly discontinuous, whereas here we can work with ρ that are continuous. We call this now modified perceptron a *neuron*. Our second change will be to use multiple neurons. Indeed, we shall arrange the neurons in *layers*. The input information will be termed the *input layer* and the output neurons the *output layer*. Any other layers will be termed *hidden layers*.

To demonstrate this, let us consider a one hidden layer network. In the hidden layer, we set up a collection of N_1 neurons which each with their own weights and biases. For reasons that will become clear later, we label the weights from the ith neuron as $W^1_{i,1}, W^1_{i,2}, \ldots W^1_{i,d}$ and the bias as b^1_i. For a given input $x \in \mathbb{R}^d$, each neuron computes $\sum_{j=1}^{d} W^1_{i,j} x + b^1_i$ and then applies the function ρ to it. We then have an

output neuron, which takes the results of the N_1 neurons and applies it's own weights $W^2_{1,1}, W^2_{1,2}, \ldots, W^2_{1,N_1}$ and a bias b^2. This is demonstrated in the top part of Figure 1.3 with $N_1 = 2$

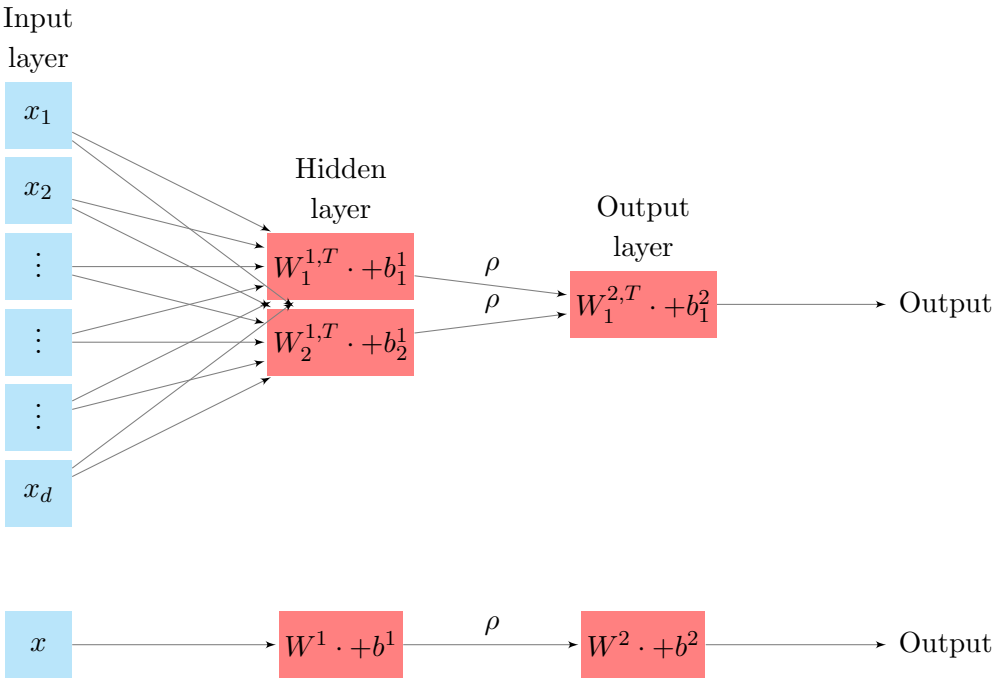


Figure 1.3: Top half: A diagram of a neural network with a two neuron single hidden layer. Bottom half: A simplified version using matrix arithmetic.

It is easy to see that for inputs x, with this model, we are computing $A^2 \rho A^1(x)$ where A^1, A^2 are the operators $A^1 = W^1 + b^1, A^2 = W^2 + b^2$ with the matrices W^1 and W^2 are formed by the entries $W^1_{i,j}, W^2_{1,j}$ respectively. The function ρ is applied componentwise. This is summarised in the bottom half of Figure 1.3. As a natural generalisation, we can introduce multiple hidden layers and even attempt to work with f that map to a multidimensional output space. With L layers including the output layer, such a network would be of the form

$$A^L \rho A^{L-1} \rho \cdots A^2 \rho A^1, \quad A^l = W^l + b^l \text{ with } W^l \in \mathbb{R}^{N_l \times N_{l-1}}, b^l \in \mathbb{R}^{N_l}$$

for fixed parameters $\mathbf{N} = (N_0, N_1, N_2, \ldots, N_L)$ with $N_0 = d$. We denote the class of all such neural networks by $\mathcal{NN}_{\mathbf{N},L}$.

A natural question is that of *expressibility*. Namely, which f can be approximated by such neural networks? It transpires that provided ρ is not a polynomial, every continuous $f : \mathbb{R}^d \to \mathbb{R}$ can be approximated arbitrarily by neural networks. Precisely, we have the following variant of the *universal approximation theorem* [95].

Theorem 1.7.1. *Let $\mathcal{NN}_{\infty,2}$ denote $\cup_{N_1=1}^{\infty} \mathcal{NN}_{\mathbf{N},2}$ with $\mathbf{N} = (d, N_1, 1)$. Then $\mathcal{NN}_{\infty,2}$ is dense (in the topology of uniform convergence on compact sets) in the space of continuous functions on $\mathbb{R}^d$ provided that ρ is a continuous function that is non-polynomial.*

The ρ that we shall examine in this thesis is the *rectified linear unit*. Specifically, we set $\rho(x) = 0$ if $x < 0$ and $\rho(x) = x$ if $x \geq 0$ [72, 93].

1.7.3 Training a neural network

So far, we have been able to define neural networks. However, it is not obvious how we should go from a set of examples to finding the correct neural network to approximate f. We therefore need to discuss *training the neural network*. At the very least, we should hope to match f closely on S. Thus we introduce a cost function $C \in \mathcal{CF}$

$$\mathcal{CF} = \{C : \mathbb{R}^{|S|} \times \mathbb{R}^{|S|} \to \mathbb{R}_+ \,|\, C(v, w) = 0 \text{ iff } v = w\} \tag{1.7.1}$$

which penalises incorrectly classified training examples. We thus attempt to solve

$$\min_{\varphi \in \mathcal{NN}_{\mathbf{N},L}} C\left[(\phi(x_1), \phi(x_2), \ldots, \phi(x_{|S|})), (f(x_1), f(x_2), f(x_3), \ldots, f(x_{|S|})\right].$$

Typically, C can be written as a sum of identical cost functions acting from $\mathbb{R} \times \mathbb{R}$ to $\mathbb{R}^+$. We write

$$C\left[(\phi(x_1), \phi(x_2), \ldots, \phi(x_{|S|})), (f(x_1), f(x_2), f(x_3), \ldots, f(x_{|S|})\right] = \sum_{x \in S} C'(\phi(x), f(x))$$

Example choices of C' include $C'(a, b) = \|a - b\|_p^p$ (most commonly when $p = 2$, termed the *quadratic cost function*) or the *cross entropy cost function*, $C'(a, b) = b \log(a) + (1 - b) \log(1 - a)$ for $a \notin \{0, 1\}$ and $C'(a, b) = 0$ otherwise.

This minimisation problem can be expressed in terms of the weights and bias. If ρ is differentiable, we can attack this minimisation problem using a form of gradient descent, assuming that we can find the derivative with respect to changes in weights and bias. This is done using *backpropagation* [107], which we shall not discuss here. Of course, the method may get stuck in a local minimum and we are trying to apply it to a function ρ that has one point of non-differentiability, but these issues are ignored in practice.

Therefore one attempts to find weights w^i such that

$$w^i = w^{i-1} - \gamma \sum_{x \in S} \nabla_w C'(\phi(x), f(x))$$

for some step size (known within the machine learning community as the *learning rate*) $\gamma > 0$ and where ∇_w represents the gradient with respect to the weights and biases. However, computing $\nabla_w C'(\phi(x), f(x))$ is generally prohibitively expensive. Thus an algorithm termed *stochastic gradient descent* [19, 113] is used in place of gradient descent. Instead of each update w^i requiring $|S|$ gradient computations, we calculate $\nabla_w C'(\phi(x), f(x))$ for a randomly chosen $x \in S$ (note that this randomness can be with or without replacement, but this is not important for the purposes of this thesis). We then immediately update w^i by $w^i = w^{i-1} - \gamma \nabla_w C'(\phi(x), f(x))$. This yields a random method; applying this method twice to the same set S and f, there is

no guarantee that the weights will converge to the same value. Thus, we must enhance our discussion of the limitation of algorithms to include *randomised algorithms*. This is done in section 2.7.

1.8 Image processing

Image processing is a vast field. Topics are as varied as denoising images taken with noise artefacts [106] to automatically 'filling in' a portion of the image using image inpainting [15, 38]. Since there are so many topics to choose from, we focus on *inverse problems* in a similar way to Section 1.5 and *image deblurring*. Both use the *total variation norm* (TV), which measures the amount of variation between neighbouring pixels of an image. Mathematically, an image is a matrix in $\mathbb{R}^{m_1 \times m_2}$, with each entry of the matrix representing a pixel. We can now define the *gradient* of an image.

Definition 1.8.1. *For $x \in \mathbb{R}^{m_1 \times m_2}$, the* gradient of x, *denoted by $\nabla(x)$,*

$$[\nabla(x)]_{i,j} = \begin{cases} (|x_{i+1,j} - x_{i,j}|, |x_{i,j+1} - x_{i,j}|) & \text{if } 0 \leq i \leq m_1 - 1 \text{ and } 0 \leq j \leq m_2 - 1 \\ (|x_{i+1,j} - x_{i,j}|, 0) & \text{if } 0 \leq i \leq m_1 - 1 \text{ and } j = m_2 \\ (0, |x_{i,j+1} - x_{i,j}|) & \text{if } i = m_1 \text{ and } 0 \leq j \leq m_2 - 1 \\ (0, 0) & \text{otherwise.} \end{cases}$$

In some sense, 'natural images' have a small gradient. The general idea behind both image deblurring and inverse problems in this setting will be to try to find images that have a small gradient. This is captured by defining the total variation norm, which can be done in either an anisotropic way or an isotropic (invariant to rotations of the gradient operator) way as follows.

Definition 1.8.2. *[37, 54, 84] Let $x \in \mathbb{R}^{m_1 \times m_2}$. The* anisotropic TV norm of x, *denoted by $\|x\|_{TV\ aniso}$, is given by*

$$\|x\|_{TV\ aniso} := \sum_{i=1}^{m_1} \sum_{j=1}^{m_2} \|[\nabla(x)]_{i,j}\|_1 .$$

The isotropic TV norm of x, *denoted by $\|x\|_{TV\ iso}$*

$$\|x\|_{TV\ iso} := \sum_{i=1}^{m_1} \sum_{j=1}^{m_2} \|[\nabla(x)]_{i,j}\|_2 .$$

Since our results apply to either of these norms, we shall simply avoid the distinction and refer to the *TV* norm $\|\cdot\|_{\mathrm{TV}}$. We will now discuss inverse problems and image deblurring in this context.

1.8.1 Total variation inverse problems

The idea that 'natural' images must somehow have a small gradient allows us to solve inverse problems in a different way to the one proposed in section 1.5. There, we were

presented with a matrix $U \in \mathbb{R}^{m\times N}$ (with $m < N$) and data $y = Ux$ for some x, and our task was to find x from the many potential candidates. To do this, we selected based on a small ℓ^1 norm after transforming to a domain in which we expect x to be sparse. If we know however that x represents an image, an alternative strategy would be ask for a small total variation norm. Specifically, we can adapt the basis pursuit problem BP to a *basis pursuit with TV regularisation* problem [96], defined as follows:

$$z \in \arg\min \|\widehat{x}\|_{\mathrm{TV}} \text{ such that } y = U\widehat{x}. \tag{BPTV}$$

Although we could ask similar questions related to understanding when solving (BPTV) exactly will yield a sensible solution to x, this is outside of the scope of this thesis. Instead we shall focus on the computational aspect, namely, can we compute (BPTV)?

1.8.2 Image deblurring

The process of image deblurring [36] is similar to working with TV inverse problems. Let us suppose that we are given an image that has been blurred by some non invertible operator $U \in \mathbb{R}^{m\times N}$ and some noise term $\eta \in \mathbb{R}^m$. Thus we are told the value of $y = Ux + \eta$. We want to deblur the image, subject to knowing how the image was blurred (there is a concept of *blind deblurring* [80] where the operator U is not known a-priori, but we shall not discuss this further here). To do so we can look for an image which, when blurred, will be roughly the same as y. Since many such images may exist, we can also ask for an image which does not feature a large total variation. Mathematically, we can adapt the lasso defined in section 1.6 to try to find z such that

$$z \in \arg\min \|\widehat{x}\|_{\mathrm{TV}} + \lambda\|y - U\widehat{x}\|^2 \tag{DeblurTV}$$

Again, we focus only on the computational properties of such a problem.

Chapter 2

Computational frameworks and the SCI hierarchy

In this chapter we discuss and generalise various models of computation and formulate the problems introduced in Chapter 1 as computational problems.

2.1 A new model for computation

There are a variety of existing approaches used to understand computability theory. Two important examples are *turing machines* and *BSS machines*. Although it would be possible to prove many of the results in this thesis with these models, we shall not do so. Instead we work with a newly introduced model known as *a general algorithm*. There are a number of reasons for doing this:

- **Generality** - BSS and turing machines form examples of general algorithms. Thus proving a result that says something cannot be computed with a general algorithm will immediately imply that it cannot be computed with either BSS machines or turing machines. By introducing this more general model for computation we are able to abstract away from specific details and thus prove results that are independent of the precise model of computation under consideration.

- **Focus** - A general algorithm does not describe the operations that a machine is permitted to use to perform computations. Instead, the focus is on the *information* that the algorithm is allowed to access. The results that are proven in this thesis all rely on the limited information that the algorithms are allowed to access to solve their problems. We therefore believe that a proof of a weaker negative result that suggests that many of these problems are non-computable with a turing machine/a BSS machine would also need to implicitly work with a general algorithm.

- **Power** - We illustrate this idea with an example. Consider the problem of training a neural network. Since the optimisation problem is non-convex, it is entirely possible for an algorithm to get 'stuck' in a local minimum. As a general algorithm does not limit the computational operations available, a general

algorithm could make use of an oracle that guarantees that the procedure avoids local minima. Therefore a result that proves that training a neural network with a given cost function is non-computable is far stronger if the result is proven with a general algorithm than with, say, a turing machine.

For these reasons, we structure this chapter as follows: firstly, we introduce the well known turing and BSS models of computation. Secondly, we describe general algorithms. Next, we discuss some famous non-computable problems. Up until this point all the material presented is not novel to this thesis. The novel material begins in Section 2.6 when we introduce the idea of an algorithm having access to only limited precision approximations of the true inputs. We also precisely define the *breakdown epsilons* to model the limited output accuracy that may be attainable. After that, we further generalise the work to *randomised algorithms* and then prove a key novel proposition on how inexact information can affect computability. With this in hand, we formulate the problems discussed in Chapter 1 in our new framework so that we are ready to prove results about them. Finally, Section 2.11 is a discussion of (well established) concepts used to understand how 'bad' a problem is from a computational perspective, known as *condition numbers*.

2.2 Turing machines

One of the most well studied and discussed models of computation is that of a *turing machine*. Informally, a turing machine consists of an (infinitely long) *tape* which are used to write and read information (one character at a time, specified by the current location of the *head*), a list of *states* which the machine can be in at any moment and a finite table of *transitions* which dictate how the machine writes information and also how it transitions between various states. Included in the list of possible states is a start state, an accept state and a reject state. For a given input ι, the turing machine is said to accept ι if it reaches the accept state with finitely many transitions and reject ι if either it reaches the reject state or it fails to terminate after finitely many operations.

There are many (equivalent) ways of defining a turing machine formally. We follow [87].

Definition 2.2.1. *A turing machine is a* 7*-tuple* $(\Sigma, \alpha, Q, q_0, q_{\text{accept}}, q_{\text{reject}}, \delta)$ *where*

- Σ *is some finite set called the* input alphabet*. The tape is initialised with some finite combination of elements of* Σ
- α *is a finite set called the* tape alphabet*. This includes a blank character. Moreover,* $\Sigma \subseteq \alpha$*.*
- Q *is a finite list of* states *that the machine can occupy at any given stage.*
- $q_0 \in Q$ *is the initial state that the machine occupies at the start.*
- $q_{\text{accept}} \in Q$ *is the accept state that allows the machine to accept a given input.*

- $q_{\text{reject}} \in Q$ *is the reject state that allows the machine to reject a given input.*
- $\delta : \alpha \times Q \to \alpha \times Q \times \{L, R, N\}$ *is the transition function.*

The transition function takes input from the tape and a state and tells the machine what the new state is, a replacement character (which can be the same) to insert at the current location on the tape and a movement instruction to move the head left, right or not move at all.

Typically, turing machines are defined so that the tape has finitely many non-blank characters each in Σ when initialised. This allows us to do computations with rational inputs, so that we can (for example) add or subtract two rational numbers using a turing machine. However, in scientific computing, it is important to be able to discuss the computability of problems with real numbers. As a simple example, let us consider the problem of adding two real numbers. There are two obvious barriers to using a turing machine in this way - firstly, since the tape alphabet has to be finite it will be impossible (in general) to add two real numbers using a turing machine. Indeed, to represent every real number we would need either an infinite tape alphabet or an infinitely long tape. Accessing each entry of this tape will prevent the algorithm from halting and thus we cannot add two real numbers together. Secondly, since the tape of the turing machine has to be initialised with only finitely many non-blank characters, it will be impossible to input our two real numbers to the machine. To address these concerns we turn to the BSS (Blum-Smale-Shub) [17] model of computation.

2.3 BSS machines

As in the previous section, BSS machines are designed to be more amenable to problems in numerical analysis. In particular, BSS machines can be set up so that they allow computations on real numbers. The definition begins with a ring (or a field) R. Informally, the BSS machine takes values from an *input space* and moves through finitely many states (in a *state space*) and instruction nodes to a value in an *output space*. Each instruction node in the state space either tells the machine to perform some operation on the state or to branch based on the current values of the state. More precisely, we have the following definition

Definition 2.3.1. *For $l, m, n \in \mathbb{N}$, a BSS machine M over R is a four tuple consisting of*

1. *An* input space $\bar{I} \subseteq R^l$.
2. *An* output space $\bar{O} \subseteq R^n$.
3. *A* state space $\bar{S} \subseteq R^m$.
4. *A finite directed graph of* instruction nodes. *Each instruction node is one of the following:*

(a) *An* input node *(of which there is only one). The input node has exactly one output edge to another node. The input node also contains a linear and injective map that takes the input to the state space.*

(b) *An* output node *which contains a linear map from the state space to the output space. Such a node must have no outgoing edges.*

(c) *A* computational node *which contains a polynomial map from the state space to the state space. Each computational node has exactly one outgoing edge.*

(d) *A* branch node *which has two outgoing edges labelled β^+ and β^-. Attached to the branch node is a polynomial from the state space to R.*

The machine works as follows: at the beginning, the machine starts with input ι at the input node. It then applies the linear map to ι to get a value in the state space (that it stores as the current state) and moves along its only edge to the next node. The machine now acts depending on what the next node is.

- If the next node is an output node, then the machine applies the linear map attached to this output node to the value of the current state and outputs this new value.
- If the next node is a computational node, the machine applies the polynomial map to the current state and stores this value as the new current state. It then moves to the next node on its outgoing edge.
- If the next node is a branch node, the machine will apply the attached polynomial to the current state. If this polynomial is less than or equal to 0, it goes along the edge labelled β^-. If the polynomial is greater than 0 then it will go along the edge labelled β^+.

The machine then repeats this process until it reaches a suitable output node.

Remark 2.3.2. It is also possible to define a BSS machine for $l, m, n = \infty$, but since this adds complexity and we shall not use this in the work that follows, we do not do this here and only note that our results will hold even for such BSS machines.

Remark 2.3.3. For our purposes, we consider R to be a field. Here, we replace the word 'polynomial' in our definition with 'rational function'. In this case it is possible that the denominator of such a function is 0. To deal with this issue, we introduce the *halting set* of inputs which is the set of inputs for which the machine halts and never divides by 0. We then consider the machine to only be operating on such inputs.

2.4 General algorithms and the SCI hierarchy

As we have seen, there are many different models of computation each with their own limitations and strengths. If we want to prove general results about computability theory, it would be useful to abstract away from the model. We thus make a definition for an *computational problem.*

Definition 2.4.1 (Computational problem)**.** *A* computational problem $\{\Xi, \Omega, \mathcal{M}, \Lambda\}$ *is comprised of the following objects:*

- Ω *is some set, which we call the* primary *or* input *set*
- Λ *is a set of complex valued functions on* Ω*, called the* evaluation *set*
- $\mathcal{M}$ *is a metric space which we call the* output *set*
- $\Xi : \Omega \to \mathcal{M}$ *is a function which we call the* problem *function*

When it is clear what $\mathcal{M}$ *and* Λ *are we will sometimes write* $\{\Xi, \Omega\}$ *for brevity.*

In essence, Ω is the set of inputs that the problem can work with. Ξ is some function that maps an input to an output (which we only require belongs to a metric space $\mathcal{M}$). This is the function that we are trying to compute with some machine (e.g. a BSS or turing machine). The set Λ acts as the set of information that is accessible to the machine.

At first glance, it may seem strange to include Λ as part of the definition of the problem. To show how important this is, we consider the following simple example:

Example 2.4.2. Let $P^1(\mathbb{Z})$ be the space of all non-constant linear polynomials from $\mathbb{R} \to \mathbb{R}$ with integer coefficients. Our task is to find the root of these polynomials. We can write this computational problem in the following way: $\{\Xi, \Omega, \mathcal{M}, \Lambda\}$ with $\Omega = P^1(\mathbb{Z})$, $\Xi(p) = -b/a$ for $p \in P^1(\mathbb{Z})$ such that $p(x) = ax + b$ and $\mathcal{M}$ is the space of rational numbers endowed with the euclidean norm. It is clear that

- If $\Lambda = \{\lambda^1, \lambda^2\}$ with $\lambda^1(p) = a, \lambda^2(p) = b$ (in other words, we have access to both coefficients of the linear polynomial) then the problem is computable with a turing machine.
- If $\Lambda = \{\lambda_n \text{ such that } n \in \mathbb{Z}\}$ with $\lambda_n(p) = p(n)$ (in other words, we can access evaluations of the polynomial at integer values) then the problem is computable with a turing machine.
- If $\Lambda = \{\lambda^1\}$ with λ^1 defined as above (in other words, we can access just the first coefficient) then the problem is not computable with a turing machine.
- If $\Lambda = \{\lambda_0\}$ with λ_0 defined as above (in other words, we can access the polynomial evaluated at 0) the problem is not computable with a turing machine.

This example highlights how the *information* that the machine has access to as well as the operations the machine is allowed to perform completely changes the nature of what is and what isn't computable. Thus it is essential to include Λ in the definition of a computational problem. Similarly, the input set Ω is also critical. Suppose that instead of $\Omega = P^1(\mathbb{Z})$ in the example above, we took $\Omega = \{p^0\}$ where $p^0(x) = x$. In this case, even if $\Lambda = \{\lambda^1\}$ or $\Lambda = \{\lambda^0\}$, the problem would become computable since a turing machine could simply output the answer 0 without considering the input. This is an example of a more general idea: suppose that Ω is some set with a subset Ω' and

that $\Xi : \Omega \to \mathcal{M}$ and Λ are such that the problem $\{\Xi, \Omega, \mathcal{M}, \Lambda\}$ is computable (in the sense of e.g. BSS or Turing). Then $\{\Xi, \Omega', \mathcal{M}, \Lambda\}$ is also computable.

We now make a natural generalisation of BSS and Turing machines for a computational problem:

Definition 2.4.3 (General Algorithm)**.** *Given a computational problem* $\{\Xi, \Omega, \mathcal{M}, \Lambda\}$*, a* general algorithm *is a mapping* $\Gamma : \Omega \to \mathcal{M}$ *such that for each* $\iota \in \Omega$

(i) there exists a finite subset of evaluations $\Lambda_\Gamma(\iota) \subset \Lambda$ *,*

(ii) $\Gamma(\iota)$ *only depends on* $\{\iota_f\}_{f \in \Lambda_\Gamma(\iota)}$ *where* $\iota_f := f(\iota)$*,*

(iii) for every $\tilde{\iota} \in \Omega$ *such that* $\tilde{\iota}_f = \iota_f$ *for every* $f \in \Lambda_\Gamma(\iota)$*, it holds that* $\Lambda_\Gamma(\tilde{\iota}) = \Lambda_\Gamma(\iota)$*.*

The three properties of the algorithm are very natural principles that we would expect any computational device to obey. Essentially, the first one says that *an algorithm cannot ask for infinite information* about an input (note that a single real number does not count as infinite information for the purpose of this definition, so a BSS machine is still a valid general algorithm). The second property says *that the output of an algorithm can only rely on the information that it has accessed* - this prevents an algorithm from bypassing Λ and automatically outputting the value $\Xi(\iota)$ for a given input ι without first accessing functions in Λ. The third property says that *an algorithm must access information in a consistent way* - if it sees the same information for two different inputs, then it cannot behave differently for those inputs.

General algorithms are far more powerful than the standard BSS or Turing machines. In particular, a general algorithm is not restricted to arithmetic or rational operations on the information it can see. This does not stop a general algorithm from being a useful concept. If one can show a lower bound with a general algorithm (that is, a result that says a computational problem is not computable using a general algorithm) then automatically that same lower bound must apply to BSS machines and Turing machines. Essentially any non-computability result shown using a general algorithm suggests that the problem is inherently non-computable not because there are not enough operations to give the correct answer, but because the information about each input available to the algorithm is insufficient to solve the problem. Thus a lower bound for a general algorithm also tells us something about the problem that a similar result for a Turing machine or a BSS machine would be unable to do.

Of primary importance are two classes Δ_0^G and Δ_1^G that represent problems that can be considered 'computable'. The most obvious of these two is the class Δ_0^G of problems which can be computed exactly with a general algorithm. More precisely:

Definition 2.4.4. *A computational problem* $\{\Xi, \Omega, \mathcal{M}, \Lambda\}$ *is said to be in* Δ_0 *if there exists a general algorithm* Γ *such that* $\Gamma(\iota) = \Xi(\iota)$ *for every* $\iota \in \Omega$*. In this case, we write* $\{\Xi, \Omega, \mathcal{M}, \Lambda\} \in \Delta_0^G$

Later we shall work with computational problems for which the input is inexact. Thus an expectation that the algorithm should be able to get the exact answer from inexact readings may be too strong. We thus consider the slightly expanded class Δ_1

of problems which can be computed up to a fixed, predefined (but arbitrarily small) error. This can be expressed formally in the following way:

Definition 2.4.5. *A computational problem* $\{\Xi, \Omega, \mathcal{M}, \Lambda\}$ *is said to be in* Δ_1 *if there exists a sequence of general algorithms* $\{\Gamma^n\}_{n=1}^{\infty}$ *such that*

$$d(\Gamma^n(\iota), \Xi(\iota)) \leq 2^{-n} \text{ for every } \iota \in \Omega. \tag{2.4.1}$$

In this case, we write $\{\Xi, \Omega, \mathcal{M}, \Lambda\} \in \Delta_1^G$

Note trivially that all problems in Δ_0^G are also in Δ_1^G.

Before continuing, let us consider an example of finding the infimum of a sequence with varying input sets to show how these definitions apply to computational problems.

Example 2.4.6. Define $\mathcal{M} = \mathbb{R}$, $\Xi(\iota) = \inf_{n\in\mathbb{N}} \iota_n$ where $\iota = (\iota_1, \iota_2, \dots)$ is a bounded below sequence of real numbers. For such ι, we also define $f_n(\iota) = \iota_n$ and set $\Lambda = \{f_n \,|\, n \in \mathbb{N}\}$. We then define:

1. $\{\Xi_1, \Omega_1, \mathcal{M}_1, \Lambda_1\}$ so that $\mathcal{M}_1 = \mathcal{M}$, Ω_1 is the space of all bounded below sequences of real numbers, $\Xi_1 = \Xi$, and $\Lambda_1 = \Lambda$.

2. $\{\Xi_2, \Omega_2, \mathcal{M}_2, \Lambda_2\}$ so that $\mathcal{M}_2 = \mathcal{M}$, Ω_2 is instead the space of all decreasing bounded below sequences ι such that if $\iota = (\iota_1, \iota_2, \dots)$ then $|\iota_n - \lim_{k\to\infty} \iota_k| \leq 2^{-n}$, Ξ_2 is the restriction of Ξ to Ω_2 and Λ_2 is the restriction of Λ to Ω_2.

3. $\{\Xi_3, \Omega_3, \mathcal{M}_3, \Lambda_3\}$ so that $\mathcal{M}_3 = \mathcal{M}$, Ω_3 is instead the space of all constant sequences, Ξ_3 is the restriction of Ξ to Ω_3 and Λ_3 is the restriction of Λ to Ω_3.

Then $\{\Xi_1, \Omega_1, \mathcal{M}_1, \Lambda_1\} \notin \Delta_1$, $\{\Xi_2, \Omega_2, \mathcal{M}_2, \Lambda_2\} \in \Delta_1 \backslash \Delta_0$ and $\{\Xi_3, \Omega_3, \mathcal{M}_3, \Lambda_3\} \in \Delta_0$. Roughly speaking, these inclusions hold for the following reasons:

1. $\{\Xi_1, \Omega_1, \mathcal{M}_1, \Lambda_1\} \notin \Delta_1$ because any general algorithm acting on an input ι must terminate after accessing finitely many of the elements $\{f(\iota) \,|\, f \in \Lambda_1\}$ which corresponds directly to accessing finitely many elements of the sequence ι. Since it is impossible to access finitely many elements of a sequence and find a neighbourhood for the infimum, the problem cannot be in Δ_1.

2. By contrast, $\{\Xi_2, \Omega_2, \mathcal{M}_2, \Lambda_2\} \in \Delta_1$ because for every $\iota \in \Omega_2$, $\inf \iota = \lim_{k\to\infty} \iota_k$ and we know that the nth element of ι is at most 2^{-n} from this limit. Hence setting $\Gamma^n(\iota) = f_n(\iota) = \iota_n$ satisfies (2.4.1). However, this problem is still not in Δ_0 since it is once again impossible to tell the exact limit of such a sequence just from finitely many evaluations.

3. Finally, it is obvious from the definitions that $\{\Xi_3, \Omega_3, \mathcal{M}_3, \Lambda_3\} \in \Delta_0$: we simply choose Γ so that $\Gamma(\iota) = f_1(\iota) = \iota_1$.

Although it is not important for the thesis, it is worth noting that the there is a generalisation of these two classes to an entire hierarchy of problems. This is known as the *SCI hierarchy* [12, 13, 64] and can be used to classify the level of difficulty for

a problem that is not in Δ_1^G. Roughly speaking, the definitions above correspond to exact computation (Δ_0^G) and computation that can be done exactly with one limit of a sequence algorithms with additional access to error control - that is, we have an upper bound for the maximum error across all inputs if we run just a finite subset of the sequence of algorithms (Δ_1^G). Informally, for $k \geq 2$ a problem is in Δ_k^G if it can be done with $k-1$ nested limits of algorithms with no necessity for error control. In many circumstances it is important to be able to classify where a problem lies in the hierarchy, but for our purposes it suffices to classify a problem as in Δ_0, in Δ_1 or outside of those two classes.

In fact, these definitions and the general SCI hierarchy are a direct continuation of the program initiated by S. Smale to establish the foundations of computational mathematics [17, 23, 24, 109–112].

An important consequence of the definition of the SCI hierarchy and its reliance on general algorithms is that, for a fixed $k \in \{1, 2\}$, if a problem is not in Δ_k^G then a similar result is also true for BSS or Turing machines. Thus any lower bound obtained is independent of the computational model. However, a positive result that a problem is in Δ_k^G does not necessarily imply something about its status on a BSS or a Turing machine.

Once again, we note that the purpose of a general algorithm is to show lower bounds. For the positive results in this thesis, we produce explicit algorithms that rely only on arithmetic operations, in the sense that the algorithms can easily be implemented on a Turing machine if each $f \in \Lambda$ has $f(\iota) \in \mathbb{Q}$ and the algorithms can be easily implemented on a BSS machine if instead each $f \in \Lambda$ has $f(\iota) \in \mathbb{R}$.

We therefore make definitions for the purpose of stating positive results. We say that $\{\Xi, \Omega, \mathcal{M}, \Lambda\} \in \Delta_1^A$ if every $f \in \Lambda$ has $f(\iota) \in \mathbb{Q}$ whenever $\iota \in \Omega$ and there exist algorithms Γ^n satisfying (2.4.1) that can be implemented using a turing machine. We also say $\{\Xi, \Omega, \mathcal{M}, \Lambda\} \in \Delta_1^A$ if every $f \in \Lambda$ has $f(\iota) \in \mathbb{R}$ whenever $\iota \in \Omega$ and there exists algorithms Γ^n satisfying (2.4.1) that can be implemented using a BSS machine. Note that in the rational case we assume that rationals that define Ω and are independent of the specific input are accessible to the algorithm. Similarly in the real case we assume that real constants that define Ω and are independent of the specific input are accessible to the algorithm. This will prove to be important for the positive results on the nullspace property. Finally, we can make analogous definitions for Δ_0^A.

Remark 2.4.7. Although here Ξ is defined for single valued computational problems, we do allow for Ξ to be multivalued in the following way: if Ξ is multivalued we consider the computational task to find an algorithm Γ for which $\Gamma(\iota) \in \Xi(\iota)$ for every $\iota \in \Omega$. There will be further discussion on this later in Section 2.10.

2.5 LPO and LLPO - basic non-computable problems

This thesis makes use of the *constructive* philosophy of mathematics. In this section we outline briefly the basic principles of constructive mathematics and discuss three re-

lated problems: the *limited principle of omniscience* (LPO), the *lesser limited principle of omniscience* (LLPO) (we follow the work in [22]) and *the halting problem.*

In classical mathematics, to prove the existence or non-existence of an object it suffices to show the opposite result. Specifically, classical mathematics makes great use of the *law of the excluded middle* i.e. for a proposition P, either P or $\neg P$. In constructive mathematics a proof is only valid if one can specifically construct a sequence of computations that can be performed in finite time to construct all relevant objects. In particular, the law of the excluded middle is rejected in constructive mathematics.

As a very simple example, consider the following classical proof that there exist a, b irrational with a^b rational. We set $a = \sqrt{2}$ and $c = \sqrt{2}^{\sqrt{2}}$. If c is irrational then $c^a = \sqrt{2}^{\sqrt{2}*\sqrt{2}} = 2$ is rational and the theorem is proven with $b = c$. Otherwise, b is rational and thus a^a is rational, so the theorem is proven with $b = a$. Although most would consider this proof to be acceptable, in constructive analysis it is not valid. The reason for this is that this proof does not give us an explicit example - to know which of (a, c) or (a, a) is acceptable as an example we first have to know whether or not c is irrational.

To understand the degree of non-constructivity of a problem, constructive mathematicians consider two key computability problems. The first of which is LPO:

Definition 2.5.1 (LPO)**.** *Given a sequence $\{x_i\}_{i=1}^{\infty}$ with each $x_i \in \{0, 1\}$, can we determine if $x_i = 0$ for every i or if there is an i such that $x_i = 1$?*

Of course, if one accepts the law of the excluded middle then one also accepts LPO. Indeed, for any sequence of binary digits, one can set P to be the proposition that $\exists i \,|\, x_i = 1$. The negation of P is that $\forall i \, x_i = 0$. Thus if one accepts that P or $\neg P$ then one also accepts LPO.

It is fair to say that LPO is viewed with considerable scepticism amongst constructivists. Indeed, it is easy to show that if one can decide LPO then one can also decide the following problem, known as the *halting problem* [57]. To state this theorem accurately, first let C_q for $q \in \mathbb{N}$ be an enumeration of all possible Turing machines (noting that by the definition of a Turing machine the set of all Turing machines is countable).

Definition 2.5.2 (Halting problem)**.** *The* halting problem *asks if there is a Turing machine H which takes inputs $a, b \in \mathbb{N}$ that halts if and only if $C_a(b)$ halts.*

Famously, Turing [120] showed that such a Turing machine H does not exist.

A slight weakening of LPO that we will consider is LLPO.

Definition 2.5.3 (LLPO)**.** *Given a sequence $\{x_i\}_{i=1}^{\infty}$ with each $x_i \in \{0, 1\}$ such that at most one x_i is non-zero, can we determine if either $x_{2i} = 0$ for all i or $x_{2i-1} = 0$ for all i?*

To gain a heuristic understanding of why LLPO and LPO are suspicious from a constructive viewpoint, consider that an algorithm that can solve either problem would need to read infinite input to determine if a given sequence is composed entirely of zeros. This means that expecting a *finite* time algorithm to solve either problem is unlikely.

2.6 Inexact input and breakdown epsilons

At present, the definition of an SCI hierarchy is idealised in the sense that Λ could be defined so that any algorithm has access to infinite precision real numbers. Even though this is convenient for analysing problems in numerical analysis, this is not a realistic assumption about how real world machines behave. Specifically, obtaining $f(\iota)$ for a given input ι and $f \in \Lambda$ may be a computational task in itself, for example if $f(\iota) = \pi$.

In such circumstances, we can also discuss an SCI hierarchy for obtaining Λ. Following from the ideas established in Section 2.4, we may then need several limits to get the precise values of $f(\iota)$. Thus we discuss the concept of *inexact input* by making the functions in Λ computational problems in their own right, for which we build a similar computational hierarchy.

As with Section 2.4, we focus on Δ_1 information. Here, instead of accessing $f(\iota)$ where $f \in \Lambda$, a general algorithm is permitted only to access mappings $f_n : \Omega \to \mathbb{F}$ for $n \in \mathbb{N}$ where f_n satisfies

$$|f_n(\iota) - f(\iota)| \leq 2^{-n} \quad \forall \iota \in \Omega. \tag{2.6.1}$$

More specifically, we can define Δ_1 information to be

Definition 2.6.1 (Δ_1-information)**.** *Let $\{\Xi, \Omega, \mathcal{M}, \Lambda\}$ be a computational problem. We say that Λ has $\Delta_{1,\mathbb{F}}$-information if each $f \in \Lambda$ is not available, however, there are mappings $f_n : \Omega \to \mathbb{F} + i\mathbb{F}$ such that* (2.6.1) *holds. Finally, if $\hat{\Lambda}$ is a collection of such functions from Ω to $\mathbb{F} + i\mathbb{F}$ described above satisfying* (2.6.1)*, we say that $\hat{\Lambda}$ provides $\Delta_{1,\mathbb{F}}$ information for Λ. Moreover, we denote the family of all such $\hat{\Lambda}$ by $\mathcal{L}^{k,\mathbb{F}}(\Lambda)$.*

Of particular interest are the case $\mathbb{F} = \mathbb{Q}$, representing inexact rational approximations to the true values or the case $\mathbb{F} = \mathbb{R}$ representing inexact real approximations to the true values. Now that we have established the notion of $\Delta_{1,\mathbb{F}}$ information, we can now define what we mean by a computational problem with $\Delta_{1,\mathbb{F}}$ information.

Definition 2.6.2 (Computational problem with $\Delta_{1,\mathbb{F}}$ information)**.** *Given $1 \in \mathbb{N}$ and a field $\mathbb{F}$, a computational problem where Λ has $\Delta_{1,\mathbb{F}}$-information is denoted by $\{\Xi, \Omega, \mathcal{M}, \Lambda\}^{\Delta_{1,\mathbb{F}}}$ and denotes the family of computational problems $\{\Xi, \Omega, \mathcal{M}, \hat{\Lambda}\}$ where $\hat{\Lambda} \in \mathcal{L}^{1,\mathbb{F}}(\Lambda)$.*

This allows us to mathematically discuss computational problems that have input that can be accessed by the algorithm to arbitrary but finite precision. We now discuss Δ_1 in this new setting.

Definition 2.6.3 (Inexact algorithms in Δ_1)**.** *We say the computational problem $\{\Xi, \Omega, \mathcal{M}, \Lambda\}^{\Delta_{1,\mathbb{F}}}$ is in Δ_1 if there exists a sequence of general algorithms Γ^n such that $d(\Gamma^n(\iota), \Xi(\iota)) \leq 2^{-n}$ for every $\iota \in \Omega$ and, crucially, for every choice of $\hat{\Lambda} \in \mathcal{L}^{m,\mathbb{F}}(\Lambda)$ providing Δ_1 information for Λ. Furthermore, to simplify exposition we abuse notation slightly by letting, for a general algorithm Γ,*

$$d(\Gamma(\iota), \Xi(\iota)), \quad \iota \in \Omega$$

denote the supremum of $d(\Gamma(\iota), \Xi(\iota))$ *over all computational problems in* $\{\Xi, \Omega, \mathcal{M}, \Lambda\}^{\Delta_{1,\mathbb{F}}}$.

We will use the notation

$$\{\Xi, \Omega, \mathcal{M}, \Lambda\}^{\Delta_{1,\mathbb{F}}} \in \Delta_k^\alpha$$

(with $k = 0$ or $k = 1$ and $\alpha = G$ for general algorithms or $\alpha = A$ for BSS/Turing machine algorithms) to denote that the computational problem is in Δ_k^α with respect to towers of algorithms with $\Delta_{1,\mathbb{F}}$-information. Since $\{\Xi, \Omega, \mathcal{M}, \Lambda\}^{\Delta_{m,\mathbb{F}}}$ is the collection of all computational problems with Λ replaced by $\hat{\Lambda} \in \mathcal{L}^{1,\mathbb{F}}(\Lambda)$, we note that the use of $\in$ is a slight abuse of notation. When $\mathcal{M}$ and Λ are obvious then we will write $\{\Xi, \Omega\}^{\Delta_{1,\mathbb{F}}} \in \Delta_k^\alpha$ for short.

Remark 2.6.4. Any result concerning $\{\Xi, \Omega, \mathcal{M}, \Lambda\}^{\Delta_1}$ (so that the underlying field $\mathbb{F}$ is not mentioned) should be interpreted as correct in both the case $\mathbb{F} = \mathbb{R}$ and the case $\mathbb{F} = \mathbb{Q}$.

One of the main questions discussed in this thesis is then the following:

$$\text{For which } k \text{ is } \{\Xi, \Omega, \mathcal{M}, \Lambda\}^{\Delta_1} \in \Delta_k^\alpha?$$

It is clear that we cannot always expect the problems discussed in the introduction with inexact input to be in Δ_0 - to do so would mean that we can take inexact input and somehow get an exact answer. Therefore the best that we should hope for is that the problems are in Δ_1. In what remains, we will talk about whether or not a problem is in Δ_1 rather than whether or not it is computable: if a problem is not in Δ_1, then there is some $\epsilon > 0$ such that for any given algorithm there must be some input for which the algorithm is unable to get within ϵ of the correct answer. We call the largest such value of ϵ the *strong breakdown epsilon*. More precisely,

Definition 2.6.5 (**Strong breakdown epsilon**). *We define the Strong Breakdown-epsilon* $\epsilon_{\mathrm{B}} \geq 0$ *as follows:*

$$\epsilon_{\mathrm{B}}^{\mathrm{s}} = \sup\{\epsilon \geq 0 \,|\, \forall \textit{general algorithms } \Gamma, \exists\, \iota \in \Omega \textit{ such that } d_{\mathcal{M}}(\Gamma(\iota), \Xi(\iota)) > \epsilon\},$$

If, for a given computational problem, $\epsilon_{\mathrm{B}}^{\mathrm{s}}$ is large, then this can mean that results obtained in practice are very different from the true solution (see for example the numerical example at the end of Section 4.2.1). If instead $\epsilon_{\mathrm{B}}^{\mathrm{s}}$ is sufficiently small, it will be possible to produce an algorithm that gives reasonably accurate results. An immediate practical consequence of a non-zero breakdown epsilon is that the algorithm should halt when it obtains a precision equal to the breakdown epsilon - any further computations may or may not do better, but the algorithm itself will be unable to tell for all inputs if this is possible. We shall see later that basis pursuit denoising acting on 'nice' (in a sense made precise in Theorem 4.2.15) is an example of a problem for which there is a non-zero breakdown epsilon (and thus traditional computability theory would suggest is impossible) but which can be computed with a high level of accuracy in practice - a fact explained by the small breakdown epsilon.

Even though a small breakdown epsilon implies the existence of an algorithm that can compute the problem to a high degree of accuracy, it is possible that such an algorithm can take arbitrarily long to compute if there is a time cost associated to more accurate approximations to functions in Λ. To characterise this, we introduce the concepts of *minimum runtime*, *runtime breakdown epsilon* and *weak breakdown epsilon*.

Definition 2.6.6 (**Minimum runtime**). *Given* $\{\Xi, \Omega, \mathcal{M}, \Lambda\}^{\Delta_1,\mathbb{F}}$ *and a general algorithm* Γ*, we define the minimum runtime* $T_\Gamma^{\mathbb{F}}(\iota)$ *for* Γ *and* $\iota \in \Omega$ *as*

$$T_\Gamma^{\mathbb{F}}(\iota) := \sup\{m \in \mathbb{N} \,|\, \hat{f}_m \in \hat{\Lambda}_\Gamma(\iota), f \in \Lambda \text{ and } \hat{\Lambda} \in \mathcal{L}^1(\Lambda)\},$$

where $\mathcal{L}^{1,\mathbb{F}}(\Lambda)$ *is defined in Definition 2.6.1. Moreover, given a general algorithm* Γ *and* $\epsilon > 0$ *we define*

$$T^{\mathbb{F}}(\Gamma, \epsilon) := \begin{cases} \sup\{T_\Gamma^{\mathbb{F}}(\iota) \,|\, \iota \in \Omega\}, & \text{if } \forall \iota \in \Omega \quad d_{\mathcal{M}}(\Gamma(\iota), \Xi(\iota)) \leq \epsilon \\ \infty & \text{otherwise.} \end{cases}$$

Definition 2.6.7 (**Limited-runtime breakdown epsilon**). *Given* $\{\Xi, \Omega, \mathcal{M}, \Lambda\}^{\Delta_1,\mathbb{F}}$*, we define the* runtime breakdown epsilon $\epsilon_B^{\mathrm{lr},\mathbb{F}}(M)$ *for each* $M \in \mathbb{N}$ *to be the number in* $\mathbb{R} \cup \{\infty\}$ *such that*

$$\epsilon_B^{\mathrm{lr},\mathbb{F}}(M) := \inf\{\epsilon \geq 0 \text{ such that there exists an algorithm } \Gamma \text{ with } T^{\mathbb{F}}(\Gamma, \epsilon) \leq M\}$$

where we recall that $\inf(\emptyset) = \infty$

Definition 2.6.8 (**Weak breakdown epsilon**). *The* weak breakdown epsilon $\epsilon_{\mathrm{B}}^{\mathrm{w}}$ *is defined by*

$$\epsilon_B^{w,\mathbb{F}} = \sup\{\epsilon \geq 0 \,|\, \forall \text{general algorithms } \Gamma \text{ and } M \in \mathbb{N}\ \exists\, \iota \in \Omega \text{ such that } d_{\mathcal{M}}(\Gamma(\iota), \Xi(\iota)) > \epsilon \text{ or } T_\Gamma^{\mathbb{F}}(\iota) > M\}.$$

Roughly speaking, for a given algorithm Γ and $\epsilon \geq 0$, the minimum runtime explains how much information on Λ is required to get a guaranteed accuracy ϵ on *all inputs*. If the minimum runtime is large then the algorithm is unable to guarantee ϵ accuracy without taking a long time. The runtime breakdown epsilon $\epsilon_{\mathrm{B}}^{\mathrm{lr}}(M)$ is the best ϵ attainable with a fixed precision 2^{-M} on functions in Λ.

The weak breakdown epsilon is the largest epsilon such that no algorithm can guarantee an accuracy smaller than or equal to epsilon and still have a finite minimum runtime. We shall see later (for example, the example presented in the proof of Theorem 4.3.4) that this can differ greatly from the strong breakdown epsilon. If a problem has a large weak breakdown epsilon and a small breakdown epsilon then there will be an algorithm that can compute the solution to a high degree of accuracy, but for any such algorithm, there will exist inputs for which it takes a very large time (for example, five hundred billion years) to get accurate results.

Remark 2.6.9. Again, whenever we discuss $\epsilon_{\mathrm{B}}^{\mathrm{lr}}(M), \epsilon_{\mathrm{B}}^{\mathrm{w}}, T(\Gamma, \epsilon), T_\Gamma(\iota)$ with no mention

of the underlying field $\mathbb{F}$ then the result holds for both $\mathbb{F} = \mathbb{Q}$ and $\mathbb{F} = \mathbb{R}$

2.7 Randomised general algorithms

The theory explained above discusses deterministic algorithms. Whilst in some areas of the mathematics of information (e.g. compressed sensing [11, 121]) the algorithms typically used are deterministic, in others (e.g. deep learning) the algorithms are *probabilistic* in the sense that running the algorithm on the same input twice can yield different results due to randomness inherent in the algorithm. We thus discuss the existing concept of a *probabilistic turing machine* and build a theory of *randomised general algorithms* that allows us to discuss the computability of various problems with probabilistic algorithms.

We first state the mathematical definition of a randomised general algorithm and then explain it informally.

Definition 2.7.1 (Randomised General Algorithm)**.** *Given a computational problem* $\{\Xi, \Omega, \mathcal{M}, \Lambda\}$*, a* randomised general algorithm *(RGA) is a set* X *of general algorithms* $\Gamma : \Omega \to \mathcal{M}$*, a sigma-algebra* $\mathcal{F}$ *on* X *and a family of probability measure* $\{\mathbb{P}\}_{\iota \in \Omega}$ *such that the following conditions hold:*

(Pi) For each ι*, the mapping* $\Gamma_\iota^{\text{ran}} : X \to \mathcal{M}$ *defined by* $\Gamma_\iota^{\text{ran}}(\Gamma) = \Gamma(\iota)$ *is measurable from* $(X, \mathcal{F})$ *to* $(\mathcal{M}, \mathcal{B})$ *where* $\mathcal{B}$ *is the borel-sigma algebra on* $\mathcal{M}$*.*

(Pii) Suppose that $\iota_1, \iota_2 \in \Omega$ *and* $X' \in \mathcal{F}$ *is a set such that for every* $\Gamma \in X'$*, we have that if* $\lambda \in \Lambda_\Gamma(\iota_1)$ *then* $\lambda(\iota_1) = \lambda(\iota_2)$*. Then* $\mathbb{P}_{\iota_1}(X') = \mathbb{P}_{\iota_2}(X')$*.*

(Piii) For each $n \in \mathbb{N}$ *and* $\iota \in \Omega$*, the sets* $T^n(\iota) := \{\Gamma \in X \,|\, T_\Gamma(\iota) \leq n\}$ *are* $\mathcal{F}$*-measurable (i.e.* $T^n(\iota) \in \mathcal{F}$*).*

Roughly speaking, a randomised general algorithm takes an input, randomly picks a general algorithm and executes it on that input. This is a similar idea to randomly choosing a path at each step of the program, but it does not require specifying any ordering of the execution (in the same way that a turing machine is defined by discussing how it proceeds given that the machine is in a certain position whereas a general algorithm does not require this to be specified). The idea of having a family of probability measures is to allow for a certain level of bias in the way it executes on a given input. To explain where this idea comes from, imagine that we want to generalise the BSS machine to a randomised BSS machine. The difference between the BSS machine and the random BSS machine would be that each branch node connects to a family of instruction nodes and executes such a branch node by picking its next instruction node according to some random distribution. To accurately model what is possible on a real computer, the random distribution would be a function of the current state (in particular, we have lost no generality by switching to a random BSS machine from a BSS machine). Since the current state is somehow a function of the input, for a randomised general algorithm we allow the underlying distribution to depend on the input.

We then have the three conditions (Pi), (Pii) and (Piii). (Pi) is simple to explain: we would somehow like to measure the distribution in $\mathcal{M}$ of the algorithm's execution on input ι. In particular, without condition (Pi), we cannot ask for the probability that the output of the algorithm with input ι has a certain error, or even ask for the probability of the algorithm executing ι and getting the correct answer. Since this is a crucial question, we define it as the first condition.

The second condition, (Pii), places a limitation on how the general algorithm is randomly selected for a given input ι. In some sense it establishes some level of consistency: if we have a measurable set X' of general algorithms which all see matching information when executed on ι_1 and ι_2 then we should select X' with the same probability for input ι_1 and ι_2. This reflects the idea that for any model of computation with a list of execution steps, the distribution of the next step should only depend on the information that the computer has observed so far (i.e. the state space) and not directly on the input itself. Thus if two inputs yield the same observations then the computer should choose its next step in the same way for both inputs. We shall see shortly that without this condition, it is possible to design a randomised general algorithm that can solve the LPO problem with probability 1.

Finally, the third condition (Piii) allows us to ask, for a fixed input, about the distribution of the runtime for that input. This is an important quantity that we would very much like to be able to measure. Without it, we cannot ask what for the probability of the algorithm taking longer than, say, fifty million years, to execute on input ι. Actually and perhaps surprisingly, the third condition is even more important than that: without it, it is again possible to design a randomised general algorithm that can solve the LPO problem with probability 1.

2.7.1 Solving LPO with relaxed conditions

In this section we shall show how losing either (Pii) or (Piii) from the list of requirements would allow us to create a randomised general algorithm that can solve the LPO problem. Let us start by considering an alternative definition of randomised general algorithms that only satisfy (Pi) and (Piii). Set $\{\Xi_3, \Omega_3, \mathcal{M}_3, \Lambda_3\}$ to be the LPO problem as discussed in Section 2.5. We set A^1 to be an algorithm that immediately outputs the value 1 and A^0 to be an algorithm that immediately outputs 0. We set $X = \{A^0, A^1\}$ and the sigma algebra $\mathcal{F}$ to be $\mathcal{F} = (\{A^0\}, \{A^1\}, \{A^0, A^1\}, \emptyset)$. For $\iota \in \Omega$, if ι has no entries with a 1, we set $\mathbb{P}_\iota(A^0) = 1$ and $\mathbb{P}_\iota(A^1) = 0$. If instead ι has an entry with a 1, we set $\mathbb{P}_\iota(A^0) = 0$ and $\mathbb{P}_\iota(A^1) = 1$. It is clear then that (Pi) is satisfied and moreover $\mathbb{P}_\iota(\Gamma_\iota^{\text{ran}} = \Xi(\iota)) = 1$. Furthermore, for each $n \in \mathbb{N}$ and $\iota \in \Omega$, $T^n(\iota) = X \in \mathcal{F}$. This construction therefore satisfies (Pi) and (Piii) and solves LPO with probability 1.

Showing that dropping condition (Piii) can lead to a randomised general algorithm that can solve LPO is more involved. We define the following family of algorithms: for $n \in \mathbb{N}$, we set A_n^1 to be an algorithm that reads the first n entries of the infinite input vector ι, and then outputs 1 and we set A_n^0 to be an algorithm that reads the first n entries of ι and then outputs 0. Let $A^0 = \cup_{n=1}^\infty A_n^0$. We define $\mathcal{F}$ to be the sigma

algebra generated by A^0 and each set $\{A^1_n\}$ for $n \in \mathbb{N}$. Crucially, sets of the form $\{A^0_n\}$ are not $\mathcal{F}$ measurable. If every entry of the input ι is 0 then we set $\mathbb{P}_\iota(S) = 1$ for any set $S \in \mathcal{F}$ such that $A^0 \subseteq S$ and $\mathbb{P}_\iota(S) = 0$ if $A^0 \not\subseteq S$. If instead ι has a 1 somewhere in its entries, we let n be the first entry of ι that is equal to 1. If $A^1_n \subseteq S$, we set $\mathbb{P}_\iota(S) = 1$ and if instead $A^1_n \not\subseteq S$ then we set $\mathbb{P}_\iota(S) = 0$. It is clear that $\Gamma^{\text{ran}}_\iota$ is $\mathcal{F}, \mathcal{B}$ measurable: $\Gamma^{\text{ran}}(\iota) = 1$ is exactly $\cup^\infty_{n=1} A^1_n$ and $\Gamma^{\text{ran}}(\iota) = 0$ is exactly $\cup^\infty_{n=1} A^0_n$ which are both measurable. Moreover, for ι with all entries set to 0 we have $\mathbb{P}_\iota(\Gamma^{\text{ran}}(\iota) = 0) = 1$ and for ι with an entry set to 1 we have $\mathbb{P}_\iota(\Gamma^{\text{ran}}(\iota) = 1) = 1$. Thus this randomised general algorithm solves LPO with probability 1.

Finally, we check the consistency condition (Pii). Let ι_1, ι_2 be distinct inputs in Ω. There are two cases for ι_1 and ι_2: either exactly one of ι_1, ι_2 is a sequence consisting entirely of 0 entries, or both ι_1 and ι_2 have at least one 1 somewhere. Let S be a set such that for all algorithms $\Gamma \in S$ and $\lambda \in \Lambda_\Gamma(\iota)_1$, we have $\lambda(\iota_1) = \lambda(\iota_2)$.

In the first case, we first suppose that $A^0 \subseteq S$. Then since A^0_n is in S for all n, the first n entries of ι_1 are equal to the first n entries of ι_2. Thus $\iota_1 = \iota_2$, contradicting ι_1, ι_2 distinct. If instead $A^0 \not\subseteq S$ then S consists only of unions of sets of the form $\{A^1_n\}$. We therefore have $\mathbb{P}_{\iota_1}(S) = 0$. Let the first non-zero entry of ι_2 be the nth entry (such an n exists because $\iota_2 \neq \iota_1$). If $A^1_n \notin S$ then $\mathbb{P}_{\iota_2}(S) = 0$. If instead $A^1_n \in S$ then by the definition of A^1_n (specifically that A^1_n always reads the first n entries of its input), $\lambda_n \in \Lambda_{A^1_n}(\iota_1)$. But then by the definition of S and the fact that ι_1 is comprised entirely of 0s, we have $0 = \lambda_n(\iota_1) = \lambda_n(\iota_2)$. Hence the nth entry of ι_2 is 0, contradicting the assumption that the nth entry of ι_2 is 1.

In the second case, both ι_1 and ι_2 are non-zero in at least one entry. Let n_1 (respectively n_2) be the first entry of ι_1 (respectively ι_2) which is non-zero. If $A^1_{n_1}, A^1_{n_2} \notin S$ then $\mathbb{P}_{\iota_1}(S) = \mathbb{P}_{\iota_2}(S) = 0$. If instead $A^1_{n_1} \in S$ then (by the definition of $A^1_{n_1}$), $\lambda_1, \lambda_2, \ldots, \lambda_{n_1} \in \Lambda_{A^1_{n_1}}(\iota_1)$. Thus $\lambda_i(\iota_1) = \lambda_i(\iota_2)$ for all $i \leq n_1$. Thus ι_2 has zeros as its first $n_1 - 1$ entries and the n_1th entry of ι_2 is 1. Therefore $n_1 = n_2$ and so $\mathbb{P}_{\iota_1}(S) = \mathbb{P}_{\iota_2}(S) = 1$. Finally, if $A^1_{n_2} \in S$ then $\lambda_1, \lambda_2, \ldots, \lambda_{n_2} \in \Lambda_{A^1_{n_2}}(\iota_1)$ by the definition of $A^1_{n_2}$. Hence $\lambda_i(\iota_1) = \lambda_i(\iota_2)$ for all $i \leq n_2$ and thus once again $n_1 = n_2$. Thus $\mathbb{P}_{\iota_1}(S) = \mathbb{P}_{\iota_2}(S) = 1$. This completes the proof that the algorithm obeys (Pii).

2.8 Probabilistic breakdown epsilons

In the same way that we can define a collection of breakdown epsilons for deterministic algorithms, we can define the *probabilistic breakdown epsilons*.

Definition 2.8.1 (Probabilistic Strong Breakdown-epsilon)**.** *We define the probabilistic Strong Breakdown-epsilon* $\epsilon^{\text{s}}_{\mathbb{P}\text{B}} : [0, 1) \to \mathbb{R}$ *as follows:*

$$\epsilon^{\text{s}}_{\mathbb{P}\text{B}}(p) = \sup\{\epsilon \geq 0, \mid \forall\, \Gamma^{\text{ran}} \in \text{RGA}\ \exists\, \iota \in \Omega \text{ such that } \mathbb{P}_\iota(d_{\mathcal{M}}(\Gamma^{\text{ran}}_\iota, \Xi(\iota)) > \epsilon) > p\}$$

where $\Gamma^{\text{ran}}_\iota$ *is defined in (Pi) in Definition 2.7.1.*

To simplify the exposition, there is a slight abuse of notation as $\mathbb{P}_\iota(d_{\mathcal{M}}(\Gamma^{\text{ran}}_\iota, \Xi(\iota)))$ here obviously refers to the the pushforward measure from $(X, \mathcal{F}, \mathbb{P}_\iota)$ to the real Borel

sigma algebra. However, the exact meaning will be obvious from the context. We will also do the same on other occasions without specifically commenting on the pushforward measure. Note that the probabilistic Strong Breakdown-epsilon of p is the largest ϵ so that the probability of failure with at least ϵ-error is greater than p.

To construct the probabilistic version of the weak breakdown epsilon (where the same abuse of notation is employed), we first need to construct the probabilistic runtime. In particular, given $\iota \in \Omega$, we denote the random variable $T_{\Gamma^{\mathrm{ran}}}(\iota)$ by

$$T_{\Gamma^{\mathrm{ran}}(\cdot)}(\iota) : X \to \mathbb{N} \cup \{\infty\} \text{ by } T_{\Gamma^{\mathrm{ran}}(\Gamma)}(\iota) = T_{\Gamma}(\iota)$$

By assumption (Piii), this random variable is measurable. We can thus define the *probabilistic weak breakdown epsilon*:

Definition 2.8.2 (Probabilistic Weak Breakdown epsilon)**.** *We define the probabilistic weak breakdown epsilon* $\epsilon^{\mathrm{w}}_{\mathbb{P}\mathrm{B}} : [0,1) \to \mathbb{R}$ *as follows:*

$$\epsilon^{\mathrm{w}}_{\mathbb{P}\mathrm{B}}(p) = \sup\{\epsilon \geq 0 \,|\, \forall\, \Gamma^{\mathrm{ran}} \in \mathrm{RGA} \text{ and } M \in \mathbb{N}\ \exists\, \iota \in \Omega \text{ such that} \\ \mathbb{P}_\iota(d_{\mathcal{M}}(\Gamma^{\mathrm{ran}}_\iota, \Xi(\iota)) > \epsilon \text{ or } T_{\Gamma^{\mathrm{ran}}}(\iota) > M) > p\},$$

where $\Gamma^{\mathrm{ran}}_\iota$ *is defined in (Pi) in Definition 2.7.1.*

The probabilistic Weak Breakdown-epsilon describes a weaker form of failure than the probabilistic Strong Breakdown-epsilon. In particular, the Weak Breakdown-epsilon of p is the largest ϵ so that for any randomised algorithm and $M \in \mathbb{N}$, the probability of either getting an error at least of size ϵ, or having spent runtime longer than M, is greater than p. Finally, just as the Strong and Weak Breakdown-epsilons have probabilistic versions, so has the Limited-runtime Breakdown-epsilon.

Note that $T(\Gamma, \epsilon)$ can be written as

$$T(\Gamma,\epsilon) = \begin{cases} \inf\{m \,|\, d_{\mathcal{M}}(\Gamma(\iota), \Xi(\iota)) \leq \epsilon \text{ and } T_\Gamma(\iota) \leq m\ \forall \iota \in \Omega\} & \text{if such an } m \\ & \text{exists} \\ \infty & \text{otherwise.} \end{cases}$$

That is to say, $T(\Gamma, \epsilon)$ is the worst-case (over possible inputs) runtime for Γ to obtain ϵ accuracy. This allows us to make a probabilistic Limited-runtime Breakdown-epsilon. In particular, for any randomised general algorithm Γ^{ran} we can define

$$T_{\mathbb{P}}(\Gamma^{\mathrm{ran}}, \epsilon, p) := \begin{cases} \inf M & \text{if } M \neq \emptyset \\ \infty & \text{otherwise} \end{cases} \tag{2.8.1}$$

$$\text{where } M := \{m \,|\, \mathbb{P}_\iota(d_{\mathcal{M}}(\Gamma^{\mathrm{ran}}_\iota, \Xi(\iota)) \leq \epsilon \text{ and } T_{\Gamma^{\mathrm{ran}}}(\iota) \leq m) > p\ \forall \iota \in \Omega\}.$$

We remark in passing that the definitions are well defined as the sets considered are clearly measurable by the continuity of the metric $d_{\mathcal{M}}$ as well as (Pi) and (Piii) in Definition 2.7.1 of an RGA.

Definition 2.8.3 (Probabilistic Limited-runtime Breakdown-epsilon)**.** *Given the computational problem* $\{\Xi, \Omega, \mathcal{M}, \Lambda\}^{\Delta_1}$*, we define the probabilistic limited-runtime break-*

down epsilon $\epsilon^{\mathrm{lr}}_{\mathbb{P}\mathrm{B}}(M,p)$ *for each* $M \in \mathbb{N}$ *as*

$$\epsilon^{\mathrm{lr}}_{\mathbb{P}\mathrm{B}}(M,p) := \inf\{\epsilon \geq 0 \text{ such that } \exists\, \Gamma^{\mathrm{ran}} \in \mathrm{RGA} \text{ with } T_{\mathbb{P}}(\Gamma^{\mathrm{ran}},\epsilon,p) \leq M\},$$

where we recall that $\inf(\emptyset) = \infty$

The connection between the different Breakdown-epsilons can be summarised in the following simple proposition which we prove in Chapter 5.

Proposition 2.8.4. *Given* $\{\Xi,\Omega,\mathcal{M},\Lambda\}^{\Delta_1}$, $M \in \mathbb{N}$ *and* $p,q \in (0,1)$ *with* $p \leq q$ *we have*

$$\epsilon^{\mathrm{s}}_{\mathbb{P}\mathrm{B}}(q) \leq \epsilon^{\mathrm{s}}_{\mathbb{P}\mathrm{B}}(p) \leq \epsilon^{\mathrm{s}}_{\mathrm{B}}, \tag{2.8.2}$$
$$\epsilon^{\mathrm{w}}_{\mathbb{P}\mathrm{B}}(q) \leq \epsilon^{\mathrm{w}}_{\mathbb{P}\mathrm{B}}(p) \leq \epsilon^{\mathrm{w}}_{\mathrm{B}}, \tag{2.8.3}$$
$$\epsilon^{\mathrm{lr}}_{\mathbb{P}\mathrm{B}}(M,p) \leq \epsilon^{\mathrm{lr}}_{\mathbb{P}\mathrm{B}}(M,q) \leq \epsilon^{\mathrm{lr}}_{\mathrm{B}}(M), \tag{2.8.4}$$
$$\epsilon^{\mathrm{s}}_{\mathbb{P}\mathrm{B}}(p) \leq \epsilon^{\mathrm{w}}_{\mathbb{P}\mathrm{B}}(p), \tag{2.8.5}$$
$$\epsilon^{\mathrm{w}}_{\mathbb{P}\mathrm{B}}(p) \leq \epsilon^{\mathrm{lr}}_{\mathbb{P}\mathrm{B}}(M,p), \quad p = [1/2,1), \tag{2.8.6}$$
$$\epsilon^{\mathrm{s}}_{\mathrm{B}} \leq \epsilon^{\mathrm{w}}_{\mathrm{B}} \leq \epsilon^{\mathrm{lr}}_{\mathrm{B}}(M). \tag{2.8.7}$$

Moreover,

$$\lim_{M\to\infty} \epsilon^{\mathrm{lr}}_{\mathrm{B}}(M) = \epsilon^{\mathrm{w}}_{\mathrm{B}}, \qquad \lim_{M\to\infty} \epsilon^{\mathrm{lr}}_{\mathbb{P}\mathrm{B}}(M,p) \begin{cases} = \epsilon^{\mathrm{w}}_{\mathbb{P}\mathrm{B}}(p) & p = 1/2 \\ \leq \epsilon^{\mathrm{w}}_{\mathbb{P}\mathrm{B}}(p) & p \in (0,1/2). \end{cases} \tag{2.8.8}$$

2.9 A key result

The following major result allows us to show that a variety of problems have non-zero breakdown epsilons.

Theorem 2.9.1. *Let* $\{\Xi,\Omega,\mathcal{M},\Lambda\}$ *be an arbitrary computational problem. Fix* $N \in \mathbb{N}\cup\{\infty\}$ *with* $N \geq 3$*, and suppose that there are two sequences* $\{\iota^1_n\}_{n=1}^N$, $\{\iota^2_n\}_{n=1}^N \subset \Omega$ *satisfying the following conditions:*

(a) There are sets $S^1, S^2 \subset \mathcal{M}$ *and* $\kappa > 0$ *such that* $\inf_{x_1\in S^1, x_2 \in S^2} d_{\mathcal{M}}(x_1,x_2) \geq \kappa$ *and* $\Xi(\iota^j_n) \subset S^j$ *for* $j = 1,2$*.*

(b) For every $f \in \Lambda$ *there is a* $c_f \in \mathbb{C}$ *such that* $|f(\iota^j_n) - c_f| \leq 1/4^n$ *for all* $n \leq N$ *and* $j = 1,2$*.*

Then, if we consider $\{\Xi,\Omega,\mathcal{M},\Lambda\}^{\Delta_1}$*, we have the following:*

(i) For all $M \leq N-2$ *and* $p \geq 1/2$ *then* $\epsilon^{\mathrm{lr}}_{\mathrm{B}}(M) \geq \epsilon^{\mathrm{lr}}_{\mathbb{P}\mathrm{B}}(M,p) \geq \kappa/2$*.*

(ii) For $N = \infty$ *and* $p \in (0,1/2)$ *then* $\epsilon^{\mathrm{w}}_{\mathrm{B}} \geq \epsilon^{\mathrm{w}}_{\mathbb{P}\mathrm{B}}(p) \geq \kappa/2$*.*

(iii) If there is an $\iota^0 \in \Omega$ *such that for every* $f \in \Lambda$ *we have that (b) is satisfied with* $c_f = f(\iota^0)$ *then* $\{\Xi,\Omega,\mathcal{M},\Lambda\}^{\Delta_1} \notin \Delta^G_1$*. Moreover,* $\epsilon^{\mathrm{s}}_{\mathrm{B}} \geq \epsilon^{\mathrm{s}}_{\mathbb{P}\mathrm{B}}(p) \geq \kappa/2$ *for* $p \in (0,1/2)$*.*

(iv) If (iii) is satisfied, $\mathcal{M} = (\mathbb{R}^d, \|\cdot\|_r)$ *for some* $r \in \mathbb{N} \cup \{\infty\}$, Λ *is finite,* S^1 *and* S^2 *in (a) are singletons, and*

$$\forall\, \iota_n^1, \iota_k^2 \in \Omega \text{ where } \iota_n^1 \neq \iota_k^2\ \exists f \in \Lambda \text{ such that } f(\iota_n^1) \neq f(\iota_k^2), \tag{2.9.1}$$

then $\{\Xi, \Omega, \mathcal{M}, \Lambda\}^{\Delta_{1,\mathbb{R}}} \in \Delta_1^A$ *implies decidability of the LLPO (Lesser Limited Principle of Omniscience), and if in addition* $\Xi(\iota^0) = S^2$ *in (iii) then* $\{\Xi, \Omega, \mathcal{M}, \Lambda\}^{\Delta_{1,\mathbb{R}}} \in \Delta_1^A$ *implies decidability of the Halting problem. Furthermore, if* $f(\iota_n^1), f(\iota_n^2)$ *are both rational for each* $f \in \Lambda$ *and* $S^1, S^2 \in \mathbb{Q}^d$ *then each reference to* $\Delta_{1,\mathbb{R}}$ *also holds for* $\Delta_{1,\mathbb{Q}}$.

To show non-computability results, it will suffice to generate examples satisfying (a) and (b). We can then obtain more precise results using (i) to (iv) depending on the situation.

The proof of Theorem 2.9.1 is highly technical. We shall delay this proof until Chapter 5. However, we shall provide here a rough justification of the valdiity of 2.9.1 for establishing that deterministic functions that satisfy (a) and (b) are not computable on a BSS machine in the case where $c_f = f(\iota^0)$ as in part (iii).

1. Set $\iota_{2n} = \iota_n^1, \iota_{2n-1} = \iota_n^2$. Roughly speaking, we choose Δ_1 information for Λ in the following way: for $f \in \Lambda$ we have $f_m(\iota_n) = f(\iota_n)$ if $n \leq m$ and $f_m(\iota_n) = c_f$ otherwise.

2. The algorithm terminates in finite time for ι_0 and hence $M := T_\Gamma(\iota)$ is finite.

3. For $m \leq M$ we have $f_m(\iota_0) = f_m(\iota_{M+1}) = f_m(\iota_{M+2})$. Thus the algorithm 'sees' the same values on ι_0, ι_{M+1} and ι_{M+2}.

4. Since the algorithm is guaranteed to behave in the same way if it sees the same input, we must have $\Gamma(\iota_0) = \Gamma(\iota_{M+1}) = \Gamma(\iota_{M+2})$. But then either $\Gamma(\iota_{M+1})$ is far from $\Xi(\iota_{M+1})$ or $\Gamma(\iota_{M+2})$ is far from $\Xi(\iota_{M+2})$, otherwise there are members of $\Xi(\iota_{M+1})$ and $\Xi(\iota_{M+2})$ which are 'close', contradicting (a). Thus Γ either gives a value far away from the true solution at ι_{M+1} or at ι_{M+2}.

Remark 2.9.2. Of course, this is an oversimplification. There are additional technicalities when working with Turing machines since c_f needs to be rational in the Turing case for this approach to produce valid Δ_1 information. Moreover, we have not shown anything about the breakdown epsilon and we have not discussed the additional difficulty of working with probabilistic algorithms. If we were to take these issues into account the proof becomes vastly more complicated and interested readers looking for the full details are advised to see Chapter 5.

2.9.1 A comparison to existing work

Theorem 2.9.1 is somewhat similar in appearance to some earlier work on computable functions for type two turing machines and similar models (for example [77, 126]). In both of these two references, continuity is shown to be a necessary condition for

computability with machines similar to type two turing machines. At first glance these results may appear to be very similar to Theorem 2.9.1. However, there are some important differences:

1. The results proven in both [77, 126] do not directly apply to models that are substantially different from a turing machine. This is important since the BSS model is crucial for analysing algorithms used in numerical analysis.

2. Theorem 2.9.1 also applies if the function Ξ is multivalued. As the problems we analyse are potentially multivalued this additional feature of Theorem 2.9.1 is absolutely necessary.

3. Standard techniques in neural networks are probabilistic, in the sense that the algorithm can adaptively change its behaviour depending on the input that it has read and its own choice of randomisation. This adaptive (i.e. depending on the input) randomisation is not covered by a theorem solely addressing deterministic computation.

4. Perhaps most importantly, since a substantial part of this thesis is to demonstrate that a collection of practical methods are non-computable but nonetheless effective, a theory that only distinguishes between computable and non-computable functions is insufficiently fine to explain the success of techniques like basis pursuit denoising observed in practice. The concept of a breakdown epsilon is used throughout the thesis to address this issue and is a key component of Theorem 2.9.1.

It is worth expanding our discussion on 2. There is a well established notion of continuity for multivalued functions known as *hemicontinuity* and defined by first explaining the concepts of upper and lower hemicontinuity (we present here a version of [6] adapted from topological spaces to metric spaces). Specifically, hemicontinuity is defined in the following way:

Definition 2.9.3. *Assume that both Ω and $\mathcal{M}$ are metric spaces and let Ξ be a multivalued mapping from Ω to $\mathcal{M}$. We say that Ξ is* upper hemicontinuous *at the point $\iota \in \Omega$ if for every sequence $\{\iota_n\}_{n=1}^{\infty}$ such that $\iota_n \to \iota$ then if $b_n \in \Xi(\iota_n)$ is such that $b_n \to b$ for some b then $b \in \Xi(\iota)$.*

Remark 2.9.4. Note that the definition used here for upper hemicontinuity is slightly stronger than the standard definition. However, for our problems $\Xi(\iota)$ is compact for each $\iota \in \Omega$ and under this condition the definition above coincides with the standard topological definition.

Definition 2.9.5. *With the same setup as in the previous definition for Ω, $\mathcal{M}$ and Ξ, we say that Ξ is* lower hemicontinuous *at $\iota \in \Omega$ if for any $b \in \Xi(\iota)$ and any sequence $\{\iota_n\}_{n=1}^{\infty}$ such that $\iota_n \to \iota$, we have a subsequence $\{\iota_{n_k}\}_{k=1}^{\infty}$ of $\{\iota_n\}_{n=1}^{\infty}$ such that there are $b_{n_k} \in \Xi(\iota_{n_k})$ with $b_{n_k} \to b$ as $k \to \infty$.*

To explain the two definitions, let us present now an example of a function which is upper but not lower hemicontinuous and an example of a function which is lower but not upper hemicontinuous

Example 2.9.6 (Upper but not lower hemicontinuous)**.** Let $\Omega = \mathcal{M} = [0,1]$. We set $\Xi(\iota) = [0,1]$ if $\iota = 1$ and $\Xi(\iota) = 0$ for $\iota \in [0,1)$. At 1 it is easy to see that Ξ is upper hemicontinuous: indeed, if $\iota_n \in [0,1]$ is not eventually identically 1 and $b_n \in \Xi(\iota_n)$ is convergent then $b_n = 0$ and hence the limit $b = 0 \in \Xi(1)$. If instead ι_n is eventually identically 1 then it is clear that eventually $\Xi(\iota_n) = \Xi(1)$ and so Ξ is upper hemicontinuous at 1. However, Ξ is not lower hemicontinuous: for the sequence $\iota_n = 1 - 1/n$ we have $\Xi(\iota_n) = 0$ and hence no subsequence of $\Xi(\iota_n)$ can converge to $b = 1$.

Example 2.9.7 (Lower but not upper hemicontinuous)**.** Again, we take $\Omega = \mathcal{M} = [0,1]$. This time, set $\Xi(\iota) = 0$ if $\iota = 1$ and $\Xi(\iota) = [0,1]$ for $\iota \in [0,1)$. It is clear that Ξ is lower hemicontinuous at 1: if $\iota_n \to 1$ then any subsequence of ι_n has $0 \in \Xi(\iota_n)$ and therefore there exists a $b_n := 0$ with $b_n \in \Xi(\iota_n)$ convergent to the single element of $\Xi(1)$. However, Ξ is not upper hemicontinuous: let $\iota_n = 1 - 1/n$ so that $1 \in \Xi(\iota_n)$. We can then set $b_n = 1$ and observe that b_n is constant (and so b_n is convergent to 1). Thus for Ξ to be upper hemicontinuous we require $1 \in \Xi(1)$. However, $\Xi(1) = 0$ and we conclude that Ξ is not upper hemicontinuous at 1.

Continuity is then defined in the following way: a multivalued function is hemicontinuous iff it is both upper and lower hemicontinuous. Natural analogues of the work in [77, 126] would be to consider hemicontinuity as a necessity for computability, and failing that, replace the word 'hemicontinuity' with either 'upper hemicontinuity' or 'lower hemicontinuity'. However, the two examples above are both computable and thus counterexamples to such a theorem: in either case, we choose $\Gamma(\iota) \equiv 0$ thus giving $\Gamma(\iota) \in \Xi(\iota)$ for each $\iota \in [0,1]$. Therefore continuity alone is insufficient for our multivalued problems and thus we require a theorem closer in nature to that of Theorem 2.9.1.

2.10 Optimisation as a computational problem

To prove results about the computability of optimisation problems using (probabilistic) general algorithms, the SCI hierarchy, imprecise information and breakdown epsilons, we must first cast the problems discussed in Chapter 1 as computational problems defined as in Section 2.4. We write $\|\cdot\|_*$ for $\|\cdot\|_1$ or $\|\cdot\|_{\mathrm{TV}}$ (see Definition 1.8.2 the definition of TV-norm) depending on the context. Note that all the statements in our theorems are true regardless of isotropic or anisotropic TV norm. In the TV case we let $V^1 = \mathbb{C}^{m_1 \times m_2}$ and $V^2 = \mathbb{C}^{m_1 m_2 \times N_1 N_2}$ to be the set of bounded linear operators between $\mathbb{C}^{N_1 \times N_2}$ to V^1, whereas in the l^1 case we let $m_1 = m$, $m_2 = 1$, $N_1 = N$, $N_2 = 1$ for some fixed integers m, N.

In all the problems, except for neural networks, $(\mathcal{M}, d)$ is either $(\mathbb{F}^d \cup \{\infty\}, \|\cdot\|)$ or $(\mathbb{F}^{m \times n} \cup \{\infty\}, \|\cdot\|)$, where $\mathbb{F}$ is either $\mathbb{R}$ or $\mathbb{C}$ and the dimensions d, m, n and the norm

$\|\cdot\|$ is the frobenius norm. The metric space for the neural network case is described below. In all cases, the primary set Ω will always contain vectors and/or matrices, thus Λ will always be the collection of coordinate functions. In particular, if $\iota \in \Omega$ is always of the form $\iota = (y, U)$ where $y \in \mathbb{C}^m$ and $U \in \mathbb{C}^{m\times n}$ then

$$\Lambda = \{f_j, f_{i,j} \mid f_{i,j} = U_{ij}, f_j(y) = y_j\}.$$

Note that Ω will change depending on the classes of Us and ys, where these classes can be described by a finite string of rational numbers. Also, Ω could be the set of pairs (y, U), where y is fixed and only U is allowed to vary. Note that lower bounds become stronger if one or more of the elements in the k-tuple inputs are fixed. The rest of the elements in the computational problems are defined below.

Definition 2.10.1 (Computational problems)**.** *We define the following for the problems listed in the introduction:*

(i) Linear Programming: $\iota \in \Omega$ is always of the form $\iota = (y, A)$, where $A \in \mathbb{R}^{m\times N}$ and $y \in \mathbb{R}^m$, and

$$\Xi_{\mathrm{LP}}(\iota) = \operatorname*{argmin}_{x\in\mathbb{R}^N} c^T x \text{ such that } Ax = y, \quad x \geq 0$$

with c a fixed vector in $\mathbb{Q}^N$.

(ii) Basis Pursuit and basis pursuit denoising: $\iota \in \Omega$ is always of the form $\iota = (y, U)$ where $y \in V^1$ and $U \in V^2$. Moreover, the parameter $\delta \geq 0$ is a fixed rational number (with $\delta > 0$ representing basis pursuit denoising and $\delta = 0$ representing basis pursuit) and

$$\Xi_{\mathrm{BP}}(\iota) = \begin{cases} \operatorname{argmin}_{x\in V^1} \|x\|_1 \text{ such that } \|Ux - y\|_2 \leq \delta, & \text{if } \iota \text{ is feasible} \\ \infty & \text{otherwise .} \end{cases}$$

(iii) Constrained Lasso: For CL, Ω is as in the BP case. Also, the parameter $\tau > 0$ is a fixed rational number and

$$\Xi_{\mathrm{CL}}(\iota) = \operatorname*{argmin}_{x\in V^1} \|Ax - y\|_2 \text{ such that } \|x\|_1 \leq \tau.$$

(iv) Unconstrained Lasso: For UL, Ω is as in the BP case. Moreover, the parameter $\lambda > 0$ is a fixed rational number and

$$\Xi_{\mathrm{UL}}(\iota) = \operatorname*{argmin}_{x\in V^1} \|Ax - y\|_2^2 + \lambda\|x\|_1.$$

(v) Basis pursuit total variation: $\iota \in \Omega$ is always of the form $\iota = (y, A)$ where $y \in V^1$

and $A \in V^2$. Ξ_{BPTV} is defined as follows

$$\Xi_{\mathrm{BPTV}}(\iota) = \begin{cases} \operatorname{argmin}_{x \in V^1} \|x\|_{\mathrm{TV}} \text{ such that } Ax = y, & \text{if } \iota \text{ is feasible} \\ \infty & \text{otherwise .} \end{cases}$$

(vi) Image deblurring: For image deblurring, Ω is as in the BPTV case. Moreover, the parameter $\lambda > 0$ is a fixed rational number and

$$\Xi_{\mathrm{DeblurTV}}(\iota) = \operatorname*{argmin}_{x \in V^1} \|Ax - y\|_2^2 + \lambda \|x\|_{\mathrm{TV}}.$$

(vii) Neural Networks (Deep Learning): We only consider lower bounds and thus concentrate on the simplest classification functions $f : \mathbb{R}^d \to \{0, 1\}$. Given such f and a training set $\mathcal{T} = \{x^1, \ldots, x^r\} \subset \mathbb{R}^d$, the task is to compute a neural network $\phi \in \mathcal{NN}_{\mathbf{N},L,d}$ and evaluate it on a classification set $\mathcal{C} = \{y^1, \ldots, y^s\}$ We let $(\mathcal{M}, d) = (\mathbb{R}^s, \|\cdot\|_\infty)$. For a given $\delta > 0$ we let Ω be the collection of

$$\iota = \left\{ \{(\hat{x}^j, f(\hat{x}^j))\}_{j=1}^r, \{\hat{y}^j\}_{j=1}^s \right\}$$

where $\hat{x}^j \in \mathcal{B}_\delta^\infty(x^j)$, $x^j \in \mathcal{T}$, $\hat{y}^j \in \mathcal{B}_\delta^\infty(y^j)$ and $y^j \in \mathcal{C}$. Finally, for a given cost function $C \in \mathcal{CF}$, where $\mathcal{CF}$ is defined in (1.7.1)*,*

$$\Xi_{\mathrm{NN}}^C(\iota) = \{\phi(\hat{y}^j)\}_{j=1}^s, \quad \phi \in \operatorname*{argmin}_{\tilde{\phi} \in \mathcal{NN}_{\mathbf{N},L,d}} C(v, w),$$

with $v = \{\tilde{\phi}(\hat{x}^j)\}_{j=1}^r, w = \{f(\hat{x}^j)\}_{j=1}^r$.

Remark 2.10.2. For the neural network case, we note that the Ξ_{NN} is the (potentially multivalued if there is more than one minimising ϕ) map to all possible sets of the form $\{\phi(\hat{y}^j)\}_{j=1}^s$ for each ϕ that minimises the cost function. Importantly, this is different from all possible sets of the form

$$\left\{ \phi_1(\hat{y}^1), \phi_2(\hat{y}^2), \ldots, \phi_s(\hat{y}^s) \,|\, \phi_1, \phi_2, \ldots, \phi_s \in \operatorname*{argmin}_{\tilde{\phi} \in \mathcal{NN}_{\mathbf{N},L,d}} C(v, w) \right\}.$$

We emphasise the fact that the computational problem is to train a neural network and then run that fixed neural network on each of the elements in the classification set, as opposed to considering the possible outputs of all neural networks for each element in the classification set.

Remark 2.10.3. As noted earlier, since there is no reason for $\Xi(\iota)$ to have a single solution for each of the minimisation problems above, we need to account for multivalued computational problems. Thus we consider our task to find either $\Gamma(\iota) \in \Xi(\iota)$ in the Δ_0 case or to find a sequence $\Gamma_n(\iota)$ with $\inf_{\xi \in \Xi(\iota)} d(\Gamma_n(\iota), \xi) \leq 2^{-n}$ in the Δ_1 case.

As an example of how the concepts and theorems developed in this section can be used to produce computability results, let us consider linear programming on 1×1

matrices with the class $\Omega = \{\iota^n \,|\, n \geq 1\} \cup \{\iota^0\}$ where $\iota^0 = (y^0, A^0)$ and $\iota^n = (y^n, A^n)$ with $y^n = 1/4^n, A^n = 1/4^n$ and $y^0 = A^0 = 0$. A simple application of Theorem 2.9.1 shows that the problem is not in Δ_1^G (so not computable with a BSS or Turing machine). We get with no additional work that the strong breakdown epsilon must be at least 1/2: this applies for both probabilistic and deterministic algorithms. Moreover, if one could design an algorithm that can solve the problem to accuracy better than 1/2 then one can also decide the halting problem. We will use a similar approach (sometimes with significantly more complicated example choices of ι) to prove each of the non-computability results in Chapter 4.

2.11 Conditioning

Given that many of the problems discussed earlier are non-computable, it is natural to try to ask if there is a criteria that can be used to 'rule out' certain problematic inputs. An initial approach might be to use *condition numbers*. Let us take for example the problem of solving a linear system of equations. Specifically, we are given $y \in \mathbb{C}^n$ and $U \in \mathbb{C}^{n\times n}$ and wish to find x such that $Ux = y$. Of course, such a problem is ill-defined if U^{-1} does not exist. Moreover, it is easy to see that even if U^{-1} exists, if $\|U\|_2$ or $\|U^{-1}\|_2$ is large then slight perturbations in y can cause large perturbations in the output solution x. Thus one can consider *the condition number of a matrix* defined in the following way:

Definition 2.11.1 (Condition number of a matrix)**.** *The classical condition number of a matrix A is given by* $\mathrm{Cond}(A) = \|A\|\|A^{-1}\|$.

One might hope that we can use a similar form of condition number to rule out problematic inputs and thus create problems that are computable. However, the situation is far more subtle. There are a myriad of different definitions of condition, see for example the pioneering work by Renegar [102, 103] and the book by Burgisser and Cucker [23], see also [42]. In this section we shall recall some of the standard definitions of condition which depend on the computational problem at hand.

Before doing so, let us consider a simple example that we will use to inform the precise definitions of condition number in the sense of a computational problem defined in Section 2.4. Suppose that $\{\Xi, \Omega, \mathcal{M}, \Lambda\}$ is such that Ω is the set of all lower triangular matrices of dimension 2×2 and $\Xi : \mathbb{R}^{2\times 2} \to \mathbb{R}$ finds the spectrum of a matrix. We set

$$\Lambda = \{f_{i,j} \,|\, f_{i,j}(\iota) = \iota_{i,j} \text{ for } i, j \in \{1, 2\}\}, \quad \mathcal{M} = (\mathbb{R}^2, \|\cdot\|_\infty).$$

In some sense, many of the upcoming definitions of condition number are designed to capture how small perturbations affect the output of Ξ. However, any algorithm designed to solve the problem above would be uninterested in the value of $f_{1,2}$: these values would be the constant 0 whenever $\iota \in \Omega$. Thus any reasonable definition of condition number would ignore small perturbations in that direction.

We therefore define the active coordinates of Ω (for $\Omega = \mathbb{R}^d$ for some dimension d)

to be $\mathcal{A}(\Omega) = \{j \,|\, \exists\, x, y \in \Omega, x_j \neq y_j\}$. Moreover, for $\nu > 0$ we define

$$\tilde{\Omega}_\nu = \{x \,|\, \exists\, y \in \Omega \text{ such that } \|x - y\|_\infty \leq \nu, x_{\mathcal{A}^c} = y_{\mathcal{A}^c}\}$$

(noting that here and throughout the remainder of this thesis, for index sets I and vectors x the notation x_I corresponds to the vector x with all entries outside of I set to 0). In other words, $\tilde{\Omega}_\nu$ is the set of ν-perturbations along the non-constant coordinates of elements in Ω. We can now recall some of the classical condition numbers.

Definition 2.11.2. *Condition numbers*

(i) (Condition of a mapping) Let $\Xi : \Omega \subset \mathbb{C}^n \to \mathbb{C}^m$ be a linear or non-linear mapping, and suppose that Ξ is also defined on $\tilde{\Omega}_\nu$ for some $\nu > 0$. Then,

$$\text{Cond}(\Xi) = \sup_{x \in \Omega} \lim_{\epsilon \to 0^+} \sup_{\substack{x+z \in \tilde{\Omega}_\nu \\ 0 < \|z\| \leq \epsilon}} \left\{ \frac{\text{dist}(\Xi(x+z), \Xi(x))}{\|z\|} \right\}, \tag{2.11.1}$$

where we allow for multivalued functions by defining

$$\text{dist}(\Xi(x), \Xi(z)) = \min_{\tilde{x} \in \Xi(x), \tilde{z} \in \Xi(z)} \|\tilde{x} - \tilde{z}\|.$$

(ii) (Distance to infeasibility) If Ξ denotes the solution map to any of the problems in Definition 2.10.1 (i), (ii), (v) with domain Ω, we define, for $(y, A) \in \Omega$

$$\begin{aligned} \rho(A, y) = \sup\{\delta \,|\, \|\hat{A}\|, \|\hat{y}\| \leq \delta, (A + \hat{A}, y + \hat{y}) \in \tilde{\Omega}_\infty \\ \Rightarrow (A + \hat{A}, y + \hat{y}) \text{ are feasible inputs}\}, \end{aligned}$$

and this yields the Feasibility Primal (FP) condition number

$$C_{\text{FP}}(A, y) := \frac{\|A\| \vee \|y\|}{\rho(A, y)}, \tag{2.11.2}$$

where the active set is defined by identifying (y, A) with $\mathbb{R}^d$.

(iii) (Distance to solution with several minimisers) If Ξ denotes the solution map to any of the problems in Definition 2.10.1 (except (vii)) with domain Ω then we define, for $(y, A) \in \Omega$,

$$\begin{aligned} \varrho(A, y) = \sup\{\delta : \|\hat{A}\|, \|\hat{y}\| \leq \delta, (A + \hat{A}, y + \hat{y}) \in \tilde{\Omega}_\infty \\ \Rightarrow (A + \hat{A}, y + \hat{y}) \text{ yields at most one solution}\}, \end{aligned}$$

and this yields the RCC condition number

$$C_{\text{RCC}}(A, y) := \frac{\|A\|_2 \vee \|y\|_2}{\varrho(A, y)}. \tag{2.11.3}$$

where again the active set is defined by identifying (y, A) with $\mathbb{R}^d$.

Finally, for any of the problems in Definition 2.10.1 (except (vii)), we define $\|\iota\| = \max(\|A\|_2, \|y\|_2)$. We usually insist that $\|\iota\|$ is bounded for each $\iota \in \Omega$ to force the input data to be bounded.

We will see that there are limitations to how much condition can help us. In particular, if we force all the inputs to be well conditioned in the sense that C_{RCC} is small, we can still end up in a situation where our problems are not in Δ_1. If instead we insist that all the inputs $(y, A) \in \Omega$ for say, basis pursuit satisfy $\mathrm{Cond}(AA^*), C_{\mathrm{FP}}(A, y), \mathrm{Cond}(\Xi) \leq 2$, then we are still unable to get Δ_1 results. Moreover, for Basis Pursuit there are important classes of inputs (y, A) such that we can get a Δ_1 result, however, there are inputs in the class such that $C_{\mathrm{RCC}}(A, y) = \infty$.

Chapter 3

Compressed sensing: the RIP and NSP in levels

Let us consider compressed sensing as in Section 1.5. The general idea is therefore to recover x from observations $y = Ux$ where $U \in \mathbb{R}^{m \times N}$ and $m \ll N$. Under certain conditions on the matrix U, every 's-sparse' vector x (i.e. any vector x with at most s non-zero entries) can be recovered by observing y and solving the basis pursuit problem with ℓ^1. Namely, one solves $\operatorname{argmin}_{x'} \|x'\|_1 Ux' = y$ and hopes that $x = x'$ provided that x has s-non-zero entries.

If any s-sparse vectors x can be perfectly recovered in this way with the same choice of U, we say that *uniform recovery of order s is possible*. More generally if x is close to an s-sparse vector then we might expect solutions to the ℓ^1 basis pursuit problem (BP) to be close to x. We can encapsulate this statement mathematically by making the following definition:

Definition 3.0.1 (Uniform recovery of order s)**.** *Let s be a positive integer. We say that* uniform recovery of order s *is possible for the matrix U if solutions $\tilde{x}$ to* (BP) *satisfy*

$$\|x - \tilde{x}\|_1 \leq C\sigma_s(x)_1, \quad \sigma_s(x)_1 := \min\{\|x - \widehat{x}_2\|_1 \text{ such that } \widehat{x}_2 \text{ is } s\text{-sparse}\}. \quad (3.0.1)$$

for some constant C independent of x. Note that (3.0.1) *implies that all s-sparse x are recovered exactly by solving* (BP)*, since if x is s-sparse then $\sigma_s(x)_1 = 0$.*

Proving that uniform recovery of order s is possible for the matrix U is an inherently complicated task. To simplify this task, the nullspace property and Restricted Isometry Property (RIP) have been introduced (see [33] and [60] for more information). More specifically, the nullspace property is defined as follows:

Definition 3.0.2 (ℓ^2 RNSP of order s)**.** *A matrix $U \in \mathbb{C}^{m \times n}$ is said to satisfy the ℓ^2* robust nullspace property *(ℓ^2 RNSP) of order s if there is a $\rho \in (0,1)$ and a $\tau > 0$ such that for all vectors $v \in \mathbb{C}^n$ and all S which are subsets of $\{1, 2, 3, \ldots, n\}$ with $|S| \leq s$, we have $\|v_S\|_2 \leq \rho \|v_{S^c}\|_1/\sqrt{s} + \tau \|Uv\|_2$.*

The Restricted Isometry Property (RIP) is defined in terms of the Restricted Isometry Constant δ_s. A matrix is said to have the RIP if $\delta_s < 1$.

Definition 3.0.3 (Restricted Isometry Property)**.** *The* Restricted Isometry Constant *(RIC) of order s for a matrix $U \in \mathbb{C}^{m\times n}$, denoted by δ_s, is the minimal $\delta \geq 0$ such that*

$$(1-\delta)\|x\|_2^2 \leq \|Ux\|_2^2 \leq (1+\delta)\|x\|_2^2 \tag{3.0.2}$$

for all s-sparse vectors $x \in \mathbb{C}^n$.

It is well known (e.g. [7] and [60]) that if U satisfies the ℓ^2 RNSP or the RIC of order s is sufficiently small, then equation (3.0.1) is satisfied when finding minimisers $\tilde{x}$ of the basis pursuit problem (BP).

Remark 3.0.4. In fact, the RIP and RNSP both imply a stronger result - suppose that instead of seeing Ux, we see a noisy version $v := Ux + \nu$ for some noise vector ν with $\|\nu\|_2 \leq \epsilon$. Instead of finding minimisers of (BP), we can try to recover x by finding minimisers to the basis pursuit denoising problem (BPDN) (so that x itself is a feasible solution). Then any minimiser $\tilde{x}$ to (BPDN) will satisfy both

$$\|x - \widetilde{x}\|_1 \leq C\sigma_s(x)_1 + D\epsilon\sqrt{s} \tag{3.0.3}$$

$$\|x - \widetilde{x}\|_2 \leq \frac{C\sigma_s(x)_1}{\sqrt{s}} + D\epsilon \tag{3.0.4}$$

for some universal constants C and D provided that U satisfies the ℓ^2 RNSP (e.g. [60], Theorem 4.22) or δ_s is sufficiently small (e.g. [26,27,30,59] and [28,48,119] for optimal conditions). In later chapters we shall introduce the ℓ^2 RNSP in levels and the RIP in levels which will also have a similar resilience to noise.

Given the recent substantial interest in uniform recovery it is natural to ask whether this intriguing mathematical concept is actually observed in many of the applications where CS is applied. Certain conditions on U, like the Restricted Isometry Property (RIP) (see Definition 3.0.2) and the nullspace property of order s (see Definition 3.0.2) imply uniform recovery of order s. However, for general matrices U it is difficult to check that these properties hold. Indeed, it is shown in [117] that verifying that the RIP holds (and thus order s uniform recovery is possible) for general U is an NP hard problem. However, there is a simple test that can be used to show that certain matrices cannot achieve uniform recovery of order s for reasonable values of s. This is called the *flip test*. As this test reveals, there are a significant number of practical applications where uniform recovery is not the correct model for compressed sensing. This list of applications includes Magnetic Resonance Imaging (MRI) [62, 86], other areas of medical imaging such as Computerised Tomography (CT) [40, 65], Nuclear Magnetic Resonance (NMR) [71], Electron Tomography [61,81], as well as other fields such as Fluorescence microscopy [104, 114], Surface scattering such as Helium Atom Scattering (HAS) [73] and Radio interferometry [89].

We will thoroughly document the lack of uniform recovery of order s in this paper, and explain why it does not hold for reasonable s in many applications. It is then natural to ask whether there might be an alternative to uniform recovery of order s that may be more suitable for the actual real world CS applications. With this in mind, we shall generalise uniform recovery of order s to a level based uniform recovery, which

we term *uniform recovery of order* $(\mathbf{s}, \mathbf{M})$. Numerical experiments will suggest that uniform recovery of order $(\mathbf{s}, \mathbf{M})$ is better suited to many of the applications where CS is used than uniform recovery of order s. We will extend the concepts of the RIP and the nullspace property to this setting with the introduction of the *RIP in levels* and the ℓ^2 robust nullspace property of order $(\mathbf{s}, \mathbf{M})$.

3.1 The absence of the uniform recovery and the flip test

3.1.1 The flip test

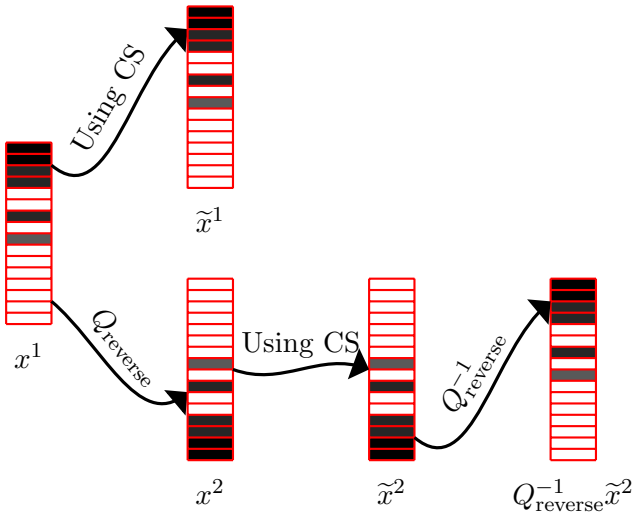


Figure 3.1: A graphical demonstration of the flip test for matrices which exhibit the uniform recovery where x^1 is a vector exactly recovered by minimisers of (BP). Darker colours denote larger values. If uniform recovery of a sufficiently high order holds, then $Q_{\text{reverse}}^{-1}\widetilde{x}^2 = \widetilde{x}^1$.

Although uniform recovery seems convenient, it is in general very difficult to verify that uniform recovery of order s is possible for a matrix U. In fact, showing that the RIC of an arbitrary matrix is below a certain value is an NP hard problem [117]. However, some special cases for U do exhibit uniform recovery of order s (e.g. with high probability, Gaussian and Bernoulli matrices can achieve uniform recovery [34]). Even though it is hard to show that uniform recovery is possible for a general matrix U, there is a simple test (the 'flip test', introduced in [2]) that shows that there are a variety of matrices used in practical applications for which uniform recovery grossly underestimates the effectiveness of compressed sensing.

Flip test (sparse vectors):

Suppose we are given $U \in \mathbb{C}^{m \times n}$, $s_1 \in \mathbb{N}$ and an s_1-sparse vector x^1 that is perfectly recovered by finding minimisers of the BP problem (BP) using U and $x = x^1$. We now want to test if this recovery is uniform.

1. Let Q be an operator that permutes the entries of x^1 and let $x^2 = Qx^1$. Run the BP problem (BP) with $x = x^2$ to try to recover x^2 from Ux^2 and obtain

a minimiser $\tilde{x}^2$. Compare x^1 and $Q^{-1}\tilde{x}^2$. If $x^1 \neq Q^{-1}\tilde{x}^2$ then we do not have uniform recovery of order s_1.

2. If the test in the first step failed, we want to test how far we were from uniform recovery of order s_1. We want to see how many coefficients s_2 of x^2 one could hope to recover uniformly. Select a nonzero coefficient of x^2 and set it to zero and call this new vector h^1. If we recover h^1 by using BP with $x = h^1$ in (BP), then set $s_2 = s_1 - 1$. If not set n non-zero coefficients of x^2 to zero to obtain h^n and repeat until h^n is recovered exactly by using BP with $x = h^n$ in (BP). Let

$$s_2 = s_1 - n.$$

3. If the first step succeeds, retry it with many different permutation matrices Q. If this succeeds for a large variety of such Q then this is an indicator (but not a mathematical proof) that we may have uniform recovery.

The particular choice of Q that was given in [2] was the permutation Q_{reverse} that reverses order - namely, if $x \in \mathbb{C}^n$ then

$$Q_{\text{reverse}}(x_1, x_2, \ldots, x_{n-1}, x_n) = (x_n, x_{n-1}, \ldots, x_2, x_1).$$

A graphical demonstration and summary of the expected results of the flip test with a matrix U that exhibits uniform recovery is given in Figure 3.1.

We have performed the flip test on Fourier and Hadamard matrices in combinations with wavelet transforms. In particular, the U used in the test is of the form

$$U = P_\Omega \text{DFT} \cdot \text{DWT}_N^{-1} \quad \text{or} \quad U = P_\Omega \text{HAD} \cdot \text{DWT}_N^{-1}$$

for different successful sampling patterns Ω. The notation DFT, HAD and DWT_N is used throughout this chapter to represent the Discrete Fourier Transform, the Hadamard Transform and the Discrete Wavelet Transform (with Daubechies wavelets with N vanishing moments) respectively. These different types of matrices are represented in a variety of applications including (but not limited to) MRI [62, 86], Radio Interferometry [89], Helium Atom Scattering [73], Electron Tomography [61, 81], CT [40, 65], Fluorescence Microscopy [104, 114], and NMR [71].

Image	Operator	s_1	s_2
College 1	$\text{DFT}\cdot\text{DWT}_3^{-1}$	121,923	329
College 2	$\text{DFT}\cdot\text{DWT}_4^{-1}$	1,850,917	143
College 3	$\text{HAD}\cdot\text{DWT}_2^{-1}$	167,772	4

Table 3.1: A table displaying the number of non-zeros that are recovered exactly by various operators. s_1 represents the number of non-zeros that can be recovered from a standard image, whereas s_2 represents the non-zeros recoverable after flipping.

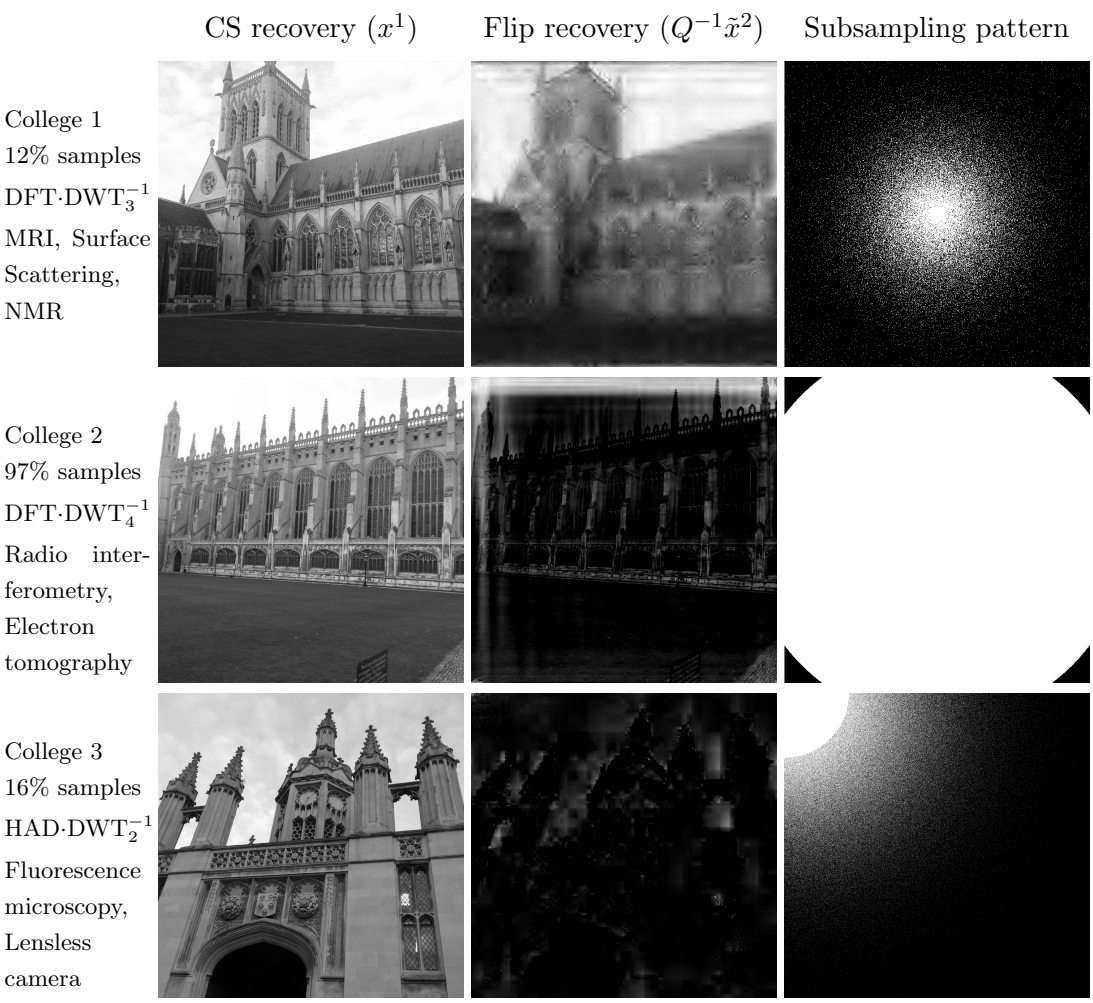
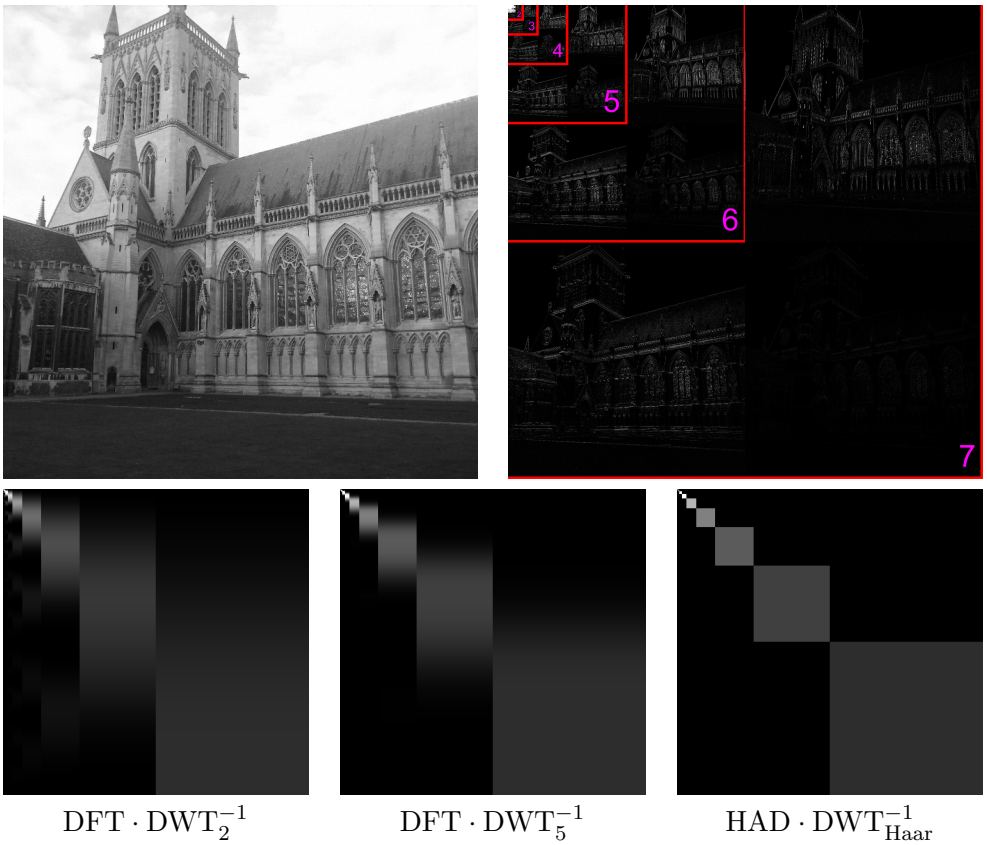


Figure 3.2: Results of the flip test for different compressed sensing matrices frequently used in applications.

In Figure 3.2 and Table 3.1 we have displayed the results of the flip test. Note the failure of uniform recovery displayed visually in Figure 3.2. More quantitatively, observe the substantial differences between s_1 and s_2 in Table 3.1. It is worth noting that even with 97% sampling as in the second row of Figure 3.2, there is still a vast difference between $\tilde{x}^1$ and $Q^{-1}x^2$. Although this may seem surprising at first, this is a consequence the near block diagonal structure of the matrix $\mathrm{DFT} \cdot \mathrm{DWT}^{-1}$ (see Figure 3.3 and Remark 3.1.2). The high Fourier frequencies (which, due to the block diagonality, correspond to the finer detail wavelet coefficients) are heavily subsampled since the finer detail coefficients are highly sparse. However, when the wavelet coefficients are flipped, we are now subsampling the Fourier frequencies corresponding to the non-sparse coarse wavelet coefficients. Thus the recovery is poor and we get the results of Figure 3.2. Note that the flip test will fail in a similar manner if we replace wavelets with other popular frames such as curvelets, contourlets or shearlets [31, 49, 79]. We thus need to consider more structure than just sparsity to explain the success of compressed sensing in these applications.



$\mathrm{DFT} \cdot \mathrm{DWT}_2^{-1}$ $\mathrm{DFT} \cdot \mathrm{DWT}_5^{-1}$ $\mathrm{HAD} \cdot \mathrm{DWT}_{\mathrm{Haar}}^{-1}$

Figure 3.3: First row: an image and its wavelet coefficients, where a brighter colour corresponds to a larger value. Second row: absolute values of a variety of compressive sensing matrices. The block diagonal structure allows us to fully sample rows that correspond to the coarser wavelet levels and subsample the rows that correspond to the finer wavelet levels.

Remark 3.1.1 (Sparsity cannot be the right model)**.** The flip test reveals that sparsity cannot be the correct model for these examples in compressed sensing. When the values of s_1 and s_2 are (for example)

$$s_1 = 121{,}923, \qquad s_2 = 329,$$

it is hard to argue that one recovers s-sparse vectors for a representative s when the location of the non-zero coefficients is arbitrary. On the contrary, as the flip test reveals, the location of the non-zero coefficients is highly important. If sparsity is not the correct model one needs to revise the model in order to find a more realistic description. Moreover, the concepts of the nullspace property of order s and the RIP no longer apply if sparsity is not the correct model for compressed sensing. Of course, it could be the case that the RIC of order 329 is sufficiently small to allow uniform recovery of order 329, however, that has nothing to do with the successful recovery of the image with $s = 121{,}923$ non-zero wavelet coefficients.

Remark 3.1.2 (Large coherence and almost block diagonality)**.** One can understand

the lack of uniform recovery from simply looking at Figure 3.3. The blocks in the matrices correspond to the different scales in the wavelet expansion that give a level structure. The wavelet coefficients get relatively sparser in the finer levels and this corresponds to the blocks in the matrices where the absolute values (coherence, see Definition 3.2.8) decrease. The coherence is not uniformly small. In fact, it is very big in the upper left corner and then decreases with the levels. As is well known [60], it is the uniform small coherence that is the key property to prove uniform recovery. It should be noted that there have been attempts in applications to change the measurements in for example MRI and Radio interferometry, in order to make the coherence smaller in the first levels. This is called the spread spectrum technique [98, 99].

3.1.2 Weighted sparsity

Consideration of a different explanation for the success of compressed sensing that includes more structure than just plain sparsity is not a novel idea. Indeed, *weighted sparsity* and the *weighted RIP* was described in [101] as a structured alternative to sparsity and the RIP. To describe this approach, we shall begin by defining *weighted sparsity*. More specifically, given a collection of weights $\omega := (\omega_1, \omega_2, \dots, \omega_n) \in \mathbb{R}^n$ with $\omega_j \geq 1$ for each j, a vector $x \in \mathbb{C}^n$ is said to be (ω, s)-weighted sparse if the weighted ℓ^0 norm, $\|x\|_{\omega,0} := \sum_{j \in \mathrm{supp}(x)} \omega_j^2$, satisfies $\|x\|_{\omega,0} < s$. We can similarly extend the ℓ^1 norm to a weighted ℓ^1 norm by defining $\|x\|_{\omega,1} := \sum_{j=1}^n \omega_j x_j$ and then examine *weighted ℓ^1 minimisation* in the same way that we can discuss ℓ^1 minimisation. A preliminary idea to deal with the difficulties raised in Section 3.1.1 is to argue that instead of expecting uniform recovery of order s as in equation (3.0.1) to hold whenever $\tilde{x}$ is a minimiser of (BP), we should hope for uniform recovery of order (ω, s) to hold. More specifically,

$$\|\tilde{x} - x\|_1 \leq C\sigma_{\omega,s}(x)_1 \tag{3.1.1}$$

where $\sigma_{\omega,s}(x)_1 := \min\{\|x - \widehat{x}_2\|_1 \text{ such that } \widehat{x}_2 \text{ is } (\omega, s)\text{-weighted sparse}\}$ and C is a fixed constant. This is further motivated by the success of such an approach to the recovery of smooth functions from undersampled measurements [101] and the improvements seen by applying weighted ℓ^1 techniques to random Gaussian matrices [76].

3.1.2.1 The insufficiency of uniform recovery of weighted sparse vectors through ℓ^1 minimisation with wavelets

Unfortunately, there are issues with this method when applied to problems involving a level based construction basis such as wavelets like in Section 3.1.1. These are more thoroughly documented in [1], but we shall provide a brief outline here. Just as the flip test demonstrates that in many examples relevant to practical applications the class of s-sparse vectors is too big and contains objects that cannot be recovered by ℓ^1 minimisation, we have the same phenomenon for weighted sparsity. We find that for problems involving a level based reconstruction basis, and for any choice of weights ω, the class of (ω, s)-sparse vectors is too large and contains vectors that cannot be recovered by either weighted-ℓ^1 or ℓ^1 minimisation.

In our specific setting above, this means that we have a 'natural' image with wavelet coefficients w that is recovered exactly and a vector w' with $\|w'\|_{(\omega,0)} \leq \|w\|_{(\omega,0)}$ which is not recovered. Therefore, either $\|w\|_{(\omega,0)} > s$ and w is not (ω, s)-sparse (so a theory based on weighted sparsity does not explain why w is recovered) or w' is (ω, s)-sparse (but not recovered, so that the class of (ω, s)-sparse vectors is too large and inequality (3.1.1) does not hold).

We can show these results by expanding the 'flip test' from Section 3.1.1. The result is the *flip test for weighted sparse vectors*:

Flip test (weighted sparse vectors):

Suppose we are given $U \in \mathbb{C}^{m\times n}$, a collection of weights $\omega \in \mathbb{C}^n$ and a vector $x^1 \in \mathbb{C}^n$ that is perfectly recovered by finding a minimiser of the BP problem (BP) using U and $x = x^1$. Set s to be the minimal value so that x^1 is (ω, s)-sparse. We now want to test if this recovery is uniform across all (ω, s)-weighted sparse vectors.

1. Let Q be an operator that permutes the entries of x^1 and let $v = Qx^1$. Repeatedly set individual coefficients of v to be 0 until v is also (ω, s)-sparse. Call this new vector x^2.

2. Run the BP problem (BP) with $x = x^2$ to try to recover x^2 from Ux^2 and obtain a minimiser $\tilde{x}^2$.

3. If $\tilde{x}^2$ is not recovered exactly with this method, we do not have uniform recovery of (ω, s)-weighted sparse vectors.

4. Retry steps 1 to 3 with many different permutation matrices Q. If this succeeds over for a large variety of such Q then this is an indicator (but not a mathematical proof) that we may have uniform recovery. A single failure, however, demonstrates that we do not have uniform recovery of (ω, s)-weighted sparse vectors.

Figure 3.4 displays some examples where the flip test implies a lack of uniform recovery of weighted sparse vectors (either because s is too small to explain the perfect recovery observed or s is too large and there are too many vectors that are (ω, s)-weighted sparse as in the previous discussion). Thus, weighted sparsity is insufficient to explain the success of compressed sensing when using wavelets and other X-lets. In Figure 3.4 we have only displayed the result of using ℓ^1 recovery, however, the results are the same when using weighted ℓ^1. See [1] for a thorough discussion of this phenomenon. We shall provide additional insight as to why weighted sparsity is insufficient in Section 3.2.1.4.

Remark 3.1.3. It must be emphasised that weighted sparsity and the weighted RIP were developed in [101] for the purpose of recovering smooth functions with polynomials. Thus, one should not expect the weighted RIP to hold for wavelets. Conversely, the RIP in levels may not work for polynomials, as unlike wavelets there is no level structure present. Moreover, in [76] the weighted approach is used in combination with random Gaussian measurements, which is very different from the setup in this paper. These facts demonstrates the finesses of compressed sensing theory and that we are

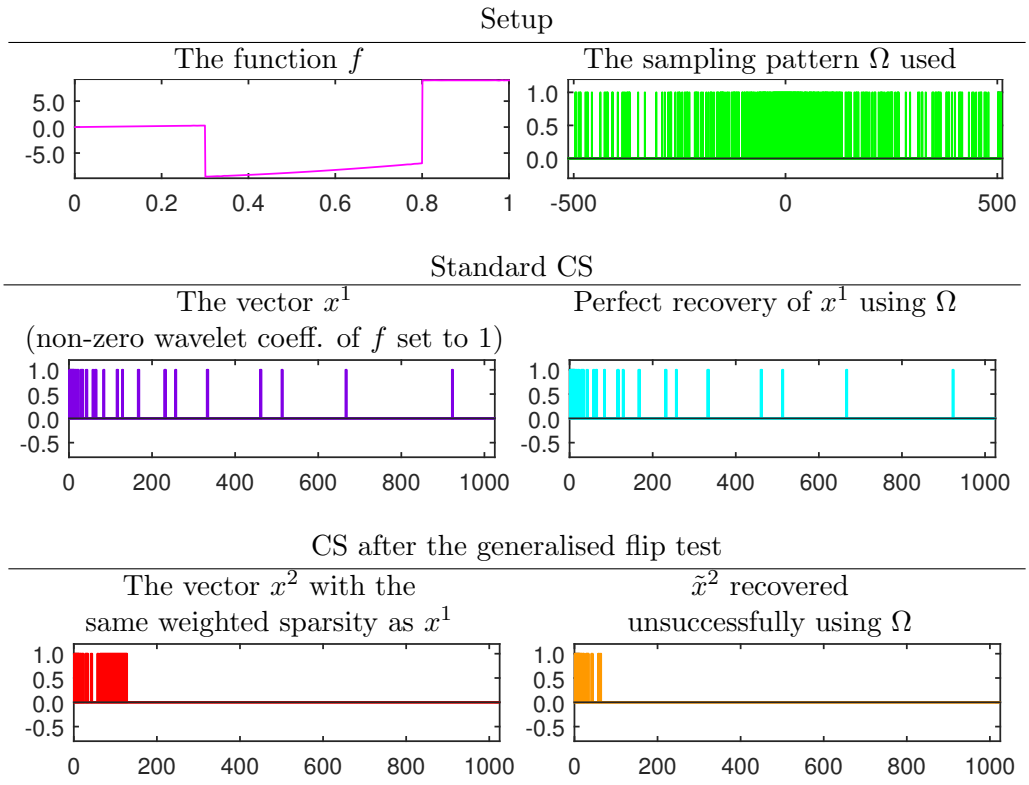


Figure 3.4: The figure displays the flip test for weighted sparse vectors with the function $f(x) = \sin(x)\mathbb{1}_{[0,0.3]} - 10\cos(x)\mathbb{1}_{(0.3,0.8]} + 9\mathbb{1}_{(0.8,1]}$ after its wavelet coefficients are thresholded so that perfect recovery is possible. Recovery was done using a subsampled 1D fourier to wavelet matrix, with Daubechies 3 wavelets and ℓ^1 minimisation. The weights on the coefficients in level i were given by 2^i. Similar results follow for other weights and also for recovery with weighted ℓ^1 minimisation.

in need for a collection of much more specific theorems using different sparsity models that depend on the problem.

Remark 3.1.4. The matrices discussed in Section 3.1.1 and Section 3.1.2.1 focused on matrices that can be used to solve finite dimensional models of the real world compressive sensing applications. In some circumstances, it has been shown that this does not match the original infinite dimensional problem and a different finite dimensional approximation is needed [3, 4, 20, 62, 63, 97]. It should be noted that the preceding flip tests could easily be adapted to this infinite dimensional setting, and thus uniform recovery of either s-sparse vectors or (ω, s)-weighted sparse vectors will still be unattainable for descriptive values of s.

3.2 An extended theory for compressed sensing

The current mathematical theory for compressive sensing revolves around a few key ideas. These are the concepts of sparsity, incoherence, uniform subsampling and uni-

form recovery of order s. In [4] and [104], it was shown that these concepts are absent for a large class of compressed sensing problems. To solve this problem, the extended concepts of *asymptotic sparsity*, *asymptotic incoherence* and *multi-level sampling* were introduced. We now introduce the fourth extended concept in the new theory of compressive sensing: uniform recovery of order $(\mathbf{s}, \mathbf{M})$. To accomplish this, we shall extend the definitions of nullspace property and restricted isometry property of order s to a pair of new concepts - the RIP in levels and the nullspace property of order $(\mathbf{s}, \mathbf{M})$.

3.2.1 A level based alternative to sparsity: $(\mathbf{s}, \mathbf{M})$-sparsity

The examples given in Figure 3.2 all involve reconstructing in a basis that is divided into various levels. It is this level based structure that prevents us from observing uniform recovery of order s and necessitates a new theory based on a different kind of sparsity. We shall demonstrate this new theory with wavelets, which we give a brief description of in the following section. Despite our focus on wavelets in the next few pages, it should be noted that our work applies equally to all level based reconstruction bases.

3.2.1.1 Wavelets

A multiresolution analysis (as defined in [44,45,85]) for $L^2(X)$ (where X is an interval or a square) is formed by constructing increasing linear spaces $(V_j)_{j=0}^{\infty}$ (known as the *scaling spaces*) and linear spaces $(W_j)_{j=0}^{\infty}$ (known as the *wavelet* spaces) with $V_j, W_j \subset L^2(X)$ so that

1. If $f(\cdot) \in V_j$ then $f(2\cdot) \in V_{j+1}$, and vice-versa.
2. $\overline{\bigcup_{j=0}^{\infty} V_j} = L^2(X)$ and $\bigcap_{j=0}^{\infty} V_j = V_0$ is the space of all constant functions on X.
3. W_j is the orthogonal complement of V_j in V_{j+1}.

The wavelet expansion of a function f is an expansion in terms of basis elements of V_0 (the *scaling level*) and W_j for $j \geq 0$ (the *wavelet levels*, said to be increasingly fine as j increases). For natural images f, the largest coefficients in the wavelet expansion of f appear in the levels corresponding to smaller j (the *coarser levels*). Closer examination of the relative sparsity in each level also reveals a pattern: let w be the collection of wavelet coefficients of f and for a given level k let S^k be the indices of all wavelet coefficients of f in the kth level. Additionally, let $\mathcal{M}_n$ be the largest (in absolute value) n wavelet coefficients of f. Given $\epsilon \in [0, 1]$, we define the functions $s(\epsilon)$ and $s_k(\epsilon)$ (as in [4]) by

$$s(\epsilon) := \min \left\{ n : \|w_{M_n}\|_2 = \sqrt{\sum_{i \in \mathcal{M}_n} |w_i|^2} \geq \epsilon \sqrt{\sum_{k \in \mathbb{N}} \sum_{i \in S^k} |w_i|^2} = \epsilon \|w\|_2 \right\},$$

$$s_k(\epsilon) := |\mathcal{M}_{s(\epsilon)} \cap S^k|.$$

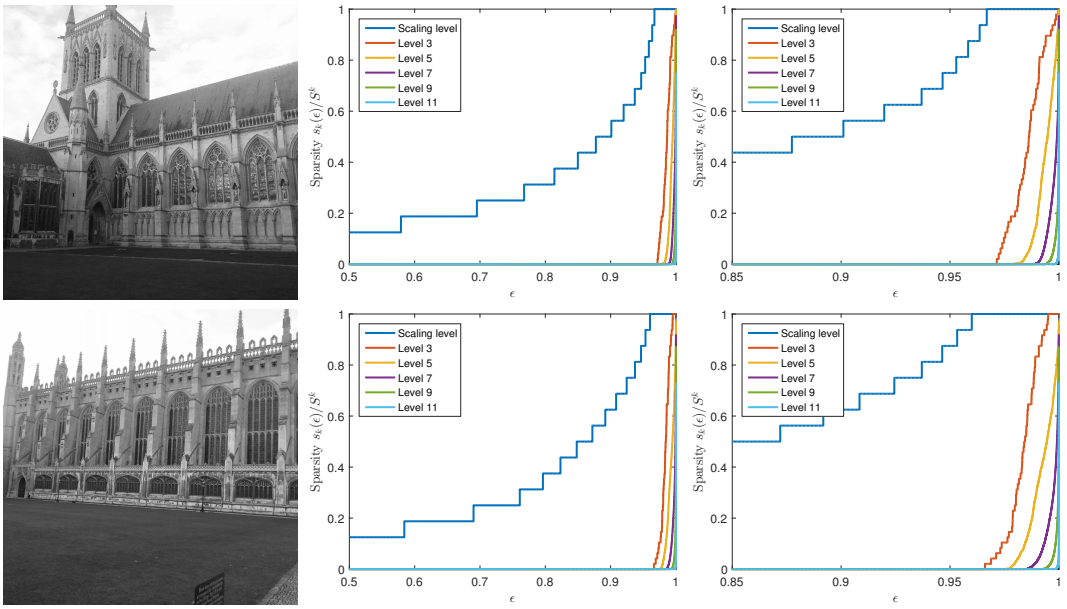


Figure 3.5: The relative sparsity of Haar wavelet coefficients of two image. The leftmost column displays the image in question. The middle and final columns display the values of $s_k(\epsilon)$ for $\epsilon \in [0.5, 1]$ and $\epsilon \in [0.85, 1]$ respectively, where k represents a wavelet level. Of particular importance is the rapid decay of $s_k(\epsilon)$ as k grows larger. 'Scaling level' denotes the case where k corresponds to the scaling level.

More succinctly, $s_k(\epsilon)$ represents the relative sparsity of the wavelet coefficients of f at the kth scale. If an image is very well represented by wavelets, we would like $s_k(\epsilon)$ to be as small as possible for ϵ close to 1. However, one can make the following observation: then the ratios $s_k(\epsilon)/|S^k|$ decay very rapidly for a fixed ϵ. Numerical examples showing this phenomenon with Haar Wavelets are displayed in Figure 3.5. Summarising, we observe that images taken from the real world are sparse with a structure which the traditional RIP ignores.

3.2.1.2 $(\mathbf{s}, \mathbf{M})$-sparsity and uniform recovery of order $(\mathbf{s}, \mathbf{M})$

Uniform recovery of order s suggests that we are able to recover all s-sparse vectors exactly, independent of which levels the s-sparse vectors are primarily supported on. Instead of such a stringent requirement, we can take advantage of the structure of our problem, a concept that is already popular from the recovery point of view [8, 67, 68, 118]. We have observed that, for wavelets, $s_k(\epsilon)/|S^k|$ decays rapidly as $k \to \infty$ (see Figure 3.5). To further understand this phenomenon, in [4] the concept of $(\mathbf{s}, \mathbf{M})$-sparsity was introduced.

Definition 3.2.1 ($(\mathbf{s}, \mathbf{M})$-sparsity)**.** *Let* $\mathbf{M} = (M_0, M_1, \ldots, M_l) \in \mathbb{N}^{l+1}$ *with* $1 \leq M_1 < M_2 < \cdots < M_l$ *and* $M_0 = 0$*, where the natural number* l *is called the* number of levels*. Additionally, let* $\mathbf{s} = (s_1, s_2, \ldots, s_l) \in \mathbb{N}^l$ *with* $s_i \leq M_i - M_{i-1}$*. We call* $(\mathbf{s}, \mathbf{M})$ *a* sparsity pattern*.*

A set Λ of integers is said to be $(\mathbf{s},\mathbf{M})$-sparse if $\Lambda \subset \{M_0+1, M_0+2, \dots, M_l\}$ and for each $i \in \{1,2,\dots,l\}$, we have $|\Lambda \cap \{M_{i-1}+1, M_{i-1}+2, \dots, M_i\}| \leq s_i$. A vector is said to be $(\mathbf{s},\mathbf{M})$-sparse if its support is an $(\mathbf{s},\mathbf{M})$-sparse set. The collection of $(\mathbf{s},\mathbf{M})$-sparse vectors is denoted by $\Sigma_{\mathbf{s},\mathbf{M}}$. We can also define $\sigma_{\mathbf{s},\mathbf{M}}(x)_1$ as a natural extension of $\sigma_s(x)_1$. Namely, $\sigma_{\mathbf{s},\mathbf{M}}(x)_1 := \min_{\widehat{x}\in\Sigma_{\mathbf{s},\mathbf{M}}} \|x-\widehat{x}\|_1$.

Remark 3.2.2. If $(\mathbf{s},\mathbf{M})$ is a sparsity pattern, we will sometimes refer to $(a\mathbf{s},\mathbf{M})$-sparse sets for some natural number a even though as_i may be larger than $M_i - M_{i-1}$. To make sense of such a statement, we define (in this context)

$$a\mathbf{s} := (\min(as_1, M_1 - M_0), \min(as_2, M_2 - M_1), \dots, \min(as_l, M_l - M_{l-1}))\,.$$

Let us now look at a specific case where $(\mathbf{s},\mathbf{M})$ represent wavelet levels (again, we emphasise that wavelets are simply one example of a level based system and that our work is more general). Roughly speaking, we can choose $\mathbf{s}$ and $\mathbf{M}$ (we set $\mathbf{M}$ so that $M_{i-1}+1$ is the first index for the i-th wavelet level)such that x is $(\mathbf{s},\mathbf{M})$-sparse if it has fewer non-zero coefficients in the finer wavelet levels. As with uniform recovery of order s, we ask for minimisers $\tilde{x}$ to (BP). Instead of asking for (3.0.1), we might expect

$$\|x-\tilde{x}\|_1 \leq C\sigma_{\mathbf{s},\mathbf{M}}(x)_1 \tag{3.2.1}$$

for some C independent of x. If these conditions are satisfied then we say that *uniform recovery of order $(\mathbf{s},\mathbf{M})$ is possible for the matrix U.*

3.2.1.3 The flip test in levels

In Section 3.1.1, we saw that there was a simple test that uniform recovery of order s is not an accurate explanation for why compressed sensing is effective with some matrices U. However, the argument in Section 3.1.1 does not apply if we expect uniform recovery of order $(\mathbf{s},\mathbf{M})$ instead of uniform recovery of order s (since equation (3.0.1) will no longer hold for minimisers $\tilde{x}$ of (BP)).

Flip test in levels ($(\mathbf{s},\mathbf{M})$-sparse vectors):

Suppose we are given $U \in \mathbb{C}^{m\times n}$, a sparsity pattern $(\mathbf{s},\mathbf{M})$ and an $(\mathbf{s},\mathbf{M})$-sparse vector x^1 that is perfectly recovered by finding a minimiser of the BP problem (BP) using U and $x = x^1$. We now want to test if this recovery is uniform.

1. Let Q be a randomly chosen permutation with $Q(\Sigma_{\mathbf{s},\mathbf{M}}) = \Sigma_{\mathbf{s},\mathbf{M}}$ and let $x^2 = Qx^1$. Run the BP problem (BP) with $x = x^2$ to try to recover x^2 from Ux^2 and obtain a minimiser $\tilde{x}^2$. Compare x^1 and $Q^{-1}\tilde{x}^2$. If $x^1 \neq Q^{-1}\tilde{x}^2$ then we do not have uniform recovery of order $(\mathbf{s},\mathbf{M})$ (since x^2 is $(\mathbf{s},\mathbf{M})$-sparse).

2. If the first step succeeds, retry it with many different permutation matrices Q satisfying $Q(\Sigma_{\mathbf{s},\mathbf{M}}) = \Sigma_{\mathbf{s},\mathbf{M}}$. If this succeeds over for a large variety of such Q then this is an indicator (but not a mathematical proof) that we may have uniform recovery.

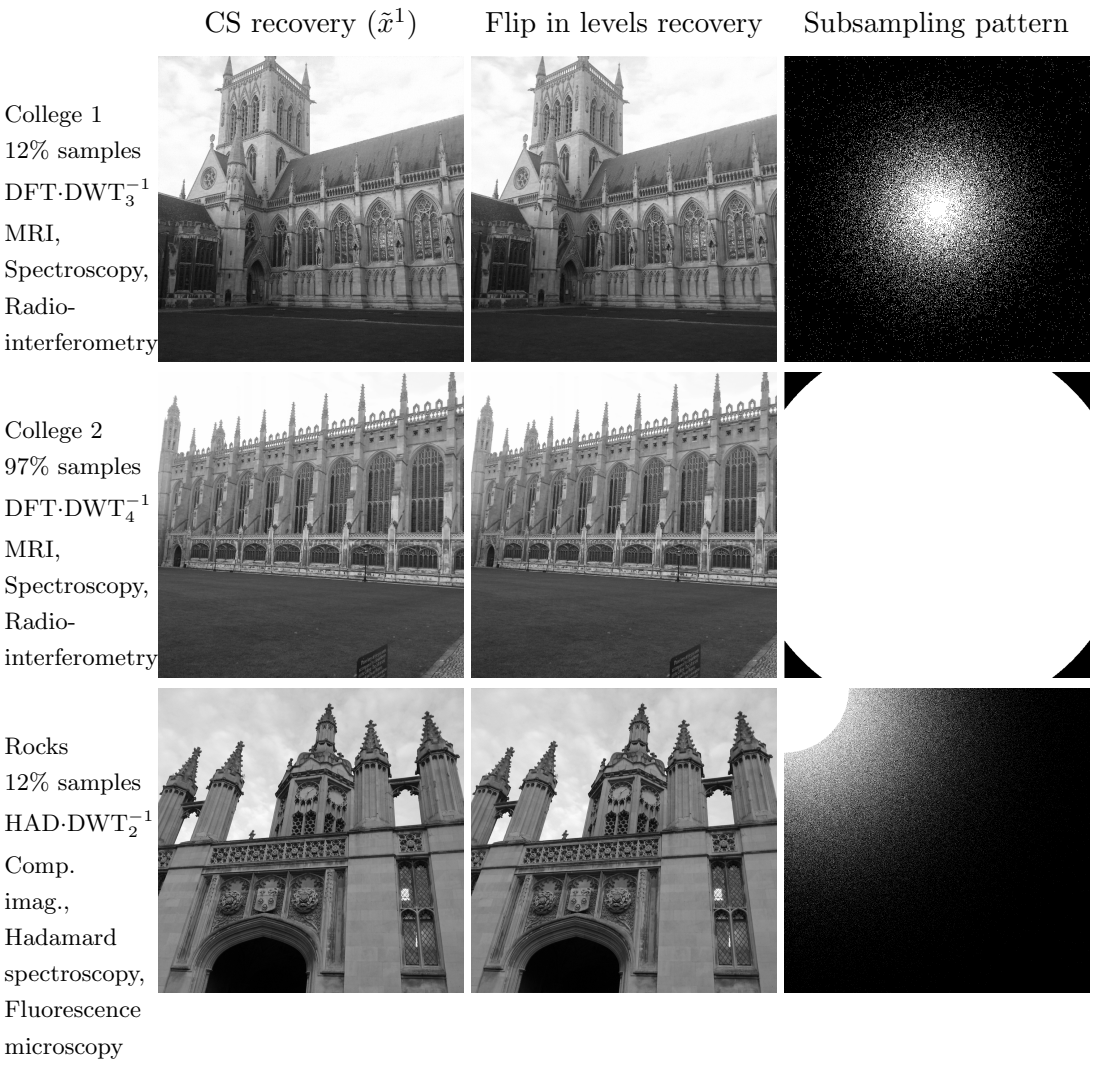


Figure 3.6: Results of the flip test when the flipping preserves the sparsity within the levels.

The requirement that $Q(\Sigma_{\mathbf{s},\mathbf{M}}) = \Sigma_{\mathbf{s},\mathbf{M}}$ now requires us to consider different permutations than a simple reverse permutation as in Section 3.1.1. A natural adaptation of Q_{reverse} to this new 'flip test in levels' is a permutation that just reverses coefficients within each wavelet level. Figure 3.6 displays what happens when we attempt to do the flip test with this permutation. In this case, we see that the performance of CS reconstruction under flipping and the performance of standard CS reconstruction are very similar. This suggests that uniform recovery within the class of $(\mathbf{s}, \mathbf{M})$-sparse vectors (as in 3.2.1) is possible with a variety of practical compressive sensing matrices. Indeed, in Table 3.2 we also consider a collection of randomly generated Q with $Q(\Sigma_{\mathbf{s},\mathbf{M}}) = \Sigma_{\mathbf{s},\mathbf{M}}$. We see that perfect recovery of Qx^1 is possible for a wide variety of permutation matrices Q.

Remark 3.2.3. Throughout this chapter we have used a variety of numerical tests to demonstrate that in a collection of compressive imaging applications the concepts of

Table 3.2: Flip test in levels with randomly generated permutations

Image	Subsampling percentage	Matrix	Permutations where x^2 was was perfectly recovered	Permutations where x^2 was was not perfectly recovered
College 1	12.48%	$\mathrm{DFT} \cdot \mathrm{DWT}_3^{-1}$	1000	0
College 2	97.17%	$\mathrm{DFT} \cdot \mathrm{DWT}_4^{-1}$	1000	0
College 3	15.54%	$\mathrm{HAD} \cdot \mathrm{DWT}_2^{-1}$	1000	0

A table displaying the flip test in levels for various images x^1 as in Figure 3.6 permuted using Q to form $x^2 = Qx^1$. Each image was processed with a fixed subsampling pattern and 1000 randomly generated permutations as described in Section 3.2.1.3.

uniform recovery of sparse vectors or uniform recovery of weighted sparse vectors are not appropriate to explain the success of compressed sensing. By contrast, there is evidence (like the flip test in levels) to suggest that uniform recovery of $(\mathbf{s}, \mathbf{M})$-sparse vectors is the right model to explain why compressive imaging works with applications using matrices such as $\mathrm{DFT} \cdot \mathrm{IWT}$ or $\mathrm{HAD} \cdot \mathrm{IWT}$. Further detail on this claim is provided in [1].

3.2.1.4 Relating $(\mathbf{s}, \mathbf{M})$-sparsity and weighted sparsity

The 'flip test in levels' suggests that for many compressed sensing problems, there are $\mathbf{s}$ and $\mathbf{M}$ such that all $(\mathbf{s}, \mathbf{M})$-sparse vectors are recovered equally well by ℓ^1 minimisation. With this in mind, we are now in a position to provide additional details on why the same is not the case for weighted sparsity. Indeed, one can easily state and prove the following theorem (see [1] for details):

Theorem 3.2.4 (The relationship between weighted sparsity and $(\mathbf{s}, \mathbf{M})$-sparsity)**.** *Let $(\mathbf{s}, \mathbf{M})$ have l levels (with $l > 2$) and fix $r < l$. Suppose that the collection of $(\mathbf{s}, \mathbf{M})$-sparse vectors are all (ω, X)-weighted sparse for some X. Then there is an l_0 with $r < l_0 < l$ such that the collection of $(\tilde{\mathbf{s}}, \mathbf{M})$-sparse are also (ω, X)-weighted sparse, where*

$$\tilde{\mathbf{s}} = (\underbrace{s_1, s_2, \ldots, s_r}_{r}, \underbrace{0, 0, \ldots, 0}_{l_0 - 1 - r}, (l-r)s_{l_0}, 0, \ldots, 0).$$

In particular, the set of $(\mathbf{s}, \mathbf{M})$-sparse and (ω, X)-weighted sparse vectors are not the same.

The use of this theorem becomes apparent if we consider Figure 3.4. As in the second row of Figure 3.3, the Fourier to Wavelet matrix in Figure 3.4 is well approximated by block diagonal matrices. This block diagonality structure means that we can design our sampling pattern so that information corresponding to coarser wavelet levels is more readily captured than the information corresponding to the finer wavelet

levels. Since the finer wavelet levels are relatively more sparse (see the first row of Figure 3.3 and Figure 3.5) we can design a sampling pattern to effectively capture images (note however that this variable density/multilevel sampling schemes have been discussed in [4,16,39,78,86,100,125] and structured sampling in [35,124]). To utilise these ideas we choose a sampling pattern so that the first r levels will be fully sampled, but after that subsampling occurs and this is where we run into difficulties with weighted sparsity. If we suppose that recovering all vectors with s_k non zero coefficients in the indices corresponding to the kth wavelet level takes Ω_k measurements in that level, then recovering all weighted sparse vectors requires $(l-r)\Omega_k$ measurements for some k. Unfortunately, this leads to weighted sparsity overestimating the number of measurements required to recover all vectors of interest. Unless we substantially oversample the finer wavelet levels then we are unable to see uniform recovery of weighted sparse vectors.

3.2.1.5 The ℓ^2 robust nullspace property of order $(\mathbf{s}, \mathbf{M})$ and the RIP in levels

Given the success of the 'flip test in levels', let us now try to find a sufficient condition on a matrix $U \in \mathbb{C}^{m\times n}$ that allows us to conclude that uniform recovery of order $(\mathbf{s}, \mathbf{M})$ is possible for U. If the RIP implies uniform recovery of order s then the obvious idea is to extend the RIP to a so-called 'RIP in levels', defined as follows:

Definition 3.2.5 (RIP in levels)**.** *For a given sparsity pattern $(\mathbf{s}, \mathbf{M})$ and matrix $U \in \mathbb{C}^{m\times n}$, the* RIP in levels *(*$\mathrm{RIP}_L$*) constant of order $(\mathbf{s}, \mathbf{M})$ (*RIC_L*), denoted by $\delta_{\mathbf{s},\mathbf{M}}$, is the smallest $\delta > 0$ such that*

$$(1-\delta)\|x\|_2^2 \leq \|Ux\|_2^2 \leq (1+\delta)\|x\|_2^2$$

for all $x \in \Sigma_{\mathbf{s},\mathbf{M}}$.

We will see that the RIP in levels allows us to obtain error estimates on $\|x-\widetilde{x}\|_1$ and $\|x-\widetilde{x}\|_2$ with $\widetilde{x}$ set to be a minimiser of (BP). Similar error estimates can be obtained if U satisfies the ℓ^2 robust nullspace property of order $(\mathbf{s}, \mathbf{M})$, a natural generalisation of the nullspace property of order s.

Definition 3.2.6 (The order $(\mathbf{s}, \mathbf{M})$ ℓ^2 RNSP)**.** *A matrix $U \in \mathbb{C}^{m\times n}$ satisfies the ℓ^2* robust nullspace property of order $(\mathbf{s}, \mathbf{M})$ *if there is a $\rho \in (0,1)$ and a $\tau > 0$ such that*

$$\|v_S\|_2 \leq \frac{\rho}{\sqrt{s}}\|v_{S^c}\|_1 + \tau\|Uv\|_2 \tag{3.2.2}$$

for all $(\mathbf{s}, \mathbf{M})$-sparse sets S and vectors $v \in \mathbb{C}^n$.

3.2.2 Matrices with a small RIP_L constant

To see how matrices similar to the ones in Figure 3.6 have a small RIC_L, we will first explain how the sampling patterns in Figure 3.6 were obtained. Earlier work on compressive sensing suggested that sampling should be done uniformly at random

(see [60]). Because of the near block diagonality of the matrices DFT · IWT and HAD · IWT for a variety of wavelets, instead of sampling uniformly at random, it is better to sample using a *multilevel* structure, where the percentage subsampling of each block depends on the relative importance of the corresponding wavelet coefficients (as done in [4], among others). More precisely, we can make the following definition:

Definition 3.2.7 (Multilevel sampling)**.** *Let N be the dimension of the ambient measurement space. We set $\mathbf{N} := (N_0, N_1, \dots, N_l)$ and $\mathbf{m} := (m_1, m_2, \dots, m_l)$ so that $0 = N_0 < N_1 < N_2 \cdots < N_l = N$ and $m_k \leq N_k - N_{k-1}$ for every $k = 1, \dots, l$. For each such k, set $\Omega_k = \{t_{k,1}, \dots t_{k,m_k}\}$ where the $t_{k,j}$ for $j = 1, \dots m_k$ are selected independently and uniformly at random from $\{N_{k-1} + 1, \dots, N_k\}$. We call $\Omega = \Omega_{\mathbf{N},\mathbf{m}} = \Omega_1 \cup \Omega_2 \cdots \cup \Omega_l$ a* $(\mathbf{N}, \mathbf{m})$-multilevel subsampling scheme

We define the *coherence* of U to determine how effective recovery is with equation (BP) for matrices $U = P_\Omega M$ where M is an isometry and where Ω is taken uniformly at random.

Definition 3.2.8 (Coherence)**.** *The* coherence *of a matrix $M \in \mathbb{C}^{N \times N}$, denoted by $\mu(M)$, is the quantity*

$$\mu(M) := \max_{i,j=1,\dots,N} |M_{i,j}|^2.$$

If $\mu(M)$ is sufficiently small and Ω is a sufficiently large set taken uniformly at random from $1, 2, \dots, N$ then with high probability $P_\Omega M$ will satisfy the RIP (and therefore exhibit uniform recovery). The related quantity with uniform multilevel subsampling is *coherence in levels.*

Definition 3.2.9 (Local Coherence)**.** *Given a sparsity pattern $(\mathbf{s}, \mathbf{M})$ and a uniform $(\mathbf{m}, \mathbf{N})$-multilevel subsampling scheme, we define the (j, k)* coherence in levels *of the matrix M with respect to $(\mathbf{s}, \mathbf{M})$ and $(\mathbf{m}, \mathbf{N})$ to be the value $\mu_{j,k}(\mathbf{N}, \mathbf{M})$ where*

$$\mu_{j,k}(M) = \max\{|M_{s,t}|^2 \text{ such that } s \in \{N_{j-1} + 1, \dots, N_j\}, t \in \{M_{k-1} + 1, \dots M_k\}\} \tag{3.2.3}$$

If the coherence in levels of a matrix is sufficiently small and Ω is a sufficiently dense uniform multilevel subsampling scheme then we can show that $P_\Omega M$ satisfies the RIP_L. Indeed in [83] the following result was shown

Theorem 3.2.10 (The existence of RIP_L matrices)**.** *There exists a constant $C > 0$ with the following property: let $M \in \mathbb{C}^{N \times N}$ be an isometry, $l \in \mathbb{N}$ and ϵ, δ real numbers such that $0 < \epsilon, \delta < 1$. Let l_0 be a natural number with $0 \leq l_0 \leq l$ and $\Omega_{\mathbf{N},\mathbf{m}}$ be an $(\mathbf{N}, \mathbf{m})$-multilevel subsampling scheme and $(\mathbf{s}, \mathbf{M})$ be a sparsity pattern. Suppose that $m_k = N_k - N_{k-1}$ for $k = 1, 2 \dots, l_0$ and that*

$$m_k \geq C\delta^{-2}(N_k - N_{k-1}) \left(\sum_{r=1}^{l} \mu_{k,r} s_r \right) (l \log(2\tilde{m}) \log(2N) \log^2(2s) + \log(\epsilon^{-1})) \tag{3.2.4}$$

$k > l_0$ where $\tilde{m} = m_{l_0+1} + \cdots m_r$. Then with probability at least $1 - \epsilon$, the matrix $U = P_\Omega M$ satisfies the RIP_L of order $(\mathbf{s}, \mathbf{M})$ with constant $\delta_{\mathbf{s},\mathbf{M}} \leq \delta$.

In particular, it is possible to use results from [5] to obtain a condition on the number of measurements taken that guarantees that matrices of the form $P_\Omega \mathrm{DFT} \cdot \mathrm{Haar}^{-1}$ satisfy the RIP_L where Ω is a uniform multilevel subsampling scheme and DFT and Haar^{-1} are the 1D Discrete Fourier Transform and 1D Inverse Haar Wavelet Transform respectively. More precisely, Corollary 3.3 from [83] says the following.

Theorem 3.2.11 (The RIP_L of the $\mathrm{DFT} \cdot \mathrm{Haar}^{-1}$ matrix)**.** *There exists a constant $C > 0$ with the following property: let $N = 2^l$ for some $l \in \mathbb{N}$, where l is the number of wavelet levels and ϵ, δ real numbers such that $0 < \epsilon, \delta < 1$. Set $S^k = 2^{\max(k-1,1)}$ to be the number of wavelet coefficients in the kth level. Suppose that m_k satisfies*

$$m_k \geq C\delta^{-2} \left(s_k + \sum_{r=l_0+1}^{l} 2^{-|k-r|} s_r \right) \left(\log(2m) \log^2(2N) \log^2(2s) + \log(\epsilon^{-1}) \right) \quad (3.2.5)$$

for $k = 1, 2, \ldots, l$ and where $s = \sum_{i=1}^{l} s_i$. Then if Ω is a $(\mathbf{N}, \mathbf{m})$-multilevel sampling scheme with $\mathbf{N} = (0, S^1, S^1 + S^2, \ldots, \sum_{r=1}^{l} S^k)$ and $\mathbf{m} = (m_1, m_2, \ldots, m_l)$ then the matrix $P_\Omega \, \mathrm{DFT} \cdot \mathrm{Haar}^{-1}$ satisfies $\delta_{\mathbf{s},\mathbf{M}} < \delta$ for $\mathbf{M} := \mathbf{N}$ and $\mathbf{s} := (s_1, s_2, \ldots, s_l)$ with probability exceeding $1 - \epsilon$.

It is possible to give further examples of matrices with a small RIP_L constant by examining their coherence in levels and employing Theorem 3.2.10. Bounds on the coherence in levels for a variety of matrices can be examined by using tools developed in papers such as [74]. Moreover, one can obtain a version of Theorem 3.2.11 for Hadamard matrices combined with Haar wavelets as the resulting matrix $U = \mathrm{HAD} \cdot \mathrm{DWT}_{\mathrm{Haar}}^{-1}$ is completely block diagonal (see Figure 3.3), however, this is beyond the scope of this paper.

3.3 Main results

If a matrix $U \in \mathbb{C}^{m \times n}$ satisfies the RIP then $(1 - \delta_s) < \|u_i\|_2^2 < (1 + \delta_s)$ for each column u_i of U. To ensure that we have similar control over $\|u_i\|_2$ with the RIP_L we make the following two definitions:

Definition 3.3.1 (Ratio constant)**.** *The* ratio constant *of a sparsity pattern $(\mathbf{s}, \mathbf{M})$, which we denote by $\eta_{\mathbf{s},\mathbf{M}}$, is given by $\eta_{\mathbf{s},\mathbf{M}} := \max_{i,j} s_i/s_j$.*

If the sparsity pattern $(\mathbf{s}, \mathbf{M})$ has l levels and there is a $j \in \{1, 2, \ldots, l\}$ for which $s_j = 0$ then we write $\eta_{\mathbf{s},\mathbf{M}} = \infty$.

Definition 3.3.2 (Covering a matrix)**.** *A sparsity pattern $(\mathbf{s}, \mathbf{M})$ is said to* cover *a matrix $U \in \mathbb{C}^{m \times n}$ if*

1. *$\eta_{\mathbf{s},\mathbf{M}} < \infty$*
2. *$M_l \geq n$ where l is the number of levels for $(\mathbf{s}, \mathbf{M})$.*

If a sparsity pattern does not cover U because it fails to satisfy either 1 or 2 from the definition of a sparsity pattern covering a matrix U then we cannot guarantee

recovery of $(\mathbf{s}, \mathbf{M})$-sparse vectors, even in the case that $\delta_{\mathbf{s},\mathbf{M}} = 0$. We shall justify the necessity of both conditions using two counterexamples. Firstly, we shall provide a matrix U, a sparsity pattern $(\mathbf{s}, \mathbf{M})$ and an $(\mathbf{s}, \mathbf{M})$-sparse vector $x_1 \in \mathbb{C}^n$ such that $\eta_{\mathbf{s},\mathbf{M}} = \infty$, $\delta_{\mathbf{s},\mathbf{M}} = 0$ and x_1 is not recovered by standard ℓ^1 minimisation. Indeed, consider the following

$$U = \begin{pmatrix} 1 & 2 \\ 0 & 0 \end{pmatrix}, \quad \mathbf{s} = (1,0), \quad \mathbf{M} = (0,1,2), \quad x_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

By the definition of $\eta_{\mathbf{s},\mathbf{M}}$, we have $\eta_{\mathbf{s},\mathbf{M}} = \infty$ and it is obvious that $\delta_{\mathbf{s},\mathbf{M}} = 0$. Furthermore, even without noise, x_1 does not solve the minimisation problem $\min \|\tilde{x}\|_1$ such that $Ux_1 = U\tilde{x}$. This can easily be seen by observing that $Ux_1 = Ux_2$ with $\|x_2\|_1 = \frac{1}{2}$ where $x_2 := (0, 1/2)^T$. It is therefore clear that Assumption 1 is necessary. We shall now provide an explanation for why Assumption 2 is also a requirement if we wish for the RIP_L to be a sufficient condition for the recovery of $(\mathbf{s}, \mathbf{M})$-sparse vectors. This time, consider the following combination of U, $(\mathbf{s}, \mathbf{M})$ and x_1:

$$U = \begin{pmatrix} 1 & 0 & 2 \\ 0 & 1 & 0 \end{pmatrix}, \quad \mathbf{s} = (1), \quad \mathbf{M} = (0,1), \quad x_1 = (1,0,0)^T.$$

and again, even though $\delta_{\mathbf{s},\mathbf{M}} = 0$, recovery is not possible because $Ux_1 = Ux_2$ with $\|x_2\|_1 = 1/2$ where $x_2 := (0, 0, 1/2)^T$.

We shall therefore try to prove that uniform recovery of order $(\mathbf{s}, \mathbf{M})$ is possible with the RIP_L under the assumption that $(\mathbf{s}, \mathbf{M})$ covers U. To do this, we need one further definition. In equation (3.0.4) the bound on $||x - \widetilde{x}||_2$ involves $\sqrt{s}$. This arises because s is the maximum number of non-zero values that could be in an s-sparse vector. The equivalent for $(\mathbf{s}, \mathbf{M})$-sparse vectors is the following:

Definition 3.3.3 (Number of elements of a sparsity pattern)**.** *The* number of elements *of a sparsity pattern* $(\mathbf{s}, \mathbf{M})$*, which we denote by* $\widetilde{s}$*, is given by* $\tilde{s} := s_1 + s_2 + \cdots + s_l$.

To prove that a sufficiently small RIP in levels constant implies uniform recovery of order $(\mathbf{s}, \mathbf{M})$, it is natural to adapt the steps used in [7] to prove that the RIP implies uniform recovery of order s. This adaptation yields a sufficient condition for recovery even in the noisy case.

Theorem 3.3.4 (RIP_L Recovery Theorem)**.** *Let* $(\mathbf{s}, \mathbf{M})$ *be a sparsity pattern with* l *levels and ratio constant* $\eta_{\mathbf{s},\mathbf{M}}$*. Suppose that the matrix* $U \in \mathbb{C}^{m \times n}$ *is covered by* $(\mathbf{s}, \mathbf{M})$ *and has a* RIP_L *constant* $\delta_{2\mathbf{s},\mathbf{M}}$ *satisfying*

$$\delta_{2\mathbf{s},\mathbf{M}} < \frac{1}{\sqrt{l\left(\sqrt{\eta_{\mathbf{s},\mathbf{M}}} + \frac{1}{4}\right)^2 + 1}}. \tag{3.3.1}$$

Furthermore, suppose that $x \in \mathbb{C}^n$ *and* $v = Ux + \nu$ *where* $\|\nu\|_2 \leq \epsilon$*. Then any* $\widetilde{x} \in \mathbb{C}^n$

which are minimisers of the noisy ℓ^1 *minimisation problem* (BPDN) *also satisfy*

$$\|x - \widetilde{x}\|_1 \leq C_1 \sigma_{\mathbf{s},\mathbf{M}}(x)_1 + D_1 \sqrt{\widetilde{s}}\, \epsilon \text{ and} \tag{3.3.2}$$

$$\|x - \widetilde{x}\|_2 \leq \frac{\sigma_{\mathbf{s},\mathbf{M}}(x)_1}{\sqrt{\widetilde{s}}} \left(C_2 + C_2' \sqrt[4]{l \eta_{\mathbf{s},\mathbf{M}}} \right) + \epsilon \left(D_2 + D_2' \sqrt[4]{l \eta_{\mathbf{s},\mathbf{M}}} \right) \tag{3.3.3}$$

where C_1, C_2, C_2', D_1, D_2 *and* D_2' *depend only on* $\delta_{2\mathbf{s},\mathbf{M}}$*. Note that the noiseless case* $\epsilon = 0$*, equation* (3.3.2) *reduces to uniform recovery of order* $(\mathbf{s}, \mathbf{M})$ *and* (BPDN) *reduces to* (BP)*.*

This result allows uniform recovery within the class of $(\mathbf{s}, \mathbf{M})$-sparse vectors but the requirement on $\delta_{2\mathbf{s},\mathbf{M}}$ depends on l and $\eta_{\mathbf{s},\mathbf{M}}$. We make the following observations:

1. If we pick a sparsity pattern that uses lots of levels then we will require a smaller RIP_L constant (note however that if we work with wavelets and $\mathbf{M}$ corresponds to wavelet levels then the number of levels l is approximately $\log_2(N)$, so that the RIP_L constant only has to shrink like $1/\sqrt{\log_2(N)}$).

2. If we pick a sparsity pattern with fewer levels then to explain the excellent observed recovery in applications we shall have choose s_i so that s_i/s_j is correspondingly larger for distinct i and j.

3. If the RIP_L constant $\delta_{2\mathbf{s},\mathbf{M}}$ is sufficiently small so that the conclusion of Theorem 3.3.4 holds, the bound on $\|x - \tilde{x}\|_2$ is weaker than the bound (3.0.4) obtained using the RIP.

As a consequence of these observations, at first glance it may appear that the results we have obtained with the RIP_L are weaker than those given using the standard RIP. However, Theorem 3.3.4 is stronger than any theorem based around uniform recovery in two senses. Firstly, if one considers a sparsity pattern with one level then the bounds (3.3.2) and (3.3.3) reduce to (3.0.3) and (3.0.4) respectively. Secondly, neither equation (3.0.1) nor (3.0.3) applies at all if we do not have uniform recovery. Therefore, for the examples given in Figure 3.2, (3.0.1) and (3.0.3) are not applicable.

Ideally, it would be possible to find a constant C such that if the RIP_L constant is smaller than C then recovery of all $(\mathbf{s}, \mathbf{M})$-sparse vectors would be possible. Unfortunately, we shall demonstrate that this is impossible in Theorems 3.3.5 and 3.3.6. Indeed, in some sense Theorem 3.3.4 is optimal in l and $\eta_{\mathbf{s},\mathbf{M}}$, as the following results confirm.

Theorem 3.3.5 (RIP_L dependence on the ratio constant)**.** *Fix* $a \in \mathbb{N}$ *and* $f : \mathbb{R} \to \mathbb{R}$ *such that* $f(\eta_{\mathbf{s},\mathbf{M}}) = o(\eta_{\mathbf{s},\mathbf{M}}^{\frac{1}{2}})$*. Then there are* $m, n \in \mathbb{N}$*, a matrix* $U' \in \mathbb{C}^{m \times n}$ *and a sparsity pattern* $(\mathbf{s}, \mathbf{M})$ *with two levels that covers* U' *such that the* RIP_L *constant* $\delta_{a\mathbf{s},\mathbf{M}}$ *and ratio constant* $\eta_{\mathbf{s},\mathbf{M}}$ *satisfy*

$$\delta_{a\mathbf{s},\mathbf{M}} \leq \frac{1}{|f(\eta_{\mathbf{s},\mathbf{M}})|} \tag{3.3.4}$$

but there is an $(\mathbf{s}, \mathbf{M})$*-sparse* z^1 *such that*

$$z^1 \notin \arg\min \|z\|_1 \text{ such that } U'z = U'z^1.$$

Roughly speaking, Theorem 3.3.5 says that if we fix the number of levels and try to replace the condition

$$\delta_{2\mathbf{s},\mathbf{M}} < \frac{1}{\sqrt{l\left(\sqrt{\eta_{\mathbf{s},\mathbf{M}}} + \frac{1}{4}\right)^2 + 1}}$$

with a condition of the form $\delta_{2\mathbf{s},\mathbf{M}} < (\eta_{\mathbf{s},\mathbf{M}})^{-\alpha/2}/(C\sqrt{l})$ for some constant C and some $\alpha < 1$ then the conclusion of Theorem 3.3.4 ceases to hold. In particular, the requirement on $\delta_{2\mathbf{s},\mathbf{M}}$ cannot be independent of $\eta_{\mathbf{s},\mathbf{M}}$. The parameter a in the statement of Theorem 3.3.5 says that we cannot simply fix the issue by changing $\delta_{2\mathbf{s},\mathbf{M}}$ to $\delta_{3\mathbf{s},\mathbf{M}}$ or any further multiple of $\mathbf{s}$.

Similarly, we can state and prove a similar theorem that shows that the dependence on the number of levels, l, cannot be ignored.

Theorem 3.3.6 (RIP_L dependence on the number of levels)**.** *Fix* $a \in \mathbb{N}$ *and* $f : \mathbb{R} \to \mathbb{R}$ *such that* $f(l) = o(l^{\frac{1}{2}})$*. Then there are* $m, n \in \mathbb{N}$*, a matrix* $U' \in \mathbb{C}^{m\times n}$ *and a sparsity pattern* $(\mathbf{s}, \mathbf{M})$ *that covers* U' *with ratio constant* $\eta_{\mathbf{s},\mathbf{M}} = 1$ *and* l *levels such that the* RIP_L *constant* $\delta_{a\mathbf{s},\mathbf{M}}$ *corresponding to* U' *satisfies* $\delta_{a\mathbf{s},\mathbf{M}} \leq 1/|f(l)|$ *but there is an* $(\mathbf{s}, \mathbf{M})$*-sparse* z^1 *such that*

$$z^1 \notin \arg\min \|z\|_1 \text{ such that } U'z = U'z^1.$$

Furthermore, Theorem 3.3.7 shows that the ℓ^2 error estimate on $\|x - \widetilde{x}\|_2$ is optimal up to constant terms.

Theorem 3.3.7 (RIP_L ℓ^2 error optimality)**.** *The* ℓ^2 *result (3.3.3) in Theorem 3.3.4 is sharp in the following sense:*

1. *For a fixed* $a \in \mathbb{N}$ *and any functions* $f, g : \mathbb{R} \to \mathbb{R}$ *such that* $f(\eta) = o(\eta^{\frac{1}{4}})$ *and* $g(\eta) = O(\sqrt{\eta})$*, there are natural numbers* m *and* n*, a matrix* $U' \in \mathbb{C}^{m\times n}$ *and a sparsity pattern* $(\mathbf{s}, \mathbf{M})$ *with two levels that such that*

   - $(\mathbf{s}, \mathbf{M})$ *covers* U'
   - *The* RIP_L *constant corresponding to the sparsity pattern* $(a\mathbf{s}, \mathbf{M})$*, denoted by* $\delta_{a\mathbf{s},\mathbf{M}}$*, satisfies* $\delta_{a\mathbf{s},\mathbf{M}} \leq 1/|g(\eta_{\mathbf{s},\mathbf{M}})|$*.*
   - *There exist vectors* z *and* z^1 *such that* $U'(z - z^1) = 0$ *and* $\|z\|_1 \leq \|z^1\|_1$ *but*

   $$\|z - z^1\|_2 > \frac{f(\eta_{\mathbf{s},\mathbf{M}})}{\sqrt{\tilde{s}}}\sigma_{\mathbf{s},\mathbf{M}}(z^1)_1.$$

2. *For a fixed* $a \in \mathbb{N}$ *and any functions* $f, g : \mathbb{R} \to \mathbb{R}$ *such that* $f(l) = o(l^{\frac{1}{4}})$ *and* $g(l) = O(\sqrt{l})$*, there are natural numbers* m *and* n*, a matrix* $U' \in \mathbb{C}^{m\times n}$ *and a sparsity pattern* $(\mathbf{s}, \mathbf{M})$ *with* $\eta_{\mathbf{s},\mathbf{M}} = 1$ *such that*

- $(\mathbf{s}, \mathbf{M})$ *covers* U'
- *The* RIP_L *constant corresponding to the sparsity pattern* $(a\mathbf{s}, \mathbf{M})$*, denoted by* $\delta_{a\mathbf{s},\mathbf{M}}$*, satisfies* $\delta_{a\mathbf{s},\mathbf{M}} \leq 1/|g(l)|$.
- *There exist vectors* z *and* z^1 *such that* $U'(z - z^1) = 0$ *and* $\|z\|_1 \leq \|z^1\|_1$ *but*

$$\|z - z^1\|_2 > \frac{f(l)}{\sqrt{\tilde{s}}} \sigma_{\mathbf{s},\mathbf{M}}(z^1)_1.$$

As with the RIP in levels, we can obtain results on recovery using the ℓ^2 robust nullspace property of order $(\mathbf{s}, \mathbf{M})$

Theorem 3.3.8 (ℓ^2 RNSP of order $(\mathbf{s}, \mathbf{M})$ recovery theorem)**.** *Suppose that a matrix* $U \in \mathbb{C}^{m \times n}$ *satisfies the* ℓ^2 *robust nullspace property of order* $(\mathbf{s}, \mathbf{M})$ *with constants* $\rho \in (0, 1)$ *and* $\tau > 0$*. Let* $x \in \mathbb{C}^n$ *and* $y \in \mathbb{C}^m$ *satisfy* $\|Ux - y\|_2 < \epsilon$*. Then any solutions* $\widetilde{x}$ *of the* ℓ^1 *minimisation problem*

$$\min_{\widehat{x} \in \mathbb{C}^n} \|\widehat{x}\|_1 \text{ subject to } \|U\widehat{x} - y\|_2 \leq \epsilon$$

satisfy

$$\|\widetilde{x} - x\|_1 \leq A_1 \sigma_{\mathbf{s},\mathbf{M}}(x)_1 + C_1 \epsilon \sqrt{\tilde{s}} \tag{3.3.5}$$

$$\|\widetilde{x} - x\|_2 \leq \frac{\sigma_{\mathbf{s},\mathbf{M}}(x)_1}{\sqrt{\tilde{s}}} \left(A_2 + B_2 \sqrt[4]{l \eta_{\mathbf{s},\mathbf{M}}} \right) + 2\epsilon \left(C_2 + D_2 \sqrt[4]{l \eta_{\mathbf{s},\mathbf{M}}} \right) \tag{3.3.6}$$

where

$$A_1 := \frac{2 + 2\rho}{1 - \rho}, \quad C_1 := \frac{4\tau}{1 - \rho}, \quad A_2 := \frac{2\rho + 2\rho^2}{1 - \rho},$$

$$B_2 := \frac{\left(2\sqrt{\rho} + 1\right)(1 + \rho)}{1 - \rho}, \quad C_2 := \frac{\rho\tau + \tau}{1 - \rho} \quad \text{and} \quad D_2 := \frac{4\sqrt{\rho}\tau + 3\tau - \rho\tau}{2 - 2\rho}.$$

This Theorem explains where the dependence on $\eta_{\mathbf{s},\mathbf{M}}$ and l in (3.3.3) emerges from. One technique for showing that the RIP implies uniform recovery of order s is to prove that a sufficiently small RIC implies the nullspace property (for example, this method is used in [60]). In a similar way, we prove the ℓ^2 error estimate in Theorem 3.3.4 by showing that a sufficiently small RIP_L constant implies the robust ℓ^2 nullspace property of order $(\mathbf{s}, \mathbf{M})$. The ℓ^2 error estimate (3.3.6) follows and we are left with a dependence on $\sqrt[4]{l \eta_{\mathbf{s},\mathbf{M}}}$ in the right hand side of (3.3.3). As before, we can show that this is optimal. We do this in Theorem 3.3.9.

Theorem 3.3.9 (ℓ^2 RNSP of order $(\mathbf{s}, \mathbf{M})$ optimality)**.** *The result in Theorem 3.3.8 is sharp, in the sense that*

1. *For any* $f : \mathbb{R}^3 \to \mathbb{R}$ *satisfying* $f(\rho, \tau, \eta) = o(\eta^{\frac{1}{4}})$ *for fixed* $\rho \in (0, 1)$ *and* $\tau > 0$*, there are natural numbers* m *and* n*, a matrix* $U' \in \mathbb{C}^{m \times n}$ *and a sparsity pattern* $(\mathbf{s}, \mathbf{M})$ *with ratio constant* $\eta_{\mathbf{s},\mathbf{M}}$ *and two levels such that*

   - $(\mathbf{s}, \mathbf{M})$ *covers* U'

- *U' satisfies the ℓ^2 robust nullspace property of order $(\mathbf{s}, \mathbf{M})$ with constants $\rho \in (0,1)$ and $\tau > 0$*
- *There exist vectors z and z^1 such that $U'(z - z^1) = 0$ and $\|z\|_1 \leq \|z^1\|_1$ but*

$$\|z - z^1\|_2 > \frac{f(\rho, \tau, \eta_{\mathbf{s},\mathbf{M}})}{\sqrt{\tilde{s}}} \sigma_{\mathbf{s},\mathbf{M}}(z^1)_1.$$

2. *For any $f : \mathbb{R}^3 \to \mathbb{R}$ satisfying $f(\rho, \tau, l) = o(l^{\frac{1}{4}})$ for fixed $\rho \in (0,1)$ and $\tau > 0$, there are natural numbers m and n, a matrix $U' \in \mathbb{C}^{m \times n}$ and a sparsity pattern $(\mathbf{s}, \mathbf{M})$ with ratio constant $\eta_{\mathbf{s},\mathbf{M}} = 1$ and l levels such that*

   - *$(\mathbf{s}, \mathbf{M})$ covers U'*
   - *U' satisfies the ℓ^2 robust nullspace property of order $(\mathbf{s}, \mathbf{M})$ with constants $\rho \in (0,1)$ and $\tau > 0$*
   - *There exist vectors z and z^1 such that $U'(z - z^1) = 0$ and $\|z\|_1 \leq \|z^1\|_1$ but*

$$\|z - z^1\|_2 > \frac{f(\rho, \tau, l)}{\sqrt{\tilde{s}}} \sigma_{\mathbf{s},\mathbf{M}}(z^1)_1.$$

The conclusions that we can draw from the above theorems are the following:

1. The RIP_L will guarantee uniform recovery of order $(\mathbf{s}, \mathbf{M})$, provided that the RIP_L constant is sufficiently small (Theorem 3.3.4).

2. The requirement on the RIP_L constant to achieve uniform recovery of order $(\mathbf{s}, \mathbf{M})$ is dependent on $\sqrt{\eta_{\mathbf{s},\mathbf{M}}}$ and $\sqrt{l}$. This is optimal up to constants (Theorem 3.3.5 and Theorem 3.3.6).

3. When compared to the error estimates obtained using the RIP, the ℓ^2 error when using the RIP_L has additional factors of the form $\sqrt[4]{l}$ and $\sqrt[4]{\eta_{\mathbf{s},\mathbf{M}}}$. Again, these are optimal up to constants (Theorem 3.3.7).

4. The same additional factors of the form $\sqrt[4]{l}$ and $\sqrt[4]{\eta_{\mathbf{s},\mathbf{M}}}$ on the ℓ^2 error estimate (Theorem 3.3.8) are also present with the robust ℓ^2 nullspace property of order $(\mathbf{s}, \mathbf{M})$.

5. These factors are optimal up to constants, so that even if we ignore the RIP_L and still try to prove results using the ℓ^2 robust nullspace property of order $(\mathbf{s}, \mathbf{M})$ then we would be unable to improve the ℓ^2 error (Theorem 3.3.9).

With these results, we have demonstrated that the RIP in levels may be able to explain why permutations within levels are possible and why more general permutations are impossible with compressed sensing for the matrices in 3.2 (similar numerical arguments can be used whenever the matrix is nearly block diagonal). The results that we have obtained give a sufficient condition on the RIP in levels constant that guarantees $(\mathbf{s}, \mathbf{M})$-sparse recovery. Furthermore, we have managed to demonstrate that this condition and the conclusions that follow from it are optimal up to constants.

3.4 Proofs

We shall present the proofs in a different arrangement to the order in which their statements were presented. The first proof that we shall present is that of Theorem 3.3.8.

3.4.1 Proof of Theorem 3.3.8

We begin with the following lemma:

Lemma 3.4.1. *Suppose that $U \in \mathbb{C}^{m\times n}$ satisfies the ℓ^2 robust nullspace property of order $(\mathbf{s},\mathbf{M})$ with constants $\rho \in (0,1)$ and $\tau > 0$. Fix $v \in \mathbb{C}^n$, and let S be an $(\mathbf{s},\mathbf{M})$-sparse set such that $|S| = \tilde{s}$ and the property that if T is an $(\mathbf{s},\mathbf{M})$-sparse set, we have $\|v_S\|_1 \geq \|v_T\|_1$. Then*

$$\|v\|_2 \leq \frac{\|v_{S^c}\|_1}{\sqrt{\tilde{s}}}\left[\rho + \left(\sqrt{\rho}+\frac{1}{2}\right)\sqrt[4]{l\eta_{\mathbf{s},\mathbf{M}}}\right] + \tau\|Uv\|_2\left[\frac{\sqrt[4]{l\eta_{\mathbf{s},\mathbf{M}}}}{2}+1\right].$$

Proof. For $i = 1,2,\ldots,l$, we define S_0^i to be $S_0^i = S \cap \{M_{i-1}+1, M_{i-1}+2,\ldots,M_i\}$ (i.e. S_0^i is the elements of S that are in the ith level). Let $m = \max_{i=1,2,\ldots,l} \min_{j\in S_0^i} |v_j|$. Since $|S_0^i| = s_i$ (otherwise $|S| < \tilde{s}$), we can see that given any $i = 1,2,\ldots,l$

$$\|v_S\|_2 = \sqrt{\sum_{n\in S}|v_n|^2} \geq \sqrt{\sum_{j\in S_0^i}|v_j|^2} \geq \sqrt{s_i}\min_{j\in S_0^i}|v_j| \geq \min_{k=1,2,\ldots,l}\sqrt{s_k}\min_{j\in S_0^i}|v_j|$$

so that $\|v_S\|_2 \geq m \min\limits_{k=1,2,\ldots,l}\sqrt{s_k}$. Furthermore, $|v_j| \leq m$ for each $j \in S^c$ otherwise there is an $(\mathbf{s},\mathbf{M})$-sparse T with $\|v_T\|_1 > \|v_S\|_1$. Therefore $\|v_{S^c}\|_2^2 = \sum_{j\in S^c}|v_j|^2 \leq \sum_{j\in S^c} m|v_j| \leq \frac{\|v_{S^c}\|_1\|v_S\|_2}{\min\limits_{k=1,2,\ldots,l}\sqrt{s_k}}$. By the ℓ^2 robust nullspace property of order $(\mathbf{s},\mathbf{M})$, $\|v_{S^c}\|_1\|v_S\|_2 \leq \frac{\rho}{\sqrt{\tilde{s}}}\|v_{S^c}\|_1^2 + \tau\|Uv\|_2\|v_{S^c}\|_1$. Since $\sqrt{a+b} \leq \sqrt{a}+\sqrt{b}$ whenever $a,b>0$,

$$\|v_{S^c}\|_2 \leq \frac{1}{\min\sqrt[4]{s_i}}\left(\frac{\sqrt{\rho}}{\sqrt[4]{\tilde{s}}}\|v_{S^c}\|_1 + \sqrt{\tau\|Uv\|_2\|v_{S^c}\|_1}\right) \tag{3.4.1}$$

Using the arithmetic-geometric mean inequality,

$$\sqrt{\tau\|Uv\|_2\|v_{S^c}\|_1} = \sqrt{\tau\|Uv\|_2\sqrt[4]{\tilde{s}}\frac{\|v_{S^c}\|_1}{\sqrt[4]{\tilde{s}}}} \leq \frac{\tau\|Uv\|_2\sqrt[4]{\tilde{s}}}{2} + \frac{\|v_{S^c}\|_1}{2\sqrt[4]{\tilde{s}}}$$

Therefore, (3.4.1) yields

$$\begin{aligned}\|v_{S^c}\|_2 &\leq \frac{1}{\min\sqrt[4]{s_i}}\left(\frac{\sqrt{\rho}}{\sqrt[4]{\tilde{s}}}\|v_{S^c}\|_1 + \frac{\|v_{S^c}\|_1}{2\sqrt[4]{\tilde{s}}} + \frac{\tau\|Uv\|_2\sqrt[4]{\tilde{s}}}{2}\right)\\ &\leq \frac{\|v_{S^c}\|_1}{\sqrt[4]{\tilde{s}}\min\sqrt[4]{s_i}}\left(\sqrt{\rho}+\frac{1}{2}\right) + \frac{\tau\|Uv\|_2\sqrt[4]{l\eta_{\mathbf{s},\mathbf{M}}}}{2}\end{aligned}$$

because $\frac{\tilde{s}}{\min s_i} \leq l\eta_{\mathbf{s},\mathbf{M}}$. Once again, employing the ℓ^2 nullspace property gives

$$\begin{aligned}\|v\|_2 \leq \|v_S\|_2 + \|v_{S^c}\|_2 &\leq \frac{\rho}{\sqrt{\tilde{s}}}\|v_{S^c}\|_1 + \tau\|Uv\|_2 + \frac{\|v_{S^c}\|_1}{\sqrt[4]{\tilde{s}}\min\sqrt[4]{s_i}}\left(\sqrt{\rho}+\frac{1}{2}\right) \\ &\quad + \frac{\tau\|Uv\|_2\sqrt[4]{l\eta_{\mathbf{s},\mathbf{M}}}}{2} \\ &\leq \frac{\|v_{S^c}\|_1}{\sqrt{\tilde{s}}}\left[\rho + \left(\sqrt{\rho}+\frac{1}{2}\right)\frac{\sqrt[4]{\tilde{s}}}{\min\sqrt[4]{s_i}}\right] + \tau\|Uv\|_2\left[\frac{\sqrt[4]{l\eta_{\mathbf{s},\mathbf{M}}}}{2}+1\right] \\ &\leq \frac{\|v_{S^c}\|_1}{\sqrt{\tilde{s}}}\left[\rho + \left(\sqrt{\rho}+\frac{1}{2}\right)\sqrt[4]{l\eta_{\mathbf{s},\mathbf{M}}}\right] + \tau\|Uv\|_2\left[\frac{\sqrt[4]{l\eta_{\mathbf{s},\mathbf{M}}}}{2}+1\right].\end{aligned}$$

□

The remaining error estimates will follow from various properties related to the ℓ^1 *robust nullspace property* (see [60], definition 4.17) holds. To be precise,

Definition 3.4.2. *A matrix $U \in \mathbb{C}^{m\times n}$ satisfies the ℓ^1 robust nullspace property relative to S with constants $\rho \in (0,1)$ and $\tau' > 0$ if*

$$\|v_S\|_1 \leq \rho\|v_{S^c}\|_1 + \tau'\|Uv\|_2 \tag{3.4.2}$$

for any $v \in \mathbb{C}^n$. We say that U satisfies the ℓ^1 robust nullspace property of order $(\mathbf{s},\mathbf{M})$ if (3.4.2) *holds for any $(\mathbf{s},\mathbf{M})$-sparse sets S.*

It is easy to see that if U satisfies the ℓ^2 robust nullspace property of order $(\mathbf{s},\mathbf{M})$ with constants ρ and τ then, for any $(\mathbf{s},\mathbf{M})$-sparse set S, U also satisfies the ℓ^1 robust nullspace property relative to S with constants ρ and $\tau\sqrt{\tilde{s}}$. Indeed, assume that U satisfies the ℓ^2 robust nullspace property of order $(\mathbf{s},\mathbf{M})$ with constants ρ and τ. Then (by the Cauchy-Schwarz inequality) $\|v_S\|_1 \leq \sqrt{\tilde{s}}\|v_S\|_2 \leq \rho\|v_{S^c}\|_1 + \tau\sqrt{\tilde{s}}\|Uv\|_2$.

An immediate conclusion of the robust nullspace property is the following, proven in [60] as Theorem 4.20.

Lemma 3.4.3. *Suppose that $U \in \mathbb{C}^{m\times n}$ satisfies the ℓ^1 robust null space property with constants $\rho \in (0,1)$ and τ' relative to a set S. Then for any complex vectors $x, z \in \mathbb{C}^n$, we have*

$$\|z-x\|_1 \leq \frac{1+\rho}{1-\rho}\left(\|z\|_1 - \|x\|_1 + 2\|x_{S^c}\|_1\right) + \frac{2\tau'}{1-\rho}\|U(z-x)\|_2.$$

We can use this lemma to show the following important result, which is similar both in proof and statement to Theorem 4.19 in [60].

Lemma 3.4.4. *Suppose that a matrix $U \in \mathbb{C}^{m\times n}$ satisfies the ℓ^1 robust nullspace property of order $(\mathbf{s},\mathbf{M})$ with constants $\rho \in (0,1)$ and $\tau' > 0$. Furthermore, suppose that $\|Ux - y\|_2 \leq \epsilon$. Then any solutions $\widetilde{x}$ to the ℓ^1 minimisation problem*

$$\min_{\widehat{x}\in\mathbb{C}^n} \|\widehat{x}\|_1 \text{ subject to } \|U\widehat{x} - y\|_2 \leq \epsilon$$

satisfy

$$\|x - \widetilde{x}\|_1 \leq \frac{2+2\rho}{1-\rho}\sigma_{\mathbf{s},\mathbf{M}}(x)_1 + \frac{4\tau'\epsilon}{1-\rho}.$$

Proof. By Lemma 3.4.3, for any $(\mathbf{s}, \mathbf{M})$-sparse set S

$$\|\widetilde{x} - x\|_1 \leq \frac{1+\rho}{1-\rho}\left(\|\widetilde{x}\|_1 - \|x\|_1 + 2\|x_{S^c}\|_1\right) + \frac{2\tau'}{1-\rho}\|U(\widetilde{x} - x)\|_2$$

Because both $\|Ux - y\|_2$ and $\|U\widetilde{x} - y\|_2$ are smaller than or equal to ϵ, $\|Ux - U\widetilde{x}\| \leq 2\epsilon$. Furthermore, because $\widetilde{x}$ has minimal ℓ^1 norm, $\|\widetilde{x}\|_1 - \|x\|_1 \leq 0$.

Thus $\|x - \widetilde{x}\|_1 \leq \frac{2+2\rho}{1-\rho}\|x_{S^c}\|_1 + \frac{4\tau'\epsilon}{1-\rho}$. If we take S to be the $(\mathbf{s}, \mathbf{M})$-sparse set which maximizes $\|x_S\|_1$, then

$$\|x - \widetilde{x}\|_1 \leq \frac{2+2\rho}{1-\rho}\sigma_{\mathbf{s},\mathbf{M}}(x)_1 + \frac{4\tau'\epsilon}{1-\rho}.$$

□

We can combine these results to complete the proof of Theorem 3.3.8. Indeed, (3.3.5) follows immediately from Lemma 3.4.4 and the fact that U satisfies the ℓ^1 robust nullspace property with constants ρ and $\tau\sqrt{\tilde{s}}$. To prove (3.3.6), we can simply set $v = x - \widetilde{x}$ in Lemma 3.4.1 to see that

$$\begin{aligned}\|x - \widetilde{x}\|_2 &\leq \frac{\|(x-\widetilde{x})_{S^c}\|_1}{\sqrt{\tilde{s}}}\left[\rho + \left(\sqrt{\rho} + \frac{1}{2}\right)\sqrt[4]{l\eta_{\mathbf{s},\mathbf{M}}}\right] + \tau\|U\left(x - \widetilde{x}\right)\|_2\left[\frac{\sqrt[4]{l\eta_{\mathbf{s},\mathbf{M}}}}{2} + 1\right] \\ &\leq \frac{\|x - \widetilde{x}\|_1}{\sqrt{\tilde{s}}}\left[\rho + \left(\sqrt{\rho} + \frac{1}{2}\right)\sqrt[4]{l\eta_{\mathbf{s},\mathbf{M}}}\right] + 2\tau\epsilon\left[\frac{\sqrt[4]{l\eta_{\mathbf{s},\mathbf{M}}}}{2} + 1\right]\end{aligned}$$

and the result follows from (3.3.5).

3.4.2 Proof of Theorem 3.3.4

It will suffice to prove that the conditions on $\delta_{\mathbf{s},\mathbf{M}}$ and $(\mathbf{s}, \mathbf{M})$ in Theorem 3.3.4 imply the ℓ^2 robust nullspace property. To show this, we begin by stating the following inequality, proven in [25]:

Lemma 3.4.5 (The norm inequality for ℓ^1 and ℓ^2)**.** *Let $v = (v_1, v_2, \ldots, v_s)$ where $v_1 \geq v_2 \geq v_3 \geq \cdots \geq v_s$. Then*

$$\|v\|_2 \leq \frac{1}{\sqrt{s}}\|v\|_1 + \frac{\sqrt{s}}{4}(v_1 - v_s)$$

We will now prove the following additional lemma which is almost identical in statement and proof to that of Lemma 6.1 in [7].

Lemma 3.4.6. *Suppose that $x, y \in \Sigma_{\mathbf{s},\mathbf{M}}$ and that*

$$\|Ux\|_2^2 - \|x\|_2^2 = t\|x\|_2^2. \tag{3.4.3}$$

Additionally, suppose that x and y are orthogonal. Then $|\langle Ux, Uy\rangle| \leq \sqrt{\delta_{2\mathbf{s},\mathbf{M}}^2 - t^2}\|x\|_2\|y\|_2$ where $\delta_{2\mathbf{s},\mathbf{M}}$ is the restricted isometry constant corresponding to the sparsity pattern $(2\mathbf{s},\mathbf{M})$ and the matrix U.

Proof. Without loss of generality, we can assume that $\|x\|_2 = \|y\|_2 = 1$. Note that for $\alpha, \beta \in \mathbb{R}$ and $\gamma \in \mathbb{C}$, the vectors $\alpha x + \gamma y$ and $\beta x - \gamma y$ are contained in $\Sigma_{2\mathbf{s},\mathbf{M}}$. Therefore,

$$\|U(\alpha x + \gamma y)\|_2^2 \leq (1 + \delta_{2\mathbf{s},\mathbf{M}})\|\alpha x + \gamma y\|_2^2 = (1 + \delta_{2\mathbf{s},\mathbf{M}})(\alpha^2 + |\gamma|^2). \tag{3.4.4}$$

where the last line follows because $\langle x, y\rangle = 0$ (from the orthogonality of x and y). Similarly,

$$-\|U(\beta x - \gamma y)\|_2^2 \leq -(1 - \delta_{2\mathbf{s},\mathbf{M}})(\beta^2 + |\gamma|^2) \tag{3.4.5}$$

We will now add these two inequalities. On the one hand (by using the assumption in (3.4.3) and the fact that α,β are real), we have

$$\begin{aligned}\|U(\alpha x + \gamma y)\|_2^2 - \|U(\beta x - \gamma y)\|_2^2 &= \alpha^2\|Ux\|_2^2 + 2\mathrm{Re}(\alpha\overline{\gamma}\,\langle Ux, Uy\rangle) + |\gamma|^2\|Uy\|_2^2 \\ &\quad - \left(\beta^2\|Ux\|_2^2 - 2\mathrm{Re}(\beta\overline{\gamma}\,\langle Ux, Uy\rangle) + |\gamma|^2\|Uy\|_2^2\right) \\ &= (1+t)\left(\alpha^2 - \beta^2\right) + 2(\alpha+\beta)\mathrm{Re}(\overline{\gamma}\,\langle Ux, Uy\rangle)\end{aligned}$$

and on the other hand (from (3.4.4) and (3.4.5))

$$\|U(\alpha x + \gamma y)\|_2^2 - \|U(\beta x - \gamma y)\|_2^2 \leq \delta_{2\mathbf{s},\mathbf{M}}\left(\alpha^2 + \beta^2 + 2|\gamma|^2\right) + \alpha^2 - \beta^2.$$

Therefore,

$$(1+t)\left(\alpha^2 - \beta^2\right) + 2(\alpha+\beta)\mathrm{Re}(\overline{\gamma}\,\langle Ux, Uy\rangle) \leq \delta_{2\mathbf{s},\mathbf{M}}\left(\alpha^2 + \beta^2 + 2|\gamma|^2\right) + \alpha^2 - \beta^2.$$

After choosing γ so that $\mathrm{Re}(\overline{\gamma}\,\langle Ux, Uy\rangle) = |\,\langle Ux, Uy\rangle\,|$ we obtain

$$|\,\langle Ux, Uy\rangle\,| \leq \frac{1}{2\alpha + 2\beta}\left[(\delta_{2\mathbf{s},\mathbf{M}} - t)\alpha^2 + (\delta_{2\mathbf{s},\mathbf{M}} + t)\beta^2 + 2\delta_{2\mathbf{s},\mathbf{M}}\right] \tag{3.4.6}$$

because $|\gamma| = 1$. By the definition of the RIP in levels constant, $\delta_{2\mathbf{s},\mathbf{M}} \geq \delta_{\mathbf{s},\mathbf{M}}$ and so

$$|t| = \left|\|Ux\|_2^2 - \|x\|_2^2\right| \leq \delta_{\mathbf{s},\mathbf{M}} \leq \delta_{2\mathbf{s},\mathbf{M}}. \tag{3.4.7}$$

If equality holds in (3.4.7), then we can set $\beta = 0$ and send $\alpha \to \infty$ in (3.4.6) to obtain the required result. Otherwise, equation (3.4.7) implies that $\sqrt{\frac{\delta_{2\mathbf{s},\mathbf{M}}+t}{\delta_{2\mathbf{s},\mathbf{M}}-t}} \in \mathbb{R}$ and so we can set $\alpha = \sqrt{\frac{\delta_{2\mathbf{s},\mathbf{M}}+t}{\delta_{2\mathbf{s},\mathbf{M}}-t}}$ and $\beta = \frac{1}{\alpha}$ in equation (3.4.6). With these values, we obtain

$$\begin{aligned}|\langle Ux, Uy\rangle| &\leq \frac{\alpha}{2\alpha^2 + 2}\left(\delta_{2\mathbf{s},\mathbf{M}} + t + \delta_{2\mathbf{s},\mathbf{M}} - t + 2\delta_{2\mathbf{s},\mathbf{M}}\right) \\ &\leq \frac{4\delta_{2\mathbf{s},\mathbf{M}}\alpha(\delta_{2\mathbf{s},\mathbf{M}} - t)}{4\delta_{2\mathbf{s},\mathbf{M}}} \leq \sqrt{\delta_{2\mathbf{s},\mathbf{M}}^2 - t^2}.\end{aligned}$$

□

Proof of Theorem 3.3.4. Let $x \in \mathbb{C}^m$ be an arbitrary m dimensional complex vector, and let

$$x^i := x_{\{M_{i-1}+1, M_{i-1}+2, \ldots, M_i\}}$$

denote the ith level of x. For an arbitrary vector $v = (v_1, v_2, \ldots, v_n)$, we define $|v|$ to be the vector $(|v_1|, |v_2|, \ldots, |v_n|)$. Let S_0^i denote the indexes of the s_ith largest elements of $|x^i|$, and $S_0 := \bigcup_{i=1}^l S_0^i$. We then define S_1^i to be the indexes of the s_ith largest elements of $|x^i|$ that are not contained in S_0^i (if there are fewer that s_i elements remaining, we simply take the indexes of any remaining elements of $|x^i|$) and define $S_1 := \bigcup_{i=1}^l S_1^i$. In general, we can make a similar definition to form a collection of index sets labelled $(S_j^i)_{i=1,2\ldots,l,j=1,2,\ldots}$ and corresponding $(\mathbf{s}, \mathbf{M})$-sparse S_j.

These definitions and the fact that $(\mathbf{s}, \mathbf{M})$ covers U implies that if $\Omega = \bigcup_{j \geq 0} S_j$ then $x_\Omega = x$. By the definition of S_0, $\|x_\Lambda\|_2 \leq \|x_{S_0}\|_2$ and $\|x_{S_0^c}\|_1 \leq \|x_{\Lambda^c}\|_1$ whenever Λ is $(\mathbf{s}, \mathbf{M})$-sparse. It will suffice to verify that

$$\sqrt{\tilde{s}}\|x_{S_0}\|_2 \leq \rho\|x_{S_0^c}\|_1 + \tau\sqrt{\tilde{s}}\|Ux\|_2 \tag{3.4.8}$$

holds for some $\rho \in (0,1)$ and $\tau > 0$. Indeed, if (3.4.8) holds then for $(\mathbf{s}, \mathbf{M})$-sparse sets Λ,

$$\begin{aligned}\sqrt{\tilde{s}}\|x_\Lambda\|_2 \leq \sqrt{\tilde{s}}\|x_{S_0}\|_2 &\leq \rho\|x_{S_0^c}\|_1 + \tau\sqrt{\tilde{s}}\|Ux\|_2 \\ &\leq \rho\|x_{\Lambda^c}\|_1 + \tau\sqrt{\tilde{s}}\|Ux\|_2\end{aligned}$$

as required in Theorem 3.3.8,. Set

$$\|Ux_{S_0}\|_2^2 = (1+t)\|x_{S_0}\|_2^2. \tag{3.4.9}$$

Clearly, $|t| \leq \delta_{\mathbf{s},\mathbf{M}}$. Then

$$\|Ux_{S_0}\|_2^2 = \langle Ux_{S_0}, Ux_{S_0}\rangle = \langle Ux_{S_0}, Ux\rangle - \sum_{j \geq 1} \left\langle Ux_{S_0}, Ux_{S_j}\right\rangle. \tag{3.4.10}$$

where we have used $x_\Omega = x$. Using the Cauchy-Schwarz inequality and (3.4.9) yields

$$|\langle Ux_{S_0}, Ux\rangle| \leq \|Ux_{S_0}\|_2\|Ux\|_2 \leq \sqrt{1+t}\,\|x_{S_0}\|_2\|Ux\|_2. \tag{3.4.11}$$

Furthermore, we can use Lemma 3.4.6 to see that

$$\begin{aligned}\left|\sum_{j \geq 1} \left\langle Ux_{S_0}, Ux_{S_j}\right\rangle\right| &\leq \sqrt{\delta_{2\mathbf{s},\mathbf{M}}^2 - t^2} \sum_{j \geq 1} \|x_{S_0}\|_2\|x_{S_j}\|_2 \\ &\leq \|x_{S_0}\|_2\sqrt{\delta_{2\mathbf{s},\mathbf{M}}^2 - t^2} \sum_{i=1}^l \sum_{j \geq 1} \|x_{S_j^i}\|_2.\end{aligned} \tag{3.4.12}$$

Combining (3.4.9),(3.4.10),(3.4.11) and (3.4.12) yields

$$(1+t)\|x_{S_0}\|_2^2 \leq \sqrt{1+t}\,\|x_{S_0}\|_2\|Ux\|_2 + \|x_{S_0}\|_2\sqrt{\delta^2_{\mathbf{2s},\mathbf{M}} - t^2}\sum_{i=1}^{l}\sum_{j\geq 1}\|x_{S_j^i}\|_2. \quad (3.4.13)$$

If $|S_j^i| = s_i$ then let $x_{i,j}^+$ (correspondingly $x_{i,j}^-$) be the largest element of $\left|x_{S_j^i}\right|$ (correspondingly the smallest element of $\left|x_{S_j^i}\right|$). If S_j^i is non-empty with fewer than s_i elements then we set $x_{i,j}^+$ to be the largest element of $\left|x_{S_j^i}\right|$ and $x_{i,j}^- = 0$. Finally, when $S_j^i = \emptyset$, we let $x_{i,j}^+ = x_{i,j}^- = 0$. It is clear then that $x_{i,j+1}^+ \leq x_{i,j}^-$.

Since $x_{S_j^i}$ contains at most s_i non-zero elements, we can apply the norm inequality for ℓ^1 and ℓ^2 (Lemma 3.4.5) to obtain

$$\|x_{S_j^i}\|_2 \leq \frac{1}{\sqrt{s_i}}\|x_{S_j^i}\|_1 + \frac{\sqrt{s_i}}{4}\left(x_{i,j}^+ - x_{i,j}^-\right)$$

for any $i = 1, 2, \ldots, l$ and $j \in \mathbb{N}$. Therefore

$$\begin{aligned}
\sum_{j\geq 1}\|x_{S_j^i}\|_2 &\leq \sum_{j\geq 1}\left(\frac{1}{\sqrt{s_i}}\|x_{S_j^i}\|_1\right) + \frac{\sqrt{s_i}}{4}\sum_{j\geq 1}\left(x_{i,j}^+ - x_{i,j}^-\right) \\
&= \sum_{j\geq 1}\left(\frac{1}{\sqrt{s_i}}\|x_{S_j^i}\|_1\right) + \frac{\sqrt{s_i}}{4}\left(x_{i,1}^+ + \sum_{j\geq 2}x_{i,j}^+ - \sum_{j\geq 1}x_{i,j}^+\right) \\
&= \sum_{j\geq 1}\left(\frac{1}{\sqrt{s_i}}\|x_{S_j^i}\|_1\right) + \frac{\sqrt{s_i}}{4}\left(x_{i,1}^+ + \sum_{j\geq 1}\left(x_{i,j+1}^+ - x_{i,j}^-\right)\right) \\
&\leq \sum_{j\geq 1}\left(\frac{1}{\sqrt{s_i}}\|x_{S_j^i}\|_1\right) + \frac{\sqrt{s_i}}{4}x_{i,1}^+
\end{aligned}$$

where the last inequality follows because $x_{i,j+1}^+ - x_{i,j}^- \leq 0$. Additionally, $\sqrt{s_i}x_{i,1}^+ = \sqrt{s_i}\|x_{S_1^i}\|_\infty \leq \|x_{S_0^i}\|_2$ because each element of $\left|x_{S_0^i}\right|$ is larger than $x_{i,1}^+$. We conclude that

$$\begin{aligned}
\sum_{i=1}^{l}\sum_{j\geq 1}\|x_{S_j^i}\|_2 &\leq \sum_{j\geq 1}\sum_{i=1}^{l}\frac{1}{\sqrt{s_i}}\|x_{S_j^i}\|_1 + \sum_{i=1}^{l}\frac{1}{4}\|x_{S_0^i}\|_2 \\
&\leq \frac{1}{\min\sqrt{s_i}}\sum_{j\geq 1}\sum_{i=1}^{l}\|x_{S_j^i}\|_1 + \frac{1}{4}\sqrt{l}\|x_{S_0}\|_2 \\
&\leq \frac{1}{\min\sqrt{s_i}}\sum_{j\geq 1}\|x_{S_j}\|_1 + \frac{1}{4}\sqrt{l}\|x_{S_0}\|_2 \leq \frac{1}{\min\sqrt{s_i}}\left\|x_{\bigcup_{j\geq 1}S_j}\right\|_1 + \frac{1}{4}\sqrt{l}\|x_{S_0}\|_2
\end{aligned}$$

where the second inequality follows from the Cauchy-Schwarz inequality applied to

$(\underbrace{1,1,\ldots,1}_{l})$ and $(\|x_{S_0^1}\|_2, \|x_{S_0^2}\|_2, \ldots, \|x_{S_0^l}\|_2)$ and the third and fourth inequalities follow from the disjoint supports of the vectors $x_{S_j^i}$ and $x_{S_{j'}^{i'}}$ whenever $i \neq i'$ or $j \neq j'$. By $x_\Omega = x$ and the disjointedness of S_i, S_j for $i \neq j$, $\bigcup_{j\geq 1} S_j = S_0^c$ so

$$\sum_{i=1}^{l}\sum_{j\geq 1}\|x_{S_j^i}\|_2 \leq \frac{1}{\min\sqrt{s_i}}\|x_{S_0^c}\|_1 + \frac{1}{4}\sqrt{l}\|x_{S_0}\|_2. \tag{3.4.14}$$

Dividing (3.4.13) by $\|x_{S_0}\|_2$ and employing (3.4.14) yields

$$(1+t)\|x_{S_0}\|_2 \leq \sqrt{1+t}\|Ux\|_2 + \sqrt{\delta_{2\mathbf{s},\mathbf{M}}^2 - t^2}\left(\frac{1}{\min\sqrt{s_i}}\|x_{S_0^c}\|_1 + \frac{1}{4}\sqrt{l}\|x_{S_0}\|_2\right). \tag{3.4.15}$$

Let $g(t) := \frac{\delta_{2\mathbf{s},\mathbf{M}}^2 - t^2}{(1+t)^2}$ for $|t| \leq \delta_{2\mathbf{s},\mathbf{M}}$. It is clear that $g(\delta_{2\mathbf{s},\mathbf{M}}) = g(-\delta_{2\mathbf{s},\mathbf{M}}) = 0$. Furthermore, g is differentiable. Therefore g attains its maximum at $t_{\max}$, where $g'(t_{\max}) = 0$. A simple calculation shows us that $t_{\max} = -\delta_{2\mathbf{s},\mathbf{M}}^2$ (note that by the assumption (3.3.1), $\delta_{2\mathbf{s},\mathbf{M}}^2 \leq \delta_{2\mathbf{s},\mathbf{M}}$). Thus $g(t) \leq g(-\delta_{2\mathbf{s},\mathbf{M}}^2) = \frac{\delta_{2\mathbf{s},\mathbf{M}}^2}{1-\delta_{2\mathbf{s},\mathbf{M}}^2}$. Additionally, $\frac{1}{\sqrt{1+t}} \leq \frac{1}{\sqrt{1-\delta_{2\mathbf{s},\mathbf{M}}}}$. Combining this with (3.4.15) yields

$$\begin{aligned}\|x_{S_0}\|_2 &\leq \frac{1}{\sqrt{1+t}}\|Ux\|_2 + \sqrt{g(t)}\left(\frac{1}{\min\sqrt{s_i}}\|x_{S_0^c}\|_1 + \frac{1}{4}\sqrt{l}\|x_{S_0}\|_2\right)\\ &\leq \frac{1}{\sqrt{1-\delta_{2\mathbf{s},\mathbf{M}}}}\|Ux\|_2 + \frac{\delta_{2\mathbf{s},\mathbf{M}}}{\sqrt{1-\delta_{2\mathbf{s},\mathbf{M}}^2}}\left(\frac{1}{\min\sqrt{s_i}}\|x_{S_0^c}\|_1 + \frac{1}{4}\sqrt{l}\|x_{S_0}\|_2\right).\end{aligned}$$

A simple rearrangement gives

$$\|x_{S_0}\|_2 \leq \frac{\sqrt{1+\delta_{2\mathbf{s},\mathbf{M}}}}{\sqrt{1-\delta_{2\mathbf{s},\mathbf{M}}^2} - \delta_{2\mathbf{s},\mathbf{M}}\sqrt{l}/4}\|Ux\|_2 + \frac{\delta_{2\mathbf{s},\mathbf{M}}}{\min\sqrt{s_i}\left(\sqrt{1-\delta_{2\mathbf{s},\mathbf{M}}^2} - \delta_{2\mathbf{s},\mathbf{M}}\sqrt{l}/4\right)}\|x_{S_0^c}\|_1 \tag{3.4.16}$$

provided

$$\sqrt{1-\delta_{2\mathbf{s},\mathbf{M}}^2} - \delta_{2\mathbf{s},\mathbf{M}}\sqrt{l}/4 > 0. \tag{3.4.17}$$

Multiplying (3.4.16) by $\sqrt{\tilde{s}}$ yields

$$\begin{aligned}\sqrt{\tilde{s}}\|x_{S_0}\| &\leq \sqrt{\tilde{s}}\frac{\sqrt{1+\delta_{2\mathbf{s},\mathbf{M}}}}{\sqrt{1-\delta_{2\mathbf{s},\mathbf{M}}^2}-\delta_{2\mathbf{s},\mathbf{M}}\sqrt{l}/4}\|Ux\|_2 \\ &\quad+\frac{\delta_{2\mathbf{s},\mathbf{M}}\sqrt{\tilde{s}}}{\min\sqrt{s_i}\left(\sqrt{1-\delta_{2\mathbf{s},\mathbf{M}}^2}-\delta_{2\mathbf{s},\mathbf{M}}\sqrt{l}/4\right)}\|x_{S_0^c}\|_1 \\ &\leq \tau\sqrt{\tilde{s}}\,\|Ux\|_2+\frac{\delta_{2\mathbf{s},\mathbf{M}}}{\sqrt{1-\delta_{2\mathbf{s},\mathbf{M}}^2}-\delta_{2\mathbf{s},\mathbf{M}}\sqrt{l}/4}\sqrt{\sum_{k=1}^{l}\frac{s_k}{\min s_i}}\|x_{S_0^c}\|_1 \\ &\leq \tau\sqrt{\tilde{s}}\,\|Ux\|_2+\frac{\delta_{2\mathbf{s},\mathbf{M}}\sqrt{l\eta_{\mathbf{s},\mathbf{M}}}}{\sqrt{1-\delta_{2\mathbf{s},\mathbf{M}}^2}-\delta_{2\mathbf{s},\mathbf{M}}\sqrt{l}/4}\|x_{S_0^c}\|_1\end{aligned}$$

where $\tau=\frac{\sqrt{1+\delta_{2\mathbf{s},\mathbf{M}}}}{\sqrt{1-\delta_{2\mathbf{s},\mathbf{M}}^2}-\delta_{2\mathbf{s},\mathbf{M}}\sqrt{l}/4}$. It is clear that (3.4.8) is satisfied if condition (3.4.17) holds and

$$\frac{\delta_{2\mathbf{s},\mathbf{M}}\sqrt{l\eta_{\mathbf{s},\mathbf{M}}}}{\sqrt{1-\delta_{2\mathbf{s},\mathbf{M}}^2}-\delta_{2\mathbf{s},\mathbf{M}}\sqrt{l}/4}<1 \quad \text{or equivalently } \delta_{2\mathbf{s},\mathbf{M}}<\frac{1}{\sqrt{l\left(\sqrt{\eta_{\mathbf{s},\mathbf{M}}}+\frac{1}{4}\right)^2+1}} \tag{3.4.18}$$

whilst (3.4.17) is equivalent to $\delta_{2\mathbf{s},\mathbf{M}}<\frac{1}{\sqrt{\frac{l}{16}+1}}$. Since

$$\frac{1}{\sqrt{l\left(\sqrt{\eta_{\mathbf{s},\mathbf{M}}}+\frac{1}{4}\right)^2+1}}\leq\frac{1}{\sqrt{\frac{l}{16}+1}}$$

it will suffice for (3.4.18) to hold, completing the proof. □

3.4.3 Proof of Theorem 3.3.5 and 3.3.6

Proof of Theorem 3.3.5. The ideas behind the counterexample in this proof are similar to those in Theorem 3.2 in [27]. We prove this theorem in three stages. First we shall construct the matrix U. Next we shall show that our construction does indeed have a RIP in levels constant satisfying (3.3.4). Finally, we shall explain why z^1 exists.

Step I: Set $n=C+C^2$, where the non-negative integer C is much greater than a (we shall give a precise choice of C later). Let $x^1\in\mathbb{C}^n$ be the vector

$$x^1:=\lambda(\underbrace{C,C,\ldots,C}_{C},\underbrace{1,1,\ldots,1}_{C^2}).$$

With this definition, the first C elements of x^1 have value $C\lambda$ and the next C^2 elements have value λ. Our $(\mathbf{s},\mathbf{M})$ sparsity pattern is given by $\mathbf{s}=(1,C^2)$ and $\mathbf{M}=(0,C,C+C^2)$. Clearly, by the definition of the ratio constant, $\eta_{\mathbf{s},\mathbf{M}}=C^2$ (in particular, $\eta_{\mathbf{s},\mathbf{M}}$ is finite). Choose $\lambda=\frac{1}{\sqrt{C^3+C^2}}$ so that $\|x^1\|_2=1$. By using this fact, we can form an orthonormal basis of $\mathbb{C}^{C+C^2}$ that includes x^1. We can write this basis as $(x^i)_{i=1}^{C+C^2}$.

Finally, for a vector $v \in \mathbb{C}^{C+C^2}$, we define the linear map U by

$$U'v := \sum_{i=2}^{C+C^2} v^i x^i \text{ where } v = \sum_{i=1}^{C+C^2} v^i x^i$$

In particular, notice that the nullspace of U is precisely the space spanned by x^1, and that $v^i = \langle v, x^i \rangle$.

Step II: Let γ be an $(a\mathbf{s}, \mathbf{M})$ sparse vector. Our aim will be to estimate $\big| \|U\gamma\|_2^2 - \|\gamma\|_2^2 \big|$. Clearly, $\|U\gamma\|_2^2 - \|\gamma\|_2^2 = -|\gamma^1|^2$, where γ^1 is the coefficient of x^1 in the expansion of γ in the basis (x^i). Therefore, to show that U satisfies the RIP_L we will only need to bound $|\gamma^1| = |\langle \gamma, x^1 \rangle|$. Let S be the support of γ. Then

$$|\langle \gamma, x^1 \rangle| = |\langle \gamma_S, x^1 \rangle| = |\langle \gamma, x_S^1 \rangle| \leq \|\gamma\|_2 \|x_S^1\|_2 \leq \lambda \|\gamma\|_2 \sqrt{aC^2 + C^2}$$

where we have used Cauchy-Schwarz in the first inequality and in the second inequality we have used the fact that x_S^1 has at most a elements of size λC and at most C^2 elements of size λ. From the definition of λ we get $\big|\langle \gamma, x^1 \rangle\big| \leq \sqrt{\frac{a+1}{C+1}} \|\gamma\|_2$. Therefore,

$$\big| \|U\gamma\|_2^2 - \|\gamma\|_2^2 \big| = |\langle \gamma, x^1 \rangle|^2 \leq \frac{a+1}{C+1} \|\gamma\|_2^2.$$

By the assumption that $f(x) = o(x^{\frac{1}{2}})$, we can find a $C \in \mathbb{N}$ sufficiently large so that $\frac{a+1}{C+1} \leq \frac{1}{|f(C^2)|}$. Then $\delta_{a\mathbf{s},\mathbf{M}} \leq \frac{1}{|f(\eta_{\mathbf{s},\mathbf{M}})|}$ as claimed.

Step III: Let

$$z^1 := (C, \underbrace{0, 0, \ldots, 0, 0}_{C-1}, \underbrace{1, 1, \ldots, 1}_{C^2}), \qquad z^2 := (0, \underbrace{C, C, \ldots, C, C}_{C-1}, \underbrace{0, 0, \ldots, 0}_{C^2}).$$

It is clear that z^1 is $(\mathbf{s}, \mathbf{M})$-sparse. Additionally, $\|z^1\|_1 = C^2 + C$ and $\| - z^2\|_1 = (C-1)\, C = C^2 - C$. Because $U(z^1 + z^2) = U(x^1)/\lambda = 0$, we have $U(-z^2) = U(z^1)$. Since the kernel of U is of dimension 1, the only vectors z which satisfy $U(z) = U(z^1)$ are $z = z^1$ and $z = -z^2$. Moreover, $\|z^1\|_1 > \| - z^2\|_1$. Consequently

$$z^1 \notin \arg\min \|z\|_1 \text{ such that } Uz = Uz^1.$$

□

Proof of Theorem 3.3.6. The proof of this theorem is almost identical to that of Theorem 3.3.5, so we shall omit details here. Again, we set x^1 so that

$$x^1 := \lambda(\underbrace{C, C, \ldots, C}_{C}, \underbrace{1, 1, \ldots, 1}_{C^2})$$

where $C \gg a$. We choose λ so that $\|x^1\|_2 = 1$. In contrast to the proof of Theorem

3.3.5, we take

$$\mathbf{s} = (\underbrace{1, 1, 1, \ldots, 1}_{C^2+1}), \qquad \mathbf{M} = (0, C, C+1, \ldots, C+C^2-1, C+C^2).$$

This time, there are C^2+1 levels and the ratio constant $\eta_{\mathbf{s},\mathbf{M}}$ is equal to 1. Once again, we produce an orthonormal basis of $\mathbb{C}^{C+C^2}$ that includes x^1, which we label $(x^i)_{i=1}^{C+C^2}$ and we define the linear map U by

$$U'v := \sum_{i=2}^{C+C^2} v^i x^i \text{ where } v = \sum_{i=1}^{C+C^2} v^i x^i.$$

The same argument as before proves that for any $(a\mathbf{s}, \mathbf{M})$-sparse γ,

$$\left| \|U'\gamma\|_2^2 - \|\gamma\|_2^2 \right| \leq \frac{a+1}{C+1} \|\gamma\|_2^2$$

and again, taking C sufficiently large so that $\frac{a+1}{C+1} \leq \frac{1}{|f(C^2+1)|}$ yields $\delta_{a\mathbf{s},\mathbf{M}} \leq \frac{1}{|f(l)|}$. The proof of the existence of z^1 is the identical to Step III in the proof of Theorem 3.3.5. □

3.4.4 Proof of Theorem 3.3.7

Proof. Once again, we prove this theorem in three stages. First we shall construct the matrix U'. Next, we shall show that the matrix U' has a sufficiently small RIP_L constant. Finally, we shall explain why both z^1 and z exist.

Step I: Let x^1 be the vector

$$x^1 := \lambda(\underbrace{0, 0, \ldots, 0}_{C^2}, \underbrace{1, 1, \ldots, 1}_{\omega(\rho,C)+1})$$

where $\omega(\rho, C) = \text{ceil}(\frac{2C}{\rho})$ for a fixed $\rho \in (0, 1)$ which we will specify later, $\text{ceil}(a)$ denotes the smallest integer greater than or equal to a and C is an integer greater than 1. In other words, the first C^2 elements of x^1 have value 0 and the next $\omega(\rho, C) + 1$ elements have value λ. We choose λ so that $\|x^1\|_2 = 1$ and C so that $C^2 > \omega(\rho, C)$ and choose our $(\mathbf{s}, \mathbf{M})$ sparsity pattern so that $\mathbf{s} = (C^2, 1)$ and $\mathbf{M} = (0, C^2, C^2+\omega(\rho, C)+1)$. By the definition of the ratio constant, $\eta_{\mathbf{s},\mathbf{M}} = C^2$ (in particular, $\eta_{\mathbf{s},\mathbf{M}}$ is finite). Because $\|x^1\|_2 = 1$, we can form an orthonormal basis of $\mathbb{C}^{C^2+\omega(\rho,C)+1}$ that includes x^1 which we can write as $(x^i)_{i=1}^{C^2+\omega(\rho,C)+1}$. Finally, for a vector $v \in \mathbb{C}^{C^2+\omega(\rho,C)+1}$, we define the linear map U' by

$$U'v := \frac{\sqrt{2}w}{\tau} \text{ where } w = \sum_{i=2}^{C^2+\omega(\rho,C)+1} v^i x^i \text{ and } v = v^1 x^1 + w$$

In particular, notice that the nullspace of U' is precisely the space spanned by x^1, and that $v^i = \langle v, x^i \rangle$.

Step II: Let γ be an $(a\mathbf{s}, \mathbf{M})$ sparse vector. For the purposes of proving Theorem 3.3.7, it will suffice to take $\tau = \sqrt{2}$. Then

$$\|U'\gamma\|_2^2 - \|\gamma\|_2^2 = -|\gamma^1|^2,$$

where γ^1 is the coefficient corresponding to x^1 in the expansion of γ in the basis (x^i). As in the proof of Theorem 3.3.5, $|\gamma^1| = |\langle \gamma, x^1 \rangle|$. Let S be the support of γ. Then

$$|\langle \gamma_S, x^1 \rangle| = |\langle \gamma, x^1_S \rangle| \leq \|\gamma\|_2 \|x^1_S\|_2 \leq \lambda \|\gamma\|_2 \sqrt{a}$$

where we have used Cauchy-Scharwz in the first inequality and in the second inequality we have used the fact that x^1_S has at most a elements of size λ. It is easy to see that $\lambda = \frac{1}{\sqrt{\omega(\rho,C)+1}}$. Therefore,

$$\left| \|U'\gamma\|_2^2 - \|\gamma\|_2^2 \right| = |\langle \gamma, x^1 \rangle|^2 \leq \frac{a}{\omega(\rho, C) + 1} \|\gamma\|_2^2 \leq \frac{\rho a}{2C}.$$

because $\omega(\rho, C) \geq \frac{2C}{\rho}$. By the assumption that $g(\eta_{\mathbf{s},\mathbf{M}}) \leq \frac{1}{A}\sqrt{\eta_{\mathbf{s},\mathbf{M}}}$ for some $A > 0$ and $\eta_{\mathbf{s},\mathbf{M}}$ sufficiently large, and the fact that $\eta_{\mathbf{s},\mathbf{M}} = C^2$, we must have $\frac{A}{C} \leq \frac{1}{g(\eta_{\mathbf{s},\mathbf{M}})}$. If we take ρ sufficiently small and C sufficiently large, then

$$\delta_{a\mathbf{s},\mathbf{M}} < \frac{\rho a}{2C} \leq \frac{A}{C} \leq \frac{1}{g(\eta_{\mathbf{s},\mathbf{M}})}.$$

as claimed.

Step III: Let $z^1 := x^1$ and set z to be the 0 vector in $\mathbb{C}^{C^2+\omega(\rho,C)+1}$. Because x^1 is in the kernel of U', $U'(z - z^1) = 0$. Furthermore, it is obvious that $\|z\|_1 \leq \|z^1\|_1$. Additionally, $\|z^1\|_2 = 1$ and

$$\frac{\sigma_{\mathbf{s},\mathbf{M}}(z^1)_1}{\sqrt{\tilde{s}}} = \lambda \frac{\omega(\rho, C)}{\sqrt{C^2+1}} \leq \frac{\omega(\rho, C)}{\sqrt{\omega(\rho, C)\,(C^2+1)}} \leq \sqrt{\frac{2C+1}{\rho\,(C^2+1)}} \leq \sqrt{\frac{3}{\rho\sqrt{\eta_{\mathbf{s},\mathbf{M}}}}}$$

since $\eta_{\mathbf{s},\mathbf{M}} = C^2$ and $\omega(\rho, C) \leq 2C/\rho + 1 \leq (2C+1)/\rho$. Because $f(\eta_{\mathbf{s},\mathbf{M}}) = o(\eta_{\mathbf{s},\mathbf{M}}^{\frac{1}{4}})$,

$$\frac{\sigma_{\mathbf{s},\mathbf{M}}(z^1)_1}{\sqrt{\tilde{s}}} f(\eta_{\mathbf{s},\mathbf{M}}) \to 0, \quad \eta_{\mathbf{s},\mathbf{M}} \to \infty.$$

The desired result follows by taking $\eta_{\mathbf{s},\mathbf{M}}$ sufficiently large so that

$$\|z - z^1\|_2 = 1 > \frac{\sigma_{\mathbf{s},\mathbf{M}}(z^1)_1}{\sqrt{\tilde{s}}} f(\eta_{\mathbf{s},\mathbf{M}}).$$

□

Proof of part 2. The proof of part 2 follows with a few alterations to the previous case.

We now use the sparsity pattern

$$\mathbf{s} = (\underbrace{1, 1, 1, \ldots, 1}_{C^2}, 1) \quad \text{and} \quad \mathbf{M} = (0, 1, 2, \ldots, C^2, C^2 + \omega(\rho, C) + 1).$$

In this case, $\eta_{\mathbf{s},\mathbf{M}} = 1$ and $l = C^2 + 1$. The result follows by simply employing the same matrix U' with this new sparsity pattern. □

3.4.5 Proofs of Theorem 3.3.9

The counterexample for Theorem 3.3.9 is the same as the one used in the proof of Theorem 3.3.7. In that case, the matrix depended on three parameters: C, τ and ρ. We show that U' satisfies the ℓ^2 robust nullspace property of order $(\mathbf{s}, \mathbf{M})$ with parameters ρ and τ. The existence of z^1 and z is identical to Step III in the proof of Theorem 3.3.7.

Proof of part 1. Firstly, if $T \subset S$ then for any $v \in \mathbb{C}^{C^2+\omega(\rho,C)+1}$, we have

$$\|v_T\|_2 \leq \|v_S\|_2 \text{ and } \frac{\rho}{\sqrt{\tilde{s}}}\|v_{T^c}\|_1 + \tau\|U'v\|_2 \geq \frac{\rho}{\sqrt{\tilde{s}}}\|v_{S^c}\|_1 + \tau\|U'v\|_2$$

so it will suffice to prove that U' satisfies (3.2.2) for $(\mathbf{s}, \mathbf{M})$-sparse sets S with $|S| = \tilde{s}$. As before, we set $U'v := \sqrt{2}w/\tau$ where w is defined as in the proof of Theorem 3.3.7. Let us consider a set S such that $|S| = \tilde{s}$. Because w_S and w_{S^c} have disjoint support, by the Cauchy-Schwarz inequality applied to the vectors $(1, 1)$ and $(\|w_S\|_2, \|w_{S^c}\|_2)$ we get $\sqrt{2}\|w\|_2 \geq \|w_S\|_2 + \|w_{S^c}\|_2$. Therefore,

$$\tau\|U'v\|_2 \geq \sqrt{2}\|w\|_2 \geq \|w_S\|_2 + \|w_{S^c}\|_2. \tag{3.4.19}$$

Furthermore, because $|S| = \tilde{s} \geq |S^c|$ (recall that $|S| = C^2 + 1$ and that C was chosen so that $C^2 > \omega(\rho, C) = |S^c|$) and $\rho \in (0, 1)$

$$\|w_{S^c}\|_2 \geq \frac{1}{\sqrt{|S^c|}}\|w_{S^c}\|_1 \geq \frac{1}{\sqrt{\tilde{s}}}\|w_{S^c}\|_1 \geq \frac{\rho}{\sqrt{\tilde{s}}}\|w_{S^c}\|_1 \tag{3.4.20}$$

Combining 3.4.19 and (3.4.20) gives

$$\begin{aligned}\tau\|U'v\|_2 + \frac{\rho}{\sqrt{\tilde{s}}}\|v_{S^c}\|_1 \geq \|w_S\|_2 + \frac{\rho}{\sqrt{\tilde{s}}}\|w_{S^c}\|_1 + \frac{\rho}{\sqrt{\tilde{s}}}\|v_{S^c}\|_1 &\geq \|w_S\|_2 + \frac{\rho}{\sqrt{\tilde{s}}}\|v_{S^c} - w_{S^c}\|_1 \\ &\geq \|w_S\|_2 + \frac{\rho}{\sqrt{\tilde{s}}}\|v^1 x^1_{S^c}\|_1\end{aligned} \tag{3.4.21}$$

We shall now aim to bound $\|v^1 x^1_S\|_2$ in terms of $\|v^1 x^1_{S^c}\|_1$. We have

$$\|v^1 x^1_S\|_2 \leq \lambda |v^1| \tag{3.4.22}$$

since at most one element of x^1_S is non-zero and its value will be at most λ. Additionally,

since each element of $x^1_{S^c}$ has value λ and there are at least $\frac{2C}{\rho}$ of them

$$\rho\|v^1 x^1_{S^c}\|_1 = \rho|v^1|\|x^1_{S^c}\|_1 \geq \frac{2\lambda C}{\rho}\rho|v^1| \geq 2\lambda C|v^1|.$$

Therefore,

$$\frac{\rho}{\sqrt{\tilde{s}}}\|v^1 x^1_{S^c}\|_1 \geq \frac{2\lambda C}{\sqrt{C^2+1}}|v^1| \geq \lambda|v^1|. \tag{3.4.23}$$

Using (3.4.22) and (3.4.23), we have $\|v^1 x^1_S\|_2 \leq \frac{\rho}{\sqrt{\tilde{s}}}\|v^1 x^1_{S^c}\|_1$. We can conclude the proof that U' satisfies the ℓ^2 robust nullspace property by combining this result with (3.4.21) as follows:

$$\|v_S\|_2 \leq \|v^1 x^1_S\|_2 + \|w_S\|_2 \leq \frac{\rho}{\sqrt{\tilde{s}}}\|v^1 x^1_{S^c}\|_1 + \|w_S\|_2 \leq \tau\|U'v\|_2 + \frac{\rho}{\sqrt{\tilde{s}}}\|v_{S^c}\|_1.$$

□

Proof of part 2. The proof of part 2 is identical. We simply adapt the sparsity pattern so that

$$\mathbf{s} = (\underbrace{1,1,1,\ldots,1}_{C^2},1) \text{ and } \mathbf{M} = (0,1,2,\ldots,C^2,C^2+\omega(\rho,C)+1).$$

We can apply the proceeding argument with this new sparsity pattern to obtain the required result. □

Chapter 4

Computational barriers in information theory

Recall from chapter 1 that we are interested in the following questions:

Q3: For any $\epsilon > 0$, is there an algorithm that executes in finite time and computes an approximate solution no further than ϵ away from a true solution? The algorithm can choose the input accuracy $\hat{\epsilon}$ to be as small as desired (as a function of ϵ and the input) to produce the output.

Q4: Can we get a positive answer to the previous question if, for a given $\epsilon > 0$, $\hat{\epsilon}$ is uniformly bounded across all possible inputs?

Q5: Suppose that the answer to the two previous questions is negative. Can we replace the requirement that the algorithm has to exist for each $\epsilon > 0$ with one that insists that the algorithm must only work for a 'small' $\epsilon > 0$ and get a positive answer?

which when phrased in the language of chapter 2 read as follows

Q3': Is $\{\Xi, \Omega, \mathcal{M}, \Lambda\}^{\Delta_1} \in \Delta_1^G$?

Q4': If the answer to the previous question is positive, can we form an algorithm Γ that solves $\{\Xi, \Omega, \mathcal{M}, \Lambda\}$ to output accuracy ϵ with runtime $T(\Gamma, \epsilon) < \infty$?

Q5': If either of the previous questions are answered in the negative, what are the breakdown epsilons?

4.1 Linear programming and linear systems

The linear programming problem is stated as (for fixed c) $\text{argmin}_{x \in \mathbb{R}^N} c^T x$ such that $Ax = y, x \geq 0$, whereas the linear systems problem is to find an x such that $Ax = y$. One of the purposes of the results in this chapter is to demonstrate and explain the general non-computability issues that can occur if there are few restrictions on the primary set and contrast them with linear systems.

Let $\Xi_{\mathrm{LP}}(\iota)$ denote the solution set, with input $\iota \in \Omega$ where $\iota = (y, A)$, of the linear programming problem and let $\Xi_{\mathrm{LS}}(\iota)$ denote the solution set, with input $\iota \in \Omega$ where $\iota = (y, A)$ of the linear systems problem.

Theorem 4.1.1. *We have contrasting results for linear programming and linear systems:*

1. *For any dimensions m, N, there exists a set of inputs Ω such that for any $\iota \in \Omega$, $\iota = (y, A)$ for some vector $y \in \mathbb{R}^m$, $A \in \mathbb{R}^{m\times N}$, $\Xi_{\mathrm{LP}}(\iota)$ has a unique minimiser and $\|\iota\| \leq 1$, yet*

$$\{\Xi_{\mathrm{LP}}, \Omega\}^{\Delta_1} \notin \Delta_1^G, \qquad \epsilon_{\mathrm{B}}^{\mathrm{s}}, \epsilon_{\mathbb{P}\mathrm{B}}^{\mathrm{s}}(p) \geq 1/2,$$

   for $p \in (0, 1/2)$. Moreover, $\{\Xi_{\mathrm{LP}}, \Omega\}^{\Delta_1} \in \Delta_1^A$ implies decidability of the Halting Problem.

2. *For any dimensions m, N let Ω be the set of $y \in \mathbb{R}^m$, $A \in \mathbb{R}^{m\times N}$ such that the linear systems problem with input y and A has a unique solution. Then $\{\Xi_{\mathrm{LS}}, \Omega\}^{\Delta_1} \in \Delta_1^A$.*

Remark 4.1.2. We should note at this point that we do not claim novelty on Theorem 4.1.1 part 2. Indeed, such a result was already shown in the computational analysis setting in [128]. We do provide proof of this variant in Chapter 5, however, our proof strategy is substantially similar to this existing work.

As the first non-computability result of this paper, it is worth examining Theorem 4.1.1 part 1 in closer detail. Firstly, we have a bounded input set which we guarantee has exactly one minimiser for every problem. The conclusion of the theorem implies that the linear programming problem is non-computable, and computability with a turing machine would imply that the halting problem is decidable. The statement about the breakdown epsilons describes the *degree of noncomputability* - firstly, the strong breakdown epsilon alludes to the fact that any algorithm will perform at least as badly as committing an error of $1/2$ for some inputs in the dataset.

For the probabilistic breakdown epsilon, the result says that a probabilistic algorithm will do no better than a deterministic one unless we are willing to accept a high probability of failing. To understand the relevance of the constant $1/2$, let us imagine a computational problem $\{\Xi, \Omega, \mathcal{M}, \Lambda\}$ for which $\cup_{\iota\in\Omega}\Xi(\iota)$ has cardinality 2. It is easy to achieve a probabilistic algorithm that outputs the correct answer with probability a half by choosing uniformly at random from the two possible values of $\Xi(\iota)$. Such an algorithm would clearly be of no practical interest - the result would be independent of the input.

Let us contrast this result with that of Theorem 4.1.1 part 2. We make the same assumptions on the input set as in part 1, but now by changing from the linear programming problem to the linear systems problem we change the computational problem from being outside of Δ_1^G to being inside Δ_1^A and thus computable. At first glance, the change from linear programming to linear systems may seem completely innocuous from a computability perspective. This result demonstrates that this intuition is wrong and shows just how subtle this theory can be.

The final result on linear programming shows that even with quite strong restrictions on the primary set, finding the minimiser of linear programming can still be a non-computable problem. By contrast, with the same set of inputs it will be perfectly possible to compute the objective function.

Theorem 4.1.3. *There exists a set of inputs Ω such that for all $\iota \in \Omega$ we can write $\iota = (y, A)$ for some vector $y \in \mathbb{R}^m$ and matrix $A \in \mathbb{R}^{m\times N}$ with bounded condition numbers in the following sense:* $\mathrm{Cond}(AA^*), C_{FP}(y, A), \mathrm{Cond}(\Xi) \leq 2$, $\|\iota\| \leq \sqrt{2}$, *yet*

$$\{\Xi_{\mathrm{LP}}, \Omega\}^{\Delta_1} \notin \Delta_1^G, \qquad \epsilon_{\mathrm{B}}^{\mathrm{s}} \geq \frac{1}{\sqrt{2}}$$

Furthermore, $\{\Xi_{\mathrm{LP}}, \Omega\}^{\Delta_1} \in \Delta_1^A$ implies decidability of LLPO. By contrast, let $\Xi_{\mathrm{LP}}^{\mathrm{Obj}}$ be defined on the same set Ω to be

$$\Xi_{\mathrm{LP}}^{\mathrm{Obj}}(\iota) := \min_{x\in\mathbb{R}^N} c^T x \text{ such that } y = Ax, \qquad \text{where} \iota = (y, A).$$

We have $\{\Xi_{\mathrm{LP}}^{\mathrm{Obj}}, \Omega\}^{\Delta_1} \in \Delta_0^A$.

This serves to highlight the important difference between finding the minimiser and computing the objective function. We stress at this point that in many applications one is more interested in the former than the latter. Thus a computability theory revolving around finding minimisers becomes of great importance.

Remark 4.1.4. Note that for most of the remaining results in this thesis, we shall discuss only the deterministic breakdown epsilons even though our techniques are easily able to show bounds on the probabilistic breakdown epsilons. This is because the typical algorithms used to solve all but the neural network problem are deterministic in nature.

4.2 Compressed sensing

As we shall see that basis pursuit and basis pursuit denoising have different computational properties, we will split our work here into two sections.

4.2.1 Basis pursuit

Firstly, recall that the basis pursuit problem is to find a solution to $\mathrm{argmin}_{x\in\mathbb{R}^N} \|x\|_1$ such that $Ux = y$.

We begin by discussing the basic non-computability results on basis pursuit. Firstly, basis pursuit is not a Δ_1 problem even when restricted to single minimiser problems and bounded input. Secondly, basis pursuit is still not a Δ_1 problem even under reasonable assumptions on all but the RCC condition number. More precisely:

Theorem 4.2.1 (Basis Pursuit)**.** *Let $\Xi(\iota)$ denote the solution set, with input $\iota \in \Omega$, of basis pursuit with l^1 norm.*

(i) For any dimensions $m, N \in \mathbb{N}$, there exists a set of inputs Ω such that for any $\iota \in \Omega$, $\iota = (y, U)$ for some vector $y \in \mathbb{R}^m$ and matrix $U \in \mathbb{R}^{m \times N}$, $\Xi(\iota)$ has a unique minimiser and $\|\iota\| \leq 1$ for all $\iota \in \Omega$, yet

$$\{\Xi, \Omega\}^{\Delta_1} \notin \Delta_1^G, \qquad \epsilon_{\mathrm{B}}^{\mathrm{s}} \geq \frac{1}{2}.$$

Moreover, $\{\Xi, \Omega\}^{\Delta_1} \in \Delta_1^A$ implies decidability of the Halting Problem.

(ii) For any dimensions $m, N \in \mathbb{N}$ with $m < N$, there exists a set of inputs Ω such that for all $\iota \in \Omega$ we can write $\iota = (y, U)$ for some vector $y \in \mathbb{R}^m$ and matrix $U \in \mathbb{R}^{m \times N}$ with bounded condition numbers in the following sense: $\mathrm{Cond}(UU^*), C_{FP}(y, U), \mathrm{Cond}(\Xi) \leq 2$, $\|\iota\| \leq \sqrt{2}$, yet

$$\{\Xi, \Omega\}^{\Delta_1} \notin \Delta_1^G, \qquad \epsilon_{\mathrm{B}}^{\mathrm{s}} \geq \frac{1}{\sqrt{2}}$$

Furthermore, $\{\Xi, \Omega\}^{\Delta_1} \in \Delta_1^A$ implies decidability of LLPO.

As a counterpart to this theorem, it is natural to ask for conditions on the input set that ensure that the problem is indeed in Δ_1^A. An initial attempt might be to show that we can establish a positive result if we assume that $\mathrm{Cond}(UU^*), C_{\mathrm{FP}}(y, U), \mathrm{Cond}(\Xi)$ and additionally C_{RCC} are all bounded.

We shall not argue whether or not such conditions lead to a computability result here. Instead, let us suppose that these conditions were able to prove a computability result. Our aim would then be to show that compressed sensing problems have these conditions and thus conclude that even though basis pursuit is not in Δ_1, if we only work with compressed sensing problems we can compute to arbitrary precision.

Thus a natural question is whether or not 'standard' compressed sensing problems have a small RCC condition number. Our next theorem shows that this is not the case, and thus a computability result that requires a small RCC would be inapplicable in explaining why compressed sensing techniques seem to perform well in practice. Moreover, we do in fact show that these 'standard' compressed sensing problems are in Δ_1, meaning that a small RCC is not necessary to guarantee computability for these important compressed sensing problems.

Theorem 4.2.2. *There exists a set of inputs Ω such that*

1. *If $y = Ux$ for some x is such that (y, U) is in Ω then the set of $U \in \mathbb{R}^{m \times N}$ so that $(Ux, U) \in \Omega$ is an open set.*
2. *There is an input $\iota = (\tilde{y}, \tilde{U}) \in \Omega$ such that $\tilde{U}$ is a Hadamard matrix and $C_{RCC}(\iota) = \infty$.*

However, despite the infinite RCC condition of the problem, we have

$$\{\Xi, \Omega\}^{\Delta_1} \in \Delta_1^A.$$

Moreover, there exists an algorithm Γ *such that for any* $\epsilon > 0$ *the algorithm has bounded minimum runtime i.e.*

$$T(\Gamma, \epsilon) < \infty.$$

Finally, the same result holds if we replace the word 'Hadamard' with 'Bernoulli'.

The issue of input sets with two minimisers is very subtle and the subject of much of the discussion in this thesis. On the one hand, 4.2.2 shows that if the inputs are arbitrarily close to two minimisers we may still be able to build an algorithm to solve the problem. On the other hand, we have the following result.

Theorem 4.2.3. *Let* $\{\Xi, \Omega\}$ *denote the computational problem of Basis Pursuit (BP). Suppose that there is input data* $\iota = (y, U) \in \Omega$ *for which there are at least two distinct elements in* $\Xi(\iota)$*. Suppose also that there is an* $\epsilon > 0$ *such that for every negative semidefinite diagonal matrix* D *with* $\|D\|_{\max} < \epsilon$ *we have* $(y, U + UD) \in \Omega$*. Then* $\{\Xi, \Omega\}^{\Delta_1} \notin \Delta_1^G$*. Moreover, if* $x^1, x^2 \in M_{min}^{BP}(\iota)$ *then* $\epsilon_{\mathrm{B}}^{\mathrm{s}} \geq \|x^1 - x^2\|_2/2$*.*

The set $M_{\min}^{\mathrm{BP}}(\iota)$ are the set of *minimisers with minimal support*. This key concept is used in several of the arguments to get lower bounds on the strong breakdown epsilon. More precisely, we define the following:

Definition 4.2.4 (Minimisers with minimal support)**.** *For feasible input* $y \in \mathbb{C}^m$*,* $U \in \mathbb{C}^{m \times N}$ *and parameters* $\delta \geq 0$*,* $\lambda > 0$*, we consider the* set $M_{\min}^{\mathrm{BP}}$ of basis pursuit denoising minimisers with minimal support $M_{\min}^{\mathrm{BP}}$ (respectively the set $M_{\min}^{\mathrm{UL}}$ of lasso minimisers with minimal support)*, defined by*

$$M_{min}^{\mathrm{BP}}(y, U, \delta) := \{x \in \Xi^{\mathrm{BP}}(y, U, \delta) \,|\, x' \in \Xi^{\mathrm{BP}}(y, U, \delta),\ supp(x') \subseteq supp(x) \Rightarrow x = x'\}$$
$$M_{min}^{\mathrm{UL}}(y, U, \lambda) := \{x \in \Xi^{\mathrm{UL}}(y, U, \lambda) \,|\, x' \in \Xi^{\mathrm{UL}}(y, U, \lambda),\ supp(x') \subseteq supp(x) \Rightarrow x = x'\}$$

Note that since we permit $\delta = 0$ *in the definition for basis pursuit denoising, so that basis pursuit is a special case.*

To give a condition that ensures that the problem is in Δ_1^A, we therefore take a different approach. We would like to guarantee that the same properties that were good for *recovery* are also good for *computation* (i.e. the RIP and nullspace property in levels from Chapter 3) We show that this is indeed the case subject to the existence of an *primal convergent* tower of algorithms:

Definition 4.2.5. *A* primal convergent tower *of algorithms for basis pursuit (denoising) is an arithmetic tower of algorithms* $\{\Gamma_n\}$ *of height one so that for given* $\epsilon > 0$*,* $\tilde{U} \in \mathbb{C}^{m \times N}$ *and* $\tilde{y} \in \mathbb{C}^m$ *we have*

$$\limsup_{n \to \infty} \|\tilde{U} x_n - \tilde{y}\|_2 \leq \epsilon, \qquad \lim_{n \to \infty} \|x_n\|_1 = BP_\epsilon(\tilde{y}, \tilde{U}), \tag{4.2.1}$$

provided that there exists x^0 *with* $\|\tilde{U} x^0 - \tilde{y}\|_2 < \epsilon$*, where* x_n *is the* n*th iteration of the algorithm with input* $(\tilde{y}, \tilde{U}, \epsilon)$ *and* $BP_\epsilon(\tilde{y}, \tilde{U})$ *is the real value given by*

$$\min \|x'\|_1 \text{ such that } \|\tilde{U} x' - \tilde{y}\| \leq \epsilon.$$

Remark 4.2.6. Note that the existence of such an algorithm does not contradict the previous theorem on the non-computability of basis pursuit as here we assume that the algorithm is fed exact information on the inputs (as opposed to Δ_1 information).

Theorem 4.2.7. *Fix natural numbers l_1 and η_1 and real constants $\tau_1 > 0, \rho_1 < 1$. Assume that $(\mathbf{s}, \mathbf{M})$ is a sparsity pattern satisfying the following conditions:*

1. *The number of levels in the sparsity pattern $(\mathbf{s}, \mathbf{M})$, denoted by l, satisfies $l < l_1$ with $l_1^{1/4} \in \mathbb{Q}$.*
2. *The ratio constant of the sparsity pattern $(\mathbf{s}, \mathbf{M})$, denoted by $\eta_{\mathbf{s},\mathbf{M}}$, satisfies $\eta_{\mathbf{s},\mathbf{M}} < \eta_1$ with $\eta_1^{1/4} \in \mathbb{Q}$.*

Let $\{\Xi_{\mathrm{BP}}, \Omega\}$ denote the Basis Pursuit problem with Ω being some (not necessarily proper) subset of possible ι with $\iota = (y, U)$ such that U satisfies the ℓ^2 robust nullspace property of order $(\mathbf{s}, \mathbf{M})$ with parameters $\rho < \rho_1$ with $\rho_1^{1/2} \in \mathbb{Q}$, $\tau < \tau_1$ and such that $y = Ux$ for some $(\mathbf{s}, \mathbf{M})$-sparse x.

Then, subject to the existence of a primal convergent tower of algorithms, there exists a collection of arithmetic algorithms $\{\Gamma_n\}_{n=1}^{\infty}$ on Ω for which $d(\Gamma_n(\iota), \Xi_{\mathrm{BP}}(\iota)) \leq 2^{-n}$. Thus $\{\Xi_{\mathrm{BP}}, \Omega\} \in \Delta_1^A$. Moreover, for any $\epsilon > 0$, $T(\Gamma_n, \epsilon) < \infty$ provided that there exists a $C \in \mathbb{R}$ such that for each $\iota \in \Omega$, $\|y\|_1 < C$.

Remark 4.2.8. The condition that $l_1^{1/4}, \eta_1^{1/4}$ and $\rho_1^{1/2}$ are all rational is merely a technical requirement and does not significantly weaken Theorem 4.2.7 because the set of $x \in \mathbb{R}$ such that $\sqrt[4]{x} \in \mathbb{Q}$ is dense in $\mathbb{R}_+$.

Remark 4.2.9. Because of the argument in the proof of Theorem 3.3.4, we get a similar Δ_1^A result as in Theorem 4.2.7 if we replace the assumptions on the nullspace property with an assumption that the input matrices have a sufficiently small RIP_L constant.

Remark 4.2.10. The strategy to proving Theorem 4.2.7 involves taking sufficiently close approximations $(\tilde{y}, \tilde{U})$ to (y, U) and then applying a primal convergent algorithm with ϵ taken to be small.

As a counterpart, we show that if the conditions given in the theorem above are slightly relaxed then we return to the territory of being outside of Δ_1.

Theorem 4.2.11. *We set $\{\Xi, \Omega\}$ to be the ℓ^1 basis pursuit problem. Fix natural numbers m, N with $N \geq 2$ and $m \geq 1$. Suppose that there is a real constant $\rho > 1$ and a set $K \subset \{1, 2, \ldots, N\}$ with $1 \leq |K| \leq (m \wedge (N-1))$ such that if a matrix $U \in \mathbb{R}^{m \times N}$ satisfies*

$$\|x_K\|_1 < \rho \|x_{K^c}\|_1 \quad \text{whenever } Ux = 0 \text{ and } x \neq 0 \tag{4.2.2}$$

then $(y, U) \in \Omega$ for all y such that $y = Uv$ for some v with support K. For such computational problems $\{\Xi, \Omega\}$, we have $\{\Xi, \Omega\} \notin \Delta_1^G$.

The requirement $1 \leq |K| \leq (m \wedge (N-1))$ is necessary for the conclusion of the theorem to hold. If this condition is omitted, then note that if $|K| > m$ then there are

no matrices satisfying (4.2.2): any such U would have $U_K \in \mathbb{C}^{m\times|K|}$ and thus U_K has a non-trivial nullspace, contradicting (4.2.2). If instead $K = \emptyset$ then the primary set Ω would consist of all pairs $(0_m, U)$ where U is a matrix and 0_m is the 0 vector, and trivially this problem is in Δ_0. Finally, if $|K| = N$ then any matrix satisfying (4.2.2) must have a trivial nullspace and therefore the basis pursuit problem is reduced to gaussian elimination.

We can prove a similar result for inputs that have a RIP in levels constant that is in some sense slightly too large:

Theorem 4.2.12. *Fix $\gamma > 1$ and $f : \mathbb{R}^+ \to \mathbb{R}^+$ such that $f(x) = o(x^{1/2})$. For parameters $p := (\delta, m, N, \mathbf{s}, \mathbf{M})$ with $\delta' > 0$, $m, N \in \mathbb{N}$ and $(\mathbf{s}, \mathbf{M})$ a valid sparsity pattern, we denote by Ω_p the set of (y, U) such that*

- *$U \in \mathbb{R}^{m\times N}$ and $y = Ux$ for some $(\mathbf{s}, \mathbf{M})$-sparse $x \in \mathbb{R}^N$.*
- *$\mathrm{cond}(UU^*) < \gamma$ and the RIP_L constant of order $(\mathbf{s}, \mathbf{M})$ for U satisfies $\delta_{\mathbf{s},\mathbf{M}} < \delta'$.*

Then there exist valid parameters p such that

(A) $(\mathbf{s}, \mathbf{M})$ has two levels and $\delta' \leq \frac{1}{f(\eta_{\mathbf{s},\mathbf{M}})}$

but $\{\Xi, \Omega_p\}^{\Delta_1} \notin \Delta_1^G$. Similarly, there exist valid parameters p such that if condition (A) is replaced by

(B) $\eta_{\mathbf{s},\mathbf{M}} = 1$ and $\delta' \leq \frac{1}{f(l)}$

then we still have $\{\Xi, \Omega_p\}^{\Delta_1} \notin \Delta_1^G$.

4.2.2 Basis pursuit denoising

As with basis pursuit, there are some basic non-computability results for basis pursuit denoising (defined as $\mathrm{argmin}_{x\in\mathbb{R}^N} \|x\|_1$ such that $\|Ux - y\|_2 \leq \delta$).

Theorem 4.2.13 (Basis Pursuit Denoising)**.** *Let $\Xi(\iota)$ denote the solution set, with input $\iota \in \Omega$, of basis pursuit with l^1 norm.*

(i) For any dimensions $m, N \in \mathbb{N}$ with $m < N$ and $\delta > 0$, there exists a set of inputs Ω such that for all $\iota \in \Omega$ we can write $\iota = (y, U, \delta)$ for some vector $y \in \mathbb{R}^m$ and matrix $U \in \mathbb{R}^{m\times N}$, $\Xi(\iota)$ has a unique minimiser and $\|\iota\| \leq 1$ for all $\iota \in \Omega$, yet

$$\{\Xi, \Omega\}^{\Delta_1} \notin \Delta_1^G, \qquad \epsilon_{\mathrm{B}}^{\mathrm{s}} \geq \frac{1}{2}.$$

Moreover, $\{\Xi, \Omega\}^{\Delta_1} \in \Delta_1^A$ implies decidability of the Halting Problem.

(ii) For any dimensions $m, N \in \mathbb{N}$ with $m < N$ and denoising parameter $\delta \in (0, 1)$, there exists a set of inputs Ω such that for all $\iota \in \Omega$ we can write $\iota = (y, U, \delta)$ for some vector $y \in \mathbb{R}^m$ and matrix $U \in \mathbb{R}^{m\times N}$ with bounded condition numbers in the following sense: $\mathrm{Cond}(UU^*), C_{FP}(U, y, \delta), \mathrm{Cond}(\Xi) \leq 2$, $\|\iota\| \leq 1$ yet

$$\{\Xi, \Omega\}^{\Delta_1} \notin \Delta_1^G, \qquad \epsilon_{\mathrm{B}}^{\mathrm{s}} \geq \frac{1-\delta}{\sqrt{2}}$$

Furthermore, $\{\Xi, \Omega\}^{\Delta_1} \in \Delta_1^A$ *implies decidability of LLPO.*

We also get a similar non-computability result on two minimiser problems:

Theorem 4.2.14. *Let* $\{\Xi, \Omega\}$ *denote the computational problem of Basis Pursuit denoising for some* $\delta > 0$ *with* l^1 *regularisation and with some set* Ω *of valid inputs. Suppose that there is input data* $\iota = (y, U, \delta) \in \Omega$ *for which there are at least two distinct elements in* $\Xi(\iota)$*. Suppose also that there is an* $\epsilon > 0$ *such that for every negative semidefinite diagonal matrix* D *with* $\|D\| < \epsilon$ *we have* $(y, U + UD, \delta) \in \Omega$*. Then* $\{\Xi_3, \Omega_3, \mathcal{M}_3, \Lambda_3\}^{\Delta_1} \notin \Delta_1^G$*. Moreover, if* $x^1, x^2 \in M_{min}^{BP}(\iota)$ *then* $\epsilon_{\mathrm{B}}^{\mathrm{s}} \geq \|x^1 - x^2\|_2/2$*.*

Thus, to get a positive result, we turn to the robust nullspace property as before. Unfortunately, this will still not be enough to give us a result in Δ_1^A, as the following statement shows.

Theorem 4.2.15. *Fix* $s \in \mathbb{N}$ *and* $\delta \in (0, 1]$ *and let* Ω *denote the set of all* ι *satisfying the following requirements:*

1. U *obeys the robust nullspace property with parameters* $\rho \leq \rho' = \frac{s+1}{2^{\lfloor \log_2(s) \rfloor + 2} - s} < 1$ *and* $\tau \leq 7\sqrt{1+\rho'}$*.*
2. $\|U\| \leq 6$ *and* $\mathrm{cond}(UU^*) \leq 36$*.*
3. $y = Ux$ *for some* s*-sparse* x *and* $y \leq 2$*.*

Then $\{\Xi_3, \Omega_3, \mathcal{M}_3, \Lambda_3\}^{\Delta_1} \notin \Delta_1^G$*. Moreover,* $\epsilon_{\mathrm{B}}^{\mathrm{s}} \geq \delta/2$*.*

However, we can also use a primal convergent algorithm to provide an upper bound for the breakdown epsilon as a counterpart to Theorem 4.2.7.

Theorem 4.2.16. *Fix natural numbers* l_1 *and* η_1 *and real constants* $\delta \geq 0, \tau_1 > 0, \rho_1 < 1$*. Assume that* $(\mathbf{s}, \mathbf{M})$ *is a sparsity pattern satisfying the following conditions:*

1. *The number of levels in the sparsity pattern* $(\mathbf{s}, \mathbf{M})$*, denoted by* l*, satisfies* $l < l_1$ *with* $l_1^{1/4} \in \mathbb{Q}$*.*
2. *The ratio constant of the sparsity pattern* $(\mathbf{s}, \mathbf{M})$*, denoted by* $\eta_{\mathbf{s},\mathbf{M}}$*, satisfies* $\eta_{\mathbf{s},\mathbf{M}} < \eta_1$ *with* $\eta_1^{1/4} \in \mathbb{Q}$*.*

Let $\{\Xi_{\mathrm{BPDN}}, \Omega\}$ *denote the Basis Pursuit problem with* Ω *being some (not necessarily proper) subset of possible* ι *with* $\iota = (y, U)$ *such that* $U \in \mathbb{R}^{m \times N}$ *satisfies the* ℓ^2 *robust nullspace property of order* $(\mathbf{s}, \mathbf{M})$ *with parameters* $\rho < \rho_1$ *with* $\rho_1^{1/2} \in \mathbb{Q}$*,* $\tau < \tau_1$ *and such that* $\|y - Ux\| \leq \delta$ *for some* $(\mathbf{s}, \mathbf{M})$*-sparse* x*.*

Then, subject to the existence of a primal convergent tower of algorithms, there exists a collection of arithmetic algorithms $\{\Gamma_n\}_{n=1}^{\infty}$ *on* Ω *for which* $d(\Gamma_n(\iota), \Xi_{\mathrm{BP}}(\iota)) \leq 2^{-n} + \epsilon'$ *where*

$$\begin{aligned}
\epsilon' &:= 2N\delta\alpha_3/(N-1) + 2\delta\left(C + D\sqrt[4]{l_1\eta_1}\right) \\
\alpha_3 &:= \tau_1(\sqrt[4]{l_1\eta_1}/2 + 1 + 2(\rho_1 + C_0(\rho_1)\sqrt[4]{l_1\eta_1}))/(1-\rho_1) \\
C &:= \frac{\rho_1\tau_1 + \tau_1}{1-\rho_1} \quad \textit{and} \quad D := \frac{4\sqrt{\rho_1}\tau_1 + 3\tau_1 - \rho_1\tau_1}{2 - 2\rho_1}.
\end{aligned}$$

Moreover, for any $\epsilon > \epsilon'$, $T(\Gamma_n, \epsilon) < \infty$ *provided that there exists a* $C \in \mathbb{R}$ *such that for each* $\iota \in \Omega$, $\|y\|_1 < C$.

These two results - one showing that the breakdown epsilon is non-zero even assuming that all inputs have the nullspace property (Theorem 4.2.15) and one showing that the breakdown epsilon is bounded above in that circumstance (Theorem 4.2.16) are important to this thesis. Firstly, they show that there are problems with a very natural input set which are not computable. Secondly, they demonstrate that even though such problems are in a sense not computable, we can compute them up to a tolerably small error.

To conclude this section, let us demonstrate a simple numerical example of an incorrect computation for basis pursuit using SPGL1. We run the following program to solve

$$\min_{x \in \mathbb{R}^2} \|x\|_1 \text{ such that } Ux = y$$

where $U = \begin{pmatrix} 1-\epsilon & 1 \end{pmatrix}, y = 1$ for some small $\epsilon > 0$. This can be expressed in the following code:

```
Large = 1e9; Epsilon = 1/Large;
y = 1;
U = [1-Epsilon, 1];
options = spgSetParms('optTol',1e-50,'bpTol',1e-50,'decTol'
   ,1e-50);
AlgBPAns = spgl1(U,y,[],[],[],opts)
```

The true solution is $x = (0, 1)^T$. Instead, MATLAB prints the following output.

```
================================================================================
SPGL1  v.1.9 (29 Apr 2015)
================================================================================
No. rows              :        1      No. columns           :        2
Initial tau           : 0.00e+00      Two-norm of b         : 1.00e+00
Optimality tol        : 1.00e-50      Target objective      : 0.00e+00
Basis pursuit tol     : 1.00e-50      Maximum iterations    :       10

Iter   Objective   Relative Gap  Rel Error      gNorm   stepG    nnzX    nnzG            tau
0  1.0000000e+00  0.0000000e+00   1.00e+00  1.000e+00     0.0       0       0  1.0000000e+00
1  7.5000000e-01  5.6250000e-01   7.50e-01  7.500e-01    -0.6       2       1
2  4.9999993e-10  2.4999993e-19   5.00e-10  5.000e-10     0.0       2       1
3  5.0000004e-10  2.5000004e-19   5.00e-10  5.000e-10     0.0       2       1
4  5.0000004e-10  2.5000004e-19   5.00e-10  5.000e-10     0.0       2       1
5  3.7500003e-10  3.2812507e-19   3.75e-10  3.750e-10    -0.6       2       1
6  0.0000000e+00  0.0000000e+00   0.00e+00  0.000e+00     0.0       2       2

EXIT -- Found a BP solution

Products with A     :      13        Total time   (secs) :     0.0
Products with A'    :       7        Project time (secs) :     0.0
Newton iterations   :       2        Mat-vec time (secs) :     0.0
Line search its     :       4        Subspace iterations :       0


AlgBPAns =

0.5000
0.5000
```

The value of $x = (0.5, 0.5)^T$ is very different to the true solution.

4.3 Statistical estimation

Recall that constrained lasso is to find a minimiser of $\operatorname{argmin}_{x\in\mathbb{R}^N} \|Ax - y\|_2$ such that $\|x\|_1 \leq \tau$. Similarly unconstrained lasso is to find a minimiser of $\operatorname{argmin}_{x\in\mathbb{R}^N} \|Ax - y\|_2^2 + \lambda\|x\|_1$. Once again, we start by proving a general non-computability result for both constrained and unconstrained lasso. However, this time we will be unable to prove a result similar to the basis pursuit/basis pursuit denoising case for single minimiser problems

Theorem 4.3.1 (Lasso)**.** *Let $\Xi_{UL}(\iota)$ (respectively Ξ_{CL}) denote the solution sets, with input $\iota \in \Omega$, of unconstrained (respectively constrained) lasso with l^1 norm. We have the following:*

(i) There exists a set of inputs Ω such that $\operatorname{Cond}(AA^*), \operatorname{Cond}(\Xi) \leq 2$, $\|\iota\| \leq 1$ for $\iota = (y, A) \in \Omega$, yet

$$\{\Xi, \Omega\}^{\Delta_1} \notin \Delta_1^G, \qquad \epsilon_{\mathrm{B}}^{\mathrm{s}} \geq \frac{2-\lambda}{2\sqrt{2}}$$

Furthermore, $\{\Xi, \Omega\}^{\Delta_1} \in \Delta_1^A$ implies decidability of LLPO.

(ii) For any dimensions $m, N \in \mathbb{N}$ with $m < N$ and any constraint parameter τ such that $\tau \in (0,1)$, there exists a set of inputs Ω such that for all $\iota \in \Omega$ we can write $\iota = (y, A, \tau)$ for some vector $y \in \mathbb{R}^m$ and matrix $A \in \mathbb{R}^{m \times N}$ with bounded condition numbers in the following sense: $\mathrm{Cond}(AA^*), C_{FP}(A, y, \tau), \mathrm{Cond}(\Xi) \leq 2$, $\|\iota\| \leq 1$ *yet*

$$\{\Xi, \Omega\}^{\Delta_1} \notin \Delta_1^G, \qquad \epsilon_{\mathrm{B}}^{\mathrm{s}} \geq \frac{\tau}{\sqrt{2}}$$

Furthermore, $\{\Xi, \Omega\}^{\Delta_1} \in \Delta_1^A$ implies decidability of LLPO.

In the previous discussion on basis pursuit and basis pursuit denoising we saw that if the input set contained an input with two minimisers and all small perturbations on that input were also in the input set then the problem is not in Δ_1. This is indeed the case for unconstrained lasso. For constrained lasso however we do not get this result. This is summarised in the following theorem:

Theorem 4.3.2. *1. Let $\{\Xi, \Omega\}$ denote the computational problem of unconstrained Lasso with l^1 regularisation. Suppose that there is input data $\iota = (y, A) \in \Omega$ for which there are at least two distinct elements in $\Xi(I)$. Suppose also that there is an $\epsilon > 0$ such that for every negative semidefinite diagonal matrix D with $\|D\|_{\max} < \epsilon$ we have $(y, A + AD) \in \Omega$. Then $\{\Xi_3, \Omega_3, \mathcal{M}_3, \Lambda_3\}^{\Delta_1} \notin \Delta_1^G$. Moreover, if $x^1, x^2 \in M_{min}^{UL}(\iota)$ then $\epsilon_{\mathrm{B}}^{\mathrm{s}} \geq \|x^1 - x^2\|_2/2$.*

2. Let $\{\Xi_4, \Omega_4, \mathcal{M}_4, \Lambda_4\}$ be the constrained lasso problem with ℓ^1 regularisation as defined in section 2.10. There exists an input set Ω and $\iota \in \Omega$ such that if $\iota = (y, A, \tau)$.

(a) There are at least two distinct vectors in $\Xi_4(\iota)$.

(b) For any matrix D with $\|D\| < 1/2$, $(y, A + AD, \tau) \in \Omega_4$.

(c) $\{\Xi_4, \Omega_4, \mathcal{M}_4, \Lambda_4\} \in \Delta_1^A$ with bounded minimum runtime so that there exist Γ^n such that $d(\Gamma_n(\iota), \Xi(\iota)) \leq 2^{-n}$ and $T(\Gamma_n, \epsilon) < \infty$.

Despite the result on the non-computability of unconstrained lasso above, in some sense two minimiser unconstrained lasso problems are 'rare' when the matrix elements have some underlying random distribution. Indeed, the following lemma is proven in [116] as Lemma 4.

Theorem 4.3.3 (Tibshirani 2012)**.** *Suppose $X \in \mathbb{R}^{m \times N}$ is a random matrix with distribution absolutely continuous with respect to the lebesgue measure on $\mathbb{R}^{m \times N}$. Then for any y and $\lambda > 0$, the unconstrained lasso problem has exactly one minimiser.*

Thus it may appear that despite the results of Theorem 4.3.2, it is unlikely that one will draw an input set with two minimiser problems. Even though this is the case, it is possible to create a probability distribution so that it takes an arbitrarily long time to solve the problems with probability greater than a half.

To state a precise theorem about this, we introduce some notation: for a subset S of $\mathbb{R}^{m \times N}$, we denote the set of lebesgue measurable sets on S by $\mathcal{L}(S)$. Let $\mathbb{R}^{1 \times 2}_{\|\cdot\|_{\max} \leq 2}$

be the measurable space of all matrices with entries bounded (in absolute value) by 2. We denote by $\mathcal{X}$ the measure space $\mathcal{X} = (\mathbb{R}^{1\times 2}_{\|\cdot\|_{\max}\leq 2}, \mathcal{L}(\mathbb{R}^{1\times 2}_{\|\cdot\|_{\max}\leq 2}))$.

Theorem 4.3.4. *For any $M \in \mathbb{N}$, there exists a probability measure $\mathbb{P}_M$ on $\mathcal{X}$ absolutely continuous with respect to the lebesgue measure and a vector $y \in \mathbb{R}$ with $|y| \leq 1$ such that for every fixed $\lambda \in (0,2)$ and set Ω comprised of three tuples (y, A, λ) at least one of the following two things occur*

1. *For the unconstrained lasso problem $\{\Xi, \Omega\}$, $\epsilon^{\mathrm{lr}}_{\mathrm{B}}(M) > (2-\lambda)/(2\sqrt{2})$.*
2. *If it is measurable, the set of all $A \in \mathbb{R}^{1\times 2}_{\|\cdot\|_{\max}\leq 2}$ such that $(y, A, \lambda) \in \Omega$ has $\mathbb{P}_M$ probability less than or equal to $1/2$.*

An initial attempt to get a positive result with lasso might use the robust nullspace property as in the section on basis pursuit. However, this will not be enough to guarantee that we are in Δ_1^A:

Theorem 4.3.5. *Fix $s \in \mathbb{N}$ and $\lambda \in (0, 1/\sqrt{k}]$ where $k = 2^{\lfloor \log_2(s) \rfloor + 1}$. Let Ω denote the set of all ι satisfying the following requirements:*

1. *A obeys the robust nullspace property with parameters $\rho \leq \rho' = \frac{s+1}{2^{\lfloor \log_2(s)\rfloor + 2 - s}} < 1$ and $\tau \leq 7\sqrt{1+\rho'}$.*
2. *$\|A\| \leq 6$ and $\mathrm{cond}(AA^*) \leq 36$.*
3. *$y = Ax$ for some s-sparse x and $y \leq 2$.*

Then $\{\Xi_4, \Omega_4, \mathcal{M}_4, \Lambda_4\}^{\Delta_1} \notin \Delta_1^G$. Moreover, $\epsilon^{\mathrm{s}}_{\mathrm{B}} \geq \frac{4\lambda\sqrt{k}}{5\sqrt{2}}$.

In the basis pursuit denoising case, we showed that we can attain close to the breakdown epsilon with an algorithm provided that we have the robust nullspace property (Theorem 4.2.16). For lasso, the robust nullspace property is mainly of importance in inverse problems and not for statistical estimation. We will thus pick a different well studied criteria - specifically, we shall work with the *dual certificate* criteria. To get a positive result with a limited runtime, we once again need the idea of a primal convergent algorithm for lasso.

Definition 4.3.6. *A* primal convergent tower *of algorithms for lasso is an arithmetic tower of algorithms $\{\Gamma_n\}$ of height one such that for given $\lambda > 0$, $\tilde{A} \in \mathbb{C}^{m\times N}$ and $\tilde{y} \in \mathbb{C}^m$ and $x_n = \Gamma_n(\tilde{y}, \tilde{A}, \lambda)$ then*

$$\lim_{n\to\infty} \|x_n\|_1 = LASSO_\lambda(\tilde{A}, \tilde{y}), \tag{4.3.1}$$

where $LASSO_\lambda(\tilde{A}, \tilde{y})$ is the real value given by $\min_{x'} \|\tilde{A}x' - \tilde{y}\| + \lambda\|x'\|_1$.

Once again, the existence of such an algorithm does not contradict any theorem on the non-computability of lasso as we assume that the primal convergent algorithm is fed exact information on the inputs (as opposed to Δ_1 information). With this in mind, we can now prove a positive result for unconstrained lasso:

Theorem 4.3.7. *Fix natural numbers m, N, positive numbers $\lambda, \alpha_1, \alpha_2, \alpha_3, \alpha_4$ and a set $S \subseteq \{1, 2, \ldots, N\}$. Let Ω be the set of all (y, A) such that $y \in \mathbb{R}^m, A \in \mathbb{R}^{m \times N}$ and the following conditions hold:*

(i) $y = A\eta$ *for some* $\eta \in \mathbb{R}^N$ *with support contained in* S.

(ii) $\|A_{S^c}^* A_S (A_S^* A_S)^{-1}\|_\infty < 1 - \alpha_1$.

(iii) $\min\{\rho \in \text{Spec}(A_S^* A_S)\} \geq \alpha_2$ *where* Spec *denotes the spectrum of a matrix.*

(iv) $\|A\|_2 \leq \alpha_3$ *and* $\|\eta\|_1 \leq \alpha_4$.

If $\{\Xi_4, \Omega_4, \mathcal{M}_4, \Lambda_4\}$*represents unconstrained lasso, then* $\{\Xi_4, \Omega_4, \mathcal{M}_4, \Lambda_4\}^{\Delta_1} \in \Delta_1^A$.

Items (i) to (iv) in Theorem 4.3.7 are the *dual certificate criteria* studied in [123] and references therein.

4.4 Neural networks - the paradoxes of deep learning

In the other sections, we began with a non-computability result and then augmented it with a collection of either positive or negative results depending on how the input set is further restricted. However, as highlighted in the introduction, there is currently very little existing work that explains when a neural network will succeed and when it will fail [46, 52].

We thus instead choose to present three separate issues with neural networks. To exclude pathological examples with training and classification sets that have elements that are arbitrary close to each other, that could make the classification function jump subject to a small perturbation, or become arbitrary large, we introduce the idea of *well separated and stable sets*. Specifically, for a given classification function $f : \mathbb{R}^d \to \{0, 1\}$, we define the family of well separated and stable sets $\mathcal{S}_\delta^f$ with separation at least δ:

$$\mathcal{S}_\delta^f = \{\{x_1, \ldots, x_r\} \,|\, \|x_j\| \leq 1, \min_{i \neq j} \|x_i - x_j\|_\infty \geq \delta, f(x_j + y) = f(x_j) \text{ for } \|y\|_\infty < \delta\}.$$

Roughly speaking, the first issue with neural networks is a non-computability result showing that there are uncountably many classification functions, training sets and classification sets for which (provided small perturbations are still valid inputs to the algorithm) it will be impossible to make an algorithm that automatically finds and applies the optimal neural network to the classification sets. This is stated accurately in the following theorem.

Theorem 4.4.1. *For $n \in \mathbb{N}$, we write $\varepsilon(n) := ((4n + 3)(2n + 2))^{-1}$. Additionally, for any $\nu > 0$ and fixed training set $\mathcal{T}$ of size K with $K \in \mathbb{N}$, let $\Omega_\nu^{\mathcal{T}}$ be the collection of $\iota = \left\{\{(\hat{x}^j, f(\hat{x}^j))\}_{j=1}^K, \{\hat{x}^j\}_{j=1}^K\right\}$, with $\hat{x}^j \in \mathcal{B}_\nu^\infty(x^j)$, $x^j \in \mathcal{T}$ and where $\mathcal{B}_\nu^\infty(x^j)$ is the open ball of radius ν about x^j in the ℓ^∞ norm.*

There are an uncountable family of classification functions $f : \mathbb{R}^{N_0} \to \{0, 1\}$ such that for any neural network dimensions $\mathbf{N} = (N_L, N_{L-1}, \ldots, N_1, N_0)$ with $N_0, L \geq 2$

and any $K \geq 3(N_1+1)\cdots(N_{L-1}+1)$ *there exist uncountably many training sets* $\mathcal{T} \in \mathcal{S}^f_{\varepsilon(K)}$ *of size* K *such that for any cost function* $C \in \mathcal{CF}$ *and* $\nu < \varepsilon(K)$, $\{\Xi^C_{\mathrm{NN}}, \Omega^{\mathcal{T}}_\nu\} \notin \Delta^G_1$. *Moreover,* $\epsilon^{\mathrm{s}}_{\mathrm{B}} \geq \epsilon^{\mathrm{s}}_{\mathbb{P}\mathrm{B}}(p) \geq 1/4$ *for* $p < 1/2$.

Note that Theorem 4.4.1 applies with a very general class of cost functions. Additionally, since we have a non-zero probabilistic breakdown epsilon, the issue cannot be resolved by working with a probabilistic algorithm (so in particular, stochastic gradient descent will be unable to solve this particular problem).

Finally, we note that the optimisation problem for neural networks is typically non-convex, and at first glance it may seem that we are providing a problem for which any algorithm will get 'stuck' in a local minima. Here we highlight specifically that the result says that $\{\Xi, \Omega\} \notin \Delta^G_1$ as opposed to $\{\Xi, \Omega\} \notin \Delta^A_1$. The immediate and important implication is that no such algorithm to perform the computation will exist *even if we provide a candidate algorithm with an oracle to detect and avoid local minima.*

The second issue says that given an arbitrarily small distance ϵ we can find an uncountable number of functions, training sets and classification sets so that there is a neural network which can train exactly on the (arbitrarily large) training set and give the correct result on the (arbitrarily large) classification set, but there will be uncountably many new vectors within ϵ of the training and classification set which this neural network will be unable to classify. This is stated precisely in the following theorem:

Theorem 4.4.2. *As in Theorem 4.4.1, for* $n \in \mathbb{N}$, *we write* $\varepsilon(n) := ((4n+3)(2n+2))^{-1}$. *There are an uncountable family of classification functions* $f : \mathbb{R}^{N_0} \to \{0,1\}$ *such that for any neural network dimensions* $\mathbf{N} = (N_L, N_{L-1}, \ldots, N_1, N_0)$ *with* $N_0, L \geq 2$, *any non-negative* $\epsilon < 1/(K+M)$ *where* M *can be made arbitrarily large and any* $K \geq 3(N_1+1)\cdots(N_{L-1}+1)$ *there exist uncountably many non-intersecting training sets* $\mathcal{T} = \{x^1, \ldots, x^K\} \in \mathcal{S}^f_{\varepsilon(K+M)}$ *and uncountably many non-intersecting classification sets* $\mathcal{C} = \{y^1, \ldots, y^M\} \in \mathcal{S}^f_{\varepsilon(K+M)}$ *such that if* $\tilde{\phi}$

$$\tilde{\phi} \in \operatorname*{argmin}_{\phi \in \mathcal{NN}_{\mathbf{N},L}} C(v,w), \quad v_j = \phi(x^j), \quad w_j = f(x^j),$$

where $1 \leq j \leq K$ *and* $C \in \mathcal{CF}$, *then*

$$\tilde{\phi}(x) = f(x) \quad \forall x \in \mathcal{T} \cup \mathcal{C}. \tag{4.4.1}$$

However, for every $\hat{\phi} \in \mathcal{NN}_{\mathbf{N},L}$ *there exists uncountably many* $v \in \mathbb{R}^{N_0}$ *such that*

$$|\hat{\phi}(v) - f(v)| \geq 1/2, \qquad \|v - x\|_\infty \leq \epsilon \text{ for some } x \in \mathcal{T}. \tag{4.4.2}$$

The final neural network related issue is that for an uncountable number of classification functions, there are an uncountable number of training sets and classification sets for which solving the optimisation problem (assuming we had an oracle which could do so) will yield a neural network that gets an incorrect answer on at least one

element of the training set and at least one element of the classification set. However, we can find arbitrarily small perturbations on the training set and classification set for which solving the optimisation problem again with this new training set will yield a neural network that will be able to exactly classify both the training set and the classification set. This is formulated mathematically in the following theorem:

Theorem 4.4.3. *There are an uncountable family of classification functions $f : \mathbb{R}^{N_0} \to \{0,1\}$ such that for any neural network dimensions $\mathbf{N} = (N_L, N_{L-1}, \ldots, N_1, N_0)$ with $N_0, L \geq 2$, any $K \geq 3(N_1+1)\cdots(N_{L-1}+1)$, any $M \geq K$ and any non-negative $\epsilon < 1/(K+M)$ there exist uncountably many non-intersecting training sets of the form $\mathcal{T}_1 = \{x^1, \ldots, x^K\} \in \mathcal{S}^f_{(K+M)^{-1}}$ and $\mathcal{T}_2 = \{\hat{x}^1, \ldots, \hat{x}^K\} \in \mathcal{S}^f_{(K+M)^{-1}}$ and uncountably many non-intersecting classification sets $\mathcal{C}_1 = \{y^1, \ldots, y^M\} \in \mathcal{S}^f_{(K+M)^{-1}}$ and $\mathcal{C}_2 = \{\hat{y}^1, \ldots, \hat{y}^M\} \in \mathcal{S}^f_{(K+M)^{-1}}$ with $M \geq K$ such that $\mathcal{T}_1 \subset \mathcal{B}^\infty_\epsilon(\mathcal{T}_2)$ and $\mathcal{C}_1 \subset \mathcal{B}^\infty_\epsilon(\mathcal{C}_2)$. Moreover, we have*

$$\tilde{\phi}_k \in \operatorname*{argmin}_{\phi \in \mathcal{NN}_{\mathbf{N},L}} C(v^k, w^k), \qquad k = 1, 2,$$

where

$$v^1_j = \phi(x^j), \quad w^1_j = f(x^j), \qquad v^2_j = \phi(\hat{x}^j), \quad w^2_j = f(\hat{x}^j),$$

such that there exist $v \in \mathcal{T}_1$ and $w \in \mathcal{C}_1$ with

$$|\tilde{\phi}_1(v) - f(v)| \geq 1/2, \qquad |\tilde{\phi}_1(w) - f(w)| \geq 1/2.$$

However,

$$\tilde{\phi}_2(x) = f(x) \quad \forall x \in \mathcal{T}_2 \cup \mathcal{C}_2.$$

Let us consider the three issues we have just highlighted. The first issue is a computability issue, demonstrating that there are neural network problems that cannot be computed. Yet, in practice, it seems that we are able to get good results when attempting to compute neural networks with backpropagation and stochastic gradient descent.

The second issue demonstrates a risk with the tactic of cross validation: namely, training a neural network and then testing it on sensible data, observing success and then assuming that it will still work well on new samples. In the absence of mathematical models to explain when a neural network will fail and when a neural network will succeed, this is seen to be an effective strategy for solving classification problems. However, we have demonstrated here that this can fail with arbitrarily large cross validation on an uncountable number of examples.

The final issue discusses a sensitivity to perturbations - even though we may fail to train on a training set, an arbitrarily small perturbation can yield a successful result. Thus it is very difficult to classify failure in the context of neural networks.

Let us examine the second issue further. Theorem 4.4.2 shows that for a given $\epsilon > 0$, there are classification functions for which the trained neural network is successful on both the training set and arbitrarily large classification. However, such neural networks have an instability on perturbations of size ϵ. Let us contrast this with Theorem

1.7.1 (the universal approximation theorem). Note that the classification functions in Theorem 4.4.2 are in $\mathcal{S}^f_{\varepsilon(K+M)}$ and are thus constant under perturbations of size $\epsilon < \varepsilon(K+M)$. An application of Theorem 1.7.1 shows that we can approximate a smoothed version of this function to arbitrary accuracy. Thus there exists a neural network that approximates the classification function arbitrarily well and is therefore stable under small perturbations. Arguably, the added stability yields a 'better' neural network that classifies well and is also stable.

We now introduce the idea of the *paradoxes of deep learning*. These relate to the inefficiency in training the network. Specifically, the paradoxes are the following ideas:

Definition 4.4.4 (The paradoxes of deep learning)**.** *Despite their success, neural networks have the following paradoxical properties:*

1. *As shown in Theorem 4.4.1, the computational problem of training a neural network and using it on a classification set is non-computable. This is even if there exist computational oracles to avoid local minima and even using random algorithms.*

2. *Even given an oracle that can train the network, the methodological process of training a neural network will yield an undesirable answer in the sense that such a neural network will be unstable. Despite this process failing, there will exist a stable alternative neural network - but such an alternative neural network cannot be trained by training the network.*

To highlight potential issues with neural networks, let us consider an example using 'Google translate'. At the time of writing, this functions by using a collection of neural networks (more detail is given [127]). We ask a simple question: translate the Norwegian word 'stortinget'. The correct translation in English is the word 'parliament' which is obtained by Google translate (see Figure 4.1). We also ask for the related translation of 'stortinget stortinget' which should translate to 'parliament parliament'. Note that this would be correctly translated by an algorithm that simply looks up words from a dictionary. Unfortunately, Google translate does not provide the correct translation (see Figure 4.2).

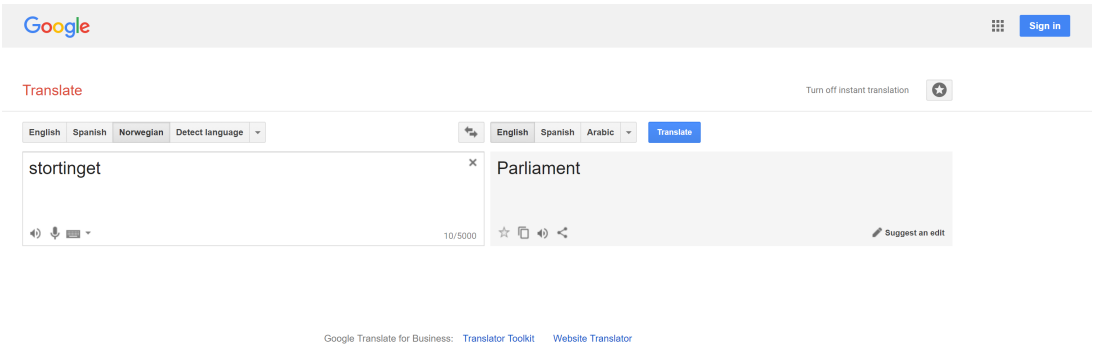


Figure 4.1: An example of google translate succeeding.

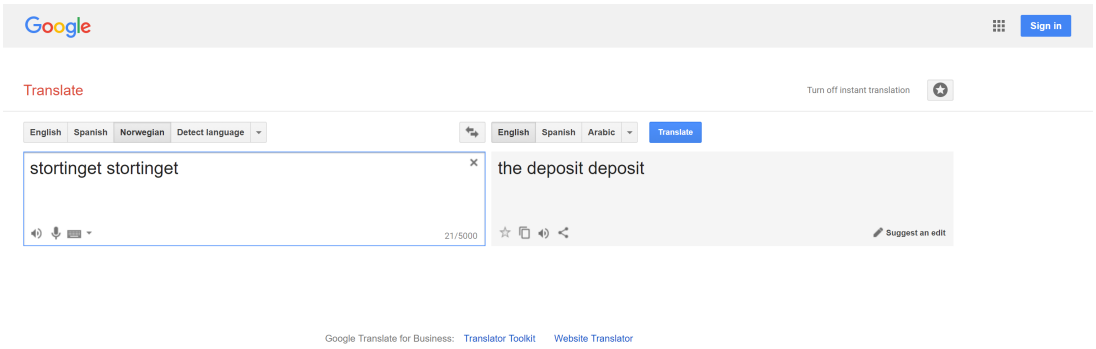


Figure 4.2: A similar example where google translate fails.

4.5 Image processing

Our results here will show that both (BPTV) (finding a minimiser to $\operatorname{argmin}_{x\in\mathbb{R}^N} \|x\|_{\mathrm{TV}}$ such that $Ax = y$) and (DeblurTV) (finding a minimiser to $\operatorname{argmin}_{x\in\mathbb{R}^N} \|Ax-y\|_2^2 + \lambda\|x\|_{\mathrm{TV}}$) are non-computable problems. This is encapsulated mathematically in the following theorems:

Theorem 4.5.1 (Image inverse problems)**.** *Let Ξ_{BPTV} denote the problem function for basis pursuit with total variation regularisation. There exists a set of inputs Ω such that*

$$\mathrm{Cond}(AA^*), C_{FP}(y,A) \leq 2\sqrt{2}, \|\iota\| \leq 2$$

for all $\iota = (y,A) \in \Omega$, yet $\{\Xi_{\mathrm{BPTV}}, \Omega\}^{\Delta_1} \notin \Delta_1^G$ with $\epsilon_{\mathrm{B}}^{\mathrm{s}} \geq \sqrt{2}$.

Theorem 4.5.2 (Image deblurring)**.** *Fix $\lambda < \frac{1}{(2+\sqrt{2})^2}$. Let Ξ_{DeblurTV} denote the problem function for image deblurring. There exists a set of inputs Ω such that*

$$\mathrm{Cond}(AA^*), \|\iota\| \leq 2$$

for all $\iota = (y,A) \in \Omega$, yet $\{\Xi_{\mathrm{DeblurTV}}, \Omega\}^{\Delta_1} \notin \Delta_1^G$ with $\epsilon_{\mathrm{B}}^{\mathrm{s}} \geq \sqrt{2}(1-2\lambda)$.

Let us end this chapter with a simple example of total variation in inverse problems. Our aim will be to recover a Shepp-Logan phantom [108] of dimension 2048^2. We choose 40% of the possible Fourier frequencies uniformly at random without replacement. Next, we try to recover using the standard ℓ^1 minimisation technique from compressed sensing with Daubechies 2 wavelets (see (BP)). Finally, we use a total variation minimisation strategy as in (BPTV) with the same samples. The results are displayed in Figure 4.3.

(a) (b) (c)

Figure 4.3: Standard ℓ^1 basis pursuit (BP) compared to total variation minimisation (BPTV) (a) The original Shepp-Logan phantom (b) Recovery using (BP) (c) Recovery using total variation minimisation.

Given the striking visual improvement observed by using (BPTV) over (BP), it is very tempting to state that total variation minimisation is a better method than ℓ^1 basis pursuit for the given sampling pattern and image. However, this conclusion cannot be drawn from Figure 4.3 alone.

Here we draw a distinction between the *method* and *algorithm used to perform the method*. Theorem 4.2.1 and Theorem 4.5.1 show that in general, it is impossible to compute the methods of (BP) or (BPTV). Thus the poor visual result demonstrated in Figure 4.3 using (BP) may be the observed result of a failure of the algorithm as opposed to a failure with the method.

This argument extends to all the techniques discussed in this thesis. Without a prior understanding of what exactly the algorithms are computing, we cannot assert anything about the underlying method from empirical evidence. Thus a comparison of various methods in practical circumstances becomes very difficult unless further analysis on when algorithms work and when they fail is completed.

Remark 4.5.3. Note that the performance observed in Figure 4.3 is far weaker than the one displayed in Figure 1.1. We emphasize here that although the sampling pattern is likely to be far from optimal, our point is merely about the importance of understanding algorithm performance when making statements that compare various methods and our choice of sampling pattern does not change this conclusion.

Chapter 5

Proofs

Note that throughout this section theorems from separate sections that share almost identical proofs are typically grouped together. Thus results may not be proven in the order that they were stated in the thesis.

5.1 Proof of Proposition 2.8.4

Proof of Proposition 2.8.4. We start with (2.8.2), and observe that $\epsilon^{\mathrm{s}}_{\mathbb{P}\mathrm{B}}(q) \leq \epsilon^{\mathrm{s}}_{\mathbb{P}\mathrm{B}}(p)$ follows directly from the definition. To see that $\epsilon^{\mathrm{s}}_{\mathbb{P}\mathrm{B}}(p) \leq \epsilon^{\mathrm{s}}_{\mathrm{B}}$ we argue by contradiction and suppose that $\epsilon^{\mathrm{s}}_{\mathbb{P}\mathrm{B}}(p) > \epsilon^{\mathrm{s}}_{\mathrm{B}}$. Then there exists an $\epsilon > 0$ such that $\epsilon^{\mathrm{s}}_{\mathbb{P}\mathrm{B}}(p) > \epsilon > \epsilon^{\mathrm{s}}_{\mathrm{B}}$. Hence,

$$\forall\, \Gamma^{\mathrm{ran}} \in \mathrm{RGA}\ \exists\, \iota \in \Omega \text{ such that } \mathbb{P}_\iota(d_{\mathcal{M}}(\Gamma^{\mathrm{ran}}_\iota, \Xi(\iota)) > \epsilon) \leq p. \tag{5.1.1}$$

However, since $\epsilon > \epsilon^{\mathrm{s}}_{\mathrm{B}}$ there exists a general algorithm Γ such that for all $\iota \in \Omega$, we have that $d_{\mathcal{M}}(\Gamma(\iota), \Xi(\iota)) \leq \epsilon$. Since any general algorithm Γ obviously is a RGA with $X = \{\Gamma\}$, the latter statement violates (5.1.1) since $p < 1$. Note that the argument to establish (2.8.3) is identical to the proof of (2.8.2). Thus, we concentrate on (2.8.4).

Indeed, note that $\epsilon^{\mathrm{lr}}_{\mathbb{P}\mathrm{B}}(M, p) \leq \epsilon^{\mathrm{lr}}_{\mathbb{P}\mathrm{B}}(M, q)$ follows directly from the definition. Moreover, to see that $\epsilon^{\mathrm{lr}}_{\mathbb{P}\mathrm{B}}(M, q) \leq \epsilon^{\mathrm{lr}}_{\mathrm{B}}(M)$ we suppose the opposite and find an $\epsilon > 0$ such that $\epsilon^{\mathrm{lr}}_{\mathbb{P}\mathrm{B}}(M, q) > \epsilon > \epsilon^{\mathrm{lr}}_{\mathrm{B}}(M)$. Hence,

$$\forall\ \Gamma^{\mathrm{ran}} \in \mathrm{RGA}\ \exists\, \iota \in \Omega \text{ such that } \mathbb{P}_\iota(d_{\mathcal{M}}(\Gamma^{\mathrm{ran}}_\iota, \Xi(\iota)) \leq \epsilon \text{ and } T_{\Gamma^{\mathrm{ran}}}(\iota) \leq M) \leq p. \tag{5.1.2}$$

However, since $\epsilon > \epsilon^{\mathrm{lr}}_{\mathrm{B}}(M)$ we have that there exists a general algorithm Γ such that

$$\forall \iota \in \Omega\ d_{\mathcal{M}}(\Gamma(\iota), \Xi(\iota)) \leq \epsilon \text{ and } T_\Gamma(\iota) \leq M. \tag{5.1.3}$$

As argued above any general algorithm is also a randomised algorithm, and thus (5.1.3) contradicts (5.1.2). To address (2.8.5) we notice that $\epsilon^{\mathrm{s}}_{\mathbb{P}\mathrm{B}}(p) \leq \epsilon^{\mathrm{w}}_{\mathbb{P}\mathrm{B}}(p)$ follows directly from the definition.

Now, to show (2.8.6) (i.e. that $\epsilon^{\mathrm{w}}_{\mathbb{P}\mathrm{B}}(p) \leq \epsilon^{\mathrm{lr}}_{\mathbb{P}\mathrm{B}}(M, p)$ for $p \in [1/2, 1)$) we argue by contradiction and choose $\epsilon > 0$ such that $\epsilon^{\mathrm{w}}_{\mathbb{P}\mathrm{B}}(p) > \epsilon > \epsilon^{\mathrm{lr}}_{\mathbb{P}\mathrm{B}}(M, p)$. Then

$$\exists\, \Gamma^{\mathrm{ran}} \in \mathrm{RGA}\ \forall \iota \in \Omega\ \mathbb{P}_\iota(d_{\mathcal{M}}(\Gamma^{\mathrm{ran}}_\iota, \Xi(\iota)) \leq \epsilon \text{ and } T_{\Gamma^{\mathrm{ran}}}(\iota) \leq M) > p, \tag{5.1.4}$$

which implies that

$$\exists\, \Gamma^{\mathrm{ran}} \in \mathrm{RGA}\ \forall \iota \in \Omega\ \mathbb{P}_\iota(d_{\mathcal{M}}(\Gamma^{\mathrm{ran}}_\iota, \Xi(\iota)) > \epsilon \text{ or } T_{\Gamma^{\mathrm{ran}}}(\iota) > M) \leq 1 - p. \tag{5.1.5}$$

However, since $\epsilon^{\mathrm{w}}_{\mathbb{P}\mathrm{B}}(p) > \epsilon$ it follows that

$$\forall\, \Gamma^{\mathrm{ran}} \in \mathrm{RGA}\, \exists\, \iota \in \Omega\ \mathbb{P}_\iota(d_{\mathcal{M}}(\Gamma^{\mathrm{ran}}_\iota, \Xi(\iota)) > \epsilon \text{ or } T_{\Gamma^{\mathrm{ran}}}(\iota) > M) > p,$$

which contradicts (5.1.5) and (5.1.4) since $p \in [1/2, 1)$. To show (2.8.7) we note that $\epsilon^{\mathrm{s}}_{\mathrm{B}} \leq \epsilon^{\mathrm{w}}_{\mathrm{B}}$ follows immediately from the definitions. Moreover, to show that $\epsilon^{\mathrm{w}}_{\mathrm{B}} \leq \epsilon^{\mathrm{lr}}_{\mathrm{B}}(M)$ one can copy the proof of (2.8.6) almost verbatim.

Finally, to show (2.8.8) we start by showing the second part and observe that $\epsilon^{\mathrm{lr}}_{\mathbb{P}\mathrm{B}}(M, p)$ is bounded below and decreases as M increases, hence, the limit exists in $[0, \infty]$. We will show the result that $\lim_{M\to\infty} \epsilon^{\mathrm{lr}}_{\mathbb{P}\mathrm{B}}(M, p) \leq \epsilon^{\mathrm{w}}_{\mathrm{B}}$ for $p \in [0, 1/2]$: this establishes immediately the first part of (2.8.8). By (2.8.6) we have that $\epsilon^{\mathrm{w}}_{\mathbb{P}\mathrm{B}}(p) \leq \epsilon^{\mathrm{lr}}_{\mathbb{P}\mathrm{B}}(M, p)$ when $p = [1/2, 1)$ establishing the last part of (2.8.8). Note that there is nothing to prove if $\epsilon^{\mathrm{w}}_{\mathrm{B}} = \infty$, so let us assume instead that $\epsilon^{\mathrm{w}}_{\mathrm{B}} < \infty$.

To finalise the proof let $\epsilon > \epsilon^{\mathrm{w}}_{\mathbb{P}\mathrm{B}}(p)$. Then $\epsilon^{\mathrm{lr}}_{\mathbb{P}\mathrm{B}}(M, p) \leq \epsilon$ for large M. Indeed, if not, then for any $M \in \mathbb{N}$ we have that (5.1.2) holds. However, since $\epsilon > \epsilon^{\mathrm{w}}_{\mathbb{P}\mathrm{B}}(p)$, it follows that

$$\exists\, \Gamma^{\mathrm{ran}} \in \mathrm{RGA}, M \in \mathbb{N}\ \forall\, \iota \in \Omega\ \ \mathbb{P}_\iota(d_{\mathcal{M}}(\Gamma^{\mathrm{ran}}_\iota, \Xi(\iota)) > \epsilon \text{ or } T_{\Gamma^{\mathrm{ran}}}(\iota) > M) \leq p,$$

which implies that

$$\forall\, \iota \in \Omega\ \ \mathbb{P}_\iota(d_{\mathcal{M}}(\Gamma^{\mathrm{ran}}_\iota, \Xi(\iota)) \leq \epsilon \text{ and } T_{\Gamma^{\mathrm{ran}}}(\iota) \leq M) > 1 - p$$

which contradicts (5.1.2) if $p \in (0, 1/2]$. Thus, we have shown that $\lim_{M\to\infty} \epsilon^{\mathrm{lr}}_{\mathbb{P}\mathrm{B}}(M, p) \leq \epsilon^{\mathrm{w}}_{\mathbb{P}\mathrm{B}}(p)$ when $p \in (0, 1/2]$. The proof that $\lim_{M\to\infty} \epsilon^{\mathrm{lr}}_{\mathrm{B}}(M) = \epsilon^{\mathrm{w}}_{\mathrm{B}}$ is almost verbatim, and we omit the details. □

5.2 Proof of Theorem 2.9.1

Proof of Theorem 2.9.1. To prove (i), we argue by contradiction and assume that

$$\epsilon^{\mathrm{lr}}_{\mathbb{P}\mathrm{B}}(M, p) < \kappa/2, \qquad p \in [1/2, 1], \quad M \leq N - 2. \tag{5.2.1}$$

To reach the contradiction we first define the sequence $\{\iota_n\}_{n=1}^{2N}$ by the following formula: $\iota_{2n} = \iota^1_n, \iota_{2n-1} = \iota^2_n$. Without loss of generality we can assume that $\Omega = \{\iota_n \,|\, n \geq 1\}$. Our aim will be to produce Δ_1 information for Ω.

For $m, n \in \mathbb{N}$ and $f \in \Lambda$, choose $d^{n,m}_f \in \mathbb{Q}$ (or $d^{n,m}_f \in \mathbb{Q} + i\mathbb{Q}$ in the complex case) such that $|f(\iota^n) - d^{n,m}_f| \leq 2^{-m}$. We also choose $c^m_f \in \mathbb{Q}$ (again $c^m_f \in \mathbb{Q} + i\mathbb{Q}$ in the complex case) such that $|c^m_f - c_f| \leq 2^{-m-1}$ and $|c^m_f - f(\iota^m)| \leq 2^{-m}$ where c_f is defined as in assumption (b). Such a c^m_f exists because $|c_f - f(\iota^m)| \leq 2^{-m}$ again by

assumption (b). We define for $\iota \in \Omega$,

$$\hat{f}_m(\iota) = d_f^{n,m} \text{ if } \iota = \iota_n \text{ for some } n \leq m, \quad \hat{f}_m(\iota) = c_f^m \text{ otherwise.} \tag{5.2.2}$$

We claim that $\hat{\Lambda} := \{\hat{f}_m \,|\, f \in \Lambda \text{ and } m \in \mathbb{N}\}$ provides Δ_1 information for Λ. Indeed, for $n > m$ with $n, m \in \mathbb{N}$, we have

$$|\hat{f}_m(\iota_n) - f(\iota_n)| = |c_f^m - f(\iota_n)| \leq |c_f^m - c_f| + |c_f - f(\iota_n)| \leq 2^{-m-1} + 2^{-n} \leq 2^{-m}$$

since $2^{-n} \leq 2^{-m-1}$, whereas for $n \leq m$ we have $|\hat{f}_m(\iota_n) - f(\iota_n)| = |d_f^{n,m} - f(\iota_n)| \leq 2^{-m}$ by the above definition of $d_f^{n,m}$. Therefore $\hat{\Lambda}$ provides Δ_1 information (in the sense of both $\Delta_{1,\mathbb{Q}}$ or $\Delta_{1,\mathbb{R}}$ defined in Definition 2.6.1) for Λ.

The rest of the argument is based on demonstrating a contradiction when the algorithms are used with this particular choice of $\hat{\Lambda}$. To do that we note that by assumption (5.2.1) there is an $\epsilon < \kappa/2$ such that

$$\exists\ \Gamma^{\text{ran}} \in \text{RGA with } T_{\mathbb{P}}(\Gamma^{\text{ran}}, \epsilon, 1/2) \leq M,$$

where we recall the definition of $T_{\mathbb{P}}(\Gamma^{\text{ran}}, \epsilon, 1/2)$ from (2.8.1). Note that since $T_{\mathbb{P}}(\Gamma^{\text{ran}}, \epsilon, 1/2) \leq M$ we have that for all $\iota \in \Omega$

$$\mathbb{P}_\iota(d_{\mathcal{M}}(\Gamma_\iota^{\text{ran}}, \Xi(\iota)) \leq \epsilon \text{ and } T_{\Gamma^{\text{ran}}}(\iota) \leq M) > 1/2.$$

Hence, for all $\iota \in \Omega$

$$\mathbb{P}_\iota(d_{\mathcal{M}}(\Gamma_\iota^{\text{ran}}, \Xi(\iota)) > \epsilon \text{ or } T_{\Gamma^{\text{ran}}}(\iota) > M) < 1/2, \tag{5.2.3}$$

which is a crucial bound that will be essential below. Next we define three sets that form the basis for the rest of the argument. In particular, for $n \in \mathbb{N}$ and $\iota \in \Omega$ let $\mathcal{T}^n(\iota) := \{\Gamma \in X \,|\, T_\Gamma(\iota) \leq n\}$, and also define

$$F_j := \{\Gamma \in X \,|\, d_{\mathcal{M}}(\Gamma(\iota_{M+j}), \Xi(\iota_{M+j})) \geq \kappa/2 \text{ or } T_\Gamma(\iota_{M+j}) > M\}, \qquad j = 1, 2,$$

and

$$\hat{F}_2 = \{\Gamma \in X \,|\, d_{\mathcal{M}}(\Gamma(\iota_{M+2}), \Xi(\iota_{M+2})) \geq \kappa/2\},$$
$$\tilde{F}_2 = \{\Gamma \in X \,|\, T_\Gamma(\iota_{M+2}) > M\}.$$

Note that it is clear, from the continuity of the metric $d_{\mathcal{M}}$ as well as (Pi) and (Piii) in Definition 2.7.1 of an RGA, that F_1, F_2, $\hat{F}_2$ and $\tilde{F}_2$ are measurable. Observe that by (5.2.3) there is a $p < 1/2$ such that $\mathbb{P}_{\iota_{M+j}}(F_j) \leq p$ for $j = 1, 2$.

Claim 1: We now claim the following.

(I) $X = F_1 \cup \hat{F}_2$.

(II) $\mathbb{P}_{\iota_{M+1}}(\hat{F}_2 \cap \tilde{F}_2^c) = \mathbb{P}_{\iota_{M+2}}(\hat{F}_2 \cap \tilde{F}_2^c)$.

(III) $\mathbb{P}_{\iota_{M+1}}(\tilde{F}_2) = \mathbb{P}_{\iota_{M+2}}(\tilde{F}_2)$

To see (I) we argue as follows. If $\Gamma \in X$ then either $T_\Gamma(\iota_{M+1}) > M$ or $T_\Gamma(\iota_{M+1}) \leq M$. If $T_\Gamma(\iota_{M+1}) > M$ then obviously $\Gamma \in F_1$ by its definition. Thus, we only need to consider the case $T_\Gamma(\iota_{M+1}) \leq M$. Proceeding with this case we let $\hat{f}_k$ be in $\hat{\Lambda}_\Gamma(\iota_{M+1})$. From the fact that $T_\Gamma(\iota_{M+1}) \leq M$, we see that $k \leq M$. Therefore, by (5.2.2) it follows that $\hat{f}_k(\iota_{M+1}) = \hat{f}_k(\iota_{M+2}) = c_f^k$.

Hence, by property (iii) in Definition 2.4.3 of a general algorithm, it follows that $\hat{\Lambda}_\Gamma(\iota_{M+1}) = \hat{\Lambda}_\Gamma(\iota_{M+2})$. Thus, by (ii) in Definition 2.4.3, $\Gamma(\iota_{M+1}) = \Gamma(\iota_{M+2})$. If $\Gamma \notin F_1$ nor $\Gamma \notin \hat{F}_2$ then $d(\Gamma(\iota_{M+1}), \Xi(\iota_{M+1})) < \kappa/2$ and $d(\Gamma(\iota_{M+1}), \Xi(\iota_{M+2})) = d(\Gamma(\iota_{M+2}), \Xi(\iota_{M+2})) < \kappa/2$. Therefore $d(\Xi(\iota_{M+1}), \Xi(\iota_{M+2})) < \kappa$, which contradicts (a), and hence we conclude that $\Gamma \in F_1$ or $\Gamma \in \hat{F}_2$, and therefore (II) holds.

To prove (II) and (III) it suffices to demonstrate that

$$\mathbb{P}_{\iota_{M+1}}(E \cap \tilde{F}_2^c) = \mathbb{P}_{\iota_{M+2}}(E \cap \tilde{F}_2^c) \qquad \forall\, E \in \mathcal{F}. \tag{5.2.4}$$

Indeed, given (5.2.4) (II) follows immediately, and (III) follows by letting $E = X$ since $\mathbb{P}_{\iota_{M+1}}(\tilde{F}_2) = 1 - \mathbb{P}_{\iota_{M+1}}(\tilde{F}_2^c) = 1 - \mathbb{P}_{\iota_{M+2}}(\tilde{F}_2^c) = \mathbb{P}_{\iota_{M+2}}(\tilde{F}_2)$. To show (5.2.4) consider $E \in \mathcal{F}$ that is not empty (if it is empty, there is nothing to prove). Let $\Gamma \in E \cap \tilde{F}_2^c$ and $\hat{f}_k \in \hat{\Lambda}_\Gamma(\iota_{M+2})$. Since

$$\tilde{F}_2^c = \{\Gamma \in X \,|\, T_\Gamma(\iota_{M+2}) \leq M\}$$

it follows that $k \leq M$. Hence, by (5.2.2) it follows that $\hat{f}_k(\iota_{M+1}) = \hat{f}_k(\iota_{M+2}) = c_f^k$. Thus, by (Pii) in Definition 2.7.1 of a randomised general algorithm, we immediately get (5.2.4), and we have proved the claim.

Armed with the claim we can now derive the final contradiction. In particular,

$$\begin{aligned}
1 &= \mathbb{P}_{\iota_{M+1}}(F_1) + \mathbb{P}_{\iota_{M+1}}(\hat{F}_2) && \text{by (I)} \\
&= \mathbb{P}_{\iota_{M+1}}(F_1) + \mathbb{P}_{\iota_{M+1}}(\hat{F}_2 \cap \tilde{F}_2^c) + \mathbb{P}_{\iota_{M+1}}(\hat{F}_2 \cap \tilde{F}_2) && \\
&\leq \mathbb{P}_{\iota_{M+1}}(F_1) + \mathbb{P}_{\iota_{M+2}}(\hat{F}_2 \cap \tilde{F}_2^c) + \mathbb{P}_{\iota_{M+1}}(\tilde{F}_2) && \text{by (II)} \\
&= \mathbb{P}_{\iota_{M+1}}(F_1) + \mathbb{P}_{\iota_{M+2}}(\hat{F}_2 \cap \tilde{F}_2^c) + \mathbb{P}_{\iota_{M+2}}(\tilde{F}_2) && \text{by (III)} \\
&\leq \mathbb{P}_{\iota_{M+1}}(F_1) + \mathbb{P}_{\iota_{M+2}}(F_2) \leq 2p < 1 && \text{since } \hat{F}_2 \subset F_2,
\end{aligned}$$

which yields the final contradiction.

To show (ii) we argue by contradiction and assume that $\epsilon^{\mathrm{w}}_{\mathbb{P}\mathrm{B}}(p) < \kappa/2$ for $p \in [0, 1/2)$. The argument is almost identical to the proof of (i). Indeed, we use the setup in (5.2.2) and by assumption there is an $\epsilon < \kappa/2$ such that

$$\exists\, \Gamma^{\mathrm{ran}} \in \mathrm{RGA}, M \in \mathbb{N} \text{ s.t. } \forall\, \iota \in \Omega\ \mathbb{P}_\iota(d_{\mathcal{M}}(\Gamma^{\mathrm{ran}}_\iota, \Xi(\iota)) \geq \epsilon \text{ or } T_{\Gamma^{\mathrm{ran}}}(\iota) > M) \leq p < 1/2,$$

which is identical to the statement in (5.2.3). Thus, the rest of the rest of the argument is identical to the proof of (i) after (5.2.3).

To prove (iii), we define the sequence $\{\iota_n\}_{n\geq 0}$ by the following formula (similar to before, but note that now ι_0 is defined): $\iota_{2n} = \iota^1_n, \iota_{2n-1} = \iota^2_n$, with $n \geq 1$, and $\iota_0 = \iota^0$. It will suffice to show that the probabilistic breakdown epsilon $\epsilon^{\mathbb{P}}_{\mathrm{B}}(p) \geq \kappa/2$. If this is the case then by Lemma 2.8.4 it immediately follows that $\epsilon^{\mathrm{s}}_{\mathrm{B}}, \epsilon^{\mathbb{P}}_{\mathrm{B}}(p) \geq \kappa/2$ and that

$\{\Xi, \Omega, \mathcal{M}, \Lambda\}^{\Delta_1} \notin \Delta_1^G$. To show the required bound on the probabilistic breakdown epsilon we shall assume instead that $\epsilon_{\mathrm{B}}^{\mathbb{P}}(p) < \kappa/2$ and argue by contradiction.

Without loss of generality it will suffice to assume that $\Omega = \{\iota_n\}_{n\geq 0}$. We set $\hat{f}_m$ as in the part (i) with the additional definition $\hat{f}_m(\iota_0) = c_f^m$ for every $m \in \mathbb{N}$. The proof that $\hat{\Lambda} := \{\hat{f}_m \,|\, f \in \Lambda \text{ and } m \in \mathbb{N}\}$ provides Δ_1 information for Λ is similar to the proof of part (i): indeed, as before we see that $|\hat{f}_m(\iota_n) - f(\iota_n)| \leq 2^{-m}$ for $n, m \in \mathbb{N}$. However, there is now an additional element in Ω: namely ι_0. We have $|\hat{f}_m(\iota_0) - f(\iota_0)| = |c_f^m - c_f| \leq 2^{-m-1}$ and this completes the proof that $\hat{\Lambda}$ is indeed Δ_1 information for Λ.

Next, for $n \in \mathbb{N}$, we define the failure sets F^n by

$$F^n := \{\Gamma \in X \,|\, d_{\mathcal{M}}(\Gamma(\iota_n), \Xi(\iota_n)) \geq \kappa/2\},$$

and the collection of general algorithms that have minimum runtime bounded by $n \in \mathbb{N}$ by $\mathcal{T}^n(\iota) := \{\Gamma \in X \,|\, T_\Gamma(\iota) \leq n\}$. Note that by arguing as above it is clear that F^n is $\mathcal{F}$ measurable. By the assumption that $\epsilon_{\mathrm{B}}^{\mathbb{P}}(p) < \kappa/2$

$$\exists\ \Gamma^{\mathrm{ran}} \in \mathrm{RGA} \text{ s.t. } \forall\, n \in \mathbb{N}\ \mathbb{P}_\iota(F^n) \leq p,$$

We will show that this leads to the desired contradiction.

Claim 2: We claim the following.

(I) There is an n such that $\mathbb{P}_{\iota_0}(T^n(\iota_0)) > 2p$.

(II) For any $n \in \mathbb{N}$, $\mathcal{T}^n(\iota_0) = (F^{n+1} \cap \mathcal{T}^n(\iota_0)) \cup (F^{n+2} \cap \mathcal{T}^n(\iota_0))$.

(III) For any $n \in \mathbb{N}$, both $\mathbb{P}_{\iota_0}(F^{n+1} \cap \mathcal{T}^n(\iota_0)) = \mathbb{P}_{\iota_{n+1}}(F^{n+1} \cap \mathcal{T}^n(\iota_0))$ and $\mathbb{P}_{\iota_0}(F^{n+2} \cap \mathcal{T}^n(\iota_0)) = \mathbb{P}_{\iota_{n+2}}(F^{n+2} \cap \mathcal{T}^n(\iota_0))$.

The contradiction arises by combining these results: indeed, by (I) and (II), $2p < \mathbb{P}_{\iota_0}(\mathcal{T}^n(\iota_0)) = \mathbb{P}_{\iota_0}((F^{n+1} \cap \mathcal{T}^n(\iota_0)) \cup (F^{n+1} \cap \mathcal{T}^n(\iota_0)))$ and by (III) we get that $\mathbb{P}_{\iota_0}(F^{n+1} \cap \mathcal{T}^n) + \mathbb{P}_{\iota_0}(F^{n+2} \cap \mathcal{T}^n) = \mathbb{P}_{\iota_{n+1}}(F^{n+1} \cap \mathcal{T}^n) + \mathbb{P}_{\iota_{n+2}}(F^{n+2} \cap \mathcal{T}^n)$. Therefore

$$\begin{aligned} 2p < \mathbb{P}_{\iota_0}(\mathcal{T}^n(\iota_0)) &= \mathbb{P}_{\iota_0}\left((\mathcal{T}^n(\iota_0) \cap F^{n+1}) \cup (\mathcal{T}^n(\iota_0) \cap F^{n+2})\right) \\ &\leq \mathbb{P}_{\iota_0}(\mathcal{T}^n(\iota_0) \cap F^{n+1}) + \mathbb{P}_{\iota_0}(\mathcal{T}^n(\iota_0) \cap F^{n+2}) \\ &= \mathbb{P}_{\iota_{n+1}}(\mathcal{T}^n(\iota_0) \cap F^{n+1}) + \mathbb{P}_{\iota_{n+2}}(\mathcal{T}^n(\iota_0) \cap F^{n+2}) \\ &\leq \mathbb{P}_{\iota_{n+1}}(F^{n+1}) + \mathbb{P}_{\iota_{n+2}}(F^{n+2}) \leq 2p \end{aligned}$$

which is the desired contradiction. Thus our problem is reduced to showing (I), (II) and (III).

For (I), suppose that we have $\mathbb{P}_{\iota_{M+1}}(\mathcal{T}^n(\iota_{M+1})) \leq 2p$ for all $n \in \mathbb{N}$. Note that we clearly have that $\mathcal{T}^n(\iota_{M+1}) \subset \mathcal{T}^{n+1}(\iota_{M+1})$. Moreover, since, for every $\Gamma \in X$ the set $\Lambda_\Gamma(\iota_{M+1})$ is finite by (i) in Definition 2.4.3 of a general algorithm, it follows that $X = \bigcup_{n=1}^{\infty} \mathcal{T}^n(\iota_{M+1})$. Hence,

$$P_{\iota_{M+1}}(X) = \lim_{n\to\infty} P_{\iota_{M+1}}(\mathcal{T}^n(\iota_{M+1})) \leq 2p < 1,$$

since $p < 1/2$, which is a contradiction.

To prove (II) and (III), we make the intermediary step of showing that if $\Gamma \in \mathcal{T}^n(\iota_0)$ then $\Gamma(\iota_{n+1}) = \Gamma(\iota_{n+2})$. The argument is similar to the part (i): if $\hat{f}_m \in \hat{\Lambda}_\Gamma(\iota_0)$ then $m < n+1, n+2$ and so $\hat{f}_m(\iota_0) = \hat{f}_m(\iota_{n+1}) = \hat{f}_m(\iota_{n+2})$. Hence, by property (iii) in Definition 2.4.3 of a general algorithm, it follows that $\hat{\Lambda}_\Gamma(\iota_0) = \hat{\Lambda}_\Gamma(\iota_{n+1})$, and similarly we get that $\hat{\Lambda}_\Gamma(\iota_0) = \hat{\Lambda}_\Gamma(\iota_{n+2})$.

However, the fact that $\hat{\Lambda}_\Gamma(\iota_0) = \hat{\Lambda}_\Gamma(\iota_{n+1}) = \hat{\Lambda}_\Gamma(\iota_{n+2})$, (ii) and (iii) in Definition 2.4.3 imply that $\Gamma(\iota_0) = \Gamma(\iota_{n+1})$ and $\Gamma(\iota_0) = \Gamma(\iota_{n+2})$, so $\Gamma(\iota_{n+1}) = \Gamma(\iota_{n+2})$.

We can now show (II). For any $\Gamma \in \mathcal{T}^n(\iota_0)$ we have shown that $\Gamma(\iota_{n+1}) = \Gamma(\iota_{n+2})$. If $\Gamma \notin F^{n+1}$ and $\Gamma \notin F^{n+2}$ then $d(\Gamma(\iota_{n+1}), \Xi(\iota_{n+1})) < \kappa/2$ and $d(\Gamma(\iota_{n+1}, \Xi(\iota_{n+2})) = d(\Gamma(\iota_{n+2}), \Xi(\iota_{n+2})) < \kappa/2$. Therefore $d(\Xi(\iota_{n+1}), \Xi(\iota_{n+2})) < \kappa$, which contradicts (a). This contradiction completes the proof of (II).

Finally, to prove (III), we note that both $\mathcal{T}^n(\iota_0) \cap F^{n+1}$ and $\mathcal{T}^n(\iota_0) \cap F^{n+2}$ are measurable because $\mathcal{T}^n(\iota_0), F^{n+1}$ and F^{n+2} are all measurable. We will show the result only for $\mathcal{T}^n(\iota_0) \cap F^{n+1}$ as the corresponding argument for $\mathcal{T}^n(\iota_0) \cap F^{n+2}$ is similar. If $\Gamma \in \mathcal{T}^n(\iota_0) \cap F^{n+1}$ then $\Gamma \in \mathcal{T}^n(\iota_0)$. As before, this implies that if $\hat{f}_m \in \hat{\Lambda}_\Gamma(\iota_0)$ then $\hat{f}_m(\iota_{n+1}) = \hat{f}_m(\iota_0)$. The result (III) follows immediately from (Pii) in Definition 2.7.1 of an RGA.

To prove (iv) we start with the LLPO statement. Clearly, we may assume without loss of generality that $\Omega = \{\iota_n\}_{n\geq 0}$. As is standard for the LLPO problem, let $\tilde{\Omega}$ be a subset of the collection of all binary sequences with the following property $\tilde{\Omega} = \{\{a_j\}_{j\in\mathbb{Z}_+} \,|\, |\{j \,|\, a_j = 1\}| \leq 1\}$, and

$$\tilde{\Xi}(\{a_j\}) = 1 \text{ if } a_{2j} = 0 \,\forall j \in \mathbb{Z}_+, \qquad \tilde{\Xi}(\{a_j\}) = 0 \text{ if } a_{2j+1} = 0 \,\forall j \in \mathbb{Z}_+.$$

Note that $\tilde{\Xi}$ is multi-valued for the case where $a_j = 0$ for all j. It suffices to show that there is a recursive mapping $\tilde{\Gamma}$ such that $\tilde{\Gamma}(\{a_j\}) = \tilde{\Xi}(\{a_j\})$. To construct $\tilde{\Gamma}$ we start by defining the real numbers $x : \tilde{\Omega} \to \mathbb{R}$, $x(\{a_j\}) = \sum_{j=0}^{\infty}(-1)^j a_j 2^{-(j+1)}$, $\quad x_n(\{a_j\}) = \sum_{j=0}^{n}(-1)^j a_j 2^{-(j+1)}$. Define $\iota : \tilde{\Omega} \to \Omega$ by

$$\iota(\{a_j\}) = \begin{cases} \iota_n^1 & \text{if } x(\{a_j\}) > 0, \text{ and } n = \min\{l \in \mathbb{N} \,|\, x_l(\{a_j\}) > 0\} \\ \iota_n^2 & \text{if } x(\{a_j\}) < 0, \text{ and } n = \min\{l \in \mathbb{N} \,|\, x_l(\{a_j\}) < 0\} \\ \iota^0 & \text{otherwise,} \end{cases} \tag{5.2.5}$$

Clearly ι is a bijection that gives an obvious identification. So, by a slight abuse of notation, we consider ι as an element in Ω. Now, for $f \in \Lambda$ and $k \in \mathbb{Z}_+$ define $\hat{f}_k : \Omega = \tilde{\Omega} \to \mathbb{R}$ (or $\hat{f}_k$ maps to $\mathbb{Q}$ in the Turing case) as follows,

$$\hat{f}_k(\iota) = \begin{cases} f(\iota_n^1) & \text{if } x_k(\{a_j\}) > 0, \text{ and } n = \min\{l \leq k \,|\, x_l(\{a_j\}) > 0\} \\ f(\iota_n^2) & \text{if } x_k(\{a_j\}) < 0, \text{ and } n = \min\{l \leq k \,|\, x_l(\{a_j\}) < 0\}, \\ f(\iota_{k+1}^1) & \text{otherwise.} \end{cases} \tag{5.2.6}$$

The key here is that $\hat{f}_k$ is clearly recursively defined from $\{a_j\}$ which will be crucial later on. Now let $\hat{\Lambda} := \{\hat{f}_k \,|\, f \in \Lambda \text{ and } k \in \mathbb{N}\}$. We claim that $\hat{\Lambda}$ provides Δ_1 information for

Λ. Indeed, for a given $k \in \mathbb{N}$, if $x_k(\{a_j\}) \neq 0$ then $f_k(\iota) = f(\iota)$. If instead $x_k(\{a_j\}) = 0$ then either a_j is the 0 sequence or $\iota = \iota_n^1$ or $\iota = \iota_n^2$ for some $n > k$. Let us consider these cases separately.

Firstly, if a_j is the 0 sequence then $\iota = \iota^0$. Moreover, $\hat{f}_k(\iota) = f(\iota_{k+1}^1)$. Thus

$$|f(\iota) - \hat{f}_k(\iota)| = |f(\iota^0) - f(\iota_{k+1}^1)| \leq 1/4^{k+1} \leq 1/2^k.$$

If instead $\iota = \iota_n^1$ for some $n > k$ then

$$|f(\iota) - \hat{f}_k(\iota)| \leq |f(\iota_n^1) - f(\iota^0)| + |f(\iota^0) - f(\iota_{k+1}^1)| \leq 1/4^n + 1/4^{k+1} \leq 1/2^k.$$

A similar argument shows that if $\iota = \iota_n^2$ then $|f(\iota) - \hat{f}_k(\iota)| \leq 1/2^k$. Thus, we have established the claim.

Since, by assumption, $\{\Xi, \Omega, \mathcal{M}, \Lambda\}^{\Delta_1} \in \Delta_1^{\mathrm{A}}$ we can choose an arithmetic tower $\{\Gamma_k\}$ such that

$$\Gamma_k(\iota) \in \mathcal{N}_{2^{-k}}(\Xi(\iota)) \quad \forall \iota \in \Omega. \tag{5.2.7}$$

where $\mathcal{N}_r(S)$ is the union of all open balls radius r about points $x \in S$. Now choose k such that $2^{-k} \leq \kappa/8$. To define our general height zero tower $\tilde{\Gamma} : \tilde{\Omega} \to \mathcal{M}$ for $\{\tilde{\Xi}, \tilde{\Omega}\}$ we proceed as follows: let

$$N := N(\iota(\{a_j\})) = \max\{m \in \mathbb{N} \,|\, \hat{f}_m \in \hat{\Lambda}_{\Gamma_k}(\iota(\{a_j\})),\, f \in \Lambda\}.$$

Now we can define $\tilde{\Gamma}$ as follows. If

$$\Gamma_k(\iota(\{a_j\})) \notin \mathcal{N}_{2^{-k}}(S^1) \cup \mathcal{N}_{2^{-k}}(S^2), \tag{5.2.8}$$

then we let $\tilde{\Gamma}(\{a_j\}) := \{0, 1\}$. If (5.2.8) is not satisfied then we define $\tilde{\Gamma}$ as follows:

$$\text{if } \hat{f}_N(\iota(\{a_j\})) = f(\iota_n^1), \text{ for some } n \leq N \; \forall f \in \Lambda \Rightarrow \tilde{\Gamma}(\{a_j\}) := 1, \tag{5.2.9}$$

$$\text{if } \hat{f}_N(\iota(\{a_j\})) = f(\iota_n^2), \text{ for some } n \leq N \; \forall f \in \Lambda \Rightarrow \tilde{\Gamma}(\{a_j\}) := 0, \tag{5.2.10}$$

and finally

$$\text{if } \hat{f}_N(\iota(\{a_j\})) = f(\iota_{N+1}^1) \; \forall f \in \Lambda \Rightarrow \tilde{\Gamma}(\{a_j\}) := \{0, 1\}. \tag{5.2.11}$$

First, note that by (5.2.6) and (2.9.1) it follows that $\tilde{\Gamma}$ is well defined. Second, we claim that $\tilde{\Gamma} = \tilde{\Xi}$.

Indeed, if (5.2.8) is satisfied then by (5.2.7) both $\Xi(\iota(\{a_j\}) \neq \{S^1\}$ and $\Xi(\iota(\{a_j\}) \neq \{S^2\}$, hence, by (5.2.5), $\iota = \iota^0$, which implies, by (5.2.5), that $\tilde{\Xi}(\{a_j\}) = \{0, 1\}$. If (5.2.9) is satisfied then, by (5.2.6) we have that $x_N(\{a_j\} > 0$, hence, $\tilde{\Xi}(\{a_j\}) = 1$. A similar argument shows that if (5.2.10) is satisfied then $\tilde{\Xi}(\{a_j\}) = 0$.

Finally, if (5.2.11) is satisfied then we claim that $\{a_j\}$ is identically the zero sequence. If the claim is true then $\tilde{\Xi}(\{a_j\}) = \{0, 1\}$, thus we only need to show the claim. Arguing by contradiction we assume the opposite. Then, by (5.2.11), there is a $j_0 > N$ such that $a_{j_0} = 1$ (so $\Xi(\iota(\{a_j\})) = S^1$ or $\Xi(\iota(\{a_j\})) = S^2$). Since (5.2.8) is

not satisfied we have (5.2.7), this leaves two options; either

$$\Gamma_k(\iota(\{a_j\})) \in \mathcal{N}_{2^{-k}}(S^1) \text{ or } \Gamma_k(\iota(\{a_j\})) \in \mathcal{N}_{2^{-k}}(S^2).$$

Let us first consider the case $\Gamma_k(\iota(\{a_j\})) \in \mathcal{N}_{2^{-k}}(S^1)$. If $\Xi(\iota(\{a_j\})) = S^2$ then by (5.2.7) we must have that

$$d_{\mathcal{M}}(S^1, S^2) \leq d_{\mathcal{M}}\left[\Xi(\iota(\{a_j\})), \Gamma_k(\iota(\{a_j\}))\right] + d_{\mathcal{M}}(S^1, \Gamma_k(\iota(\{a_j\}))) \leq 2^{-k} + 2^{-k} = 2^{-k+1}$$

and since $2^{-k} \leq \kappa/8$, we conclude that $d(S^1, S^2) < \kappa$, contradicting (a). Thus $\Xi(\iota(\{a_j\})) = S^1$.

We conclude that j_0 is an odd number. Now choose a different sequence $\{\tilde{a}_j\} \in \tilde{\Omega}$ such that $\tilde{a}_{j_1} = 1$ with $j_1 = j_0 + 1$. In this case $\Xi(\iota(\{\tilde{a}_j\}) = S^2$. Thus, since $\{\Xi, \Omega, \mathcal{M}, \Lambda\}^{\Delta_1} \in \Delta_1^{\mathrm{A}}$,

$$d_{\mathcal{M}}(\Gamma_k(\iota(\{\tilde{a}_j\})), S^2) \leq 2^{-k} \leq \kappa/8.$$

However, since $j_0 > N$ we have that $\hat{f}_k(\iota(\{a_j\}) = \hat{f}_k(\iota(\{\tilde{a}_j\})$ for all $k \leq N$. Hence, by (iii) in 2.4.3 of a general algorithm

$$\hat{\Lambda}_{\Gamma_k}(\iota(\{a_j\})) = \hat{\Lambda}_{\Gamma_k}(\iota(\{\tilde{a}_j\})). \tag{5.2.12}$$

Thus, by (ii) in Definition 2.4.3 of a general algorithm it follows that $\Gamma_k(\iota(\{a_j\})) = \Gamma_k(\iota(\{\tilde{a}_j\}))$ which contradicts that $\Gamma_k(\iota(\{a_j\})) \subset \mathcal{N}_{2^{-k}}(S^1)$ and (a). This finishes the case when $\Gamma_k(\iota(\{a_j\})) \subset \mathcal{N}_{2^{-k}}(S^1)$, however, the case $\Gamma_k(\iota(\{a_j\})) \subset \mathcal{N}_{2^{-k}}(S^2)$ follows almost verbatim from the argument above.

To finish the argument we only need to show that $\tilde{\Gamma}$ is recursive. Since Γ_k is recursive and the $\hat{f}_k$s are defined recursively, we finish the proof by observing that all the requirements in (5.2.8), (5.2.9), (5.2.10), (5.2.11) can be checked recursively. In particular, to check (5.2.9) for $r < \infty$ it suffices to calculate $\|x - S^1\|_r^r, \|x - S^2\|_r^r$ and compare with 2^{-kr}, which can all be done recursively. If $r = \infty$ then it suffices to compare element-wise values of $|x - S^2|$ against 2^{-k}.

To prove that if $\Xi(\iota^0) = S_2$ then $\{\Xi, \Omega, \mathcal{M}, \Lambda\}^{\Delta_1} \in \Delta_1^A$ implies decidability of the Halting problem, we show that $\{\Xi, \Omega, \mathcal{M}, \Lambda\}^{\Delta_1} \in \Delta_1^A$ implies decidability of LPO which implies decidability of the Halting problem. Define the sequence $\{\iota_n\}_{n\geq 0}$ as follows: $\iota_n = \iota_n^1$, for $n \geq 1$, and $\iota_0 = \iota^0$. As above we may assume without loss of generality that $\Omega = \{\iota_n\}_{n\geq 0}$ Let $\hat{\Omega} = \tilde{\Omega}$ as above and define

$$\hat{\Xi}(\{a_j\}) = 0 \text{ if } a_j = 0 \, \forall j \in \mathbb{Z}_+, \quad \hat{\Xi}(\{a_j\}) = 1 \text{ otherwise}$$

The rest of the proof is similar to the LLPO case and it suffices to show that there is a recursive mapping $\hat{\Gamma}$ such that $\hat{\Gamma}(\{a_j\}) = \hat{\Xi}(\{a_j\})$. To construct $\hat{\Gamma}$ we define the real numbers $x : \hat{\Omega} \to \mathbb{R}$, $x(\{a_j\}) = \sum_{j=0}^{\infty} a_j 2^{-(j+1)}$, $\quad x_n(\{a_j\}) = \sum_{j=0}^{n} a_j 2^{-(j+1)}$. Define

$\iota : \hat{\Omega} \to \Omega$ by

$$\iota(\{a_j\}) = \begin{cases} \iota^1_n & \text{if } x(\{a_j\} > 0, \text{ and } n = \min\{l \in \mathbb{N} \,|\, x_l(\{a_j\}) > 0\} \\ \iota^0 & \text{otherwise.} \end{cases} \tag{5.2.13}$$

Clearly ι is a bijection that gives an obvious identification. So, by a slight abuse of notation, where we consider ι as an element in Ω we can, for $f \in \Lambda$ and $k \in \mathbb{Z}_+$ define $\hat{f}_k : \Omega = \hat{\Omega} \to \mathbb{R}$ as follows,

$$\hat{f}_k(\iota) = f(\iota_n) \text{ if } x_k(\{a_j\}) > 0 \text{ and } \iota = \iota_n \text{ for some } n \leq k, \ \hat{f}_k(\iota) = f(\iota^1_{k+1}) \text{ otherwise.} \tag{5.2.14}$$

If we define $\hat{\Lambda} := \{\hat{f}_k \,|\, f \in \Lambda \text{ and } k \in \mathbb{Z}_+\}$, similar reasoning as above in the proof of the LLPO case yields that $\hat{\Lambda}$ provides Δ_1 information for Λ. Since, by assumption, $\{\Xi, \Omega, \mathcal{M}, \Lambda\}^{\Delta_1} \in \Delta^{\mathrm{A}}_1$ we can choose an arithmetic tower $\{\Gamma_k\}$ such that $\Gamma_k(\iota) \in \mathcal{N}_{2^{-k}}(\Xi(\iota)) \quad \forall \iota \in \Omega$. Now we choose k such that $2^{-k} \leq \kappa/8$. To define our general height zero tower $\hat{\Gamma} : \hat{\Omega} \to \{0, 1\}$ for $\{\tilde{\Xi}, \tilde{\Omega}\}$ we proceed as follows. Let

$$N := N(\iota(\{a_j\})) = \max\{m \in \mathbb{N} \,|\, \hat{f}_m \in \hat{\Lambda}_{\Gamma_k}(\iota(\{a_j\})), \, f \in \Lambda\}.$$

Note that N must be finite since Γ_k is a general algorithm and Λ is finite. If $\Gamma_k(\iota(\{a_j\})) \in \mathcal{N}_{2^{-k}}(S^1)$ then we let $\hat{\Gamma}(\{a_j\}) = 1$ and otherwise let $\hat{\Gamma}(\{a_j\}) = 0$.

We claim that $\hat{\Gamma}(\{a_j\}) = \hat{\Xi}(\{a_j\})$ and that $\hat{\Gamma}$ is recursive. Indeed, $\Gamma_k(\iota(\{a_j\})) \in \mathcal{N}_{2^{-k}}(S^1)$ implies that $\Xi(\iota(\{a_j\}) = S^1$ (since otherwise $d_{\mathcal{M}}(S^1, S^2) < \kappa$ as in the LLPO case), which means that $\iota(\{a_j\}) = \iota^1_n$ for some n. However, that implies that $x(\{a_j\}) > 0$ and hence $\hat{\Gamma}(\{a_j\}) = 1$. If $\Gamma_k(\iota(\{a_j\})) \notin \mathcal{N}_{2^{-k}}(S^1)$, then, since $\Gamma_k(\iota(\{a_j\})) \in \mathcal{N}_{2^{-k}}(\Xi(\iota(\{a_j\})))$, we must have that $\Xi(\iota(\{a_j\})) = S^2$ (again, otherwise $d_{\mathcal{M}}(S^1, S^2) < \kappa$). Moreover, this implies, by (5.2.13) that $\iota(\{a_j\}) = \iota^0$ and hence by (5.2.13), it follows that $\hat{\Gamma}(\{a_j\}) = 1$. To see that $\hat{\Gamma}$ is recursive we observe that by (5.2.14) $\hat{f}_k(\iota)$ is recursively defined for all $f \in \Lambda$ and all $k \in \mathbb{N}$. Since $\{\Gamma_n\}$ is an arithmetic tower it now follows that $\hat{\Gamma}$ is recursive. □

5.3 A key proposition on two minimisers

Our aim in this section will be to prove the following result (which is valid for complex inputs):

Proposition 5.3.1. *Let $\lambda > 0$, $y \in \mathbb{C}^m$ and $U \in \mathbb{C}^{m \times N}$. Let $\Xi^{\mathrm{BP}}(y, U, \delta)$ and $\Xi^{\mathrm{UL}}(y, U, \lambda)$ denote the solution sets of the l^1 Basis Pursuit denoising (since $\delta \geq 0$, basis pursuit is a subcase) and Unconstrained Lasso respectively defined in section 2.10 (extended in the obvious way for complex inputs). If $|\Xi^{\mathrm{BP}}(y, U, \delta)| \geq 2$, then there are distinct $x^1, x^2 \in M^{\mathrm{BP}}_{min}(y, U, \delta)$ such that for any $\epsilon > 0$ there are negative semidefinite diagonal matrices $E^1 = E^1(\epsilon)$ and $E^2 = E^2(\epsilon)$ with $\|E^1\|_{max}, \|E^2\|_{max} < \epsilon$ such that $x^1 \in \Xi^{\mathrm{BP}}(y, U + UE^1, \delta)$, $x^2 \in \Xi^{\mathrm{BP}}(y, U + UE^2, \delta)$ and $|\Xi^{\mathrm{BP}}(y, U + UE^1, \delta)| = |\Xi^{\mathrm{BP}}(y, U + UE^2, \delta)| = 1$. The same result will hold if in the previous statement $\Xi^{\mathrm{BP}}(y, U, \delta)$ is replaced by $\Xi^{\mathrm{UL}}(y, U, \lambda)$.*

To prove 5.3.1, we require two subsequent lemmas. The first lemma we state (but since it is well known, we relegate its proof to the appendix) lists fundamental properties of unconstrained lasso and basis pursuit denoising.

Lemma 5.3.2. *Let $M^{BP}(y,U,\delta)$ be the set of minimisers to the ℓ^1 basis pursuit denoising problem with input (y,U,δ) ($\delta \geq 0$) and let $M^{UL}(y,U,\lambda)$ be the set of minimisers to the ℓ^1 unconstrained lasso problem with input (y,U,λ), where ℓ^1 basis pursuit or unconstrained lasso are defined as in section 2.10. If $M = M^{BP}(y,U,\delta)$ or $M = M^{UL}(y,U,\lambda)$ and there are distinct $v^1, v^2 \in M$ then M satisfies the following properties*

1. *If $v \in M$ then $Uv = Uv^1$.*
2. *If instead $v \in \mathbb{C}^N$ and $Uv = Uv^1$ then $\|v\|_1 \geq \|v^1\|_1 = \|v^2\|_1$.*
3. *If $v \in \mathbb{C}^N$ and $Uv = Uv^1$ and $\|v\|_1 = \|v^1\|_1$ then $v \in M$.*

The next result is about the sets of minimisers with minimal support. We shall show in Lemma 5.3.3 that $M^{\mathrm{BP}}_{\min}(y,U,\delta)$ and $M^{\mathrm{UL}}_{\min}(y,U,\delta)$ are never empty. Indeed, we prove a stronger result:

Lemma 5.3.3. *If there is an $x \in M(\iota) \setminus M_{min}(\iota)$ then there are $w^1, w^2 \in M_{min}$ such that $supp(w^1) \not\subseteq supp(w^2)$ and $supp(w^2) \not\subseteq supp(w^1)$ (where ι is either valid basis pursuit denoising input or lasso input, M can be either M^{BP} or M^{UL} and likewise $M_{min} = M^{BP}_{min}$ or $M_{min} = M^{UL}_{min}$).*

Proof. The proof is divided into several steps.

Step I: Suppose that there are distinct $v^1, v^2 \in M$ with $\mathrm{supp}(v^2) \subseteq \mathrm{supp}(v^1)$, where M is either $\Xi^{\mathrm{BP}}(\iota)$ or $\Xi^{\mathrm{UL}}(\iota)$. We will produce a $v^3 \in M$ with $\mathrm{supp}(v^3) \subseteq \mathrm{supp}(v^1)$ and a $k \in \mathbb{N}$ such that $v^2_k \neq 0$ but $v^3_k = 0$. The proof is based in the three properties in Lemma 5.3.2 that are satisfied for both $M^{\mathrm{BP}}(\iota)$ and $M^{\mathrm{UL}}(\iota)$. Thus, the proofs for M^{BP} and M^{UL} are identical and so we shall write $M = M^{\mathrm{BP}}(\iota)$ or $M = M^{\mathrm{UL}}(\iota)$ without concerning ourselves with the underlying problem. First, we note that $\mathrm{sgn}(v^1_{\mathrm{supp}(v^2)}) = \mathrm{sgn}(v^2_{\mathrm{supp}(v^2)})$. where for a complex number $z = re^{i\theta}$ with $\theta \in [0, 2\pi)$ and $r \geq 0$, $\mathrm{sgn}(z) = e^{i\theta}$ whenever $r > 0$ and $\mathrm{sgn}(0) = 0$. Indeed, otherwise let $j \in \mathrm{supp}(v^2)$ be such that $\mathrm{sgn}(v^1_j) \neq \mathrm{sgn}(v^2_j)$ and set $\tilde{v} = (v^1 + v^2)/2$. Then $U\tilde{v} = (Uv^1 + Uv^2)/2 = Uv^1$ (by property (1) in Lemma 5.3.2) and

$$2\|\tilde{v}\|_1 = \sum_{i=1}^{n} |v^1_i + v^2_i| \leq |v^1_j + v^2_j| + \sum_{i=1, i\neq j}^{n} |v^1_i + v^2_i| < |v^1_j| + |v^2_j| + \sum_{i=1, i\neq j}^{n} |v^1_i| + |v^2_i|,$$

where the last sum is bounded by $\|v^1\|_1 + \|v^2\|_1$. Hence, since $\|v^1\|_1 + \|v^2\|_1 = 2\|v^1\|_1$ then $\|\tilde{v}\|_1 = \|v^1\|_1$ contradicting property (2) in Lemma 5.3.2. Furthermore, we claim that

$$S = \{i \in \{1, \ldots, N\} \,|\, |v^2_i| > |v^1_i|\} \neq \emptyset.$$

Arguing by contradiction, suppose that $S = \emptyset$. It is clear then that $\mathrm{supp}(v^2) = \mathrm{supp}(v^1)$, otherwise $\|v^1\|_1 > \|v^2\|_1$, contradicting property (2) in Lemma 5.3.2.

Therefore, we must have $|v_i^1| = |v_i^2|$ for every $i \in \text{supp}(v^2)$, because otherwise $\|v^1\|_1 = \sum_{i\in\text{supp}(v^2)} |v_i^1| > \sum_{i\in\text{supp}(v^2)} |v_i^2| = \|v^2\|_1$ contradicting property (2) in Lemma 5.3.2. But then $v_i^1 = v_i^2$ since $\text{sgn}(v_i^1) = \text{sgn}(v_i^2)$ for all $i \in \text{supp}(v^2)$, contradicting $v^1 \neq v^2$, and we have established the claim.

Let us set $e = v^1 - v^2$. Because $\{i \,|\, |v_i^2| > |v_i^1|\} \neq \emptyset$, we let k be any natural number satisfying

$$k \in \underset{1\leq i\leq N}{\text{argmin}} |v_i^1|/|e_i| \text{ such that } |v_i^2| > |v_i^1|.$$

Next, we set $\alpha = v_k^1/e_k$. Finally, we set $v^3 = v^1 - \alpha e$ and show that v^3 and k have the desired properties. To do this, we begin by observing that $\text{supp}(v^3) \subseteq \text{supp}(v^1)$. Furthermore, v_k^2 is non-zero (since $|v_k^2| > |v_k^1|$) but $v_k^3 = v_k^1 - e_k v_k^1/e_k = 0$. Since e is in the kernel of U we also have $Uv^3 = Uv^1$. It remains to show that $\|v^3\|_1 = \|v^1\|_1$ and the result will follow by property (3) in Lemma 5.3.2. Note that by the fact that $\text{sgn}(v^1_{\text{supp}(v^2)}) = \text{sgn}(v^2_{\text{supp}(v^2)})$., which we showed above, we have $\text{sgn}(v_k^2) = \text{sgn}(v_k^1)$. Thus, since $|v_k^2| > |v_k^1|$, it follows that $\text{sgn}(e_k) = -\text{sgn}(v_k^1)$ and so α is a negative real number. We will show that $|v_i^1| - \alpha|v_i^1| + \alpha|v_i^2| \geq 0$ for all i. Indeed, if $|v_i^1| \geq |v_i^2|$ then the claim follows because $-\alpha|v_i^1| + \alpha|v_i^2| = |\alpha|(|v_i^1| - |v_i^2|)$. If instead $|v_i^2| > |v_i^1|$ then by the definition of α we have $|v_i^1|/|v_i^1 - v_i^2| \geq |\alpha|$ so

$$|v_i^1| \geq |\alpha||v_i^1 - v_i^2| \geq |\alpha|(|v_i^2| - |v_i^1|) = -\alpha(|v_i^2| - |v_i^1|).$$

We conclude that

$$\begin{aligned}
\|v^3\|_1 = \sum_{i=1}^N |v_i^1 - \alpha e_i| &= \sum_{i=1}^N |\text{sgn}(v_i^1)|v_i^1| - \text{sgn}(v_i^1)\alpha(|v_i^1| - |v_i^2|)| \\
&= \sum_{i=1}^N ||v_i^1| - \alpha|v_i^1| + \alpha|v_i^2|| \\
&= \sum_{i=1}^N |v_i^1| - \alpha|v_i^1| + \alpha|v_i^2| = \|v^1\|_1 - \alpha\|v^1\|_1 + \alpha\|v^2\|_1 = \|v^1\|_1
\end{aligned}$$

where the first equality follows because $\text{sgn}(v^2) = \text{sgn}(v^1)$ on $\text{supp}(v^2)$, and the third equality follows from the fact that $|v_i^1| - \alpha|v_i^1| + \alpha|v_i^2| \geq 0$. Therefore $v^3 \in M$ and the required properties for v^3 and k are satisfied.

Step II: Using Step I we can now finish the proof. Set $x^0 = x$. We produce a collection of vectors x^n for $n = 0, 1, \ldots, N_0$ such that x^{N_0} is in $M_{\min}$. Recursively, suppose x^i is not in $M_{\min}$. Then we can find (by the definition of not being a member of $M_{\min}$) $\tilde{x}^{i+1} \in M$, such that $\text{supp}(\tilde{x}^{i+1}) \subseteq \text{supp}(x^i)$. and $\tilde{x}^{i+1} \neq x^i$. If $\text{supp}(\tilde{x}^{i+1}) \subsetneq \text{supp}(x^i)$ we set $x^{i+1} = \tilde{x}^{i+1}$. Otherwise, we apply Step I to the vectors x^i and $\tilde{x}^{i+1}$ (with $v^1 = x^i, v^2 = \tilde{x}^{i+1}$) to find x^{i+1} and k with $\text{supp}(x^{i+1}) \subseteq \text{supp}(x^i)$ and $x_k^{i+1} = 0, \tilde{x}_k^{i+1} \neq 0$. Since $\text{supp}(\tilde{x}^{i+1}) = \text{supp}(x^i)$, this implies that $x_k^i \neq 0$, so that $\text{supp}(x^{i+1})$ is a proper subset of $\text{supp}(x^i)$. In particular, since $\text{supp}(x^{i+1})$ is a proper subset of $\text{supp}(x^i)$, it is easy to see by induction that $|\text{supp}(x^i)| \leq |\text{supp}(x^0)| - i$. If the process does not terminate (i.e. every $x^i \in M \setminus M_{\min}$) then $\lim_{i\to\infty} |\text{supp}(x^i)| = -\infty$

which is a clear contradiction. Therefore the process terminates at some N_0. We set $w^1 = x^{N_0} \in M_{\min}$ and note that $\mathrm{supp}(w^1)$ is a proper subset of $\mathrm{supp}(x)$, or else $w^1 \notin M_{\min}$.

To find w^2, we apply Step I to the vectors x and w^1 (with $v^1 = x$, $v^2 = w^1$, noting that $w^1 \in M_{\min}$ and $x \notin M_{\min}$ implies that $x \neq w^1$) to produce a vector $\hat{x} \in M$ and a natural number k such that $\mathrm{supp}(\hat{x}) \subseteq \mathrm{supp}(x)$, $w^1_k \neq 0$ and $\hat{x}_k = 0$. We apply the same process as before to produce a collection of vectors $(\hat{x}^n)_{n=0,1,\ldots,N_1}$ such that $\hat{x}^{N_1} \in M_{\min}$ with $\mathrm{supp}(\hat{x}^{n+1}) \subseteq \mathrm{supp}(\hat{x}^n)$. We set $w^2 = \hat{x}^{N_1}$. Since $\mathrm{supp}(w^2) \subseteq \mathrm{supp}(\hat{x})$, $w^2_k = 0$. Therefore $\mathrm{supp}(w^1) \not\subseteq \mathrm{supp}(w^2)$. Finally,

$$w^2 \in M,\ w^1 \in M_{\min},\ \mathrm{supp}(w^2) \subseteq \mathrm{supp}(w^1) \Rightarrow w^2 = w^1.$$

But $w^2_k \neq w^1_k$, so $\mathrm{supp}(w^2) \not\subseteq \mathrm{supp}(w^1)$. □

The existence of minimisers with minimal support allows us to prove 5.3.1

Proof of Proposition 5.3.1. Let x^1 and x^2 be distinct vectors in $M^{\mathrm{BP}}(y, U, \delta)$ for the basis pursuit problem or $M^{\mathrm{UL}}(y, U, \lambda)$ for the lasso problem. Note that we may assume without loss of generality that x^1 and x^2 are both in $M^{\mathrm{BP}}_{\min}(y, U, \delta)$ (respectively $x^1, x^2 \in M^{\mathrm{UL}}_{\min}(y, U, \lambda)$ for the lasso problem). Otherwise, we can use Lemma 5.3.3 to find new distinct vectors which are in $M^{\mathrm{BP}}_{\min}$ (respectively $M^{\mathrm{lasso}}_{\min}$). We define the $N \times N$ diagonal negative semidefinite matrices

$$E^1(\epsilon) := -\beta_\epsilon(\mathbb{1}_{1\notin\mathrm{supp}(x^1)} \oplus \mathbb{1}_{2\notin\mathrm{supp}(x^1)} \oplus \ldots \oplus \mathbb{1}_{N\notin\mathrm{supp}(x^1)})$$
$$E^2(\epsilon) := -\beta_\epsilon(\mathbb{1}_{1\notin\mathrm{supp}(x^2)} \oplus \mathbb{1}_{2\notin\mathrm{supp}(x^2)} \oplus \ldots \oplus \mathbb{1}_{N\notin\mathrm{supp}(x^2)})$$

where $\beta_\epsilon = \min(1, \epsilon)/2$ and $\mathbb{1}_{i\notin\mathrm{supp}(x^1)}$ is 1 if $i \notin \mathrm{supp}(x^1)$ and 0 otherwise. Note that both $E^1(\epsilon)$ and $E^2(\epsilon)$ are nonzero because of Lemma 5.3.3. We now need to show for $j = 1, 2$ that x^j is the unique vector in $M(U + UE^j)$. We will argue for $j = 1$: the proof for $j = 2$ is analogous. Additionally, we will discuss basis pursuit and lasso separately.

Case 1 (Basis pursuit): Firstly, since $(U+UE^1)v = Uv$ on vectors v with $\mathrm{supp}(v) = \mathrm{supp}(x^1)$, $\|(U+UE^1)x^1 - y\|_2 \leq \delta$. Let us suppose that there is a vector $\tilde{x}^1$ with $\|(U + UE^1)\tilde{x}^1 - y\|_2 \leq \delta$ and $\|\tilde{x}^1\|_1 \leq \|x^1\|_1$. Let $\hat{x}^1 = \tilde{x}^1 + E^1\tilde{x}^1$. Clearly $\|U\hat{x}^1 - y\|_2 \leq \delta$. For every $j \in \mathrm{supp}(\hat{x}^1)$ with $j \notin \mathrm{supp}(x^1)$, it is easy to see that $|\hat{x}^1_j| = (1-\beta_\epsilon)|\tilde{x}^1_j| < |\tilde{x}^1_j|$ and for $j \in \mathrm{supp}(x^1)$ we have $\hat{x}^1_j = \tilde{x}^1_j$. Therefore if $\mathrm{supp}(\hat{x}^1)$ is not a subset of $\mathrm{supp}(x^1)$, then $\|\hat{x}^1\|_1 < \|\tilde{x}^1\|_1 \leq \|x^1\|_1$, contradicting the fact that $x^1 \in M^{\mathrm{BP}}(U)$. Thus $\mathrm{supp}(\hat{x}^1)$ is a subset of $\mathrm{supp}(x^1)$ so $\tilde{x}^1 = \hat{x}^1$. But then $\hat{x}^1 = x^1$ because $x^1 \in M^{\mathrm{BP}}_{\min}(U)$. Therefore x^1 is the unique vector in $M^{\mathrm{BP}}(y, U + UE^1, \delta)$.

Case 2 (Lasso): Let us suppose that $\tilde{x}^1 \in \mathbb{C}^N$ satisfies $\|(U + UE^1)\tilde{x}^1 - y\|_2^2 + \lambda\|\tilde{x}^1\|_1 \leq \|(U + UE^1)x^1 - y\|_2^2 + \lambda\|x^1\|_1$. As in the basis pursuit case, we set $\hat{x}^1 = \tilde{x}^1 + E^1\tilde{x}^1$. If $\mathrm{supp}(\hat{x}^1)$ is not a subset of $\mathrm{supp}(x^1)$, then $\|\hat{x}^1\|_1 < \|\tilde{x}^1\|_1$. Therefore, by

the above assumption,

$$\|U\hat{x}^1 - y\|_2^2 + \lambda\|\hat{x}^1\|_1 < \|(U + UE^1)\tilde{x}^1 - y\|_2^2 + \lambda\|\tilde{x}^1\|_1 \\ \leq \|(U + UE^1)x^1 - y\|_2^2 + \lambda\|x^1\|_1 = \|Ux^1 - y\|_2^2 + \lambda\|x^1\|_1$$

contradicting the fact that $x^1 \in M^{\mathrm{UL}}$. Thus $\mathrm{supp}(\hat{x}^1) = \mathrm{supp}(x^1)$. But then $\mathrm{supp}(\tilde{x}^1) = \mathrm{supp}(x^1)$, so $\|U\tilde{x}^1 - y\|_2^2 + \lambda\|\tilde{x}^1\|_1 = \|(U + UE^1)\tilde{x}^1 - y\|_2^2 + \lambda\|\tilde{x}^1\|_1$ $\leq \|(U + UE^1)x^1 - y\|_2^2 + \lambda\|x^1\|_1 = \|(U + UE^1)x^1 - y\|_2^2 + \lambda\|x^1\|_1$. We conclude that $\tilde{x}^1 = x^1$ because $x^1 \in M^{\mathrm{UL}}_{\min}(y, U, \lambda)$. Therefore x^1 is the unique vector in $M^{\mathrm{UL}}(y, U + UE^1, \lambda)$. □

5.4 Proof of Theorem 4.1.1

Part 1: As with the other computability results, our aim will be to construct inputs so as to employ proposition 2.9.1. Let $M^{\mathrm{LP}}(y, U, c)$ denote the solution to a linear programming problem of the form $\mathrm{argmin}_{x\in\mathbb{R}^N} c^T x$ such that $Ux = y, \quad x \geq 0$. We will consider inputs of the form

$$U^\epsilon = \epsilon \oplus \mathbf{0}_{m-1\times N-1}, \quad (y^\epsilon)^T = \epsilon \oplus 0_{m-1}, \quad c = (1, 1, \ldots, 1)$$

for $\epsilon \in [0, 1]$. It is clear that if $\iota = (y^\epsilon, U^\epsilon, c)$ then $\|\iota\|_2 \leq \epsilon \leq 1$. Moreover, the solution to $y^\epsilon = U^\epsilon x$ is uniquely given by $x = 1$ for $\epsilon > 0$. For $\epsilon = 0$, every $x \in \mathbb{R}$ is a solution to $U^\epsilon x = y^\epsilon$. However, we seek a non-negative x which minimises $c^T x = x$. Thus the solution for $\epsilon = 0$ to the linear programming is given by 0. To summarize, we have the following result

$$M^{\mathrm{LP}}(y^\epsilon, U^\epsilon, c) = \begin{cases} 1 & \text{if } \epsilon \in (0, 1] \\ 0 & \text{otherwise.} \end{cases}$$

In the setting of proposition 2.9.1, we set the sequence $\iota_n^1 := (y^{1/4^n}, U^{1/4^n}, c)$, the constant sequence $\iota_n^2 = (y^0, U^0, c)$, and $\iota^0 = (y^0, U^0, c)$. Therefore with $S^1 = \{e_1\}$, $S^2 = \{0\}$, we have $\Xi(\iota_n^1) = S^1, \Xi(\iota_n^2) = S^2$ and $\Xi(\iota^0) = S^2$. Therefore (a) is satisfied with $\kappa = 1$.

Note also that for $f \in \Lambda$, we have $f(\iota_n^2) = f(\iota^0)$ and $|f(\iota_n^1) - f(\iota^0)| \leq 1/4^n$. We conclude that (b) is satisfied with $c_f = f(\iota^0)$ and so $\epsilon_B^s, \epsilon_{\mathbb{P}\mathrm{B}}^s(p) \geq 1/2$ for $p \in (0, 1/2)$.

The only thing left to show is that $\{\Xi, \Omega\}^{\Delta_1} \in \Delta_1^A$ implies the decidability of the halting problem. Clearly the output space is exactly $\mathbb{R}$ with the ℓ^2 norm, Λ is finite and S_1 and S_2 are singletons by definition. (2.9.1) is obvious because for $\iota = (y, U, c)$ the function $f : \iota \to \mathbb{R}$ defined by $f(y) = y_1$ is in Λ. The result follows immediately by proposition 2.9.1 part (iv).

Part 2: Here, as for the other positive results, we will construct an algorithm that solves the problem and then prove its correctness. We assume the existence of recursive subroutines $\mathsf{PosDef}(M)$, $\mathsf{SolLin}(M, y)$, $\mathsf{FindV}(\iota, \epsilon)$ and $\mathsf{FindM}(\iota, \epsilon)$ that work in the following way:

- $\mathsf{PosDef}(M)$ returns true iff the matrix $M \in \mathbb{R}^{m\times m}$ is positive definite (i.e. for

all x, $\langle x, Mx\rangle \geq 0$ with equality iff $x = 0$). This can be done with an algorithm that is based on the cholesky decomposition.

- $\mathsf{SolLin}(M, y)$ solves for x the linear system of equations $y = Mx$ (where y and M are stored with exact precision). This can be achieved through gaussian elimination.

- $\mathsf{FindV}(\iota, \epsilon)$ defined for $\iota = (y, U)$ and $\epsilon > 0$ finds a vector v such that $\|v - y\| \leq \epsilon$. This can be achieved by making use of the Δ_1 information available to the algorithm.

- $\mathsf{FindM}(\iota, \epsilon)$ defined for $\iota = (y, U)$ and $\epsilon > 0$ finds a matrix M such that $\|M - U\| \leq \epsilon$. Again, this can be achieved through the use of the Δ_1 information available to the algorithm.

We then consider the following algorithm:
Algorithm $\mathsf{LinearSystems}(\iota, n)$
Input: $\iota \in \Omega$, $n \in \mathbb{N}$
Output: $\Gamma_n(\iota) \in \mathbb{R}^N$ with $\mathrm{dist}(\Gamma_n(\iota), \Xi(\iota)) \leq 2^{-n}$.
Subroutines: $\mathsf{PosDef}(M)$, $\mathsf{SolLin}(M, y)$ $\mathsf{FindV}(\iota, \epsilon)$, $\mathsf{FindM}(\iota, \epsilon)$.
We set the following parameters:

$$y^0 := \mathsf{FindV}(\iota, 1),\ M^0 := \mathsf{FindM}(\iota, 1),\ \varepsilon := 2^{-n},\ \delta := \frac{\varepsilon}{6}$$

$$C := \|y^0\|_1 \vee \|M^0\|_F^2 \vee 1, \quad N := \lceil C \rceil + 2,$$

Then we proceed in the following way:

$k = N$

do-while $\mathsf{PosDef}((M^k)^* M^k - \frac{I}{k}) = 0$

$k = k + 1$

$M^k = \mathsf{FindM}(I, \frac{\delta}{k^5})$

end

$y^k = \mathsf{FindV}(I, \frac{\delta}{k^5})$

$v^k = \mathsf{SolLin}((M^k)^* M^k, (M^k)^* y^k)$

$\Gamma_n(I) := v^k$

We begin by showing that the above *while-loop* will terminate. For $y = Ux$ to have a unique solution, U must be invertible. Hence for vectors v, $Uv = 0$ if and only if $v = 0$.

Suppose that the while loop does not terminate. Then there are a sequence of vectors $(x^k)_{k=N_0}^{\infty}$ with $x^k \in \mathbb{R}^N$, $\|x^k\|_2 = 1$ and $\langle x^k, \left[(M^k)^* M^k - \frac{I}{n}\right] x^k\rangle \leq 0$. By passing to a convergent subsequence using the compactness of the unit ball in $\mathbb{R}^N$,

we assume that $x^k \to x$ for some $x \in \mathbb{R}^N$ with $\|x\|_2 = 1$. Furthermore, because $\|M^k - U\|_2 \leq \delta/k^5$, we must have $M^k \to U$ as $k \to \infty$, since ϵ is defined independently of k. Thus $\langle x, U^*Ux\rangle = \lim_{k\to\infty} \langle x^k, \left((M^k)^*M^k - \frac{I}{k}\right) x^k\rangle \leq 0$. We conclude that x is in the nullspace of U. Thus $Ux = 0$ and so $x = 0$, contradicting he fact that $\|x\|_2 = 1$.

We conclude that Γ_n is an arithmetic tower of algorithms, and we are left with the problem of proving that $\mathrm{dist}(\Gamma_n(\iota), \Xi_{\mathrm{LS}}(\iota)) \leq 2^{-n}$ for any $\iota \in \Omega$. We will therefore show that $\|v^n - x\|_2 \leq 2^{-n}$ where $Ux = y$. With $X = U^*U$, $X^k = (M^k)^*M^k$, we have (since $x = (U^*U)^{-1}U^*Ux = (U^*U)^{-1}U^*y$),

$$\begin{aligned}\|v^k - x\|_2 &= \|(X^k)^{-1}(M^k)^*y^k - X^{-1}U^*y\|_2 \\ &\leq \|(X^k)^{-1}(M^k)^* - X^{-1}U^*\|_2\|y\|_2 + \|(X^k)^{-1}(M^k)^*\|_2\|y^k - y\|_2. \quad (5.4.1)\end{aligned}$$

Our aim will be to bound both $\|(X^k)^{-1}(M^k)^* - X^{-1}U^*\|_2\|y\|_2$ and $\|(X^k)^{-1}(M^k)^*\|_2\|y^k - y\|_2$. To do this, we show the following:

$$\|M^k\|_2, \|U\|_2, \|y\|_2, \|(X^k)^{-1}\|_2 \leq k, \quad \|M^k - U\|_2 \leq \frac{\delta}{k^5}, \quad \|y^k - y\|_2 \leq \frac{\delta}{k^2}. \quad (5.4.2)$$

The observation that $\|(X^k)^{-1}\|_2 \leq k$ is a consequence of the fact that (since the condition of the while loop is met) $X^k - I/k$ is positive definite and thus $\langle x - A^{-1}x/k, A^{-1}x\rangle \geq 0$. Therefore $\|x\|_2\|A^{-1}x\|_2 \geq \langle x, A^{-1}x\rangle \geq \|A^{-1}x\|^2/k$. The result follows by taking supremums over $\|x\|_2 = 1$.

Furthermore, by the initial parameter setup we have $\|M^0 - U\|_2, \|y^0 - y\|_2 \leq 1$ and $C \geq \|M^0\|_F^2 \vee 1 \geq \|M^0\|_2^2 \vee 1 \geq \|M^0\|_2$. We also have that $C \geq \|y^0\|_1 \geq \|y^0\|_2$. Therefore we must have $\|U\|_2, \|y\|_2 \leq C + 1$. Additionally, $\|M^k - U\|_2 \leq \frac{\delta}{k^5} \leq 1$ by the definition of FindM. Thus $\|M^k\|_2 \leq C + 2$. (5.4.2) follows immediately from the fact that $k \geq C + 2$ and the definition of M^k and y^k using FindM and FindV.

Next, we use (5.4.2) to bound $\|(X^k)^{-1}(M^k)^* - X^{-1}U^*\|_2$. We have

$$\begin{aligned}\|(X^k)^{-1}(M^k)^* - X^{-1}U^*\|_2 &\leq \|(X^k)^{-1}\|_2 \left\|\left(M^k - U\right)^*\right\|_2 + \|(X^k)^{-1} - X^{-1}\|_2\|U^*\|_2 \\ &\leq \frac{\delta}{k^4} + k\|(X^k)^{-1} - X^{-1}\|_2.\end{aligned}$$

To bound $\|(X^k)^{-1} - X^{-1}\|_2$, again we use (5.4.2) and note the following:

$$\begin{aligned}\|(X^k)^{-1} - X^{-1}\|_2 &\leq \|X^{-1}\|_2\|(X^k)^{-1}\|_2\|X - X^k\|_2 \\ &\leq k\|X^{-1}\| \left(\|(M^k)^*M^k - (M^k)^*U\|_2 + \|(M^k)^*U - U^*U\|_2\right) \\ &\leq k\left(\|(X^k)^{-1} - X^{-1}\|_2 + \|(X^k)^{-1}\|_2\right)\left(\|M^k\|_2 + \|U\|_2\right)\|M^k - U\|_2 \\ &\leq \frac{2\delta k^2}{k^5}\left(\|(X^k)^{-1} - X^{-1}\|_2 + k\right) \leq \frac{\|(X^k)^{-1} - X^{-1}\|_2}{2} + \frac{2\delta}{k^2}\end{aligned}$$

since $4\delta \leq 1 \leq k^3$. Thus $\|(X^k)^{-1} - X^{-1}\|_2 \leq \frac{4\delta}{k^2}$ and we conclude that

$$\|(X^k)^{-1}M^k - X^{-1}U\|_2 \leq \frac{\delta}{k^4} + \frac{4\delta}{k} \leq \frac{5\delta}{k}.$$

Combining this with (5.4.1) and (5.4.2) yields

$$\|v^k - x\|_2 \leq \frac{5\delta}{k}\|y\|_2 + \|(X^k)^{-1}M^k\|_2\|y^k - y\|_2 \leq 5\delta + \delta = 2^{-n}$$

by the definition of δ.

5.5 Proof of Theorems 4.1.3, 4.2.1, 4.2.13

Throughout this section we assume $\varepsilon = (\epsilon_1, \epsilon_2)$ with both $\epsilon_1, \epsilon_2 \in [0, 1)$ and at most one of $\epsilon_1, \epsilon_2 \neq 0$. Our aim will be to produce problems that obey the prerequisites of proposition 2.9.1. We will make use of the matrix A^1_ϵ defined in the following way: $A^1_\epsilon := \begin{pmatrix} 1-\epsilon_1 & 1-\epsilon_2 \end{pmatrix}$. A simple calculation shows that $A^1_\epsilon(A^1_\epsilon)^* = (1-\epsilon_1)^2 + (1-\epsilon_2)^2$. Thus $\operatorname{cond}(A^1_\epsilon(A^1_\epsilon)^*) = 1$ as $A^1_\epsilon(A^1_\epsilon)^*$ is a 1×1 matrix.

We also set $y^1 = 1$. Our first aim will be to prove a result on the solutions to linear programming and basis pursuit (denoising) with this input. Actually, we prove a slightly more general result so that we can make statements about the condition of a map condition number.

Lemma 5.5.1. *Fix $\delta \in [0, 1)$ and $y > \delta$. Set $U = \begin{pmatrix} \alpha & \beta \end{pmatrix}$. Let $\Xi_{\mathrm{LP}}, \Xi_{\mathrm{BP}}, \Xi_{\mathrm{BPDN}}$ denote the problem functions for linear programming, basis pursuit and basis pursuit denoising respectively. We have*

$$\Xi_{\mathrm{BP}}(y, U) = \begin{cases} \frac{y}{\alpha}e_1 & \text{if } \alpha > \beta \\ \frac{y}{\beta}e_2 & \text{if } \beta > \alpha \end{cases}, \quad \Xi_{\mathrm{BPDN}}(y, U) = \begin{cases} \frac{y-\delta}{\alpha}e_1 & \text{if } \alpha > \beta \\ \frac{y-\delta}{\beta}e_2 & \text{if } \alpha < \beta \end{cases}$$
$$\Xi_{\mathrm{LP}}(y, U) = \begin{cases} \frac{y}{\alpha}e_1 & \text{if } \alpha > \beta \\ \frac{y}{\beta}e_2 & \text{if } \beta > \alpha \end{cases} \tag{5.5.1}$$

Moreover, for $\alpha = \beta$ we have $\Xi_{\mathrm{LP}}(y, U) = \Xi_{\mathrm{BP}}(y, U)$ and

$$\Xi_{\mathrm{BP}}(y, U) = \operatorname{Conv}\left(\frac{ye_1}{\alpha}, \frac{ye_2}{\alpha}\right)$$
$$\Xi_{\mathrm{BPDN}}(y, U) = \operatorname{Conv}\left(\frac{(y-\delta)e_1}{\alpha}, \frac{(y-\delta)e_2}{\alpha}\right)$$

Proof of Lemma 5.5.1. We prove the results in (5.5.1) only for $\alpha > \beta$: the argument for $\alpha < \beta$ is identical.

We start with basis pursuit denoising. Assume that $\alpha > \beta$. Since $U(y-\delta)/\alpha e_1 = y - \delta$ and $\|(y-\delta)/\alpha e_1\|_1 = (y-\delta)/\alpha$, if $x \in \Xi_{\mathrm{BPDN}}(y, U)$ then $\|x\|_1 \leq (y-\delta)/\alpha$. Thus for such an x we have

$$-\delta \leq \alpha x_1 + \beta x_2 - y \leq \alpha|x_1| + \beta|x_2| - y \leq \alpha\|x\|_1 - y \leq -\delta \tag{5.5.2}$$

Notice that the second inequality is an equality if and only if x_1 and x_2 are both positive, the third inequality is an equality if and only if $|x_2| = 0$ and the final inequality is an equality if and only if $\|x\|_1 = (y-\delta)/\alpha$. All inequalities in (5.5.2) must however

be equalities since otherwise (5.5.2) implies that $-\delta < -\delta$, which is a contradiction. Thus $x \in \Xi_{\mathrm{BPDN}}(y, U)$ if and only if $x_2 = 0$ and $x_1 \geq 0$ and $\|x\|_1 = (y - \delta)/\alpha$. Such an x is clearly of the form $x = (y - \delta)e_1/\alpha$.

The argument for $\alpha = \beta$ is similar. Let us assume that $\tilde{x}$ is a minimiser. For any $x \in \mathrm{Conv}\left((y - \delta)e_1/\alpha, (y - \delta)e_2/\alpha\right)$ we have $\|Ux - y\|_2 = \delta$ and $\|x\|_1 = (y - \delta)/\alpha$. Hence $\|\tilde{x}\|_1 \leq (y - \delta)/\alpha$ and $\|U\tilde{x} - y\|_2 \leq \delta$. Thus (in the same way as before)

$$-\delta \leq \alpha\tilde{x}_1 + \alpha\tilde{x}_2 - y \leq \alpha|\tilde{x}_1| + \beta|\tilde{x}_2| - y = \alpha\|\tilde{x}\|_1 - y \leq -\delta$$

where now equality holds if and only if $\tilde{x}_1, \tilde{x}_2 \geq 0$ and $\|\tilde{x}\|_1 = (y - \delta)/\alpha$. However, such a $\tilde{x}$ would be an element of $\mathrm{Conv}\left((y - \delta)e_1/\alpha, (y - \delta)e_2/\alpha\right)$. Hence

$$\Xi_{\mathrm{BPDN}}(y, A) = \mathrm{Conv}\left(\frac{(y - \delta)e_1}{\alpha}, \frac{(y - \delta)e_2}{\alpha}\right)$$

which completes the proof of Lemma 5.5.1 for basis pursuit denoising.

Next, note that the result for basis pursuit follows immediately from the basis pursuit denoising case when $\delta = 0$. The only result that remains is the one for Ξ_{LP}: we will show the inclusion $\Xi_{\mathrm{BP}}(y, U) \subseteq \Xi_{\mathrm{LP}}(y, U)$ and the inclusion $\Xi_{\mathrm{LP}}(y, U) \subseteq \Xi_{\mathrm{BP}}(y, U)$ which will imply $\Xi_{\mathrm{LP}}(y, U) = \Xi_{\mathrm{BP}}(y, U)$.

Firstly, for $\Xi_{\mathrm{BP}}(y, U) \subseteq \Xi_{\mathrm{LP}}(y, U)$ note that we have already shown that every $\hat{x}$ in $\Xi_{\mathrm{BP}}(y, U)$ has $\hat{x}_1, \hat{x}_2 \geq 0$ and $U\hat{x} = y$. Thus $\hat{x}$ is feasible for the linear programming problem. Let us suppose that $\hat{x} \notin \Xi_{\mathrm{LP}}(y, U)$. Then there exists x with $x_1, x_2 \geq 0$ such that $Ux = y$ and $c^T x < c^T \hat{x}$. However, $c^T x = x_1 + x_2 = |x_1| + |x_2| = \|x\|_1$ and similarly $c^T \hat{x} = \|\hat{x}\|_1$. Thus x is a vector with $\|x\|_1 < \|\hat{x}\|_1$ and $Ux = y$. We conclude that $\hat{x} \notin \Xi_{\mathrm{BP}}(y, U)$, which is a contradiction.

Finally, to show that $\Xi_{\mathrm{LP}}(y, U) \subseteq \Xi_{\mathrm{BP}}(y, U)$, let

$$\Xi^{\mathrm{Obj}}_{\mathrm{LP}}(y, U) = \min \|x\|_1 \text{ such that } Ux = y \text{ and } x_1, x_2 \geq 0$$
$$\Xi^{\mathrm{Obj}}_{\mathrm{BP}}(y, U) = \min \|x\|_1 \text{ such that } Ux = y$$

noting that $\Xi^{\mathrm{Obj}}_{\mathrm{LP}}(y, U)$ is the same value as the objective function for linear programming since $c = (1, 1)^T$ and $x_1, x_2 \geq 0$ for linear programming feasible x. :et $\hat{x} \in \Xi_{\mathrm{LP}}(y, U)$. Then $\|\hat{x}\|_1 = \Xi^{\mathrm{Obj}}_{\mathrm{LP}}(y, U)$ and $Ux = y$. Thus $\hat{x}$ is feasible for $\Xi_{\mathrm{BP}}(y, U)$. If we assume that $\hat{x} \notin \Xi_{\mathrm{BP}}(y, U)$ then $\|\hat{x}\|_1 > \Xi^{\mathrm{Obj}}_{\mathrm{BP}}(y, U)$. Thus $\Xi^{\mathrm{Obj}}_{\mathrm{LP}}(y, U) > \Xi^{\mathrm{Obj}}_{\mathrm{BP}}(y, U)$, which is a clear contradiction since the set of feasible points for linear programming is a subset of the set of feasible points for basis pursuit. □

With Lemma 5.5.1 in hand, we can prove Theorem 4.1.3, Theorem 4.2.1 and Theorem 4.2.13. We start with Theorem 4.1.3

Proof of Theorem 4.1.3. To prove the non-computability result we appeal to Theorem

2.9.1. We set

$$\begin{aligned}\iota_n^1 &= (y^1, A_\epsilon^1) \quad \text{with } \varepsilon = (4^{-n}, 0)\\ \iota_n^2 &= (y^1, A_\epsilon^1) \quad \text{with } \varepsilon = (0, 4^{-n})\\ \iota^0 &= (y^1, A_\epsilon^1) \quad \text{with } \varepsilon = (0, 0).\end{aligned}$$

By Lemma 5.5.1, we have $\Xi(\iota_n^1) = \{e_2\}$ and $\Xi(\iota_n^2) = \{e_1\}$. Thus requirement (a) in Proposition 2.9.1 is satisfied with $S^1 = \{e_2\}, S^2 = \{e_1\}$ and $\kappa = \sqrt{2}$. It is also obvious (since $A_\epsilon^1 \to A_{0,0}^1$ as $\epsilon \to (0,0)$) that (b) is satisfied with $c_f = f(\iota^0)$. We immediately conclude by Proposition 2.9.1, part (iii) that $\epsilon_{\mathrm{B}}^{\mathrm{s}} \geq \sqrt{2}/2 = 1/\sqrt{2}$ and that $\{\Xi, \Omega\}^{\Delta_1} \notin \Delta_1^G$.

Next, we prove that $\{\Xi, \Omega\}^{\Delta_1} \in \Delta_1^A$ implies the decidability of LLPO. That $\mathcal{M} = \mathbb{R}^N$, Λ is finite and S^1, S^2 are singletons is clear by the definition and the previous argument. It is also clear that for each ι_n^1, ι_k^2, there exists an $f \in \Lambda$ such that $f(\iota_n^1) \neq f(\iota_k^2)$. Indeed, take $f : \iota \to \mathbb{R}$ to be the function such that $f(y, U) = U_{1,1}$. Then $f(\iota_n^1) = 1 - 4^{-n} \neq 1 = f(\iota_k^2)$. The result then follows by Proposition 2.9.1, part (iv).

Finally, we show the results on the condition of elements $\iota \in \Omega$. Since $\mathrm{cond}(A_\epsilon^1 (A_\epsilon^1)^*) = 1$ we immediately obtain that $\mathrm{cond}(A_\epsilon^1 (A_\epsilon^1)^*) \leq 2$.

The result $\|\iota\| \leq \sqrt{2}$ follows from the fact that

$$\|A_\epsilon^1\|_2 = \sqrt{\|A_\epsilon^1 (A_\epsilon^1)^*\|_2} \leq \sqrt{\mathrm{cond}(A_\epsilon^1 (A_\epsilon^1)^*)} \leq \sqrt{2}$$

and that $\|y^1\|_2 \leq 1$.

To show the bound on C_{FP}, note that if $\iota = (y, A) \in \Omega$ and $\hat{A}, \hat{y}$ are such that $(y + \hat{y}, A + \hat{A}) \in \tilde{\Omega}_\infty$ then y is non-negative provided that $\|\hat{y}\|_2 \leq 1$. Moreover, A has at least one entry that is exactly 1. Thus if $\|\hat{A}\|_2 < 1$, $A + \hat{A}$ has a positive entry. Without loss of generality, let us suppose that the second entry of $A + \hat{A}$ is positive. Then $(y + \hat{y})/(A_2 + \hat{A}_2)$ is a feasible point for the linear programming problem. Thus $\rho(y, A) \geq 1$ and so $C_{\mathrm{FP}}(\iota) \leq 2$.

The one remaining condition number to check is $\mathrm{cond}(\Xi)$. Fix $\nu > 0$ and let us assume that $\iota = \iota_n^1 = (y, A)$ for some n. Suppose that $\hat{\iota} = (\hat{y}, \hat{A})$ is such that $\iota + \hat{\iota} \in \tilde{\Omega}_\nu$. Since changes in y are not in the active set $\mathcal{A}(\Omega)$, $\hat{y} = 0$. In addition, since $\iota = \iota_n^1$, we have $A_{1,1} < 1 = A_{1,2}$. Thus for sufficiently small $\epsilon > 0$ if $\|\hat{A}\|_2 \leq \epsilon$ then $(A + \hat{A})_{1,1} < (A + \hat{A})_{1,2}$. For such $\hat{\iota}$, we have by Lemma 5.5.1 that $\Xi_{\mathrm{LP}}(\iota + \hat{\iota}) = (1 + \hat{A}_{1,2})^{-1} e_2$ and $\Xi_{\mathrm{BP}}(\iota) = e_2$.

Thus

$$\mathrm{dist}(\Xi_{\mathrm{LP}}(\iota + \hat{\iota}), \Xi_{\mathrm{LP}}(\iota)) = \left| 1 - \frac{1}{1 + \hat{A}_{1,2}} \right| \leq \left| \frac{\hat{A}_{1,2}}{1 + \hat{A}_{1,2}} \right| \leq \frac{\epsilon}{1 - \epsilon}$$

and we conclude that for such ι,

$$\lim_{\epsilon \to 0^+} \sup_{\iota + \hat{\iota} \in \tilde{\Omega}_\nu, 0 < \|\hat{\iota}\| \leq \epsilon} \frac{\mathrm{dist}(\Xi_{\mathrm{LP}}(\iota + \hat{\iota}), \Xi_{\mathrm{LP}}(\iota))}{\|\hat{\iota}\|} \leq 1$$

and the same argument shows that the same result if $\iota = \iota_n^2$ for some n.

The last case to consider is $\iota = \iota^0$. Again, suppose that $\hat{\iota} = (\hat{y}, \hat{A})$ is such that $\iota + \hat{\iota} \in \tilde{\Omega}_\nu$. Since changes in y are not in the active set $\mathcal{A}(\Omega)$, $\hat{y} = 0$. We consider two cases: firstly, $\hat{A}_{1,1} > \hat{A}_{1,2}$ and secondly $\hat{A}_{1,2} \geq \hat{A}_{1,1}$.

If $\hat{A}_{1,1} > \hat{A}_{1,2} > -1$ then $(A+\hat{A})_{1,1} > (A+\hat{A})_{1,2} > 0$ and thus Lemma 5.5.1 shows that $(1+\hat{A}_{1,1})^{-1}e_1 \in \Xi_{\mathrm{LP}}(\iota+\hat{\iota})$. Again by Lemma 5.5.1, $e_1 \in \Xi_{\mathrm{LP}}(\iota)$. Thus (provided $\|\hat{A}\|_2 < 1$)

$$\mathrm{dist}(\Xi_{\mathrm{LP}}(\iota+\hat{\iota}), \Xi_{\mathrm{LP}}(\iota)) = \left|1 - \frac{1}{1+\hat{A}_{1,1}}\right| \leq \left|\frac{\hat{A}_{1,1}}{1+\hat{A}_{1,1}}\right| \leq \frac{\epsilon}{1-\epsilon}.$$

Similarly, if $\hat{A}_{1,2} \geq \hat{A}_{1,2} > -1$ then Lemma 5.5.1 shows that $(1+\hat{A}_{1,2})^{-1}e_2 \in \Xi_{\mathrm{LP}}(\iota+\hat{\iota})$. Again by Lemma 5.5.1, $e_2 \in \Xi_{\mathrm{LP}}(\iota)$. Thus

$$\mathrm{dist}(\Xi_{\mathrm{LP}}(\iota+\hat{\iota}), \Xi_{\mathrm{LP}}(\iota)) = \left|1 - \frac{1}{1+\hat{A}_{1,2}}\right| \leq \left|\frac{\hat{A}_{1,2}}{1+\hat{A}_{1,1}}\right| \leq \frac{\epsilon}{1-\epsilon}.$$

Thus

$$\lim_{\epsilon\to 0^+} \sup_{\iota+\hat{\iota}\in\tilde{\Omega}_\nu, 0<\|\hat{\iota}\|\leq\epsilon} \frac{\mathrm{dist}(\Xi_{\mathrm{LP}}(\iota+\hat{\iota}), \Xi_{\mathrm{LP}}(\iota))}{\|\hat{\iota}\|} \leq 1.$$

We conclude that $\mathrm{Cond}(\Xi) \leq 1$. This completes the proof of the non-computability result.

All that remains is the computability result for $\Xi_{\mathrm{LP}}^{\mathrm{Obj}}$. However, for each $\iota \in \Omega$, we can use Lemma 5.5.1 to see that if $x \in \Xi_{\mathrm{LP}}(\iota)$, then $\|x\|_1 = c^T x = 1$. Thus for every $\iota \in \Omega$, $\Xi_{\mathrm{LP}}^{\mathrm{Obj}}(\iota) = 1$. It will therefore suffice to create the arithmetic algorithm $\Gamma : \Omega \to \mathbb{R}$ defined by $\Gamma(\iota) \equiv 1$, which has $\Gamma(\iota) = \Xi_{\mathrm{LP}}^{\mathrm{Obj}}(\iota)$. □

Next, we prove Theorem 4.2.1.

Proof of Theorem 4.2.1. **Part (i)**: The proof of part (i) is very similar to the proof of Theorem 4.1.1. Indeed, we use the same choice of U^ϵ and y^ϵ as in that argument. In a similar way, we set the sequence $\iota_n^1 := (y^{1/4^n}, U^{1/4^n})$, the constant sequence $\iota_n^2 := (y^0, U^0)$ and $\iota_0 := (y^0, U^0)$. The same argument as before shows that $\Xi(\iota_n^1) = \{(1 \oplus 0_{N-1})^T\}$. To see that $\Xi(\iota_0) = 0_N$, note that $U^0 0_N = 0_m = y^0$. Moreover, every non-zero $x \in \mathbb{R}^N$ has $\|x\|_1 > 0_N$. Thus $\Xi(\iota_0) = \{0_N\}$. The remainder of the argument is identical to the one in Theorem 4.1.1.

Part (ii): After replacing Ξ_{LP} with Ξ_{BP}, the proof reads identically to the one for Theorem 4.1.3. □

We use a similar approach (setting up an example problem and employing Proposition 2.9.1) as before to prove Theorem 4.2.13:

Proof of Theorem 4.2.13. **Part (i)**: The proof for part (i) differs from the proof of part (i) in Theorem 4.1.1. We choose U^ϵ in the same way as before but now we choose $y^{\epsilon,\delta} = (y^{\epsilon,\delta})^T = (\epsilon+\delta) \oplus 0_{m-1}$. It is easy to see that if $\|U^\epsilon x - y^{\epsilon,\delta}\|_2 \leq \delta$ then we require $x_1 \geq 1$ whenever $\epsilon > 0$. Moreover, $\|U^\epsilon e_1 - y^{\epsilon,\delta}\| = \delta$ and thus all minimisers

x must have $\|x\|_1 \leq 1$, $x_1 \geq 1$. The only such vector is e_1, thus $\Xi(y^{\epsilon,\delta}, U^{\epsilon,\delta}) = e_1$ if $\epsilon > 0$.

If instead $\epsilon = 0$ we note that $\|U^{\epsilon}0 - y^{\epsilon,\delta}\| = \delta$ and so 0 is feasible for the basis pursuit problem. But every non-zero vector has an ℓ^1 norm strictly larger than 0, so 0 is the unique minimiser. We can now apply proposition 2.9.1 with $S^1 = e_1$, $S^2 = 0$ as in the proof of Theorem 4.1.1.

Part (ii): Up until the discussion on the condition of a map, the proof for part (ii) is identical to that of Theorem 4.1.3 with two exceptions: firstly, we now have $\Xi_{\mathrm{BPDN}}(\iota_n^1) = \{(1-\delta)e_2\} = S^1, \Xi_{\mathrm{BPDN}}(\iota_n^2) = \{(1-\delta)e_1\} = S^2$ by using Lemma 5.5.1. The second change comes from this new definition of S^1, S^2: we now have that $\kappa = (1-\delta)\sqrt{2}$. Aside from these minor changes the proof (up to the statements on the condition of a map) follows verbatim as in Theorem 4.1.3 part (ii).

For the statement on the condition of a map, the proof is almost the same, with one notable difference. Following the argument in the same way as before, we now see that for any $\iota \in \Omega$

$$\lim_{\epsilon \to 0^+} \sup_{\iota + \hat{\iota} \in \tilde{\Omega}_\nu, 0 < \|\hat{\iota}\| \leq \epsilon} \frac{\mathrm{dist}(\Xi_{\mathrm{LP}}(\iota + \hat{\iota}), \Xi_{\mathrm{LP}}(\iota))}{\|\hat{\iota}\|} \leq 1 - \delta$$

We conclude that $\mathrm{Cond}(\Xi) \leq 1 - \delta$ (the previous result was the same except that the value $1 - \delta$ is instead replaced by the value 1). □

5.6 Proof of Theorem 4.2.7 and Theorem 4.2.16

It is easy to see that Theorem 4.2.7 is a subcase of Theorem 4.2.16. We thus start with Theorem 4.2.16.

Proof of Theorem 4.2.16. Here, as for the other positive results, we will construct an algorithm that solves the problem and then prove its correctness. We assume the existence of recursive subroutines $\mathsf{Sqrt}(x, \epsilon)$, $\mathsf{FindV}(\iota, \epsilon)$, $\mathsf{FindM}(\iota, \epsilon)$ and $\mathsf{PCon}(y, U, \epsilon, k)$ that work in the following way:

- $\mathsf{Sqrt}(x, \epsilon)$ defined for $x \geq 0$ and $\epsilon > 0$ returns an approximation $\tilde{x}$ to $\sqrt{x}$ such that $|(\tilde{x} - \sqrt{x})|/\sqrt{x} \leq \epsilon$. This can be done using the Babylonian algorithm.
- $\mathsf{FindV}(\iota, \epsilon)$ defined for $\iota = (y, U)$ and $\epsilon > 0$ finds a vector v such that $\|v - y\| \leq \epsilon$. This can be achieved by making use of the Δ_1 information available to the algorithm.
- $\mathsf{FindM}(\iota, \epsilon)$ defined for $\iota = (y, U)$ and $\epsilon > 0$ finds a matrix M such that $\|M - U\| \leq \epsilon$. Again, this can be achieved through the use of the Δ_1 information available to the algorithm.
- $\mathsf{PCon}(y, U, \epsilon, k)$ for $y \in \mathbb{R}^m$, $U \in \mathbb{R}^{m \times N}$, $\epsilon > 0$ and $k \in \mathbb{N}$ returns the nth iteration of some preselected primal convergent basis pursuit denoising algorithm with measurement vector y, matrix U and denoising parameter ϵ.

We now describe an algorithm for basis pursuit denoising:
Algorithm $\mathsf{BPursuitDN}(\iota,n,\varphi)$

Input: $\iota \in \Omega$, $n \in \mathbb{N}$, $\varphi = (\rho_1, \tau_1, l_1, \eta_1) \in \mathbb{R}^4$ with $l_1 = r_l^4, \eta_1 = r_\eta^4, \rho_1 = r_\rho^2$ for some a-priori known constants $r_l, r_\eta, r_\rho \in \mathbb{Q}$.

Output: $\Gamma_n(\iota) \in \mathbb{R}^N$ with $\mathrm{dist}(\Gamma_n(\iota), \Xi(\iota)) \leq 2^{-n}$.

Subroutines: $\mathsf{Sqrt}(\iota,\epsilon), \mathsf{FindV}(\iota,\epsilon), \mathsf{FindM}(\iota,\epsilon), \mathsf{PCon}(y,U,\epsilon,k)$.

We set the parameters $\varepsilon := 2^{-n}$,

$$\begin{aligned} \alpha_2 &:= (\rho_1 + C_0(\rho_1))\sqrt[4]{l_1\eta_1}(1+\rho_1)/(1-\rho_1), \\ \alpha_3 &:= \tau_1(\sqrt[4]{l_1\eta_1}/2 + 1 + 2(\rho_1 + C_0(\rho_1)\sqrt[4]{l_1\eta_1}))/(1-\rho_1), \end{aligned} \tag{5.6.1}$$

$$\begin{aligned} &\epsilon_1 := \varepsilon/(8\alpha_2) \wedge 1, \quad \epsilon_4 := \varepsilon/(16\alpha_3) \wedge 1, \quad \epsilon_3 := \epsilon_4/3 \wedge \varepsilon/(8\alpha_3), \\ &b := \mathsf{FindV}(\iota,\epsilon_3), \quad \alpha_1 := \tau_1\left(\epsilon_3 + \|b\|_1 + \delta\right), \\ &\epsilon_2 := 1/(N\alpha_3) \wedge \varepsilon/(8\alpha_1\alpha_3) \wedge \epsilon_4/(3\alpha_1), \quad A := \mathsf{FindM}(\iota,\epsilon_2). \end{aligned} \tag{5.6.2}$$

$k := 0, m := 0, m := \infty, \tilde{G}_0 = \infty$.
Then we proceed in the following way

if $\|b\|_2^2 \leq (\delta + \epsilon_4)^2$, set $\Gamma_n(\iota) = 0$.

while $\tilde{G}_k > \epsilon_1/2$ or $\epsilon_4^k \geq (2\epsilon_4 + \delta)^2$

$k = k + 1$

$v^k = \mathsf{PCon}(A, b, \epsilon_4 + \delta, k)$

$e^k = Av^k - b$, $\epsilon_4^k = \|e^k\|_2^2$

if $A^*e^k \neq 0$ **and** $\epsilon_4^k \leq (2\epsilon_4 + \delta)^2$ **then**

$w_k = \mathsf{Sqrt}(\sum_{i=1}^m |e_i^k|^2, \frac{\epsilon_1 \|A^*e^k\|_\infty}{3(2\epsilon_4+\delta)\delta})$

$m = m \wedge \|A^*e^k\|_\infty^{-1} \left[\langle b, e_k\rangle + w_k(\delta + \epsilon_4)\right], \tilde{G}_k = \|v^k\|_1 + m$

else

$m = m \wedge 0$, $\tilde{G}_k = \|v^k\|_1 + m$

endif

end

$\Gamma_n(I) := v^k$

To show that $\mathsf{BPursuitDN}$ will indeed give us the required arithmetic tower Γ_n, we must show that (both in the case that $\|b\|_2 \leq \delta + \epsilon_4$ and $\|b\|_2 > \delta + \epsilon_4$)

1. The algorithm uses only arithmetic operations.
2. The algorithm uses finitely many operations for each $\iota \in \Omega$.

3. $\operatorname{dist}(\Gamma_n(\iota), \Xi_{\mathrm{BPDN}}(\iota)) \leq 2^{-n} + 2N\delta\alpha_3/(N-1) + 2\delta\left(C + D\sqrt[4]{l_1\eta_1}\right)$ for any $\iota \in \Omega$.

To address this third point, note that by the definition of Ω, if $\iota = (y, U) \in \Omega$ then there exists an $(\mathbf{s}, \mathbf{M})$-sparse x such that $\|Ux - y\|_2 \leq \delta$. By Theorem 3.3.8 it follows that for any $v \in \Xi_{\mathrm{BPDN}}(\iota)$, $\|v - x\|_2 \leq 2\delta\left(C + D\sqrt[4]{l_1\eta_1}\right)$ where $C := \frac{\rho_1\tau_1+\tau_1}{1-\rho_1}$ and $D := \frac{4\sqrt{\rho_1}\tau_1+3\tau_1-\rho_1\tau_1}{2-2\rho_1}$. Therefore, instead of proving the third point, it will suffice to show that

4. $\|\Gamma_n(\iota) - x\| \leq 2^{-n} + 2N\delta\alpha_3/(N-1)$.

We now consider two cases. Firstly, we will show 1, 2 and 4 for the case $\|b\|_2 \leq \delta + \epsilon_4$. Next, we shall show 1, 2 and 4 if $\|b\|_2 > \delta + \epsilon_4$.

If $\|b\|_2 \leq \delta + \epsilon_4$ then it is obvious that the algorithm uses finitely many arithmetic operations. Indeed, $\|b\|_2^2$ can be calculated using finitely many operations and the algorithm simply outputs 0 at that point. By the definition of b and FindV, we have $\|y - b\|_2 \leq \epsilon_3$. Thus 4 follows because

$$\begin{aligned}\|x\|_2 \leq \tau\|Ux\|_2 &\leq \tau_1\left(\|Ux - y\|_2 + \|y - b\|_2 + \|b\|_2\right)\\ &\leq \tau_1\left(2\delta + \epsilon_3 + \epsilon_4\right) \leq 2\delta\alpha_3 + \alpha_3(\epsilon_3 + \epsilon_4) \leq 2\delta\alpha_3 + 2^{-n}\end{aligned}$$

which is itself a consequence of the fact that $\epsilon_3 + \epsilon_4 \leq 3\epsilon/16\alpha_3$ and $\alpha_3 \geq \tau_1$.

We now consider the more complicated case, where $\|b\| > \delta + \epsilon_4$. The construction of the values $\alpha_2, \alpha_3, \epsilon_1, \epsilon_4, \epsilon_3, \alpha_1$ and ϵ_2 in the algorithm is clearly done using arithmetic operations, given that the ℓ^1 norm can be calculated using arithmetic operations and each of $C_0(\rho_1), \tau_1$ and $\sqrt[4]{(l_1\eta_1)}$ are known a-priori to the algorithm. The only stage of the algorithm itself that may be ill-defined (noting that the square of the ℓ^2 norm can also be calculated with arithmetic operations) is the line where v^k is set to be $\mathsf{PCon}(A, b, \epsilon_4 + \delta, k)$. We must show here that there is an x^0 with $\|Ax^0 - b\| < \delta + \epsilon_4$ so that we can apply the definition of a primal convergent algorithm.

Since $\iota = (y, U) \in \Omega$, we must have $\|y - Ux\|_2 \leq \delta$ for some $(\mathbf{s}, \mathbf{M})$-sparse x. We claim that $\|Ax - b\| < \delta + \epsilon_4$. To see this, note firstly that from the nullspace property on U and the fact that x is $(\mathbf{s}, \mathbf{M})$-sparse, we have

$$\|x\|_2 \leq \tau_1\|Ux\|_2 \leq \tau_1(\|y\|_2 + \delta) \leq \tau_1\left(\epsilon_3 + \|b\|_2 + \delta\right) = \alpha_1. \tag{5.6.3}$$

Furthermore, by the definition of the constants above, $\epsilon_2\alpha_1 + \epsilon_3 < \epsilon_4$. Therefore $\|Ax - b\| \leq \|(A - U)x\| + \|Ux - y\| + \|y - b\| \leq \epsilon_2\|x\|_2 + \delta + \epsilon_3 < \epsilon_4 + \delta$.

Next, we need to show that the algorithm uses only finitely many operations. This will be obvious if we can show that $\tilde{G}_k > \epsilon_1/2$. To do so, we show that $|\tilde{G}_k - G_k| \leq \epsilon_1/3$ where

$$\begin{aligned}G_k &= \|v^k\|_1 + \min_{m \leq k}\left(\langle b, p^m\rangle + (\delta + \epsilon_4)\|p^m\|_2\right)\\ p^m &= \begin{cases}\frac{e^m}{\|A^*e^m\|_\infty} & \text{if } A^*e^m \neq 0 \text{ or } \|e^m\|_2 > 2\epsilon_4 + \delta\\ 0 & \text{otherwise.}\end{cases}\end{aligned}$$

Then we can apply Proposition A.1.1 (with $\epsilon = \epsilon_4$) and so $G_k \to 0$. Therefore eventually $G_k < \epsilon_1/6$ so that $\tilde{G}_k \leq \epsilon_1/6 + \epsilon_1/3 \leq \epsilon_1/2$.

If $A^*e^k = 0$ or $\epsilon_4^k > (2\epsilon_4 + \delta)^2$ then it is obvious that $\langle b, p^m \rangle + (\delta + \epsilon_4)\|p^m\|_2 = 0$ by the definition of p^m. We claim that $\langle b, p^m \rangle + (\delta + \epsilon_4)\|p^m\|_2$ is within $\epsilon_1/3$ of $(\|A^*e^k\|_\infty)^{-1}\left[\langle b, e^k \rangle + w^k(\delta + \epsilon_4)\right]$ whenever $A^*e_k \neq 0$ and $\epsilon_4^k > (2\epsilon_4 + \delta)^2$. The result follows from the fact that if the real numbers $\{a_i\}_{i=1}^k$ and $\{\tilde{a}_i\}_{i=1}^k$ satisfy $|\tilde{a}_i - a_i| \leq \epsilon_1/3$ then $\min_{i \leq k} \tilde{a}_i$ is within $\epsilon_1/3$ of $\min_{i \leq k} a_i$.

By the definition of Sqrt, $|w_k - \|e^k\|_2|/\|e^k\|_2 \leq \epsilon_1\|A^*e^k\|_\infty/3(2\epsilon_4 + \delta)\delta$. Therefore, for each k with $\|e^k\|_2^2 = \epsilon_4^k \leq (2\epsilon_4 + \delta)^2$ we have

$$\begin{aligned}\left|\|A^*e^k\|_\infty^{-1}(\langle b, e^k \rangle + w_k\delta) - \|A^*e^k\|_\infty^{-1}\left(\left\langle b, e^k \right\rangle + \delta\|e^k\|_2\right)\right| &= \delta\|A^*e^k\|_\infty^{-1}|(w_k - \|e^k\|_2)| \\ &\leq \frac{\delta\|A^*e^k\|_\infty^{-1}\epsilon_1\|A^*e^k\|_\infty\|e^k\|_2}{3(2\epsilon_4 + \delta)\delta} \leq \epsilon_1/3.\end{aligned}$$

Therefore $|\widetilde{G}_k - G_k| \leq \epsilon_1/3$ and hence the while loop terminates in finite time.

All that remains is to show that $\|\Gamma_n(\iota) - x\| \leq 2^{-n} + 2N\delta\alpha_3/(N-1)$. Note that it suffices to show that

$$\|\Gamma_n(\iota) - x\|_2 \leq \frac{\epsilon_1\alpha_2 + 2\epsilon_4\alpha_3 + 2\delta\alpha_3 + \epsilon_3\alpha_3 + \epsilon_2\alpha_1\alpha_3}{1 - \epsilon_2\alpha_3}. \tag{5.6.4}$$

Indeed, the definitions in (5.6.2) in BPursuitDN imply the following conditions for $\epsilon_1, \epsilon_2, \epsilon_3$ and ϵ_4: $\epsilon_1\alpha_2 \leq \frac{2^{-n}}{8}$, $\epsilon_2\alpha_3 \leq \frac{1}{N}$, $\epsilon_2\alpha_1\alpha_3 \leq \frac{2^{-n}}{8}$, $\epsilon_2\alpha_1 < \frac{\epsilon_4}{2}$, $\epsilon_3 < \frac{\epsilon_4}{2}$, $\epsilon_3\alpha_3 \leq \frac{2^{-n}}{8}$, $2\epsilon_4\alpha_3 \leq \frac{2^{-n}}{8}$. Therefore, by plugging these values into the right hand side of (5.6.4) we get

$$\begin{aligned}\|\Gamma_n(\iota) - x\|_2 &\leq \frac{N-1}{N}(\epsilon_1\alpha_2 + 2\epsilon_4\alpha_3 + 2\delta\alpha_3 + \epsilon_3\alpha_3 + \epsilon_2\alpha_1\alpha_3) \\ &\leq \frac{N-1}{N}(2^{-n}/8 + 2^{-n}/8 + 2\delta\alpha_3 + 2^{-n}/8 + 2^{-n}/8) \\ &= \frac{N-1}{N}\left(2^{-n-1} + 2\delta\alpha_3\right) \leq 2^{-n} + 2\delta\alpha_3(N-1)/N.\end{aligned}$$

For any $z \in \mathbb{R}^n$, by Lemma 3.4.3 the nullspace property of order $(\mathbf{s}, \mathbf{M})$ with parameters $\rho < 1$ and $\tau > 0$ implies the following result: $\|z - x\|_1 \leq \frac{1+\rho}{1-\rho}(\|z\|_1 - \|x\|_1) + \frac{2\tau\sqrt{\tilde{s}}}{1-\rho}\|U(z-x)\|_2$. Moreover, by Lemma 3.4.1 we have that

$$\|z - x\|_2 \leq \frac{\|z - x\|_1}{\sqrt{\tilde{s}}}\left(\rho + C_0(\rho)\sqrt[4]{l\eta_{\mathbf{s},\mathbf{M}}}\right) + \tau\|U(z-x)\|_2\left(\sqrt[4]{l\eta_{\mathbf{s},\mathbf{M}}}/2 + 1\right).$$

Combining these two results yields

$$\begin{aligned}\|z-x\|_2 &\leq \frac{\left[\rho + C_0(\rho)\sqrt[4]{l\eta_{\mathbf{s},\mathbf{M}}}\right]}{\sqrt{\tilde{s}}}\left(\frac{1+\rho}{1-\rho}(\|z\|_1 - \|x\|_1) + \frac{2\tau\sqrt{\tilde{s}}}{1-\rho}\|U(z-x)\|_2\right) \\ &\quad + \tau\|U(z-x)\|_2\left(\sqrt[4]{l\eta_{\mathbf{s},\mathbf{M}}}/2 + 1\right) \\ &\leq \alpha_2(\|z\|_1 - \|x\|_1) + \tau\|U(z-x)\|_2\left(\sqrt[4]{l\eta_{\mathbf{s},\mathbf{M}}}/2 + 1 + \frac{2\left(\rho + C_0(\rho)\sqrt[4]{l\eta_{\mathbf{s},\mathbf{M}}}\right)}{1-\rho}\right) \\ &\leq \alpha_2(\|z\|_1 - \|x\|_1) + \alpha_3\|U(z-x)\|_2 \qquad (5.6.5)\end{aligned}$$

where the final line follows because

$$\frac{\left[\rho + C_0(\rho)\sqrt[4]{l\eta_{\mathbf{s},\mathbf{M}}}\right]}{\sqrt{\tilde{s}}}\left[\frac{1+\rho}{1-\rho}\right] \text{ and } \tau\left[\frac{\sqrt[4]{l\eta_{\mathbf{s},\mathbf{M}}}}{2} + 1 + \frac{2\left(\rho + C_0(\rho)\sqrt[4]{l\eta_{\mathbf{s},\mathbf{M}}}\right)}{1-\rho}\right]$$

are increasing functions of ρ, l and $\eta_{\mathbf{s},\mathbf{M}}$ and $\rho < \rho_1$, $l < l_1$ and $\eta_{\mathbf{s},\mathbf{M}} < \eta_1$.

Note that $\Gamma_n(I) = v^k$ where v^k is the final output of the while loop in BPursuitDN and that $\|Av^k - b\|_2 \leq 2\epsilon_4 + \delta$ by the halting criteria in the while loop. For this v^k, we therefore have

$$\begin{aligned}\|U(v^k - x)\|_2 &= \|(Av^k - b) + (Uv^k - Av^k) + (b - y) + y - Ux\|_2 \\ &\leq 2\epsilon_4 + \delta + \|(A-U)v^k\|_2 + \epsilon_3 + \delta \\ &\leq 2\epsilon_4 + 2\delta + \|(A-U)(v^k - x)\|_2 + \|(A-U)x\|_2 + \epsilon_3 \\ &\leq 2\epsilon_4 + 2\delta + \epsilon_2\left(\|v^k - x\|_2 + \|x\|_2\right) + \epsilon_3.\end{aligned}$$

and since $\|x\|_2 \leq \alpha_1$, we obtain

$$\|U(v^k - x)\|_2 \leq 2\epsilon_4 + 2\delta + \epsilon_2\|v^k - x\|_2 + \epsilon_2\alpha_1 + \epsilon_3. \qquad (5.6.6)$$

Combining (5.6.5) with $z = v^k$ and (5.6.6), we obtain $\|v^k - x\|_2 \leq \alpha_2(\|v^k\|_1 - \|x\|_1) + \alpha_3\left(2\epsilon_4 + 2\delta + \epsilon_2\|v^k - x\|_2 + \epsilon_2\alpha_1 + \epsilon_3\right)$ and so by rearranging

$$\|v^k - x\|_2 \leq \frac{\alpha_2(\|v^k\|_1 - \|x\|_1) + 2\epsilon_4\alpha_3 + 2\delta\alpha_3 + \epsilon_3\alpha_3 + \epsilon_2\alpha_1\alpha_3}{1 - \epsilon_2\alpha_3}$$

At the conclusion of the while loop, $\tilde{G}_k \leq \epsilon_1/2$. In the proof that the algorithm terminates after finitely many operations, we showed that $|\tilde{G}_k - G_k| \leq \epsilon_1/3$. Thus when the algorithm has terminated $G_k < \epsilon_1$ and so by Proposition A.1.1 we have $\|v^k\|_1 - \|x\|_1 \leq \|v^k\|_1 - \mathrm{BP}_{\epsilon_4+\delta}(A,b) \leq G_k \leq \epsilon_1$. Equation 5.6.4 follows since the final step of the algorithm is to set $\Gamma_n(\iota) = v^k$. □

Finally, we discuss briefly Theorem 4.2.7.

Proof of Theorem 4.2.7. To see the result of Theorem 4.2.7, note that the set Ω defined in Theorem 4.2.16 is identical to the set Ω defined in Theorem 4.2.7. Furthermore,

Theorem 4.2.16 includes the case $\delta = 0$, from which the result of Theorem 4.2.7 follows immediately. □

5.7 Proof of Theorem 4.2.2

Let us begin by proving the following result that allows us to create examples of problems with infinite condition number.

Lemma 5.7.1. *Let $U \in \mathbb{R}^{m\times N}$, where we assume that $\tilde{\Omega}_\infty \supseteq \{(y', U) \,|\, y' \in \mathbb{R}^m\}$. Suppose that there is a non-empty set $S \subseteq \{1, 2, \ldots, N\}$ and a vector v in the row span of U such that U, S and v satisfy the following properties:*

1. *There is a non-zero vector ξ^1 such that $supp(\xi^1) \subseteq S$ and $U\xi^1 = 0$*
2. *There is a constant $c > 0$ such that $\max_{i\in S} |v_i| = \min_{i\in S} |v_i| = 1/c$ and $\|v_{S^c}\|_\infty \leq 1/c$.*

If $T \subseteq S$ is a set with the following property:

3. *If $U\rho = 0$ and $supp(\rho) \cap T \neq \emptyset$ for some vector $\rho \in \mathbb{C}^N$ then there is a non-zero ρ' with $supp(\rho) \cap T \subseteq supp(\rho') \subseteq S$ and $U\rho' = 0$.*

then there is a vector x with $supp(x) = T$ so that the distance to several minimisers $\varrho(Ux, U) = 0$ and $\|U\|_2 > 0$. Note that condition (3) is trivially satisfied if $S = \{1, 2, \ldots, N\}$ (with $\rho' = \rho$) or $T = \emptyset$ (where there is nothing to show).

Proof. To prove this lemma, we will construct a non-zero vector $\xi \in \mathbb{R}^N$ with $U\xi = 0, \mathrm{supp}(\xi) \subseteq S$ and ξ has maximal support in S, in the sense that

$$\text{If } \xi' \text{ with support in } S \text{ is in the nullspace of } U \text{ then } \mathrm{supp}(\xi') \subseteq \mathrm{supp}(\xi). \tag{5.7.1}$$

Once ξ is constructed, we will construct non-empty disjoint sets J_+ and J_- along with the vector x such that for any $\epsilon > 0$,

(I) $\mathrm{supp}(\xi) = J_+ \cup J_-$

(II) $U(x + \epsilon\xi_{J_+}) = U(x - \epsilon\xi_{J_-})$.

(III) $\|x + \epsilon\xi_{J_+}\|_1 = \|x - \epsilon\xi_{J_-}\|_1$.

(IV) If $\hat{\xi} \in \mathbb{R}^N$ is such that $U(\hat{\xi} + x + \epsilon\xi_{J_+}) = U(x + \epsilon\xi_{J_+})$ then $\|\hat{\xi} + x + \epsilon\xi_{J_+}\|_1 \geq \|x + \epsilon\xi_{J_+}\|_1$.

This will imply that $(x+\epsilon\xi_{J_+}), (x+\epsilon\xi_{J_-}) \in \mathrm{argmin}\, \|w\|_1$ such that $Uw = U(x+\epsilon\xi_{J_+})$ Indeed, from (IV), $x + \epsilon\xi_{J_+}$ is a minimiser. (II) implies that $x - \epsilon\xi_{J_-}$ is feasible, so we can use (III) to see that $x - \epsilon\xi_{J_-}$ is also a minimiser. (I) will be used to prove (III) and (IV). Consequently, $\varrho(U, Ux) = \sup\{\delta : \|\hat{U}\|, \|\hat{y}\| \leq \delta \Rightarrow (U + \hat{U}, Ux + \hat{y}) \text{ has exactly one solution}\} \leq \epsilon$. Since ϵ was arbitrary and independent of x, we must have $\varrho(U, Ux) = 0$ and the proof is completed by noticing that (2) implies that $\|U\|_2 > 0$.

Step 1: Construction of ξ, J_+, J_-, x and the verification of (I): To construct ξ, we inductively construct $(\xi^k)_{k=1}^{N+1}$ starting from ξ^1 as defined in the statement of the lemma (property (1)) with $U\xi^k = 0$ and $\text{supp}(\xi^k) \subseteq S$. We set $\xi = \xi^N$. ξ^k is constructed as follows: if ξ^k satisfies (5.7.1) then set $\xi^{k+1} = \xi^k$. Otherwise, let $\tilde{\xi}$ have support in S, $U\tilde{\xi} = 0$ and $\tilde{\xi}_i \neq 0, \xi_i^k = 0$ for some i (the existence of $\tilde{\xi}$ is exactly the definition of ξ^k not satisfying (5.7.1)). We set $m^k = \min\{|\xi_j^k| \text{ such that } |\xi_j^k| > 0\}$ and we define $\xi^{k+1} := \xi^k + \frac{m^k\tilde{\xi}}{2\|\tilde{\xi}\|_\infty}$. It is clear that $U\xi^{k+1} = 0$. We claim that $\text{supp}(\xi^k)$ and $\text{supp}(\tilde{\xi})$ are both subsets of $\text{supp}(\xi^{k+1})$, that $\text{supp}(\xi^{k+1}) \subseteq S$ and that in this case, $|\text{supp}(\xi^{k+1})| > |\text{supp}(\xi^k)|$. To see this, note that

$$|\xi_j^{k+1}| \geq |\xi_j^k| - \frac{|m^k||\tilde{\xi}_j|}{2\|\tilde{\xi}\|_\infty} \geq |\xi_j^k| - \frac{|m^k|}{2} \geq \frac{|\xi_j^k|}{2}, \quad j \in \text{supp}(\xi^k),$$
$$\xi_j^{k+1} = m^k\tilde{\xi}_j/(2\|\tilde{\xi}\|_\infty) \neq 0, \quad j \in \text{supp}(\tilde{\xi}) \setminus \text{supp}(\xi^k).$$

The fact that $\text{supp}(\xi^{k+1}) \subseteq S$ follows from the fact that ξ^{k+1} is a linear combination of the vectors ξ^k and $\tilde{\xi}$, both of which have support contained in S. Finally, since $\xi_i^k = 0$ and $\tilde{\xi}_i \neq 0$ and $\text{supp}(\xi^k), \text{supp}(\tilde{\xi}) \subseteq \text{supp}(\xi^{k+1})$, if ξ^k does not satisfy (5.7.1) then $|\text{supp}(\xi^{k+1})| > |\text{supp}(\xi^k)|$. If there is a k_1 such that ξ^{k_1} satisfies (5.7.1) then $\xi^{N+1} = \xi^{k_1}$ and so ξ^{N+1} satisfies (5.7.1). If instead, for each k, ξ^k does not satisfy (5.7.1) then for each k $|\text{supp}(\xi^k)| > |\text{supp}(\xi^{k-1})| > |\text{supp}(\xi^{k-2})| > \cdots > |\text{supp}(\xi^1)|$.| In particular, $|\text{supp}(\xi^{N+1})| > N$, contradicting the fact that $\xi^{N+1} \in \mathbb{R}^N$. Therefore ξ^{N+1} satisfies (5.7.1).

Next, we define the sets J_+ and J_- by

$$J_+ = \{i \in S \,|\, v_i\xi_i > 0\}, \qquad J_- = \{i \in S \,|\, v_i\xi_i < 0\}.$$

Because v is non-zero on S (by property (2)) and ξ has support contained in S (by the construction above), (I) holds (in particular, J_+ and J_- cannot both be empty so $\xi \neq 0$). The vector x is constructed by setting

$$x_i = \begin{cases} \xi_i & \text{if } i \in J_+ \cap T \\ -\xi_i & \text{if } i \in J_- \cap T \\ 1 & \text{if } i \in J_+^c \cap J_-^c \cap T \\ 0 & \text{otherwise.} \end{cases}$$

Clearly $\text{supp}(x) = T$.

Step 2: Verification that (II) and (III) hold: From the fact that $U\xi = 0$ and (I), it follows that $U\xi_{J_+} = -U\xi_{J_-}$ and (II) follows immediately. To show (III), we first show that $\|\xi_{J_+}\|_1 = \|\xi_{J_-}\|_1$. Since ξ is in the nullspace of U and v is in the row span of U, $\langle v, \xi\rangle = 0$. By (I), $0 = \langle v, \xi_{J_+}\rangle + \langle v, \xi_{J_-}\rangle$. Finally, by the definition of J_+, J_- and (2), we must have $0 = \langle v, \xi_{J_+}\rangle + \langle v, \xi_{J_-}\rangle = \frac{\|\xi_{J_+}\|_1 - \|\xi_{J_-}\|_1}{c}$ which proves that $\|\xi_{J_+}\|_1 = \|\xi_{J_-}\|_1$. Next, because J_+, $T \cap J_+^c \cap J_-^c$ and J_- are all disjoint,

$$\|x + \epsilon\xi_{J_+}\|_1 = \|x_{J_+} + \epsilon\xi_{J_+}\|_1 + \|x_{T\cap J_+^c \cap J_-^c}\|_1 + \|x_{J_-}\|_1.$$

Because $\text{supp}(x) = T$, $\text{sgn}(x_{J_+\cap T}) = \text{sgn}(\xi_{J_+\cap T})$ and $\|\xi_{J_+}\|_1 = \|\xi_{J_-}\|_1$

$$\|x_{J_+} + \epsilon\xi_{J_+}\|_1 = \|x_{J_+\cap T}\|_1 + \epsilon\|\xi_{J_+}\|_1 = \|x_{J_+\cap T}\|_1 + \epsilon\|\xi_{J_-}\|_1.$$

Therefore $\|x + \epsilon\xi_{J_+}\|_1 = \|x_{J_+\cap T}\|_1 + \epsilon\|\xi_{J_-}\|_1 + \|x_{T\cap J_+^c\cap J_-^c}\|_1 + \|x_{J_-}\|_1$. Furthermore, $\text{sgn}(x_{J_-\cap T}) = -\text{sgn}(\xi_{J_-\cap T})$ and $\text{supp}(x) = T$ so $\|x + \epsilon\xi_{J_+}\|_1 = \|x_{J_+\cap T}\|_1 + \|x_{T\cap J_+^c\cap J_-^c}\|_1 + \|x_{J_-} - \epsilon\xi_{J_-}\|_1$. We conclude that $\|x+\epsilon\xi_{J_+}\|_1 = \|x-\epsilon\xi_{J_-}\|_1$, as claimed.

Step 3: Verification that (IV) holds: For any such $\hat{\xi}$, we must have $U\hat{\xi} = 0$. We claim that $\hat{\xi}_{T\cap J_+^c\cap J_-^c} = 0$. Otherwise, property (3) implies that we can find $\rho' \neq 0$ and $t \in T \cap J_+^c \cap J_-^c$ satisfying

$$t \in \text{supp}(\hat{\xi}) \cap T \subseteq \text{supp}(\rho') \subseteq S$$

and $U\rho' = 0$. By (I), $t \notin \text{supp}(\xi)$. But then $\text{supp}(\rho') \not\subseteq \text{supp}(\xi)$, contradicting (5.7.1). Therefore $\hat{\xi}_{T\cap J_+^c\cap J_-^c} = 0$ as claimed. Consequently, using the fact that $\text{supp}(x) = T$,

$$\begin{aligned}\|\hat{\xi} + x + \epsilon\xi_{J_+}\|_1 &= \|\hat{\xi}_{J_+} + x_{J_+\cap T} + \epsilon\xi_{J_+}\|_1 + \|\hat{\xi}_{J_-} + x_{J_-\cap T}\|_1 + \|\hat{\xi}_{J_+^c\cap J_-^c} + x_{J_+^c\cap J_-^c}\|_1 \\ &= \|\hat{\xi}_{J_+} + x_{J_+\cap T} + \epsilon\xi_{J_+}\|_1 + \|\hat{\xi}_{J_-} + x_{J_-\cap T}\|_1 + \|x_{T\cap J_+^c\cap J_-^c}\|_1 + \|\hat{\xi}_{J_+^c\cap J_-^c}\|_1\end{aligned}$$

Now by (2) $\|v\|_\infty = 1/c$ and so for any vector $z \in \mathbb{C}^N$, $\|z\|_1 \geq c\langle v, z\rangle$. Therefore

$$\begin{aligned}\|\hat{\xi} + x + \epsilon\xi_{J_+}\|_1 &\geq c\langle v, \hat{\xi}_{J_+} + x_{J_+\cap T} + \epsilon\xi_{J_+}\rangle + c\langle v, \hat{\xi}_{J_-} + x_{J_-\cap T}\rangle \\ &\quad + \|x_{T\cap J_+^c\cap J_-^c}\|_1 + c\langle v, \hat{\xi}_{J_+^c\cap J_-^c}\rangle \\ &= c\langle v, \hat{\xi}\rangle + c\langle v, x_{J_+\cap T} + \epsilon\xi_{J_+}\rangle + c\langle v, x_{J_-\cap T}\rangle + \|x_{T\cap J_+^c\cap J_-^c}\|_1 \\ &= c\langle v, x_{J_+\cap T} + \epsilon\xi_{J_+}\rangle + c\langle v, x_{J_-\cap T}\rangle + \|x_{T\cap J_+^c\cap J_-^c}\|_1\end{aligned}$$

where the last equality follows because $\hat{\xi}$ is in the nullspace of U and v is in the row span of U, so $\langle v, \hat{\xi}\rangle = 0$. Finally, by property (2) and the definition of x,

$$c\langle v, x_{J_+\cap T} + \epsilon\xi_{J_+}\rangle = c\langle v, \xi_{J_+\cap T} + \epsilon\xi_{J_+}\rangle = \|\xi_{J_+\cap T} + \epsilon\xi_{J_+}\|_1 = \|x_{J_+\cap T} + \epsilon\xi_{J_+}\|_1$$

Similar arguments show that $c\langle v, x_{J_-\cap T}\rangle = \|\xi_{J_-\cap T}\|_1 = \|x_{J_-\cap T}\|_1$. Therefore

$$\begin{aligned}\|\hat{\xi} + x + \epsilon\xi_{J_+}\|_1 &\geq \|x_{J_+\cap T} + \epsilon\xi_{J_+}\|_1 + \|x_{J_-\cap T}\|_1 + \|x_{T\cap J_+^c\cap J_-^c}\|_1 \\ &= \|x + \epsilon\xi_{J_+}\|_1,\end{aligned}$$

and we have shown (IV). □

Although we do not use this in the thesis, it is worth examining how Lemma 5.7.1 can be applied to produce examples of Hadamard and Bernoulli data with infinite condition number. Let $H_n \in \mathbb{R}^{2^n\times 2^n}$ be the 1D Hadamard transform with the sequency ordering and let $W_n \in \mathbb{R}^{2^n\times 2^n}$ be the 1D discrete Haar wavelet transform. It is easy to see that $H_nW_n^{-1} = \bigoplus_{i=0}^{n-1} 2^{-i}X_i$ where $X_i \in \mathbb{R}^{2^{i+1}\times 2^{i+1}}$ is a Hadamard matrix. An efficient way to subsample is by defining projections P_{Ω_i} onto sampling sets Ω_i with for $i = 0, 1, 2, \ldots, n-1$, where $|\Omega_i| \leq 2^{i-1}$ for $i \geq 1$ and $|\Omega_1| \leq 1$. Our subsampled

matrix is then

$$A = \bigoplus_{i=0}^{n-1} 2^{-i} P_{\Omega_i} X_i. \tag{5.7.2}$$

However, even though such matrices are very effective in compressive sensing and will with high probability satisfy conditions that guarantee that the l^1 optimisation problem can be computed with error control (i.e. it is in Δ_1), it is easy to show that we end up with an infinite condition number.

Proposition 5.7.2. *The following are examples of matrix-vector pairs with infinite condition number (where we assume as in Lemma 5.7.1 that for each case the set $\tilde{\Omega}_\infty$ satisfies $\tilde{\Omega}_\infty \supseteq \{(y', A) \,|\, y' \in \mathbb{R}^m\}$):*

(i) Let $A \in \mathbb{R}^{m \times N}$ with $m < N$ and $m > 0$ be a subsampled Hadamard or Bernoulli random matrix. Then we have the following: for any set $T \subseteq \{1, 2, \ldots, N\}$, there is a vector x (depending on A) with support T such that $\varrho(Ax, A) = 0$.

(ii) If instead A is as in (5.7.2) with $m = \sum_{i=1}^{n-1} |\Omega_i| < N$ and for every i we have $\Omega_i \neq \emptyset$. Then, for any set $T \subseteq \{1, 2, \ldots, N\}$, there is a vector x with support T such that $C_{RCC}(A, Ax) = \infty$.

(iii) If A and m is as in (ii) such that there is an $i \geq 2$ with $|\Omega_i| \in [1, 2^{i-1} - 1]$. Then for any set $T \subseteq \{2^{i-1} + 1, 2^{i-1} + 2, \ldots, 2^i\}$, there is a vector x with support T such that $C_{RCC}(A, Ax) = \infty$.

Proof of Proposition 5.7.2. Part (i): Since $m < N$, the nullspace of A is non-trivial. Let $\eta^1 \neq 0$ satisfy $A\eta^1 = 0$. Because A is a subsampled Hadamard or Bernoulli matrix, all the entries of A have constant modulus. Moreover, since $m \geq 1$, the matrix A has at least one row, and so we can set $S = \{1, 2, \ldots, N\}$. Thus, (1) and (2) in Lemma 5.7.1 are satisfied. We can now apply Lemma 5.7.1 after observing that condition (3) is met trivially for such an S.

Part (ii): As above, since $m < N$, the nullspace of A is non-trivial, thus let$\eta^1 \neq 0$ such that $A\eta^1 = 0$. Since

$$A = \bigoplus_{i=0}^{n-1} 2^{-i} P_{\Omega_i} X_i,$$

where X_i is a Hadamard matrix it is clear that the entries of A have constant modulus within each block. As a consequence, by the block diagonality of A, there is a vector v in the rowspan of A with $S := \{1, 2, \ldots, N\} = \mathrm{supp}(v)$ such that $\max_{i \in S} |v_i| = \min_{i \in S} |v_i| = 1/c$ for some constant $c > 0$. Hence, the theorem follows from Lemma 5.7.1.

Part (iii): Since $|\omega_i| \in [1, 2^{i-1} - 1]$ and $X_i \in \mathbb{R}^{2^{i-1} \times 2^{i-1}}$, the matrix $P_{\omega_i} X_i$ has a non-trivial nullspace. Let w be a non-zero vector in the nullspace of $P_{\omega_i} X_i$, and write $n^1_S = w$, $n^1_{S^c} = 0_{S^c}$. Because A is block diagonal, $An^1 = 0$. Set v to be a row taken from the $P_{\omega_i} X_i$ block of A. On S, v has non-zero entries. Suppose ρ is a vector such that $\mathrm{supp}(\rho) \cap T \neq \emptyset$. If $A\rho = 0$ then $P_{\omega_i} X_i \rho_S = 0$. Therefore by block diagonality $\rho' = \rho_S$ is non-zero and $A\rho' = 0$. The proof is complete by Lemma 5.7.1. □

We will now prove Theorem 4.2.2. To do this, we will choose Ω to be a subset of all (y, U) such that $y = Ux$ for some sparse x and U obeys the RIP. To guarantee that such a set includes some Hadamard and the Bernoulli matrices, we quote the following two results from [60]. In particular, [60, Theorem 12.31] and the surrounding discussion yields

Theorem 5.7.3. *If $A \in \mathbb{R}^{m\times N}$ is a scaled Hadamard matrix with randomly selected rows then there exists a constant C independent of m, N, δ (for some $\delta > 0$) such that if $m \geq C \log^4(N) s\delta^{-2}$ then A obeys the RIP of order s with constant $\delta_s \leq \delta$ with probability at least $1 - N^{-\log^3(N)}$.*

Furthermore, [60, Theorem 9.2] tells us that

Theorem 5.7.4. *There is a constant $C > 0$ such that if A is a scaled Bernoulli random matrix then with probability at least $1 - \epsilon$ the matrix A obeys the RIP of order s with constant $\delta_s \leq \delta$ provided*

$$m \geq C\delta \left[s \log(eN/s) + \log(2\epsilon^{-1}) \right].$$

Proof of Theorem 4.2.2. For m, N, δ which shall be chosen later, we let Ω be the set of all possible (y, U) with RIP constant of order 2 strictly smaller than 2δ and $y = Ux$ for some 1-sparse x. Note that for any $(y, U) \in \Omega$, there exists $\epsilon'_U > 0$ so that if $\|U' - U\|_2 \leq \epsilon'_U$ then $(y, U') \in \Omega$ Take δ sufficiently small so that the RIP in levels (with one level and a ratio constant of 1) constant $\delta_{\mathbf{s},\mathbf{M}}$ holds so that U has the nullspace property in levels as in Theorem 4.2.7. Note that this will occur for sufficiently small δ by the argument used to show Theorem 4.2.14.

In the case of the Hadamard matrices, we choose $m, N > 1$ sufficiently large so that $m \geq 4C \log^4(N)\delta$. There exists a Hadamard matrix A such that $A \in \mathbb{R}^{m\times N}$ obeys the RIP of order 2 with constant $\delta_2 < 2\delta$ by Theorem 5.7.3.

It is clear from the definition of a Hadamard matrix that for each natural number $i \leq m$ there exists one-sparse $x^{1,i}, x^{2,i} \in \mathbb{R}^N$ so that if $y^j = Ax^{j,i}$ for $j = 1, 2$ then $y^1_i, y^2_i \neq 0$ and $y^1_i = -y^2_i$. Thus since $(y^j, A) \in \Omega$, $\tilde{\Omega}_\infty$ satisfies $\tilde{\Omega}_\infty \supseteq \{(y', A) \,|\, y' \in \mathbb{R}^m\}$. Therefore by Proposition 5.7.2 there exists x with support of size 1 such that if $b = Ax$ then $\varrho(b, A) = 0$. Thus $C_{\mathrm{RCC}}(b, A) = \infty$.

In the case of the Bernoulli matrices, we choose m, N sufficiently large so that $m \geq C\delta \left[2 \log(eN/2) + \log(2\epsilon^{-1}) \right]$ where $\epsilon = 1/2$. We claim that there exists a matrix $A \in \mathbb{R}^{m\times N}$ such that A is a Bernoulli matrix with RIP constant δ_2 satisfying $\delta_2 \leq \delta$ and such that for each natural number $i \leq m$ there exists one-sparse $x^{1,i}, x^{2,i} \in \mathbb{R}^N$ so that if $y^j = Ax^{j,i}$ for $j = 1, 2$ then $y^1_i, y^2_i \neq 0$ and $y^1_i = -y^2_i$.

Indeed, let A be chosen at random from all Bernoulli matrices in $\mathbb{R}^{m\times N}$. By Theorem 5.7.4, A obeys the RIP of order $\delta_2 \leq \delta$ with probability greater than or equal to $1/2$ (and thus there is at least one such A). Moreover, we have $Ae_1 = -Ae_1$ and each entry of Ae_i is non-zero. Hence it suffices to choose $x^{1,i} = e_i, x^{2,i} = -e_i$.

As before, by Proposition 5.7.2 there exists x with support of size 1 such that if $b = Ax$ then $\varrho(b, A) = 0$ and so $C_{\mathrm{RCC}}(b, A) = \infty$.

All that remains is to show that in either case, $\{\Xi, \Omega\}^{\Delta_1} \in \Delta_1^A$. But, as stated before, the RIP (with sufficiently small δ) implies the nullspace property and thus the nullspace property in levels (with one level and a ratio constant of 1). Therefore we can conclude the argument by applying Theorem 4.2.7. □

5.8 Proof of Theorem 4.2.3 and Theorem 4.2.14

The proofs of these two theorems are very similar and based on combining Proposition 2.9.1 and Proposition 5.3.1.

Proof of Theorem 4.2.3. Set $\epsilon_n = \epsilon \wedge (N4^n\|U\|_{\max})^{-1}$. We find negative semidefinite diagonal matrices $E_n^1 := E^1(\epsilon_n), E_n^2 := E^2(\epsilon_n)$ and vectors x^1 and x^2 as in Proposition 5.3.1. Set $\iota^0 = (y, U)$ and $\iota_n^j := (y, U + UE_n^j)$ for $j = 1, 2$. Next, set $S^1 = \{x^1\}, S^2 = \{x^2\}$. Our aim will be to use Proposition 2.9.1. The desired result will follow immediately if we can show that Proposition 2.9.1 condition (a) and condition (b) hold.

By the definition of x^1, x^2 and E^1, E^2 we have that $S^1 = \Xi^{\mathrm{BP}}(y, U + UE^1)$ and $S^2 = \Xi^{\mathrm{BP}}(y, U + UE^2)$. Furthermore, since $x^1 \neq x^2$ we have that

$$\inf_{v \in S^1, w \in S^2} d_{\mathcal{M}}(v, w) = d_{\mathcal{M}}(x^1, x^2) = \|x^1 - x^2\|_2 = \kappa > 0.$$

Thus we have Proposition 2.9.1 condition (a).

Secondly, we will show that Proposition 2.9.1 condition (b) holds. We set $c_f = f(\iota^0)$. It will suffice to consider f such that $f(\iota) = M_{a,b}$ for some fixed a and b whenever $\iota = (y, M)$: for all other f it is obvious that $f(\iota_n^j) = f(\iota^0)$. We have that $|f(\iota^0) - f(\iota_n^j)| = |(UE_n^j)_{a,b}|$. However,

$$\|UE_n^j\|_{\max} \leq \max_{a \leq m, b \leq N} \left| \sum_{i=1}^N U_{a,i}(E_n^j)_{i,b} \right| \leq \max_{a \leq m, b \leq N} \sum_{i=1}^N \|U\|_{\max}\|E_n^j\|_{\max} \leq N\|U\|_{\max}\epsilon_n.$$

We conclude that $|f(\iota^0) - f(\iota_n^j)| \leq 4^{-n}$ since $N\|U\|_{\max}\epsilon_n \leq 4^{-n}$ and thus Proposition 2.9.1 condition (b) holds. The result follows from an application of Proposition 2.9.1. □

Proof of Theorem 4.2.14. The proof of Theorem 4.2.14 is identical to the proof of Theorem 4.2.3. □

5.9 Proof of Theorem 4.2.11

We shall use Theorem 4.2.3 by constructing a matrix U (depending on K) that obeys (4.2.2) and a vector y so that if $\iota = (y, U)$ then there are two minimisers to $\Xi(\iota)$. We assume without loss of generality that $K = \{1, 2, \ldots, |K|\}$, since otherwise we can instead replace U by the matrix UQ where Q is a permutation matrix mapping the set K to $\{1, 2, \ldots, |K|\}$ and then UQ will satisfy (4.2.2). Define

$$U = \left(\begin{array}{ccc|c} 1 & \mathbf{0}_{1\times(|K|-1)} & 1 & \\ \mathbf{0}_{(|K|-1)\times 1} & \mathbf{I}_{(|K|-1)\times(|K|-1)} & \mathbf{0}_{(|K|-1)\times 1} & \mathbf{0}_{|K|\times(N-|K|-1)} \\ \hline & \mathbf{0}_{(m-|K|)\times N} & & \end{array}\right)$$

$$y = 1 \oplus \mathbf{0}_{m-1}, \quad v^1 = 1 \oplus \mathbf{0}_{N-1}, \quad v^2 = \mathbf{0}_K \oplus 1 \oplus \mathbf{0}_{N-K-1}.$$

where for non-negative a and b, $\mathbf{0}_{a\times b}$ is the (possibly empty for a or b equal to 0) $a \times b$ matrix of 0s and $\mathbf{I}_{a\times a}$ is the $a \times a$ identity matrix. It is easy to see that for all $x^1 \in \ker U$, $\|x^1_K\|_1 \leq \|x^1_{K^c}\|_1$ since if $x^1 \in \ker U$ then $x^1_{\{2,3,\dots,K\}} = 0$ and $|x^1_1| = |x^1_{K+1}|$.

Since $Uv^1 = y$, we see that $(y, U) \in \Omega_4$. Note also that $v^1, v^2 \in \Xi(y, U)$. Hence, if we can show that for sufficiently small $\epsilon > 0$, $(y, U+UD) \in \Omega$ whenever D is a diagonal matrix with $\|D\|_2 < \epsilon$, the result will follow immediately by Theorem 4.2.3. Let D be a diagonal matrix with $\|D\|_2 < \epsilon$ (so that $\|D\|_{\max} < \epsilon$) and let $x \in \ker U + UD$. Then $x + Dx \in \ker U$, so

$$(1-\epsilon)\|x_K\|_1 \leq \|x_K + Dx_K\|_1 \leq \|x_{K^c} + Dx_{K^c}\|_1 \leq (1+\epsilon)\|x_{K^c}\|_1$$

therefore $\|x_K\|_1 \leq \frac{(1+\epsilon)}{(1-\epsilon)}\|x_{K^c}\|_1 < \rho\|x_{K^c}\|_1$ for small ϵ so that $(U + UD)$ satisfies (4.2.2). Moreover, since $\mathrm{supp}(v^1 + Dv^1) = \mathrm{supp}(v^1)$ it follows that $(y, U + UD)$ is in Ω. We now apply Theorem 4.2.3.

5.10 Proof of Theorem 4.2.12

Proof. For condition (A), we set $\mathbf{s} = (1, C^2 - 2C)$ and $\mathbf{M} = (0, C, C^2 - C)$ and for condition (B) we set $\mathbf{s} = \mathbf{1}_{C^2-2C+1}$ and $\mathbf{M} = (0, C, C+1, \dots, C^2 - C - 1, C^2 - C)$, with C chosen to be an integer such that

$$\frac{aC^2 + C^2 - 2C}{C^3 + C^2 - 2C} \leq \frac{1}{2f(C^2 - 2C + 1)} \vee \frac{1}{2f(C^2 - 2C)} \tag{5.10.1}$$

(C exists by the assumption on the decay of f). Set $N = C^2 - C$, $m = C^2 - C - 1$ and $g(\eta_{\mathbf{s},\mathbf{M}}, l) = \frac{1}{f(\eta_{\mathbf{s},\mathbf{M}})}$ for condition (A) and $g(\eta_{\mathbf{s},\mathbf{M}}, l) = \frac{1}{f(l)}$ for condition (B). Let x^1, x^2 and b^1 be the vectors

$$x^1 := \beta(C \oplus \mathbf{0}_{C-1} \oplus \mathbf{1}_{C^2-2C}),\ x^2 := \beta(0 \oplus C\mathbf{1}_{C-1} \oplus \mathbf{0}_{C^2-2C}),\ b^1 = x^1 - x^2,$$

where $\beta = \sqrt{C^3 + C^2 - 2C}$ (note that x^1 is $(\mathbf{s}, \mathbf{M})$-sparse and that $\|b^1\| = 1$ with this choice of β). We form an orthornormal basis of $\mathbb{R}^N$ which we denote by the row vectors $(b^i)_{i=1,2,\dots,N}$ (note that b^1 is a unit vector). Next, we construct the $\mathbb{R}^{(N-1)\times N}$ matrix U by setting $U = [b^2, b^3, \dots, b^N]^T$ Note that $U^*U = I$ so that if $\|X\|_2$ is

sufficiently small $(U+X)^*(U+X)$ is invertible and

$$\frac{\|(U+X)^*(U+X)\|_2}{\|((U+X)^*(U+X))^{-1}\|_2} < \gamma. \tag{5.10.2}$$

Furthermore, let $\tilde{z} = \lambda_1 b^1 + \lambda_2 b^2 + \cdots + \lambda_N b^N$ be an $(a\mathbf{s}, \mathbf{M})$-sparse vector. Denote the support of $\tilde{z}$ by S. Then $|\,\|Uz\|^2 - \|z\|^2| = |\lambda_1^2| \leq |\langle \tilde{z}_S, b^1\rangle|^2 = |\langle \tilde{z}, b_S^1\rangle|$. However, $\|b_S^1\|^2 \leq \frac{aC^2+C^2-2C}{C^3+C^2-2C} \leq g/2$ by (5.10.1) and the defintion of g. We conclude that W does indeed satisfy the RIP in levels with constant $\delta_{a\mathbf{s},\mathbf{M}} \leq g(\eta_{\mathbf{s},\mathbf{M}}, l)$. We set $y = Ux^1$ and note that $Ux^1 = Ux^2$. Furthermore, it is easy to see that the set of matrices with sufficiently small RIPL constant is an open set, therefore by (5.10.2) the set of acceptable inputs includes $(y, U(I+D))$ for diagonal D with $\|D\|_2$ sufficiently small. We can thus apply Theorem 4.2.3 to see that $\{\Xi_4, \Omega_4, \mathcal{M}_4, \Lambda_4\}^{\Delta_1} \notin \Delta_1^G$. □

5.11 Proof of Theorem 4.2.15 and Theorem 4.3.5

We require the following basic lemma which, since it is well known, we prove in the appendix.

Lemma 5.11.1. *As usual, for any $A \in \mathbb{R}^{m\times N}$, $y \in \mathbb{R}^m$ and $\delta \geq 0$ set*

$$M^{BP}(y, A, \delta) = \arg\min_{x\in\mathbb{R}^N} \|x\|_1 \text{ such that } \|Ax - y\|_2 \leq \delta.$$

We have that $0 \in M^{BP}(y, A, \delta)$ if and only if $\|y\|_2 \leq \delta$. Furthermore, if $0 \notin M^{BP}(y, A, \delta)$ then for any $x \in M^{BP}(y, A, \delta)$ we have $\|Ax - y\|_2 = \delta$.

We begin this section by constructing a family of robust nullspace matrices which have two minimisers when performing basis pursuit denoising. The proof of Theorem 4.2.15 will proceed by choosing a specific example from this family and then employing the argument of Theorem 4.2.14.

Proposition 5.11.2. *Fix a natural number s and real numbers $\alpha, \gamma > 0$ with $\alpha > \gamma(\alpha^2+1)$. Set $k = 2^{\lfloor \log_2(s)\rfloor+1}$. Then there is a matrix $A \in \mathbb{R}^{m\times N}$ and a vector $y \in \mathbb{R}^m$ with $N := N(k) = 2k + k/2$ and $m := m(k) = 2k + k/2 - 1$ such that the following properties hold:*

1. $\|AA^*\|_2 \leq \sqrt{1 + 2\alpha^2 + \frac{1}{\gamma^4}\left(4 + \alpha^2\gamma^2\right)^2}$, $\|(AA^*)^{-1}\|_2 \leq \sqrt{\left(1 + \frac{\alpha^2\gamma^2}{4}\right)^2 + \frac{2\alpha^2\gamma^4}{16} + \frac{\gamma^4}{16}}$.
2. *A obeys the ℓ^2 Robust Nullspace Property of order s with parameters $\rho = \frac{s}{2^{\lfloor \log_2(s)\rfloor+2}-s}$ and $\tau = \sqrt{1+\rho}\|A\|_2\|(AA^*)^{-1}\|_2$.*
3. *There is a $y = Ax$ for some s-sparse x with $\|y\|_2 \leq \sqrt{1+\alpha^2}$, such that if $M^{BP}(y, A, \delta) = \operatorname{argmin}\|x\|_1$ such that $\|Ax - y\|_2^2 \leq \delta$, then*

$$M^{BP}(y, A, 1) = \mathrm{Conv}(c_1\mathbf{1}_k \oplus \mathbf{0}_k \oplus c_2\mathbf{1}_{k/2}, \mathbf{0}_k \oplus -c_1\mathbf{1}_k \oplus c_2\mathbf{1}_{k/2}),$$
$$M_{min}^{\mathrm{BP}}(y, A) = \{c_1\mathbf{1}_k \oplus \mathbf{0}_k \oplus c_2\mathbf{1}_{k/2}, \mathbf{0}_k \oplus -c_1\mathbf{1}_k \oplus c_2\mathbf{1}_{k/2}\},$$

where

$$c_1 = \sqrt{\frac{2}{k}}\frac{\left(-\gamma - \gamma\alpha^2 + \alpha\right)}{C}, \quad c_2 = \frac{2\sqrt{2}}{\sqrt{k}\gamma}\left[\frac{\alpha\gamma - 1}{C} + 1\right], \quad C = \sqrt{\gamma^2 + (1-\alpha\gamma)^2}.$$

4. *There is a* $y = Ax$ *for some* s*-sparse* x*, such that if* $M^{UL}(y, A, \lambda) = \operatorname{argmin} \lambda\|x\|_1 + \|Ax - y\|_2^2$ *then*

$$M^{UL}(y, A, \lambda) = \operatorname{Conv}(c_1\mathbf{1}_k \oplus \mathbf{0}_k \oplus c_2\mathbf{1}_{k/2}, \mathbf{0}_k \oplus -c_1\mathbf{1}_k \oplus c_2\mathbf{1}_{k/2}),$$
$$M^{\mathrm{BP}}_{min}(y, A) = \{c_1\mathbf{1}_k \oplus \mathbf{0}_k \oplus c_2\mathbf{1}_{k/2}, \mathbf{0}_k \oplus -c_1\mathbf{1}_k \oplus c_2\mathbf{1}_{k/2}\},$$

where

$$c_1 = \lambda\left(\frac{\alpha}{\gamma} - \alpha^2 - 1\right), \quad c_2 = \frac{2\lambda}{\gamma}\left(\alpha - \gamma^{-1}\right) + \frac{2\sqrt{2}}{\gamma\sqrt{k}}$$

whenever $\beta := \frac{\lambda\sqrt{k}}{\sqrt{2}} < \gamma(1-\alpha\gamma)^{-1}$.

Proof. Note that for some $n \geq 1$, s can be written as $s = 2^{n-1} + c$ where $0 \leq c < 2^{n-1}$ (so that $n = \lfloor \log_2(s) \rfloor + 1$). Let $k = 2^n$ and let H_n be the Hadamard matrix with entries ± 1 of dimension $k \times k$ defined by the recurrence relation

$$H_j = H_1 \otimes H_{j-1}, \quad H_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad H_0 = 1, \quad j \in \mathbb{N}.$$

Set $m = 2k + k/2 - 1$ and $N = 2k + k/2$. Let A be the $\mathbb{R}^{m\times N}$ matrix and y be the $\mathbb{R}^{m\times 1}$ vector defined by

$$A = \frac{1}{\sqrt{2k}} P \begin{pmatrix} H_1 \otimes H_n & \eta \otimes H_{n-1} \\ \mathbf{0}_{k/2\times 2k} & \gamma H_{n-1} \end{pmatrix}, \quad \eta = (0, 0, \alpha\gamma, 0)^T \tag{5.11.1}$$

$$P \in \mathbb{R}^{m\times N}, \quad Pe_j = e_{j-1},\, Pe_1 = 0, \qquad y = \mathbf{0}_{k-1} \oplus \alpha \oplus \mathbf{0}_{k-1} \oplus 1 \oplus \mathbf{0}_{k/2-1}, \tag{5.11.2}$$

where $\{e_j\}$ represents the canonical basis. To show (1) note that

$$\begin{pmatrix} H_1 \otimes H_n & \eta \otimes H_{n-1} \\ \mathbf{0}_{k/2\times 2k} & \gamma H_{n-1} \end{pmatrix} \begin{pmatrix} H_1 \otimes H_n & \eta \otimes H_{n-1} \\ \mathbf{0}_{k/2\times 2k} & \gamma H_{n-1} \end{pmatrix}^*$$
$$= \begin{pmatrix} 2kI_{2k} + \eta \otimes \eta^* \otimes \frac{k}{2} I_{k/2} & \gamma\eta \otimes H_{n-1}H_{n-1}^* \\ \gamma(\eta \otimes H_{n-1}H_{n-1}^*)^* & \gamma^2 H_{n-1}H_{n-1}^* \end{pmatrix}$$
$$= \begin{pmatrix} 2kI_{2k} + \alpha^2\gamma^2\xi \otimes \xi^* \otimes \frac{k}{2} I_{k/2} & \alpha\gamma^2\xi \otimes \frac{k}{2} I_{k/2} \\ \alpha\gamma^2(\xi \otimes \frac{k}{2} I_{k/2})^* & \gamma^2 \frac{k}{2} I_{k/2} \end{pmatrix} = 2k(I_k \oplus (S \otimes I_{k/2})),$$

where $\xi = (0, 0, 1, 0)^T$ and

$$S = \begin{pmatrix} (1 + \frac{\alpha^2\gamma^2}{4}) & 0 & \frac{\alpha\gamma^2}{4} \\ 0 & 1 & 0 \\ \frac{\alpha\gamma^2}{4} & 0 & \frac{\gamma^2}{4} \end{pmatrix}, \quad S^{-1} = \begin{pmatrix} 1 & 0 & -\alpha \\ 0 & 1 & 0 \\ -\alpha & 0 & \frac{1}{\gamma^2}(4 + \alpha^2\gamma^2) \end{pmatrix}. \tag{5.11.3}$$

Thus, by the definition of P it follows that

$$AA^* = I_{k-1} \oplus \left(S \otimes I_{k/2}\right), \quad (AA^*)^{-1} = I_{k-1} \oplus \left(S^{-1} \otimes I_{k/2}\right). \tag{5.11.4}$$

Thus, we only need to estimate $\|S\|$ and $\|S^{-1}\|$ from (5.11.3). To do that, note that for a matrix of the form $M := \begin{pmatrix} a & b \\ b & c \end{pmatrix}$ with fixed constants a, b, c and vectors $x \in \mathbb{C}^k$ with $x = x^1 \oplus x^2$ and $x^1, x^2 \in \mathbb{C}^{k/2}$, we have $\|Mx\|_2^2 = \|ax^1 + bx^2\|_2^2 + \|bx^1 + cx^2\|_2^2$. The Cauchy-Schwartz inequality yields for $i = 1, 2, \ldots, k/2$ $(|ax_i^1 + bx_i^2|)^2 \leq (a^2+b^2)(|x_i^1|^2 + |x_i^2|^2)$. Thus (by summing and applying a similar idea to $\|bx^1 + cx^2\|_2^2$) we see that $\|Mx\|_2^2 \leq (a^2 + 2b^2 + c^2)(\|x^1\|_2^2 + \|x^2\|_2^2)$. Therefore $\|M\| \leq \sqrt{a^2 + 2b^2 + c^2}$. Applying this inequality to AA^* and $(AA^*)^{-1}$ using (5.11.3) and (5.11.4), we get $\|AA^*\|_2 \leq 1 \vee \sqrt{\left(1 + \alpha^2\gamma^2/4\right)^2 + \alpha^2\gamma^4/8 + \gamma^4/16}$ and $\|(AA^*)^{-1}\|_2 \leq 1 \vee \sqrt{1 + 2\alpha^2 + \frac{1}{\gamma^4}\left(4 + \alpha^2\gamma^2\right)^2}$, and the result follows.

To prove (2) let $v \in \mathbb{R}^N$ be an arbitrary vector, and write $v = \xi + A^*w$ where ξ is in the nullspace of A and let $K = \{1, 2, \ldots, 2k\}$. Note that the nullspace of A is exactly the vectors of the form $\beta(\mathbf{1}_{2k} \oplus \mathbf{0}_{k/2})$ where $\mathbf{1}_{2k} = (1, \ldots, 1) \in \mathbb{R}^{2k}$, $\mathbf{0}_{k/2} = (0, \ldots, 0) \in \mathbb{R}^{k/2}$ and $\beta \in \mathbb{R}$. Therefore, for any s-sparse set S, we have $\|\xi_S\|_2 \leq \beta\sqrt{s}$ and $\|\xi_{S^c \cap K}\|_1 \geq \beta|S^c \cap K|$. Consequently

$$\begin{aligned}
\|\xi_S\|_2 \leq \frac{\sqrt{s}}{|S^c \cap K|}\|\xi_{S^c \cap K}\|_1 &\leq \frac{\sqrt{s}}{|S^c \cap K|}\left(\|v_{S^c \cap K}\|_1 + \|(A^*w)_{S^c \cap K}\|_1\right) \\
&\leq \frac{s\|v_{S^c}\|_1}{|S^c \cap K|\sqrt{s}} + \sqrt{\frac{s}{|S^c \cap K|}}\frac{\|(A^*w)_{S^c \cap K}\|_1}{\sqrt{|S^c \cap K|}} \\
&\leq \frac{\rho\|v_{S^c}\|_1}{\sqrt{s}} + \sqrt{\frac{s}{|S^c \cap K|}}\|(A^*w)_{S^c \cap K}\|_2
\end{aligned}$$

with $\rho = \frac{s}{2k-s}$, where the last line follows because $|S^c \cap K| \geq 2k - s$. Hence,

$$\|v_S\|_2 \leq \frac{\rho\|v_{S^c}\|_1}{\sqrt{s}} + \sqrt{\rho}\|(A^*w)_{S^c \cap K}\|_2 + \|(A^*w)_S\|_2 \leq \frac{\rho\|v_{S^c}\|_1}{\sqrt{s}} + \sqrt{1+\rho}\|A^*w\|_2$$

where the inequality $\sqrt{\rho}\|(A^*w)_{S^c \cap K}\|_2 + \|(A^*w)_S\|_2 \leq \sqrt{1+\rho}\|A^*w\|_2$ is an application of the cauchy-schwartz inequality with the vectors $(\sqrt{\rho}, 1)$ and $(\|(A^*w)_{S^c \cap K}\|_2, \|(A^*w)_S\|_2)$. Since $s = k/2 + c$ where $c < k/2$, $s < k$ we see that $\rho < 1$. To bound $\|A^*w\|_2$, we see that

$$\|A^*w\|_2 \leq \|A\|_2\|(AA^*)^{-1}(AA^*)w\|_2 \leq \|A\|_2\|(AA^*)^{-1}\|_2\|AA^*w\|_2 = \|A\|_2\|(AA^*)^{-1}\|_2\|Av\|_2$$

because $Av = AA^*w + A\xi = AA^*w$ since ξ is in the nullspace of A. We conclude that

$$\|v_S\|_2 \leq \frac{\rho\|v_{S^c}\|_1}{\sqrt{s}} + \sqrt{1+\rho}\|A^*w\|_2 \leq \frac{\rho\|v_{S^c}\|_1}{\sqrt{s}} + \tau\|Av\|_2.$$

where $\tau = \sqrt{1+\rho}\|A\|_2\|(AA^*)^{-1}\|_2$.

To establish (3) we begin by letting $x = \frac{k\sqrt{2k}}{2\gamma}\mathbf{0}_{2k} \oplus \mathbf{1}_{k/2}$. Then $Ax = y$. Moreover,

since x has $k/2$ non-zero entries and $k/2 = 2^{n-1} \leq s$, x is s-sparse. In order to establish that $|M^{\mathrm{BP}}(y, A, 1)| \geq 2$ and $|M^{\mathrm{UL}}(y, A, \lambda^*)| \geq 2$ we will split the argument into several sub-steps. In particular, we show the following:

(3a) The sets $M^{\mathrm{BP}}(y, A, 1)$ and $M^{\mathrm{UL}}(y, A, \lambda^*)$ are non-empty. Moreover any vector v in $M^{\mathrm{UL}}(y, A, \lambda^*)$ satisfies $\lambda^*\|v\|_1 + \|Av - y\|_2^2 \leq \lambda^*\alpha\sqrt{2k} + 1$.

(3b) Let $v \in M^{\mathrm{BP}}(y, A, 1)$ (similarly, for lasso let $v \in M^{\mathrm{UL}}(y, A, \lambda^*)$) where $v = v^1 \oplus v^2 \oplus v^3$ with $v^1, v^2 \in \mathbb{R}^k$ and $v^3 \in \mathbb{R}^{k/2}$. Then are $c^1, c^2, c^3 \in \mathbb{R}$ such that $v^1 = c^1\mathbf{1}_k$, $v^2 = c^2\mathbf{1}_k$ and $v^3 = c^3\mathbf{1}_{k/2}$.

(3c) For basis pursuit, there is a $t \in [0, 1]$ such that $v = t(c_1\mathbf{1}_k, \mathbf{0}_k, c_2\mathbf{1}_{K/2}) + (1 - t)(\mathbf{0}_k, -c_1\mathbf{1}_k, c_2\mathbf{1}_{k/2})$ where

$$c_1 = \sqrt{\frac{2}{k}}\frac{\left(-\gamma - \gamma\alpha^2 + \alpha\right)}{C}, \quad c_2 = \frac{2\sqrt{2}}{\sqrt{k}\gamma}\left[\frac{\alpha\gamma - 1}{C} + 1\right], \quad C = \sqrt{\gamma^2 + (1 - \alpha\gamma)^2}.$$

(3d) For lasso, there is a $t \in [0, 1]$ such that $v = t(c_1\mathbf{1}_k, \mathbf{0}_k, c_2\mathbf{1}_{K/2}) + (1 - t)(\mathbf{0}_k, -c_1\mathbf{1}_k, c_2\mathbf{1}_{k/2})$ where

$$c_1 = \lambda\left(\frac{\alpha}{\gamma} - \alpha^2 - 1\right), \quad c_2 = \frac{2\lambda}{\gamma}\left(\alpha - \gamma^{-1}\right) + \frac{2\sqrt{2}}{\gamma\sqrt{k}}$$

provided that $\beta := \frac{\lambda\sqrt{k}}{\sqrt{2}} < \gamma(1 - \alpha\gamma)^{-1}$.

We begin by showing (3a). For lasso, the existence of minimizers is obvious, whereas for basis pursuit, we let

$$v' = \frac{\alpha\sqrt{2k}}{k}\mathbf{1}_k \oplus \mathbf{0}_{k+k/2} \quad \text{so that} \quad Av' = \mathbf{0}_{k-1} \oplus \alpha \oplus \mathbf{0}_{k-1+k/2}.$$

Since $\|Av' - y\|_2 = 1$, the set of w with $\|Aw - y\|_2^2 \leq 1$ is non-empty, and the existence of minimizers follows immediately. Property (3a) follows for lasso and basis pursuit by noting that $\|v'\|_1 = \alpha\sqrt{2k}$. To show (3b), let $\bar{v}^1 = \bar{c}^1\mathbf{1}_k$, $\bar{v}^2 = \bar{c}^2\mathbf{1}_k$, $\bar{v}^3 = \bar{c}^3\mathbf{1}_{k/2}$, with $\bar{v} = \bar{v}^1 \oplus \bar{v}^2 \oplus \bar{v}^3$, $\bar{c}^1 = \frac{\sum_{i=1}^k v_i^1}{k}$, $\bar{c}^2 = \frac{\sum_{i=1}^k v_i^2}{k}$, $\bar{c}^3 := \frac{2\sum_{i=1}^{k/2} v_i^3}{k}$. Note that (3b) holds if and only if $v^1 = \bar{v}^1$, $v^2 = \bar{v}^2$ and $v^3 = \bar{v}^3$. We argue by contradiction - suppose that (3b) does not hold and set $y^v = Av$. We consider two cases: (i) $v^3 \neq \bar{v}^3$ and (ii) $v^3 = \bar{v}^3$, but we have that at least one of v^1 and v^2 satisfies $v^1 \neq \bar{v}^1$ or $v^2 \neq \bar{v}^2$. Starting with case (i), if $v^3 \neq \bar{v}^3$ then, by the definition of the Hadamard matrix, it follows that $(H_{n-1}v^3)_j \neq 0$ for some $j > 1$. Thus, by (5.11.1), $y_i^v \neq 0$ for some $i \in \{2k + 1, 2k + 2, \ldots, 2k + k/2 - 1\}$. If we consider case (ii) then $(H_{n-1}v^3)_j = 0$ except for $j = 1$. Note also that, by the definition of the Hadamard matrix, we must have either

$$\begin{pmatrix} PH_n & PH_n \end{pmatrix}(v^1 \oplus v^2) \neq 0 \text{ or } \begin{pmatrix} PH_n & -PH_n \end{pmatrix}(v^1 \oplus v^2) \neq 0,$$

where P is as in (5.11.2), except adjusted to the right dimensions. Thus, there is an $i \in \{1, \ldots, 2k - 1\} \setminus \{k\}$ such that $y_i^v \neq 0$. Hence, in both case (i) and (ii), there is an

$i \in \{1, \ldots, 2k + k/2 - 1\} \setminus \{k, 2k\}$ such that $y_i^v \neq 0$. However,

$$\frac{1}{\sqrt{2k}} \begin{pmatrix} H_1 \otimes H_n & \eta \otimes H_{n-1} \\ \mathbf{0}_{k/2 \times 2k} & \gamma H_{n-1} \end{pmatrix} (\bar{v}^1 \oplus \bar{v}^2 \oplus \bar{v}^3) = \mathbf{0}_k \oplus y_k^v \oplus \mathbf{0}_{k-1} \oplus y_{2k}^v \oplus \mathbf{0}_{k/2-1},$$

hence

$$A\bar{v} = \mathbf{0}_{k-1} \oplus y_k^v \oplus \mathbf{0}_{k-1} \oplus y_{2k}^v \oplus \mathbf{0}_{k/2-1}, \tag{5.11.5}$$

so $\|A\bar{v} - y\|^2 < \|Av - y\|_2^2$. Additionally, $\|\bar{v}^i\|_1 \leq \|v^i\|_1$ so $\|\bar{v}\|_1 \leq \|v\|_1$. We conclude that $\|\bar{v}\|_1 \leq \|v\|_1$ and $\|A\bar{v} - y\|_2^2 < \|Av - y\|_2^2$. For lasso, this is an immediate contradiction: these inequalities imply immediately that $\lambda^*\|\bar{v}\|_1 + \|A\bar{v} - y\|^2 < \lambda^*\|v\|_1 + \|Av - y\|_2^2$. For basis pursuit, these inequalities imply that $\bar{v} \in M^{\mathrm{BP}}(y, A, 1)$ and here we obtain a contradiction in the following way: since $\bar{v} \neq v$ but both are elements of $M^{\mathrm{BP}}(y, A, 1)$, $0 \notin M^{\mathrm{BP}}(y, A, 1)$. Therefore we can apply Lemma 5.11.1 to get $\|Av - y\|_2 = 1, \|A\bar{v} - y\|_2^2 = 1$. But this contradicts $\|A\bar{v} - y\|_2 < \|Av - y\|_2$. Thus $v = \bar{v}$.

Next, we prove (3c). Note that by (3b) and (5.11.5) it follows that $Av - y = \mathbf{0}_{k-1} \oplus (y_k^v - \alpha) \oplus \mathbf{0}_{k-1} \oplus (y_{2k}^v - 1) \oplus \mathbf{0}_{k/2-1}$, where $y^v = Av$. Moreover, a quick calculation gives $y_k^v = \frac{1}{\sqrt{2k}}\left(kc_1 - kc_2 + \frac{\alpha\gamma k c_3}{2}\right)$ and $y_{2k}^v = \frac{\gamma c_3 k}{2\sqrt{2k}}$. Hence, we have that

$$\begin{aligned} \|Av - y\|_2^2 &= \left[\frac{1}{\sqrt{2k}}\left(kc_1 - kc_2 + \frac{\alpha\gamma k c_3}{2}\right) - \alpha\right]^2 + \left(\frac{\gamma c_3 k}{2\sqrt{2k}} - 1\right)^2, \\ \|v\|_1 &= |c_1|k + |c_2|k + |c_3|\frac{k}{2}. \end{aligned} \tag{5.11.6}$$

Thus, we can rewrite the problem of determining v by in terms of a weighted l^1 problem. In particular, by a change of variables where we let $x_1 = \sqrt{k}c_1/(\sqrt{2}\alpha)$, $x_2 = \sqrt{k}c_2/(\sqrt{2}\alpha)$ and $x_3 = k\gamma c_3/(2\sqrt{2k})$, obtaining v is equivalent to finding

$$z \in \operatorname*{argmin}_{x \in \mathbb{R}^3} \sqrt{2k}(\alpha|x_1| + \alpha|x_2| + \frac{1}{\gamma}|x_3|) \text{ s. t. } \alpha^2(x_1 - x_2 + x_3 - 1)^2 + (x_3 - 1)^2 \leq 1.$$

Note that we deliberately choose this kind of change of variables instead of writing the problem directly in terms of c_1, c_2, c_3 in (5.11.6) as this gives easier to manage KKT conditions. In particular, the KKT conditions for this problem are

$$\begin{aligned} &0 \in \frac{\sqrt{k}}{\sqrt{2}}\partial\,\psi(x_1) + \mu_1\omega_1(x)\alpha, \quad \psi(t) = |t| \\ &0 \in \frac{\sqrt{k}}{\sqrt{2}}\partial\,\psi(x_2) - \mu_1\omega(x)\alpha, \qquad 0 \in \frac{\sqrt{k}}{\sqrt{2}\gamma}\partial\,\psi(x_3) + \mu_1\left(\omega_1(x)\alpha^2 + \omega_2(x)\right) \\ &\mu_1(\omega_1(x)^2\alpha^2 + \omega_2(x)^2 - 1) = 0, \quad \mu_1 \geq 0 \\ &\omega_1(x)^2\alpha^2 + \omega_2(x)^2 \leq 1, \qquad \omega_1(x) = x_1 - x_2 + x_3 - 1 \quad \omega_2(x) = x_3 - 1. \end{aligned} \tag{5.11.7}$$

Note also that there are x_1', x_2', x_3' with $\alpha^2(x_1' - x_2' + x_3' - 1)^2 + (x_3' - 1)^2 < 1$. Thus Slater's condition applies, so these KKT conditions are both necessary and sufficient. We make the following claims: $\mu_1 \neq 0$, so $\omega_1(x)^2\alpha^2 + \omega_2(x)^2 - 1 = 0$. Additionally, at least one

of the inequalities $x_1 \leq 0, x_2 \geq 0$ is false, and finally we claim that $x_3 > 0$. For the first part, if $\mu_1 = 0$ then the first three equations imply that $0 \in \text{sgn}(x_1), \text{sgn}(x_2), \text{sgn}(x_3)$. Thus $x_1 = x_2 = x_3 = 0$, violating the final condition that $\omega_1(x)^2\alpha^2 + \omega_2(x)^2 \leq 1$ (since $\alpha > 0$). Therefore $\mu_1 \neq 0$.

We claim that we cannot have both $x_1 \leq 0, x_2 \geq 0$. Suppose otherwise. Then $x_1 - x_2 \leq 0$, so $\omega_2(x) \geq \omega_1(x)$. If $x_3 \leq 0$ then $\omega_2(x) \leq -1$, so $\omega_1(x)^2\alpha^2 + \omega_2(x)^2 \geq \alpha^2 + 1 > 1$ contradicting the final condition. Thus $x_3 > 0$ and so the third KKT equation reads $0 = \frac{\sqrt{k}}{\sqrt{2}\gamma} + \mu_1\left(\omega_1(x)\alpha^2 + \omega_2(x)\right) \geq \frac{\sqrt{k}}{\sqrt{2}\gamma} + \mu_1\omega_1(x)\left(\alpha^2 + 1\right)$. Since $\alpha^2 + 1 \geq 0$, this implies that $\mu_1\omega_1(x) < 0$. Hence (from the first KKT equation), $\mu_1\omega_1(x) = \frac{\sqrt{k}K^1}{2\alpha}$ for some $K^1 \in [-1, 0)$. Thus, $0 \geq \frac{\sqrt{k}}{2\gamma} + \frac{\sqrt{k}K^1}{2\alpha}(\alpha^2 + 1)$. This implies that $\frac{-1}{\gamma} \geq \frac{K^1}{\alpha}(\alpha^2 + 1)$. Therefore (since $\gamma > 0$) $\frac{1}{\gamma} \leq \frac{|K^1|}{\alpha}(\alpha^2 + 1) \leq \frac{\alpha^2+1}{\alpha}$, contradicting the premise that $\alpha > \gamma(\alpha^2 + 1)$.

To see that $x_3 > 0$, by the second property at least one of x_1, x_2 are non-zero. Therefore, from the first two KKT equations, we must have that $\omega_1(x) \neq 0$. Since $\alpha^2\omega_1(x)^2 + \omega_2(x)^2 = 1$ we get $\omega_2(x) < 1$. Thus $x_3 > 0$.

Our last goal is to solve for x_1, x_2 and x_3 now armed with the knowledge that at $x_3 > 0$ and that at least one of $x_1 > 0$ or $x_2 < 0$. Note that this implies that either $\text{sgn}(x_1) = 1$ or $\text{sgn}(x_2) = -1$. Therefore $\mu_1\omega_1(x)\alpha + \sqrt{\frac{k}{2}} = 0$. Consequently, the KKT equations are equivalent to

$$0 = \frac{\sqrt{k}}{\sqrt{2}} + \mu_1\omega_1(x)\alpha,\ 0 = \frac{\sqrt{k}}{\sqrt{2}\gamma} + \mu_1\left(\omega_1(x)\alpha^2 + \omega_2(x)\right),\ \omega_1(x)^2\alpha^2 + \omega_2(x)^2 = 1,\ \mu_1 > 0$$

$x_2 \leq 0$ and $x_1, x_3 \geq 0$ with at least one of $x_1 > 0, x_2 < 0$.

Solving the first two equations yields $\omega_1(x) = -\mu_1'/\alpha$, $\omega_2(x) = \mu_1'(\alpha - 1/\gamma)$, where $\mu_1' = \sqrt{\frac{k}{2\mu_1^2}}$. Using $\omega_1(x)^2\alpha^2 + \omega_2(x)^2 = 1$ yields $\mu_1' = \gamma/C$. Therefore $\omega_1(x) = -\gamma/(C\alpha)$ and $\omega_2(x) = (\alpha\gamma - 1)/C$. The substitutions $\omega_1(x) = x_1 - x_2 + \omega_2(x)$ and $\omega_2(x) = x_3 - 1$ gives $x_1 - x_2 = (-\alpha^2\gamma - \gamma + \alpha)/C\alpha$ and $x_3 = (\alpha\gamma - 1)/C + 1$. The values of x_1, x_2 and x_3 in terms of c^1, c^2 and c^3 yield the final result.

Our argument for LASSO is similar: as before, we still have

$$\begin{aligned} \|Av - y\|_2^2 &= \left[\frac{1}{\sqrt{2k}}\left(kc_1 - kc_2 + \frac{\alpha\gamma k c_3}{2}\right) - \alpha\right]^2 + \left(\frac{\gamma c_3 k}{2\sqrt{2k}} - 1\right)^2, \\ \|v\|_1 &= |c_1|k + |c_2|k + |c_3|\frac{k}{2}. \end{aligned} \tag{5.11.8}$$

With the same change of variables $x_1 = \sqrt{k}c_1/(\sqrt{2}\alpha)$, $x_2 = \sqrt{k}c_2/(\sqrt{2}\alpha)$ and $x_3 = k\gamma c_3/(2\sqrt{2k})$, obtaining v for the LASSO case is equivalent to finding

$$z \in \underset{x\in\mathbb{R}^3}{\text{argmin}}\ \lambda\sqrt{2k}(\alpha|x_1| + \alpha|x_2| + \frac{1}{\gamma}|x_3|) + \alpha^2(x_1 - x_2 + x_3 - 1)^2 + (x_3 - 1)^2.$$

Taking subdifferentials yields the following KKT conditions

$$\begin{aligned} &0 \in \beta\partial\,\psi(x_1) + \omega_1(x)\alpha, \quad \psi(t) = |t| \\ &0 \in \beta\partial\,\psi(x_2) + \omega_1(x)\alpha, \qquad 0 \in \frac{\beta}{\gamma}\partial\,\psi(x_3) + \omega_1(x)\alpha^2 + \omega_2(x) \\ &\omega_1(x) = x_1 - x_2 + x_3 - 1, \quad \omega_2(x) = x_3 - 1, \quad \beta = \frac{\lambda\sqrt{k}}{\sqrt{2}}. \end{aligned} \tag{5.11.9}$$

Although these conditions are similar to the ones given in (5.11.7), we will no longer be able to conclude easily that $\omega_1(x)^2\alpha^2 + \omega_2(x)^2 - 1 = 0$. Our proof will therefore be slightly different and will rely on the fact that $\beta < \gamma(1-\alpha\gamma)^{-1}$. We will initially make three claims: firstly, that $\beta < \alpha$, secondly that $x_3 > 0$ and thirdly that at least one of the inequalities $x_1 \leq 0, x_2 \geq 0$ is false.

It is easy to see the first claim: we have $\alpha > \gamma(\alpha^2+1)$. Thus $\alpha(1-\alpha\gamma) > \gamma$ and the claim follows from the fact that $\beta < \gamma(1-\alpha\gamma)^{-1}$. The second claim is more involved: note that if $x_3 \leq 0$, the first KKT condition reads $\alpha \leq \beta K^1 + (x_1 - x_2)\alpha$ for some $K^1 \in \partial\,\psi(x_1)$. If $(x_1 - x_2) \leq 0$ then $\alpha \leq \beta K^1 \leq \beta$, contradicting the fact that $\beta < \alpha$. Thus either $x_1 > 0$ or $x_2 < 0$ (or both), and so at least one of the first two KKT conditions can be written as $0 = \beta + \omega_1(x)\alpha$. If we substitute this into

$$0 \in \frac{\beta}{\gamma}\partial\,\psi(x_3) + \omega_1(x)\alpha^2 + \omega_2(x),$$

we get $0 = \beta\left(\gamma^{-1}K^3 - \alpha\right) + \omega_2(x)$ for some $K^3 \in \partial\,\psi(x_3)$. Again using the assumption that $x_3 \leq 0$, we have $\omega_2(x) \leq -1$. Thus $1 \leq \beta\gamma^{-1}\left(K^3 - \alpha\gamma\right) \leq \beta\gamma^{-1}(1-\alpha\gamma)$ contradicting the condition $\beta < \gamma(1-\alpha\gamma)^{-1}$. Thus $x_3 > 0$.

For the third claim, we argue similarly to the basis pursuit case. Let us assume otherwise (i.e. that $x_1 \leq 0$ and $x_2 \geq 0$): then $\omega_1(x) \leq \omega_2(x)$. Using the second claim and the third KKT equation, we see that $0 = \frac{\beta}{\gamma} + \omega_1(x)\alpha^2 + \omega_2(x) \geq \frac{\beta}{\gamma} + \omega_1(x)(\alpha^2+1)$. From the first KKT equation there is a $K^1 \in \partial\,\psi(x_1)$ with $\omega_1(x) = -\frac{\beta K^1}{\alpha}$. Combining these two equations gives $\frac{\beta(\alpha^2+1)K^1}{\alpha} \geq \frac{\beta}{\gamma}$. But $K^1 \leq 1$, and so $\frac{\beta(\alpha^2+1)K^1}{\alpha} \geq \frac{\beta}{\gamma}$ implies that $\gamma(\alpha^2+1) \geq \alpha$ yielding the desired contradiction.

Therefore the KKT equations are equivalent to $0 = \beta + \alpha\omega_1(x)$, $0 = \frac{\beta}{\gamma} + \alpha^2\omega_1(x) + \omega_2(x)$ and $x_2 \leq 0$ and $x_1, x_3 \geq 0$ with at least one of $x_1 > 0, x_2 < 0$. We can now solve these simultaneous equations for ω_1 and ω_2 to get $\omega_1(x) = -\beta/\alpha$ and $\omega_2(x) = \alpha\beta - \beta/\gamma$. Thus from the definition of ω_1 and ω_2 in terms of x_1, x_2 and x_3, we get $x_3 = \alpha\beta - \beta/\gamma + 1$ and $x_1 - x_2 = -\alpha\beta + \beta(1/\gamma - 1/\alpha)$. We can then use the substitutions for c_1, c_2, c_3 in terms of x_1, x_2, x_3 to get the final result. □

We will now choose sensible parameters in Proposition 5.11.2 to prove Theorem 4.2.15.

Proof of Theorem 4.2.15. We would like to apply Proposition 5.11.2 immediately. Unfortunately, that only covers the case $\delta = 1$. However, we can extend the result there by making the following claim: for any $y \in \mathbb{R}^m$ and $U \in \mathbb{R}^{m\times N}$ and real numbers $\delta^1, \delta^2 > 0$ we have that $x \in M^{\mathrm{BP}}(y, U, \delta^1)$ if and only if $\delta^2 x/\delta^1 \in M^{\mathrm{BP}}(\delta^2 y/\delta^1, U, \delta^2)$. To see

this, note it suffices to argue in only the forward direction (since the other direction is identical to the forward direction with a change of parameters). Let $x \in M^{\mathrm{BP}}(y, U, \delta^1)$ and let v be such that $\|Uv - \delta^2 y/\delta^1\|_2 \leq \delta^2$. Then $\|U\frac{\delta^1 v}{\delta^2} - \delta^1 y\|_2 \leq \delta^1$, so $\frac{\delta^1 \|v\|_1}{\delta^2} \geq \|x\|_1$. Thus $\frac{\delta^2 x}{\delta^1}$ satisfies $\|U\frac{\delta^2 x}{\delta^1} - \frac{\delta^2 y}{\delta^1}\|_2 = \frac{\delta^2}{\delta^1}\|Ux - y\| \leq \delta^2$, and for all u such that $\|Uv - \delta^2 y/\delta^1 y\| \leq \delta^2$, $\|u\|_1 \geq \|\frac{\delta^2 x}{\delta^1}\|_1$ and so $\frac{\delta^2 x}{\delta^1}$ is indeed a minimiser.

With $\alpha = 1.4, \gamma = 0.37$, Proposition 5.11.2 gives us the existence of y', A such that

1.
$$\|AA^*\|_2 \leq \sqrt{1 + 2\alpha^2 + \frac{1}{\gamma^4}\left(4 + \alpha^2\gamma^2\right)^2} \approx 31.26$$
$$\|(AA^*)^{-1}\|_2 \leq \sqrt{\left(1 + \frac{\alpha^2\gamma^2}{4}\right)^2 + \frac{2\alpha^2\gamma^4}{16} + \frac{\gamma^4}{16}} \approx 1.14.$$
and thus $\mathrm{cond}(AA^*) \leq 31.26 \times 1.14 \approx 35.64 < 36$ and $\|A\|_2 \leq \sqrt{31.26} < 6$.

2. A obeys the ℓ^2 Robust Nullspace Property of order s with parameters $\rho = \frac{s}{2^{\lfloor \log_2(s) \rfloor + 2} - s} < \frac{s+1}{2^{\lfloor \log_2(s) \rfloor + 2} - s}$ and
$$\tau = \sqrt{1+\rho}\|A\|_2\|(AA^*)^{-1}\|_2 \leq \sqrt{1+\rho}\sqrt{31.26} \times 1.14 \approx 6.4\sqrt{1+\rho} < 7\sqrt{1+\rho'}.$$

3. $y' = Ax$ for some s-sparse x with $\|y\|_2 \leq \sqrt{1+\alpha^2} \approx 1.7205 < 2$, such that
$$M^{\mathrm{BP}}(y', A, 1) = \mathrm{Conv}(c_1\mathbf{1}_k \oplus \mathbf{0}_k \oplus c_2\mathbf{1}_{k/2}, \mathbf{0}_k \oplus -c_1\mathbf{1}_k \oplus c_2\mathbf{1}_{k/2}),$$
$$M^{\mathrm{BP}}_{\min}(y', A, 1) = \{c_1\mathbf{1}_k \oplus \mathbf{0}_k \oplus c_2\mathbf{1}_{k/2}, \mathbf{0}_k \oplus -c_1\mathbf{1}_k \oplus c_2\mathbf{1}_{k/2}\},$$
where
$$c_1 \times C\sqrt{\frac{k}{2}} \approx 0.3048\sqrt{\frac{k}{2}} \leq 0.3049\sqrt{\frac{k}{2}}, \quad C \approx 0.6076 \geq 0.6077.$$
so that $c_1 \approx 0.5017\sqrt{\frac{2}{k}} > (\sqrt{2k})^{-1}$.

Note that all calculations (with the exception of C for which the opposite occured) were done by rounding up. Furthermore, the assertion that $\frac{s+1}{2^{\lfloor \log_2(s) \rfloor + 2} - s} = \rho' < 1$ follows by letting $s = 2^{n-1} + c$ with $n \geq 1$ and c an integer between 0 and $2^{n-1} - 1$ inclusive. We then have that $s + 1 = 2^{n-1} + c + 1$ and that $2^{\lfloor \log_2(s) \rfloor + 2} - s = 2^{n+1} - 2^{n-1} - c$. Thus

$$s+1-(2^{\lfloor \log_2(s) \rfloor + 2} - s) = 2^{n-1} + c + 1 - 2^{n+1} + 2^{n-1} + c \leq -2^n + 2c + 1 \leq -2^n + 2^n - 2 + 1 < 0.$$

We thus get a matrix A satisfying the requirements for Theorem 4.2.15 with each inequality satisfied in a strict sense. It is easy to see that since the inequalities are satisfied in a strict sense, if we form $B = A + \epsilon A'$ where $\|A'\|_2 = 1$ and ϵ is sufficiently small then $\mathrm{cond}(BB^*) < 36$, $\|B\|_2 < 6$, B will still obey the nullspace property with possibly increased parameters ρ and τ (but still with $\rho < \rho'$ and $\tau < 7\sqrt{1+\rho'}$). Moreover, if we set $y = \delta y'$ we note that $\|y\|_2 = \delta\|y'\|_2 \leq \delta$.

We define $\iota = (y, U, \delta)$ and note that $|M^{\mathrm{BP}}(y, U, \delta)| \geq 2$. The conclusion follows by Theorem 4.2.14 with the statement about the breakdown epsilon following from the fact that if $x^1, x^2 \in M^{\mathrm{BP}}_{\min}(\iota)$ then

$$x^1 - x^2 = \pm\delta \left(c_1 \mathbf{1}_k \oplus c_1 \mathbf{1}_k \oplus \mathbf{0}_{k/2}\right)$$

and so $\|x^1 - x^2\|_2/2 = \delta c_1 \sqrt{\frac{k}{2}} > \delta/2$. □

The proof of Theorem 4.3.5 is very similar, making use of a specific choice of parameters in Proposition 5.11.2.

Proof of Theorem 4.3.5. As in the proof of Theorem 4.2.15, we make the substitution of $\alpha = 1.4, \beta = 0.37$ in Proposition 5.11.2. The same calculations as before show us that we get y, A such that

1. $\mathrm{cond}(AA^*) < 36$ and $\|A\|_2 < 6$.
2. A obeys the ℓ^2 Robust Nullspace Property of order s with parameters $\rho < \frac{s+1}{2^{\lceil \log_2(s) \rceil + 2} - s}$ and $\tau < 7\sqrt{1+\rho'}$.
3. $y = Ax$ for some s-sparse x with $\|y\|_2 < 2$, such that

$$M^{\mathrm{UL}}(y, A, \lambda) = \mathrm{Conv}(c_1 \mathbf{1}_k \oplus \mathbf{0}_k \oplus c_2 \mathbf{1}_{k/2}, \mathbf{0}_k \oplus -c_1 \mathbf{1}_k \oplus c_2 \mathbf{1}_{k/2}),$$
$$M^{\mathrm{BP}}_{\min}(y, A) = \{c_1 \mathbf{1}_k \oplus \mathbf{0}_k \oplus c_2 \mathbf{1}_{k/2}, \mathbf{0}_k \oplus -c_1 \mathbf{1}_k \oplus c_2 \mathbf{1}_{k/2}\},$$

where

$$c_1 = \lambda \left(\frac{\alpha}{\gamma} - \alpha^2 - 1 \right) \approx 0.8239 > 0.8$$

whenever $\beta := \frac{\lambda\sqrt{k}}{\sqrt{2}} < \gamma(1 - \alpha\gamma)^{-1} \approx 0.7676$. Since $\sqrt{2} \times 0.7676 \approx 1.085$ (rounded down), this condition will be satisfied whenever $\lambda < 1/\sqrt{k}$.

We set $\iota = (y, A)$ and note as before that if we consider matrices B which are sufficiently small perturbations of A then $(y, B) \in \Omega$. We then employ Theorem 4.3.2, part 1. To get the breakdown epsilon, note that if $x^1, x^2 \in M^{\mathrm{UL}}_{\min}(\iota)$ then

$$x^1 - x^2 = \pm \left(c_1 \mathbf{1}_k \oplus c_1 \mathbf{1}_k \oplus \mathbf{0}_{k/2}\right)$$

and so $\|x^1 - x^2\|_2/2 = c_1\sqrt{2k}/2 > \frac{4\lambda\sqrt{k}}{5\sqrt{2}}$. □

5.12 Proof of Theorem 4.3.1

We start by proving the following scaling argument for unconstrained lasso

Lemma 5.12.1. *For $y \in \mathbb{R}^m$, $U \in \mathbb{R}^{m \times N}$ and $\lambda > 0$, set*

$$M^{UL}(y, U, \lambda) := \arg\min_{x \in \mathbb{R}^N} \|Ux - y\|_2^2 + \lambda \|x\|_1$$

Then for $\gamma_1, \gamma_2 > 0$, we have

$$M^{UL}(\gamma_1 y, \gamma_2 A, \lambda) = \gamma_1 \gamma_2^{-1} M^{UL}(y, A, \gamma_1^{-1}\gamma_2^{-1}\lambda).$$

Proof. We prove this in two steps, namely, we will show

$$M^{\mathrm{UL}}(\gamma_1 y, \gamma_2 A, \lambda) = \gamma_2^{-1} M^{\mathrm{UL}}(\gamma_1 y, A, \gamma_2^{-1}\lambda) \tag{5.12.1}$$
$$M^{\mathrm{UL}}(\gamma_1 y, A, \gamma_2^{-1}\lambda) = \gamma_1 M^{\mathrm{UL}}(y, A, \gamma_1^{-1}\gamma_2^{-1}\lambda). \tag{5.12.2}$$

To see (5.12.1), let $\hat{x} \in \operatorname{argmin}_{x\in\mathbb{R}^N} \|\gamma_2 Ax - \gamma_1 y\|^2 + \lambda\|x\|_1$. Then for $x \in \mathbb{R}^N$ we have

$$\|\gamma_2 A\hat{x} - \gamma_1 y\|_2^2 + \lambda\|\hat{x}\|_1 \le \|\gamma_2 Ax - \gamma_1 y\|_2^2 + \lambda\|x\|_1$$

so with $\hat{v} = \gamma_2\hat{x}$ and $v = \gamma_2 x$ we have

$$\begin{aligned}\|A\hat{v} - \gamma_1 y\|_2^2 + \gamma_2^{-1}\lambda\|\hat{v}\|_1 &= \|\gamma_2 A\hat{x} - \gamma_1 y\|_2^2 + \lambda\|\hat{x}\|_1 \\ &\le \|\gamma_2 Ax - \gamma_1 y\|_2^2 + \lambda\|x\|_1 = \|Av - \gamma_1 y\|_2^2 + \gamma_2^{-1}\lambda\|v\|_1.\end{aligned}$$

Thus if $\hat{x} \in M^{\mathrm{UL}}(\gamma_1 y, \gamma_2 A, \lambda)$ then $\gamma_2\hat{x} \in M^{\mathrm{UL}}(\gamma_1 y, A, \gamma_2^{-1}\lambda)$. Since the argument is completely reversible we have that if $\gamma_2\hat{x} \in M^{\mathrm{UL}}(\gamma_1 y, A, \gamma_2^{-1}\lambda)$ then $\hat{x} \in M^{\mathrm{UL}}(\gamma_1 y, \gamma_2 A, \lambda)$. The claim follows immediately.

The argument for (5.12.2) is similar. Let $\hat{x} \in \operatorname{argmin}_{x\in\mathbb{R}^N} \|Ax - \gamma_1 y\|^2 + \gamma_2^{-1}\lambda\|x\|_1$. Then for $x \in \mathbb{R}^N$ we have

$$\|A\hat{x} - \gamma_1 y\|_2^2 + \gamma_2^{-1}\lambda\|\hat{x}\|_1 \le \|Ax - \gamma_1 y\|_2^2 + \gamma_2^{-1}\lambda\|x\|_1$$

so with $\hat{v} = \gamma_1^{-1}\hat{x}$ and $v = \gamma_1^{-1}x$ we have

$$\begin{aligned}\gamma_1^2\|A\hat{v} - y\|_2^2 + \gamma_1\gamma_2^{-1}\lambda\|\hat{v}\|_1 &= \|A\hat{x} - \gamma_1 y\|_2^2 + \gamma_2^{-1}\lambda\|\hat{x}\|_1 \\ &\le \|Ax - \gamma_1 y\|_2^2 + \gamma_2^{-1}\lambda\|x\|_1 \\ &= \gamma_1^2\|Av - \gamma_1 y\|_2^2 + \gamma_1\gamma_2^{-1}\lambda\|v\|_1.\end{aligned}$$

Therefore (dividing by γ_1^2)

$$\|A\hat{v} - y\|_2^2 + \gamma_1^{-1}\gamma_2^{-1}\lambda\|\hat{v}\|_1 \le \|Av - \gamma_1 y\|_2^2 + \gamma_1^{-1}\gamma_2^{-1}\lambda\|v\|_1.$$

Thus if $\hat{x} \in M^{\mathrm{UL}}(\gamma_1 y, A, \gamma_2^{-1}\lambda)$ then $\gamma_1^{-1}\hat{x} \in M^{\mathrm{UL}}(y, A, \gamma_1^{-1}\gamma_2^{-1}\lambda)$. Since the argument is completely reversible we have that if $\gamma_1^{-1}\hat{x} \in M^{\mathrm{UL}}(y, A, \gamma_1^{-1}\gamma_2^{-1}\lambda)$ then $\hat{x} \in M^{\mathrm{UL}}(\gamma_1 y, A, \gamma_2^{-1}\lambda)$. The claim follows immediately. □

Lemma 5.12.2. *Fix $\tau \in (0,1)$, $\lambda \in (0,2)$ and $\alpha, \beta, y > 0$ such that $\alpha \vee \beta \ge \lambda/2$ and $1 \ge \alpha\tau \vee \beta\tau$. Set $U = \begin{pmatrix}\alpha & \beta\end{pmatrix}$. Let $\Xi_{\mathrm{CL}}, \Xi_{\mathrm{UL}}$ denote the problem functions for constrained lasso and unconstrained lasso respectively. We have*

$$\Xi_{\mathrm{CL}}(1, U) = \begin{cases}\tau e_1 & \text{if } \alpha > \beta \\ \tau e_2 & \text{if } \beta > \alpha\end{cases}, \quad \Xi_{\mathrm{UL}}(1, U) = \begin{cases}\frac{2\alpha-\lambda}{2\alpha^2} e_1 & \text{if } \alpha > \beta \\ \frac{2\beta-\lambda}{2\beta^2} e_2 & \text{if } \beta > \alpha.\end{cases} \tag{5.12.3}$$

Moreover, for $\alpha = \beta$ we have

$$\Xi_{\mathrm{CL}}(1, U) = \mathrm{Conv}\,(\tau e_1, \tau e_2)$$
$$\Xi_{\mathrm{UL}}(1, U) \supseteq \left\{ \left(\frac{2\alpha - \lambda}{2\alpha^2} \right) e_1, \left(\frac{2\alpha - \lambda}{2\alpha^2} \right) e_2 \right\}.$$

Proof. We start with constrained lasso. As in the proof of Lemma 5.5.1, we consider the case $\alpha > \beta$, noting that the argument for $\beta > \alpha$ is identical. We have for $x \in \mathbb{R}^2$ with $\|x\|_1 \leq \tau$,

$$y - (\alpha x_1 + \beta x_2) \geq y - (\alpha|x_1| + \beta|x_2|) \geq y - \alpha(|x_1| + |x_2|) \geq y - \alpha\tau$$

with equality if and only if $|x_2| = 0$ and $x_1 = \tau$. Moreover, since $y \geq \alpha\tau$, we have

$$0 < y - \alpha\tau \leq y - (\alpha x_1 + \beta x_2) = |y - (\alpha x_1 + \beta x_2)| = \|Ux - y\|_2.$$

Thus over $x \in \mathbb{R}^2$ with $\|x\|_1 \leq \tau$, $\|Ux - y\|_2$ is minimised exactly when $x = \tau e_1$.

To see the result for $\alpha = \beta$, we work similarly: now if $\|x\|_1 \leq \tau$ we have $y - (\alpha x_1 + \beta x_2) \geq y - \alpha\tau > 0$. As before, we get that $\|Ux - y\|_2 \geq y - \alpha\tau$ with equality iff x_1, x_2 are non-negative and $\|x\|_1 = \tau$ and this gives exactly the claimed formula for the minimisers.

Finally, we prove the result for unconstrained lasso. This proof is somewhat more involved than for the other computational problems. We start by considering the problem for $y = 1$, $\alpha = 1$ and $\beta = (1 - \epsilon)$ for some positive $\epsilon < 1$. Our aim will be to prove that for such an input (y, U) that the minimiser is $(1 - \lambda/2)e_1$ provided $\lambda \leq 2$. We will then transform the problem to obtain the result for general U.

Suppose that $(1 - \lambda/2)e_1$ is not the unique minimiser: then there is an $\tilde{x} = (x_1, x_2)^T \neq (1 - \lambda/2)e_1$ such that the $\tilde{x}$ is a solution of $\operatorname{argmin} \|Ux - 1\|_2^2 + \lambda\|x\|_1$. Set $F_\epsilon(a, b) := (a + (1-\epsilon)b - 1)^2 + \lambda(|a| + |b|)$, so that $\|U(a, b)^T - 1\|_2^2 + \lambda\|(a, b)^T\|_1 = F_\epsilon(a, b)$.

We now analyse F_ϵ. Firstly, for $F_\epsilon((x_1, x_2)) \leq F_\epsilon(0, 1 - \lambda/2) = \lambda(1 - \lambda/4)$ we must have $\lambda|x_1| + |x_2| \leq \lambda(1 - \lambda/4) < \lambda$. Thus $x_1 + (1-\epsilon)x_2 - 1 \leq |x_1| + (1-\epsilon)|x_2| - 1 < 0$ with the first inequality an inequality iff $|x_1| = x_1$ and $|x_2| = x_2$. Therefore $F_\epsilon(x_1, x_2) \geq F_\epsilon(|x_1|, |x_2|)$ with equality iff $|x_1|, |x_2| \geq 0$. Thus to minimise $F_\epsilon(x_1, x_2)$ the entries x_1 and x_2 must be non-negative. We can then set $x_1 + x_2 = \alpha$ with $\alpha < 1$ to obtain $F_\epsilon((x_1, x_2)) = (\alpha - \epsilon x_2 - 1)^2 + \lambda\alpha \geq (\alpha - 1)^2 + \lambda\alpha$ with equality iff $x_2 = 0$. We conclude that $x_2 = 0$ and x_1 is non-negative. Finally, it is easy to see that minimisers of $(\alpha - 1)^2 + \lambda\alpha$ for non-negative α occur when $\alpha = 1 - \lambda/2$. Therefore $x_1 = 1 - \lambda/2$ and $x_2 = 0$. Thus $\tilde{x} = (1 - \lambda/2)e_1$, which contradicts the erroneous original assumption that $\tilde{x} \neq (1 - \lambda/2)e_1$.

Now we consider the more general case for α, β not necessarily equal to 1. The aim will be to transform this new lasso problem into the original one.

We prove the result only for $\alpha > \beta$: the other case $\beta > \alpha$ follows by symmetry. Write $A = \alpha \begin{pmatrix} 1 & \beta\alpha^{-1} \end{pmatrix}$. Since $\alpha > \beta$ and both α, β are positive we can write $\beta\alpha^{-1} =$

$(1-\epsilon)$ for some $\epsilon \in (0,1)$. Thus $\alpha A = \begin{pmatrix} 1 & 1-\epsilon \end{pmatrix}$ for some $\epsilon \in (0,1)$. By Lemma 5.12.1, $M^{\mathrm{UL}}(1, A, \lambda) = \alpha^{-1} M^{\mathrm{UL}}(1, \begin{pmatrix} 1 & 1-\epsilon \end{pmatrix}, \alpha^{-1}\lambda)$.

The result for $\alpha > \beta$ follows from the argument above that because $\alpha^{-1}\lambda \leq 2$, the unique solution to $M^{\mathrm{UL}}(y, \begin{pmatrix} 1 & 1-\epsilon \end{pmatrix}, \alpha^{-1}\lambda)$ is $(1-\alpha^{-1}\lambda/2)e_1$.

It remains to show that both ce_1 and ce_2 are minimisers when $\alpha = \beta$, where $c = \frac{2\alpha-\lambda}{2\alpha^2}$. Again, it will suffice to show the first of these two statements, since the proof is analogous for the second. Let $u \in \mathbb{R}^N$. For positive $\beta' < \beta$, set $U' = U'(\beta') = \begin{pmatrix} \alpha & \beta' \end{pmatrix}$. Since ce_1 is the minimiser to $M^{\mathrm{UL}}(1, U, \lambda)$ by (5.12.3), we see that $\|U'u - y^1\|^2 + \lambda\|u\|_1 \geq \|U'ce_1 - y^1\|^2 + \lambda c\|e_1\|_1$. Noting that $\|\cdot - y\|_2^2$ is continuous and that $U' \to U$ as $\beta' \uparrow \beta$, we have $\|Uu - y^1\|^2 + \lambda\|u\|_1 \geq \|Uce_1 - y^1\|^2 + \lambda\|ce_1\|_1$. Since u was arbitrary we conclude that ce_1 is indeed a minimiser. As previously stated, the second statement that ce_2 is a minimiser is analogous by instead considering positive $\alpha' < \alpha$, defining U' instead to be $U' = \begin{pmatrix} \alpha' & \beta \end{pmatrix}$ and sending $\alpha' \uparrow \alpha$. □

Proof of Theorem 4.3.1. For both parts, we will use the same proof as the one in Theorem 4.1.3. Since the proofs are mostly identical with the exception of the discussion on condition numbers, we only list the changes necessary in this new setting.

Part (i) Instead of appealing to Lemma 5.5.1, we now use Lemma 5.12.2. We will now have $S^1 := \Xi(\iota_n^1) = \{(1-\lambda/2)e_2\}$, $S^2 := \Xi(\iota_n^2) = \{(1-\lambda/2)e_1\}$. This changes κ to $\kappa = \frac{2-\lambda}{\sqrt{2}}$.

For $\iota \in \Omega$. The bound on $\|\iota\|$ is identical, and C_{FP} is not relevant for the lasso problems. Our argument for $\mathrm{cond}(\Xi_{\mathrm{UL}})$ is similar to the proof of Theorem 4.1.3. Again, we see that if $\hat{\iota} = (\hat{y}, \hat{A})$ is such that $\iota + \hat{\iota} \in \tilde{\Omega}_\nu$ then $\hat{y} = 0$. We consider two cases: $\iota = \iota_n^1$, and $\iota = \iota^0$. We set $\hat{\iota} = (\hat{y}, \hat{A})$ and $\iota = (y, A)$. As before, if ϵ is sufficiently small, if $\|\hat{A}\|_2 \leq \epsilon$ then $(A+\hat{A})_{1,1} < (A+\hat{A})_{1,2}$. Additionally, once again choosing sufficiently small $\epsilon > 0$, we get $(\hat{A}_{1,1} + 1) \geq \lambda/2$ since $\lambda < 2$. For such $\hat{\iota}$, we have by Lemma 5.12.2 that $\Xi_{\mathrm{UL}}(\iota + \hat{\iota}) = (2\beta - \lambda)/(2\beta^2)$ where $\beta = 1 + \hat{A}_{1,2}$ and $\Xi_{\mathrm{UL}}(\iota) = (2-\lambda)/2e_2$.

Thus

$$\begin{aligned} \mathrm{dist}(\Xi_{\mathrm{UL}}(\iota+\hat{\iota}), \Xi_{\mathrm{UL}}(\iota)) = \left| \frac{2\beta-\lambda}{2\beta^2} - \frac{2-\lambda}{2} \right| &= \left| \frac{2\beta(1-\beta) - \lambda(1-\beta^2)}{2\beta^2} \right| \\ &\leq \left| \frac{\epsilon\,[2\beta - \lambda(1+\beta)]}{2\beta^2} \right| \end{aligned}$$

and we conclude that for such ι (since $\beta \to 1$ as $\epsilon \downarrow 0$),

$$\lim_{\epsilon \downarrow 0} \sup_{\iota+\hat{\iota} \in \tilde{\Omega}_\nu, 0 < \|\hat{\iota}\| \leq \epsilon} \frac{\mathrm{dist}(\Xi_{\mathrm{LP}}(\iota+\hat{\iota}), \Xi_{\mathrm{LP}}(\iota))}{\|\hat{\iota}\|} \leq 1 - \lambda$$

and the same argument shows that the same result if $\iota = \iota_n^2$ for some n.

The last case to consider is $\iota = \iota^0$. This argument is identical to Theorem 4.1.3, except that as above we have that $(A+\hat{A})_{1,1} \vee (A+\hat{A})_{1,2} \geq \lambda/2$ for $\hat{A}$ sufficiently small since $A_{1,1}, A_{1,2} = 1$ and $\lambda < 2$. We conclude as above $\mathrm{Cond}(\Xi) \leq 1 - \lambda$.

Part (ii) We will now have $\Xi(\iota_n^1) = \{\tau e_2\} = S^1$, $\Xi(\iota_n^2) = \{\tau e_1\} = S^2$. This changes κ to $\kappa = \sqrt{2}\tau$. The argument for condition is as in part (i) except that now the result

that $(A+\hat{A}) \vee (A+\hat{A})_{1,2} \geq \lambda/2$ for sufficiently small $\hat{A}$ is replaced by the argument that $(A+\hat{A})_{1,1}\tau \vee (A+\hat{A})_{1,2}\tau \leq 1$ since $A_{1,1}, A_{1,2} \leq 1$ and $\tau < 1$. Additionally, we now get that $\Xi_{\mathrm{CL}}(\iota+\hat{\iota}) = \Xi_{\mathrm{CL}}(\iota) = \tau e_1$ or $\Xi_{\mathrm{CL}}(\iota+\hat{\iota}) = \Xi_{\mathrm{CL}}(\iota) = \tau e_2$ depending on the choice of ι provided that $\hat{\iota}$ is sufficiently small. The conclusion then becomes that $\mathrm{cond}(\Xi_{\mathrm{CL}}) = 0$, which completes the proof of the boundedness of each of the condition numbers. □

5.13 Proof of Theorem 4.3.2

We prove the two parts separately. First we show the negative result for unconstrained lasso:

Theorem 4.3.2, part 1. As with the proof of Theorem 4.2.14, the strategy and wording is almost verbatim the same as the proof of Theorem 4.2.3. This time, we replace every mention of the input $(y, U+UE_n^j)$ with $(y, U+UE_n^j, \lambda)$ and every mention of (y,U) with (y,U,λ) in the proof of Theorem 4.2.3. The proof of Theorem 4.3.2, part 1 is otherwise identical to the proof of Theorem 4.2.3. □

Next, we prove the positive result for constrained lasso.

Theorem 4.3.2, part 2. We choose the primary set Ω to be all (y, U', τ) of the form $y = 1, U' = \begin{pmatrix} a & b \end{pmatrix}, \tau = 3$ where $a \in (1/2, 1+1/2)$ and $b \in (-1/2, 1/2)$. Set $U = \begin{pmatrix} 1 & 0 \end{pmatrix}$ and $\iota = (1, U, 3)$. Next, we prove that each of Properties (a) to (c) hold for this choice of Ω. For (a), we note that if $x = (1,\alpha)^T$ with $\alpha \in [0,2]$, $Ux = y$ and $\|x\|_1 = 1+\alpha \leq 3$. Therefore there are infinitely many distinct minimisers of the optimisation problem. To see that (b) is true, note that if D is a negative semidefinite diagonal matrix with $D < 1/2$, $U + UD = (1+\beta, 0)$ where β is some number in $[0, \frac{1}{2})$. Thus $(1, U+UD, 3) \in \Omega_4$. To establish (c) and show that $\{\Xi_4, \Omega_4, \mathcal{M}_4, \Lambda_4\} \in \Delta_1^A$, note that for any input $\iota' = (1, U', 3)$ with $U' = \begin{pmatrix} a & b \end{pmatrix}$ and $a \in (1/2, 1+1/2)$, the vector $x' = (1/a, 0)^T \in \Xi_4(I)$ is a solution, since $\|x'\|_1 \leq 2$ and $U'x' = 1$. We will produce a family of algorithms Γ^n such that given any Δ_1 information on ι', $\|\Gamma^n(\iota') - x'\|_2 \leq 2^{-n}$. To construct Γ^n, we read the first entry of U to precision $\hat{\epsilon} \geq 0$, where $\hat{\epsilon} \leq \min(3 \cdot 2^{-n-3}, 1/4)$. This gives us a number $\hat{a}$, and we output the vector $(1/\hat{a}, 0)^T$, which can be done with one arithmetic operation. It is clear that Γ^n has bounded minimum runtime from $T(\Gamma^n, 2^{-n}) < \log_2(3 * 2^{-n-3}) \leq n-2 < \infty$. Finally,

$$\|(1/\hat{a}, 0)^T - x'\|_2 = \frac{|\hat{a}-a|}{|a\hat{a}|} \leq \frac{\hat{\epsilon}}{|a(a+\hat{\epsilon})|} \leq \frac{2\hat{\epsilon}}{|3a^2|} \leq \frac{8\hat{\epsilon}}{3} \leq 2^{-n}.$$

□

5.14 Proof of Theorem 4.3.4

The proof of Theorem 4.3.4 will make use of Lemma 5.12.2.

Proof of Theorem 4.3.4. We choose $\mathbb{P}_M$ so that a random $\mathbb{P}_M$ matrix is such that both entries are distributed according to the uniform distribution on $(1, 1 + 4^{-M-2})$. It is clear that such a probability measure is absolutely continuous with respect to the lebesgue measure since it has a density function. Moreover with probability 1 the entries of a $\mathbb{P}_M$ matrix are bounded by $1 + 4^{-M-2} < 2$. Next, we choose $y = 1$. For a given Ω, we denote the set of all $U \in \mathbb{R}^{1\times 2}_{\|\cdot\|_{\max}\leq 2}$ such that $(y, U, \lambda) \in \Omega$ by S. We assume that S is lebesgue measurable - if not 2 holds trivially. There are three cases to consider:

(a) For all $U \in S$, $U_{1,1} \geq U_{1,2}$

(b) For all $U \in S$, $U_{1,1} < U_{1,2}$

(c) There exist $U^1 \in S$ and $U^2 \in S$ such that $U^1_{1,1} > U^1_{1,2}$ and $U^2_{1,1} < U^2_{2,2}$.

For either case (a) or case (b), we will claim that $\mathbb{P}_M(S) \leq 1/2$. Let us start with case (a). Let S^+ be the set of all $U \in \mathbb{R}^{1\times 2}_{\|\cdot\|_{\max}\leq 2}$ with $U_{1,1} \geq U_{1,2}$. It is clear that $S \subseteq S^+$. Additionally, $\mathbb{P}_M(S^+) = 1/2$ since

$$\begin{aligned}\mathbb{P}_M(S^+) &= 4^{2M+4}\int_1^{1+4^{-M-2}}\int_{x_2}^{1+4^{-M-2}} d\mathcal{L}(x_1)d\mathcal{L}(x_2)\\ &= 4^{2M+2}\int_1^{1+4^{-M-2}}(1 + 4^{-M-2} - x_2)\, d\mathcal{L}(x_2) = \frac{1}{2}\end{aligned}$$

Thus $\mathbb{P}_M(S) \leq 1/2$. The argument for (b) is the same with S^+ replaced by the set S^- (defined to be the set of all U such that $U_{1,1} < U_{1,2}$). Therefore for either case (a) or (b) we have $\mathbb{P}_M(S) \leq 1/2$ and therefore 2 holds.

The only remaining case is (c). For $N = M + 2$ and $n \leq N$, we set $\iota^1_n = (1, U^1, \lambda), \iota^2_n = (1, U^2, \lambda)$ with an aim to using Proposition 2.9.1 part (i). We need to show that Proposition 2.9.1 (a) and (b) hold.

Proposition 2.9.1 condition (a)

We use Lemma 5.12.2. By the distribution of U we can assume that $U^1_{1,1}, U^1_{1,2}, U^2_{1,1}, U^2_{2,2} \in [1, 1 + 4^{-M-2}]$. An application of Lemma 5.12.2 tells us that

$$\Xi(\iota^1_n) = \frac{2U^1_{1,1} - \lambda}{2(U^1_{1,1})^2}e_1, \quad \Xi(\iota^1_n) = \frac{2U^2_{1,2} - \lambda}{2(U^2_{1,2})^2}e_2.$$

Let us examine the function $f(x) = (2x - \lambda)/(2x^2)$. Our task will to be to find a lower bound for f on the interval $[1, 2]$ for $\lambda \in (0, 2)$. Note that since f is continuous on $[1, 2]$ and differentiable everywhere in $[1, 2]$ either f attains its minimum on $[1, 2]$ at some point $\tilde{x}$ with $f'(\tilde{x}) = 0$ or f attains its minimum on the boundary (i.e. either at 0 or at 1).

We have $f'(x) = -1/x^2 + \lambda/x^3$. Thus f attains its critical point at $x = \lambda$ and a simple calculation shows that $f(\lambda) = (2\lambda)^{-1}$. On the other hand, at the boundary point $x = 1$ we have $f(x) = 1 - \lambda/2$ and at the boundary point $x = 2$ we have $f(x) = 1 - \lambda/4$. Clearly for $\lambda \in (0, 2)$, $1 - \lambda/2 < 1 - \lambda/4$.

Furthermore, we claim that $1 - \lambda/2 \leq (2\lambda)^{-1}$. Indeed, suppose otherwise. Then $2\lambda - \lambda^2 > 1$ and so $0 > \lambda^2 - 2\lambda + 1 = (\lambda - 1)^2$. This is a clear contradiction. Thus for $x \in [1, 2]$ we have that $f(x) \geq 1 - \lambda/2$.

Therefore if we define $S^1 = \{te_1 \,|\, t \in [1 - \lambda/2, \infty)\}$ and $S^2 = \{te_2 \,|\, t \in [1 - \lambda/2, \infty)\}$ then $\Xi(\iota_n^1) \subseteq S^1$ and $\Xi(\iota_n^2) \subseteq S^2$. Moreover, with $\kappa = (1 - \lambda/2)\sqrt{2}$ we have $\inf_{x_1 \in S^1, x_2 \in S^2} d_{\mathcal{M}}(x_1, x_2) \geq \kappa$. This concludes the proof that Proposition 2.9.1 condition (a) holds.

Proposition 2.9.1 condition (b)

This part of the proof is fairly straightforward. Let $f \in \Lambda$ and $i \in \{1, 2\}$. By the assumption that $U_{1,1}^1, U_{1,2}^1, U_{1,1}^2, U_{2,2}^2 \in [1, 1 + 4^{-M-2}]$, if $f : \Omega \to \mathbb{R}$ is such that for each $\iota = (y, U, \lambda) \in \Omega$ we have $f(\iota) = U_{j,k}$ set $c_f = 1$. Then $f(\iota_n^i) - c_f \in [0, 4^{-M-2}]$. Hence $|f(\iota_n^i) - c_f| \leq 1/4^N \leq 1/4^n$. If instead $f(\iota) = y = 1$ we set $c_f = 1$ and so $|f(\iota_n^i) - c_f| = 0$. Finally, if $f(\iota) = \lambda$ we set $c_f = \lambda$ and again see that $|f(\iota_n^i) - c_f| = 0$. We have thus shown that Proposition 2.9.1 condition (b) holds.

Since we have now shown both of the requirements for Proposition 2.9.1 the result follows immediately.

□

5.15 Proof of Theorem 4.3.7

Our argument will be split into three main ideas. Firstly, we need to approximate the modulus of continuity for lasso. Next, we need to establish stopping criteria for primal convergent algorithms so that we can get an approximate solution on perturbed input data. Finally, we need to combine these points to obtain the Δ_1^A family of algorithms.

Proposition 5.15.1. *Let $A \in \mathbb{R}^{m \times N}$ and $S \subseteq \{1, 2, \ldots N\}$. Suppose that $y = A\xi$ where $\xi \in \mathbb{R}^{m \times N}$ has $supp(\xi) \subseteq S$. Let $v \in M^{UL}(y, A, \lambda)$. We assume that the matrix A satisfies the following conditions:*

(i) $\|A_{S^c}^* A_S (A_S^* A_S)^{-1} sgn(v_S)\|_\infty < 1 - \alpha_1$ *for some fixed* $\alpha_1 > 0$.

(ii) $\min(\text{Spec}(A_S^* A_S)) \geq \alpha_2$ *for some fixed* $\alpha_2 > 0$, *where* Spec *denotes the spectrum of a matrix.*

Fix h_1, h_2 and h_3 with $h_1, h_2, h_3 \geq 0$. If $\tilde{y}$ and $\tilde{A}$ are such that $\|A - \tilde{A}\|_2 \leq h_1$ $\|y - \tilde{y}\|_2 \leq h_2$ and the vector w satisfies $\|\tilde{A}(v + w) - \tilde{y}\|_2^2 + \lambda\|v + w\|_1 \leq h_3 + \min_{x \in \mathbb{R}^N} \|\tilde{A}x - \tilde{y}\|_2^2 + \lambda\|x\|_1$ then

$$\|w\|_2 \leq \alpha_2^{-1/2} \left(\sqrt{\gamma_4 + \|A\|_2^2 (\lambda\alpha_1)^{-2} (\gamma_4)^2} + \|A\|_2 (\lambda\alpha_1)^{-1} \gamma_4 \right) + (\lambda\alpha_1)^{-1} \gamma_4,$$

where we have

$$\begin{aligned}
\gamma_1 &:= \lambda^{-1} \left(h_3 + (h_1\|\xi\|_2 + h_2)^2 + \lambda\|\xi\|_1 \right) \\
\gamma_2 &:= 2\gamma_1 \left(h_2\|A\|_2 + h_1(\|A\|_2\gamma_1 + \|A\|_2\|\xi\|_2 + h_2) \right) \\
\gamma_3 &:= 2\|A\|_2(\|\xi\|_1 + \|\xi\|_2)(h_1\|\xi\|_1 + h_2) + (h_1\|\xi\|_1 + h_2)^2 \\
\gamma_4 &:= \gamma_2 + \gamma_3 + h_3.
\end{aligned}$$

Proof. We will estimate $\|w_S\|_2$ and $\|w_{S^c}\|_2$, where the key to estimating the former is through bounding $\|A_S w_S\|_2$, as assumption (ii) then easily can be applied. To obtain the estimates we first need the following claim.

Claim: We claim that $\|Aw\|_2^2 + \lambda\alpha_1\|w_{S^c}\|_1 \leq \gamma_4$. Note that if the claim holds we have that $\|w_{S^c}\|_2 \leq \|w_{S^c}\|_1 \leq (\lambda\alpha_1)^{-1}(\gamma_4)$ and $\|Aw\|_2^2 \leq \gamma_4$. Consequently

$$\begin{aligned}\gamma_4 &\geq \|A_S w_S\|_2^2 + 2\langle A_S w_S, A_{S^c} w_{S^c}\rangle + \|A_{S^c} w_{S^c}\|_2^2 \\ &\geq \|A_S w_S\|_2^2 - 2\|A_S w_S\|_2 \|A\|_2 (\lambda\alpha_1)^{-1}(\gamma_4) \\ &\geq (\|A_S w_S\|_2 - \|A\|_2(\lambda\alpha_1)^{-1}(\gamma_4))^2 - \|A\|_2^2(\lambda\alpha_1)^{-2}(\gamma_4)^2\end{aligned}$$

so that $\sqrt{\gamma_4 + \|A\|_2^2(\lambda\alpha_1)^{-2}(\gamma_4)^2} + \|A\|_2(\lambda\alpha_1)^{-1}\gamma_4 \geq \|A_S w_S\|_2$. Now from assumption (ii) on A, we have $\|A_S w_S\|_2^2 \geq \alpha_2\|w_S\|_2^2$. The result follows by observing that $\|w\|_2 \leq \|w_S\|_2 + \|w_{S^c}\|_2$.

Hence, to complete the proof we only need to prove the claim. The conditions on A imply that $\mathrm{supp}(v) \subseteq S$ (this is proven in [123] as a combination of Lemma 2 and Lemma 3). Thus (since $v = M^{\mathrm{UL}}(y, A, \lambda)$) we have $2A_S^*(Av - y) + \lambda\mathrm{sgn}(v_S) = 0$. Since $y = A\xi$, and ξ has support S, this implies that $2(v - \xi) = -\lambda(A_S^* A_S)^{-1}\mathrm{sgn}(v_S)$, so that $2Av - y = -\lambda A_S(A_S^* A_S)^{-1}\mathrm{sgn}(v_S)$. We thus have

$$\begin{aligned}2\langle Aw, Av - y\rangle &= 2\left(\langle A_S w_S, Av - y\rangle + \langle A_{S^c} w_{S^c}, Av - y\rangle\right) \\ &= -\lambda\left(\langle w_S, \mathrm{sgn}(v_S)\rangle + \langle w_{S^c}, A_{S^c}^* A_S (A_S^* A_S)^{-1}\mathrm{sgn}(v_S)\rangle\right) \\ &\geq -\lambda\langle w_S, \mathrm{sgn}(v_S)\rangle + \lambda(\alpha - 1)\|w_{S^c}\|_1. \qquad (5.15.1)\end{aligned}$$

To simplify the expressions we will throughout the argument use, for $T \in \mathbb{R}^{m\times N}$, $\eta_1 \in \mathbb{R}^N$ and $\eta_2 \in \mathbb{R}^m$, $\Psi(T, \eta_1, \eta_2) = \|T\eta_1 - \eta_2\|_2^2$. We will now obtain a simple bound on $\|v + w\|_1$ (and thus $\|v + w\|_2$). Let $\eta = v + w$. We must have

$$\begin{aligned}\lambda\|\eta\|_1 &\leq h_3 + \Psi(\tilde{A}, \xi, \tilde{y}) + \lambda\|\xi\|_1 \leq h_3 + \Psi(\tilde{A} - A, \xi, \tilde{y} - y) + \lambda\|\xi\|_1 \\ &\leq h_3 + (h_1\|\xi\|_2 + h_2)^2 + \lambda\|\xi\|_1,\end{aligned}$$

so that $\|\eta\|_2, \|\eta\|_1 \leq \gamma_1$. Similarly, $\lambda\|v\|_1 \leq \Psi(A, \xi, y) + \lambda\|\xi\|_1 = \lambda\|\xi\|_1$. Next, we must have

$$\Psi(\tilde{A}, \eta, \tilde{y}) + \lambda\|\eta\|_1 \leq h_3 + \Psi(\tilde{A}, v, \tilde{y}) + \lambda\|v\|_1. \qquad (5.15.2)$$

Working with the left hand side of (5.15.2), we must have $\Psi(\tilde{A}, \eta, \tilde{y}) = \Psi(A, \eta, \tilde{y}) + 2\langle(\tilde{A} - A)\eta, A\eta - \tilde{y}\rangle + \Psi(\tilde{A} - A, \eta, 0)$. By some manipulation we get

$$\begin{aligned}2\langle(\tilde{A} - A)\eta, A\eta - \tilde{y}\rangle + \Psi(\tilde{A} - A, \eta, 0) &\geq -2h_1\|\eta\|_2\|A\eta - \tilde{y}\|_2 \\ &\geq -2\gamma_1 h_1(\|A\|_2\gamma_1 + \|A\|_2\|\xi\|_2 + h_2) = -\gamma_2 \qquad (5.15.3)\end{aligned}$$

Similarly, the right hand side of (5.15.2) can be controlled as follows: $\Psi(\tilde{A}, v, \tilde{y}) =$

$\Psi(A, v, y) + 2\langle(\tilde{A} - A)v + y - \tilde{y}, Av - y\rangle + \Psi(\tilde{A} - A), v, \tilde{y} - y)$. Moreover,

$$\begin{aligned} &2\langle(\tilde{A} - A)v + y - \tilde{y}, Av - y\rangle + \Psi(\tilde{A} - A), v, y - \tilde{y}) \\ &\quad \leq 2\|Av - y\|_2(h_1\|v\|_2 + h_2) + (h_1\|v\|_2 + h_2)^2 \\ &\quad \leq 2\|A\|_2(\|v\|_2 + \|\xi\|_2)(h_1\|v\|_2 + h_2) + (h_1\|v\|_2 + h_2)^2 \\ &\quad \leq 2\|A\|_2(\|\xi\|_1 + \|\xi\|_2)(h_1\|\xi\|_1 + h_2) + (h_1\|\xi\|_1 + h_2)^2 = \gamma_3. \end{aligned} \tag{5.15.4}$$

where in the last line we bound $\|v\|_2$ by

$$\lambda\|v\|_2 \leq \lambda\|v\|_1 + \|Av - y\|_2^2 \leq \lambda\|\xi\|_1 + \|A\xi - y\|_2^2 = \lambda\|\xi\|_1.$$

Substituting these calculations using (5.15.3) and (5.15.4) into (5.15.2) yields

$$\lambda\|\eta\|_1 + \Psi(A, \eta, y) \leq \lambda\|v\|_1 + \Psi(A, v, y) + \gamma_3 + \gamma_2 + h_3 = \lambda\|v\|_1 + \Psi(A, v, y) + \gamma_4. \tag{5.15.5}$$

Now using 5.15.1 $\Psi(A, \eta, y) - \Psi(A, v, y) = \|Aw\|_2^2 + 2\langle Aw, Av - y\rangle \geq \|Aw\|_2^2 - \lambda\langle w_S, \operatorname{sgn}(v_S)\rangle + \lambda(\alpha - 1)\|w_{S^c}\|_1$, which combined with (5.15.5) gives

$$\begin{aligned} \gamma_4 &\geq \|Aw\|_2^2 + \lambda\left(\|v + w\|_1 - \langle w_S, \operatorname{sgn}(v_S)\rangle + (\alpha_1 - 1)\|w_{S^c}\|_1 - \|v\|_1\right) \\ &\geq \|Aw\|_2^2 + \lambda\left(\|(v + w)_S\|_1 + \alpha\|w_{S^c}\|_1 - \langle w_S, \operatorname{sgn}(v_S)\rangle - \langle v_S, \operatorname{sgn}(v_S)\rangle\right) \\ &\geq \|Aw\|_2^2 + \lambda\alpha_1\|w_{S^c}\|_1 \end{aligned}$$

where the second line follows because $(v + w)_{S^c} = w_{S^c}$ and the third line because $\langle v_S + w_S, \operatorname{sgn}(v_S)\rangle \leq \|(v + w)_S\|_1$. □

Next, we build a stopping criteria for lasso. In doing so we must identify when a lasso solution is 0. Thus we state two lemmas, both of which are proven in the appendix.

Lemma 5.15.2. *We have the following result:* $\|A^*b\|_\infty \leq \frac{\lambda}{2}$ *if and only if* $0 \in \operatorname{argmin}_{x\in\mathbb{R}^N} \|Ax - y\|^2 + \lambda\|x\|_1$.

Lemma 5.15.3. *Suppose that algorithm produces a sequence* x_n *such that* $\|Ax_n - b\|_2^2 + \lambda\|x_n\|_1 \to \|Ax - b\|_2^2 + \lambda\|x\|_1$ *where the right hand side is minimised at* x^*. *Furthermore suppose that* $\|A^*b\|_\infty > \frac{\lambda}{2}$. *Then*

$$0 \leq \|Ax_n - b\|^2 + \lambda\|x_n\|_1 - \|Ax - b\|_2^2 - \lambda\|x\|_1 \leq G_n$$

where $G_n = \|Ax_n - b\|_2^2 + \lambda\|x_n\|_1 + \min_{m\leq n} \frac{\|p_m\|_2^2}{4} + \langle p_m, b\rangle$ *and* $p_m = \lambda(Ax_m - b)/\|A^*(Ax_m - b)\|_\infty$. *Moreover* $\lim_{n\to\infty} G_n = 0$.

With this continuity estimate in place, we can now move onto the proof of 4.3.7.

Proof of Theorem 4.3.7. For $\epsilon \in (0, 1)$, our first aim will be to find sufficient conditions on h_1, h_2 and h_3 (as functions of $\epsilon, \alpha_1, \alpha_2, \alpha_3$ and α_4) that imply that $\|w\|_2 < \epsilon$ (where w is defined as in proposition 5.15.1). It will prove useful to define additional parameters

$\beta_3 = \max(\alpha_3, 1)$, $\beta_4 = \max(\alpha_4, 1)$. Firstly, let us assume that we can show that γ_4 is less than δ for some $\delta \in (0, 1]$. Then (using the result of proposition 5.15.1):

$$\begin{aligned}\|w\|_2 &\leq \alpha_2^{-1/2}\left(\sqrt{\delta + \|A\|_2^2(\lambda\alpha_1)^{-2}\delta^2} + \|A\|_2(\lambda\alpha_1)^{-1}\delta\right) + (\lambda\alpha_1)^{-1}\delta \\ &\leq \alpha_2^{-1/2}\left(\sqrt{\delta} + 2\|A\|_2(\lambda\alpha_1)^{-1}\delta\right) + (\lambda\alpha_1)^{-1}\sqrt{\delta} \\ &\leq \sqrt{(\alpha_2)^{-1}\delta} + 2\sqrt{(\alpha_2)^{-1}\beta_3\delta^2}(\lambda\alpha_1)^{-1} + \sqrt{\beta_3\delta}(\lambda\alpha_1)^{-1}\end{aligned}$$

where the second line follows from the inequality $\sqrt{x+y} \leq \sqrt{x} + \sqrt{y}$ and the fact that $\delta \leq 1$ implies $\sqrt{\delta} \geq \delta$, and the third line follows from $\beta_3 \geq 1$. For positive x, y the arithmetic geometric mean inequality implies that $2xy \leq x^2 + y^2$ and so $2\sqrt{(\alpha_2)^{-1}\beta_3\delta^2}(\lambda\alpha_1)^{-1} \leq \delta/\alpha_2 + (\lambda\alpha_1)^{-2}\beta_3\delta$.

If we assume additionally that $\sqrt{\delta} < C\epsilon\min(\lambda\alpha_1(\beta_3)^{-1/2}, \sqrt{\alpha_2})$ for some constant $C \in (0, 1)$ that will be determined later, we get

$$\begin{aligned}\|w\|_2 &\leq \sqrt{(\alpha_2)^{-1}\delta} + \delta/\alpha_2 + (\lambda\alpha_1)^{-2}\beta_3\delta + \sqrt{\beta_3\delta}(\lambda\alpha_1)^{-1} \\ &\leq C\epsilon + (C\epsilon)^2 + (C\epsilon)^2 + C\epsilon < 4C\epsilon.\end{aligned}$$

We thus choose $C = 1/4$ to get $\|w\|_2 < \epsilon$. So far, we have shown that it will suffice to ensure that

$$\gamma_4 \leq \delta, \quad \text{provided that } \delta < \frac{\epsilon^2}{16}\min(\frac{\lambda^2\alpha_1^2}{\beta_3}, \alpha_2, 1)$$

to conclude that $\|w\|_2 < \epsilon$. We will now find parameters h_1, h_2, h_3 that ensure that $\gamma_4 \leq \delta$.

Firstly, we have $\gamma_1 \leq \lambda^{-1}(h_3 + (h_1\beta_4 + h_2)^2 + \lambda\beta_4)$. We can use the inequality $(x+y)^2 \leq 2x^2 + 2y^2$ to get $\gamma_1 \leq \lambda^{-1}(h_3 + 2(h_1\beta_4)^2 + 2h_2^2 + \lambda\beta_4)$. Thus if

$$2(h_1\beta_4)^2 \leq \frac{\lambda\beta_4}{2}, \quad 2h_2^2 \leq \frac{\lambda\beta_4}{2}, \quad h_3 \leq \lambda\beta_4 \tag{5.15.6}$$

we get $\gamma_1 \leq 3\beta_4$. Under these conditions, we have

$$\begin{aligned}&\gamma_2 \leq 6\beta_4\left[h_2\beta_3 + h_1(3\beta_3\beta_4 + \beta_3\beta_4 + h_2)\right] \leq 6h_2\beta_3\beta_4 + 24h_1\beta_3\beta_4^2 + 6h_1h_2\beta_4. \\ &\gamma_3 \leq 2\beta_3(2\beta_4)(h_1\beta_4 + h_2) + (h_1\beta_4 + h_2)^2 \leq 4h_1\beta_3\beta_4^2 + 4h_2\beta_3\beta_4 + (h_1\beta_4 + h_2)^2.\end{aligned}$$

Therefore

$$\begin{aligned}\gamma_2 + \gamma_3 &\leq 28h_1\beta_3\beta_4^2 + 10h_2\beta_3\beta_4 + 6h_1h_2\beta_4 + (h_1\beta_4 + h_2)^2 \\ &\leq 28h_1\beta_3\beta_4^2 + 10h_2\beta_3\beta_4 + 5(h_1\beta_4)^2 + 5h_2^2\end{aligned}$$

where again we have used the inequalities $2xy \leq x^2 + y^2$ and $(x+y)^2 \leq 2x^2 + 2y^2$. Consequently to ensure that $\gamma_4 \leq \delta$ it suffices to ensure that

$$h_3 \leq \delta/2, \quad 28h_1\beta_3\beta_4^2 \leq \delta/8, \quad 10h_2\beta_3\beta_4 \leq \delta/8, \quad 5(h_1\beta_4)^2 \leq \delta/8, \quad 5h_2^2 \leq \delta/8 \tag{5.15.7}$$

and the conditions in (5.15.6) are satisfied.

In particular, we choose (noting that $224 = 8 \times 28$)

$$h_1 := \min\left(\frac{\lambda}{4}, \frac{\delta}{224\beta_3\beta_4^2}\right), \quad h_2 := \min\left(\frac{\lambda\beta_4}{4}, \frac{\delta}{80\beta_3\beta_4}\right), \quad h_3 := \min\left(\frac{\delta}{2}, \lambda\beta_4\right). \tag{5.15.8}$$

To see that these values imply (5.15.6), note that $h_1 \leq \frac{\delta}{224\beta_3\beta_4^2}$ implies that $h_1\beta_4 \leq 1$. Thus $2(h_1\beta_4)^2 \leq 2h_1\beta_4 \leq 2\lambda\beta_4/4 = \lambda\beta_4/2$. In the same way, $h_2 \leq \delta/(80\beta_3\beta_4)$ implies that $h_2 \leq 1$ and so $2h_2^2 \leq 2h_2 \leq 2\lambda\beta_4/4 = \lambda\beta_4/2$. The final requirement that $h_3 \leq \lambda\beta_4$ is obvious from the definition of h_3.

Next, we show that the choice of h_1, h_2 and h_3 in (5.15.8) imply (5.15.7). That $h_3 \leq \delta_2$ is obvious from the definition. Again, since $224 = 8 \times 28$, $28h_1\beta_3\beta_4^2 \leq \delta/8$ follows from the definition of h_3. Similarly, $10h_2\beta_3\beta_4 \leq \delta/8$ is immediate. For $5(h_1\beta_4)^2 \leq \delta/8$, we note as before that $h_1\beta_4 \leq 1$. Thus $5(h_1\beta_4)^2 \leq 5h_1\beta_4 \leq 5\delta/(224\beta_3\beta_4) \leq \delta/8$ since $224\beta_3\beta_4 \geq 8 \times 5$. Finally, $h_2 \leq 1$ gives $5h_2^2 \leq 5h_2 \leq 5\delta/(80\beta_3\beta_4) \leq \delta/8$.

We have shown that with the choice of h_1, h_2, h_3 in (5.15.8), if we compute $\tilde{A}, \tilde{y}$ such that $\|\tilde{A} - A\|_2 \leq h_1$, $\|\tilde{y} - y\| \leq h_2$ and we compute $v + w$ such that $\lambda\|v + w\|_1 + \|\tilde{A}(v + w) - \tilde{y}\|_2^2 \leq h_3 + \min_{x\in\mathbb{R}^N} \|\tilde{A}x - \tilde{y}\|_2^2 + \lambda\|x\|_1$ then $\|w\|_2 \leq \epsilon$. This motivates the following algorithm:

Algorithm $\mathsf{LASSO}(\iota,n,\varphi)$

Input: $\iota \in \Omega$, $n \in \mathbb{N}$, $\varphi = (\alpha_1, \alpha_2, \alpha_3, \alpha_4) \in \mathbb{R}^4$.

Output: $\Gamma^n(\iota) \in \mathbb{R}^N$ with $\mathrm{dist}(\Gamma^n(\iota), \Xi(\iota)) \leq 2^{-n}$.

Subroutines: $\mathsf{PCon}(U, y, \lambda, k)$, $\mathsf{FindV}(\iota, \epsilon)$, $\mathsf{FindM}(\iota, \epsilon)$.

We set the parameters $\epsilon := 2^{-n}$ and δ, h_1, h_2, h_3 as in 5.15.8.

Then

$k := 0, m := 0, G_0 := \infty$.

$A = \mathsf{FindM}(\iota, h_1), b = \mathsf{FindV}(\iota, h_2)$.

if $\|A^*b\|_\infty \leq \lambda/2$, set $\Gamma_n(\iota) = 0$.

while $\tilde{G}_k > h_3$

$k = k + 1$

$v^k = \mathsf{PCon}(A, b, \lambda, k)$

$e^k = Av^k - b$

if $A^*e^k \neq 0$ **then**

$p_k = \lambda e^k / \|A^*e^k\|_\infty$

$m = m \wedge (\|p_k\|_2^2/4 + \langle b, e_k\rangle)$, $\tilde{G}_k = \|Av^k - b\|_2^2 + \lambda\|v^k\|_1 + m$

endif

end

$\Gamma_n(\iota) := v^k$

To see that the algorithm is correct, note that each line uses arithmetic operations and if the algorithm terminates then by the previous argument we know that $\text{dist}(\Gamma_n(\iota), \Xi(\iota)) \leq \epsilon$ since $\Gamma(\iota) \in w + \Xi(\iota)$ with $\|w\|_2 \leq \epsilon$. Thus it remains to show that the algorithm does indeed terminate. This will only fail to occur if $\tilde{G}_k > h_3$ for each k. But this contradicts 5.15.3, completing the proof.

□

5.16 Proof of Theorem 4.4.1

Proposition 5.16.1. *Fix $m \in \mathbb{N}, n \in \mathbb{N}, A \in \mathbb{R}^{N \times N_0}, B \in \mathbb{R}^{m \times N}$ and $w \in \mathbb{R}^N$. Suppose that $R = \{x^q, x^{q+1}, x^{q+2}, \ldots, x^{q+r-1}\}$ is a set such that $|R|$ is divisible by $N+1$ and the x^k are each vectors in $\mathbb{R}^{N_0}$ such that such that the sequence $\{x_1^k\}_{k=q}^{q+r-1}$ is strictly increasing and $x_j^k = 0$ for $j > 1$. Then there is a matrix $C \in \mathbb{R}^{m \times 2}$ a vector $v \in \mathbb{R}^m$ and a set $S \subseteq R$ with $|S| \geq R/(N+1), S = \{x^s, x^{s+1}, \ldots, x^{s+t-1}\}$ such that $B\rho(Ax + w) = Cx + v$ for all $x \in S$.*

Proof. Write $B = (b_{j,k})_{j=1,k=1}^{j=m,k=N}$, $A = (a_{j,k})_{j=1,k=1}^{j=N,k=2}$. We claim that there are at most $N+1$ entries in the set

$$SS = \{(\text{sgn}(a_{1,1}x_1 + w_1), \text{sgn}(a_{2,1}x_1 + w_2), \ldots, \text{sgn}(a_{N,1}x_1 + w_N)) \mid x \in S\}.$$

To see this, note that if we allow x_1 to vary over $\mathbb{R}$ then each of the lines $y = a_{1,1}x_1 + w_1, y = a_{2,1}x_1 + w_2, \ldots y = a_{N,1}x_1 + w_N$ intersect the line $y = 0$ at most once. Between each of the (at most N) intersections (allowing x_1 to vary) the vector $(\text{sgn}(a_{1,1}x_1 + w_1), \text{sgn}(a_{2,1}x_1 + w_2), \ldots, \text{sgn}(a_{N,1}x_1 + w_N))$ is constant. The proof is complete by noticing that any line divided into at most N intersections has at most $N+1$ regions between intersections.

We can now define S: by the pigeonhole principle, there exists a subset of R with cardinality at least $|R|/(N+1)$ such that the vector

$$\text{sgn}(a_{\cdot,1}x_1 + w) = (\text{sgn}(a_{1,1}x_1 + w_1), \text{sgn}(a_{2,1}x_1 + w_2), \ldots, \text{sgn}(a_{N,1}x_1 + w_N))$$

is constant over x in this subset. Let S be the maximal (in cardinality) such subset. The first requirement in the construction follows from the fact that $|S| \geq |R|/(N+1)$.

To see that for some s and t, $S = \{x^s, x^{s+1}, \ldots, x^{s+t-1}\}$, suppose otherwise. Then there are j_1, k_1 such that $j_1 + 1 < k_1$, $x^{j_1}, x^{k_1} \in S$ and $x^{j_1+1} \notin S$. But then for some l we must have $\text{sgn}(a_{l,1}x_1^{j_1} + w_1) = \text{sgn}(a_{l,1}x_1^{k_1} + w_1) \neq \text{sgn}(a_{l,1}x_1^{j_1+1} + w_1)$. However, because $\{x^j\}_{j=j_1}^{k_1}$ is an increasing sequence, we see that if $a_{l,1} \geq 0$ then $a_{l,1}x_1^j + w_1 \leq a_{l,1}x_1^{j+1} + w_1 \leq a_{l,1}x_1^k + w_1$ and similarly if $a_{l,1} < 0$ then $a_{l,1}x_1^j + w_1 > a_{l,1}x_1^{j+1} + w_1 > a_{l,1}x_1^k + w_1$ which is a contradiction. Thus the second requirement is satisfied and the construction of S is complete.

We now show how to construct C and v. For any $x \in S$ we have $x_2 = x_3 = \cdots = x_{N_0} = 0$ and so for such x the ith row of $B\rho(Ax+w)$ is given by $\sum_{j=1}^N b_{i,j}\rho(a_{j,1}x_1 + w_j)$. Since $\text{sgn}(a_{j,1}x_1 + w_j)$ is constant over $x \in S^{i+1}$, we must have that for each j either $\rho(a_{j,1}x_1 + w_j) = 0$ or $\rho(a_{j,1}x_1 + w_j) = a_{j,1}x_1 + w_j$ for all such x. In the former case,

we define $d_{i,j} = 0, y_{i,j} = 0$ and in the latter case we define $d_{i,j} = b_{i,j}a_{j,1}, y_{i,j} = b_{i,j}w_j$. Therefore, by definition, the ith row of $B\rho(Ax+w)$ is given by $\sum_{j=1}^{N} d_{i,j}x_1 + y_{i,j}$. Now we define the matrix $C = (c_{i,j})_{i=1,j=1}^{i=m,j=N_0}$ and the vector v by

$$c_{i,1} = \sum_{j=1}^{N} d_{i,j},\ c_{i,j} = 0 \text{ for } j > 1 \quad \text{and} \quad v_i = \sum_{j=1}^{N} y_{i,j}$$

Then the ith row of $B\rho(Ax + w)$ is $\sum_{j=1}^{N} d_{i,j}x_1 + y_{i,j} = c_{i,1}x_1 + c_{i,2}x_2 + v_i$ which is exactly the ith row of $Cx + v$. □

Proof of Theorem 4.4.1. Let $a \in [1/2, 1]$ and define $f_a : \mathbb{R}^{N_0} \to \{0, 1\}$ by $f_a(x) = 1$ if $\lceil a/x_1 \rceil$ is an odd integer and $f_a(x) = 0$ otherwise. Note that this definition of f_a yields uncountably many unique classification functions. As with the other non-computability results, we will appeal to Proposition 2.9.1. To do that we start by defining, for $\kappa \in [1/4, 3/4]$, $\mathcal{T}_0^\kappa := \{x^1, x^2, \ldots, x^K\}$, where $x^k = (a(k + 1 - \kappa)^{-1}, 0, 0, \ldots, 0) \in [0, 1]^{N_0}$. Define also for any $\delta > 0$, $\mathcal{T}_\delta^\kappa = \{x^{1,\delta}, x^{2,\delta}, \ldots, x^{K,\delta}\}$, where $x^{k,\delta} = (a(k + 1 - \kappa)^{-1}, \delta, 0, 0, \ldots, 0) \in [0, 1]^{N_0}$ if k is even and $x^{k,\delta} = x^k$ otherwise. Note that to assure that $\mathcal{T}_0^\kappa, \mathcal{T}_\delta^\kappa \in \mathcal{S}_{\varepsilon(K)}^{f_a}$ we have to check that

$$\min_{x,y \in \mathcal{T}_0^\kappa, x \neq y} \|x - y\|_\infty \geq \varepsilon(K) \text{ and for } x \in \mathcal{T}_0^\kappa, f_a(x + y) = f_a(x) \text{ when } \|y\|_\infty < \varepsilon(K),$$

and similarly for $\mathcal{T}_\delta^\kappa$. Let us start with $\mathcal{T}_0^\kappa$. For the first part, note that for distinct $x^i, x^j \in \mathcal{T}_0^\kappa$ we have

$$\|x^i - x^j\|_\infty = \left| \frac{a}{i + 1 - \kappa} - \frac{a}{j + 1 - \kappa} \right| = \frac{|a(j - i)|}{(i + 1 - \kappa)(j + 1 - \kappa)} \geq \frac{1}{2(K + 1 - \kappa)(K - \kappa)}$$

since $a|j - i| \geq a \geq 1/2$ and the condition that $i, j \leq K$ with at least one bounded by $K - 1$ implies that $(i + 1 - \kappa)^{-1}(j + 1 - \kappa)^{-1} \geq (K + 1 - \kappa)^{-1}(K - \kappa)^{-1}$. Since $\kappa \geq 1/4$, we get $\|x^i - x^j\|_\infty \geq [2(K + 1 - 1/4)(K - 1/4)]^{-1} \geq \varepsilon(K)$.

Next, let us show the second part (i.e. that $f_a(x + y) = f_a(x)$ whenever $x \in T_0^\kappa$ and y is such that $\|y\|_\infty < \varepsilon(K)$). Let $x = x^k$ for some $k \in \{1, 2, \ldots, K\}$. We have

$$\frac{a(1 - \kappa)}{(k + 1 - \kappa)k} > \frac{1}{(4K + 3)(2K + 2)} \geq y_1 \geq \frac{-1}{(4K + 3)(2K + 2)} \geq \frac{-a\kappa}{(k + 1 - \kappa)(k + 1)}.$$

We claim this implies that $a(x_1^k + y_1)^{-1} \in (k, k + 1]$. For the upper bound, note that

$$\frac{y_1}{a} \geq \frac{-\kappa}{(k + 1 - \kappa)(k + 1)} = \frac{1}{k + 1} - \frac{1}{k + 1 - \kappa} = \frac{1}{k + 1} - \frac{x_1}{a}.$$

Similarly, for the lower bound, we have

$$\frac{y_1}{a} < \frac{1 - \kappa}{k(k + 1 - \kappa)} = k^{-1}\left(\frac{k + 1 - \kappa}{k + 1 - \kappa} - \frac{k}{k + 1 - \kappa} \right) = \frac{1}{k} - \frac{x_1}{a}.$$

Therefore $\lceil a/(x_1^k + y_1)\rceil = k+1$. Since this applies even when $y = 0$, we have, $f_a(x^k + y) = f_a(x^k) = 1$ if and only if $k+1$ is odd for y with $\|y\|_\infty < \varepsilon(K)$ which is what we wanted to show. The argument for $\mathcal{T}_\delta^\kappa$ is identical.

We have now constructed an uncountable family of training sets $\mathcal{T}_0^\kappa, \mathcal{T}_\delta^\kappa$. Let

$$\phi_0 \in \operatorname*{argmin}_{\phi\in\mathcal{NN}_{\mathbf{N},L,d}} C(v^1, w^1), \quad v_j^1 = \phi(x^j), \quad w_j^1 = f(x^j), \quad 1 \le j \le K, \tag{5.16.1}$$

$$\phi_\delta \in \operatorname*{argmin}_{\phi\in\mathcal{NN}_{\mathbf{N},L,d}} C(v^2, w^2), \quad v_j^2 = \phi(x^{j,\delta}), \quad w_j^2 = f(x^{j,\delta}), \quad 1 \le j \le K. \tag{5.16.2}$$

We now make three claims. The first relates to ϕ_δ.

Claim I: $\phi_\delta(x^{k,\delta}) = f(x^{k,\delta})$ for all $k \in \{1,\dots,K\}$. Indeed, to see why this is the case we start by defining the neural network

$$\tilde{\phi} = W^L \rho W^{L-1} \rho W^{L-2} \dots \rho W^1$$

where $W^l x = A^l x + b^l$ and $A^l \in \mathbb{R}^{N_l \times N_{l-1}}$, $b^l \in \mathbb{R}^{N_l}$ are defined as follows: let $A^1_{1,1} = 0$, $A^1_{1,2} = \delta^{-1}$ and $A^1_{i,j} = 0$ otherwise, $A^l_{1,1} = 1$ for $l > 1$ and $A^l_{i,j} = 0$ otherwise, and $b^l = 0$ for every l. Clearly

$$W^1 x^{k,\delta} = \begin{cases} e_1 \in \mathbb{R}^{N_1} \text{ if } k+1 \text{ is odd} \\ \mathbf{0} \in \mathbb{R}^{N_1} \text{ if } k+1 \text{ is even} \end{cases}$$

and it is therefore easy to see that $\tilde{\phi}(x^{k,\delta}) = 1$ if $k+1$ is odd and 0 otherwise. We have shown already that $f_a(x^{k,\delta}) = 1$ if $k+1$ is odd and $f_a(x^{k,\delta}) = 0$ otherwise so that $\tilde{\phi}(x^{k,\delta}) = f_a(x^{k,\delta})$. Since $C(t,s) = 0$ if $t = s$, we must have that the objective function of (5.16.1) at the minimiser is at most 0. Thus (since C is non-negative), the objective function is exactly 0. However, since $C(t,s) \neq 0$ if $t \neq s$, if there is a k such that $\phi_\delta(x^{k,\delta}) \neq f_a(x^{k,\delta})$ then the objective function evaluated at ϕ_δ is greater than 0, which contradicts the fact that ϕ_δ is defined to be a minimiser.

Next, we shall prove two claims about ϕ_0. Firstly, we claim the following:

Claim II: We now claim that there is a set

$$\mathcal{S} = \{x^s, x^{s+1}, x^{s+2}\} \subset \mathcal{T}_0^\kappa$$

for some $s \in \mathbb{N}$ with $s \le K-2$ such that there is an $M \in \mathbb{R}^{1\times 2}$ and $z \in \mathbb{R}$ such that for all $x \in S$ we have that $\phi_0(x) = Mx + z$. Suppose for the moment that the claim is true (we will prove it later on). We can then show the following claim:

Claim III: $\max_{x\in\mathcal{T}_0^\kappa} |\phi_0(x) - f_a(x)| \ge 1/2$. Indeed, to show the claim, suppose otherwise. Note that $\phi(x^{s+1}) = aM_{1,1}(s+2-\kappa)^{-1} + y_1 \in [\alpha, \beta]$ where

$$\alpha := (aM_{1,1}(s+1-\kappa)^{-1} + y_1) \wedge (aM_{1,1}(s+3-\kappa)^{-1} + y_1) = \phi_0(x^s) \wedge \phi_0(x^{s+2})$$
$$\beta := (aM_{1,1}(s+1-\kappa)^{-1} + y_1) \vee (aM_{1,1}(s+3-\kappa)^{-1} + y_1) = \phi_0(x^s) \vee \phi_0(x^{s+2})$$

since the function g defined by $g(x) := aM_{1,1}x^{-1} + y_1$ is monotonic away from zero.

By our earlier work, $f_a(x^k) = 1$ if k is even and 0 otherwise. We will now consider

two cases: firstly, the case where s is even and secondly the case where s is odd. Suppose that s is even. Then we have $f_a(x^s) = f_a(x^{s+2}) = 1$ and thus we require $\phi_0(x^s), \phi_0(x^{s+2}) > 1/2$. But then $\alpha > 1/2$ and so $\phi_0(x^{s+1}) > 1/2$, contradicting $f_a(x^{s+1}) = 0$.

If instead s is odd, we have $f_a(x^s) = f_a(x^{s+2}) = 0$ and thus we require $\phi_0(x^s), \phi_0(x^{s+2}) < 1/2$. Therefore $\beta < 1/2$ and so $\phi_0(x^{s+1}) < 1/2$, contradicting $f_a(x^{s+1}) = 1$. This completes the proof of Claim III.

Now let

$$\iota_0 = \left\{ \{(\hat{x}^j, f(\hat{x}^j))\}_{j=1}^K, \{\hat{x}^j\}_{j=1}^K \right\}, \qquad \hat{x}^j \in \mathcal{T}_0^\kappa,$$

and for $n \in \mathbb{N}$ we let $\iota_n^1 = \iota_0$ and

$$\iota_n^2 = \left\{ \{(\tilde{x}^j, f(\tilde{x}^j))\}_{j=1}^K, \{\tilde{x}^j\}_{j=1}^K \right\}, \qquad \tilde{x}^j \in \mathcal{T}_{\nu/n}^\kappa.$$

Moreover, define $S_1 = \Xi(\iota_0)$, and $S_2 = \bigcup_{n\in\mathbb{N}} \Xi(\iota_n^2)$. Note that for any $\{\phi_0(\hat{x}^j)\}_{j=1}^K \in S_1$ we have that ϕ_0 satisfies (5.16.1). Thus, by Claim III $\max_{x\in\mathcal{T}_0^\kappa} |\phi_0(x) - f(x)| \geq 1/2$. However, for any $\{\phi_{\nu/n}(\hat{x}^j)\}_{j=1}^K \in S_2$ we have that $\phi_{\nu/n}$ satisfies (5.16.2) with $\delta = \nu/n$. Hence, by Claim I it follows that $\max_{x\in\mathcal{T}_{\nu/n}^\kappa} |\phi_{\nu/n}(x) - f(x)| = 0$. Since $\mathcal{T}_0^\kappa, \mathcal{T}_{\nu/n}^\kappa \in \mathcal{S}_{\varepsilon(K)}^{f_a}$ it follows that $f(\hat{x}^j) = f(\tilde{x}^j)$ for all $\hat{x}^j \in \mathcal{T}_0^\kappa$ and $\tilde{x}^j \in \mathcal{T}_{\nu/n}^\kappa$. Hence,

$$\inf_{\xi_1\in S^1, \xi_2\in S^2} d_{\mathcal{M}}(\xi_1, \xi_2) \geq 1/2.$$

Hence, we are now in the situation where it is clear that that conditions (a) and (b) are satisfied in Proposition 2.9.1. Hence, the theorem follows from (ii) in Proposition 2.9.1.

Thus, to finish the proof we only need to verify the Claim II about the existence of the set $\mathcal{S}$. We show the existence of $\mathcal{S}$ inductively by showing that there are sets $S_l \subset \{x^1, x^2, \ldots, x^K\}$, matrices $M^l \in \mathbb{R}^{N_L \times N_{l-1}}$ and vectors z^l for $l = 1, \ldots, L$ such that

(i) $|\mathcal{S}_l| \geq 3 \times (N_{l-1} + 1) \times \cdots \times (N_{L-1} + 1)$

(ii) $\mathcal{S}_l = \{x^{s_l}, x^{s_l+1}, \ldots, x^{s_l+t_l}\}$ for some $s_l, t_l \in \mathbb{N}$.

(iii) $\phi(x) = W^L \rho W^{L-1} \rho W^{L-2} \ldots W^{l+1} \rho(M(x) + z^l)$ whenever $x \in \mathcal{S}_l$.

The first step of the induction is obvious by taking $M^1 = A^1$ and $z^1 = b^1$. If we assume the existence of S_l, M^l and z^l for some $l < L$, the existence of S_{l+1} is guaranteed by Proposition 5.16.1. Indeed, we apply Proposition 5.16.1 with $B = A^{l+1}, A = M^l, R = S_l$ and $w = z^l$ to obtain some set S_{l+1}, a matrix M^{l+1} and a vector v^{l+1} then we see that $A^{l+1}\rho(M^l x + b) = M^{l+1}x + v^{l+1}$ on $x \in S_{l+1}$ and thus $W^{l+1}\rho(M^l x + b) = M^{l+1}x + z^{l+1}$ where $z^{l+1} = v^{l+1} + b^{l+1}$. This completes the proof of the claim, and we are finally done. □

5.17 Proof of Theorem 4.4.2

Proof. To prove this theorem we stay close to the proof of Theorem 4.4.1. To start with we define the uncountable collection of functions $f_a : \mathbb{R}^{N_0} \to \{0,1\}$ for $a \in [1/2,1]$ by $f_a(x) = 1$ if $\lceil a/x_1 \rceil$ is an odd integer and $f_a(x) = 0$ otherwise. Again, for $\kappa_1 \in [1/4, 3/4]$ and for $\delta > 0$, we define

$$\mathcal{T}_\delta^{\kappa_1} = \{x^{1,\delta}, x^{2,\delta}, \ldots, x^{K,\delta}\}, \quad \mathcal{C}_\delta^{\kappa} = \left\{x^{K+1,\delta}, \ldots, x^{K+M,\delta}\right\}. \tag{5.17.1}$$

where $x^{k,\delta} = (a(k+1-\kappa_1)^{-1}, \delta, 0, 0, \ldots, 0) \in [0,1]^{N_0}$ if k is even and $x^{k,\delta} = (a(k+1-\kappa_1)^{-1}, 0, 0, 0, \ldots, 0)$ otherwise.

We choose $\delta < \epsilon$. By arguing as in the proof of Theorem 4.4.1 we get that $\mathcal{T}_\delta^{\kappa_1}, \mathcal{C}_\delta^{\kappa_1} \in \mathcal{S}^{f_a}_{\varepsilon(K+M)}$. Note that this gives us uncountably many training and classification sets. To pick an arbitrary element in this family we choose $\kappa_1 \in [1/4, 3/4]$. We can now construct $\tilde{\phi}$. As in the proof of Theorem 4.4.1, we set $\phi_\delta = W^L \rho W^{L-1} \rho W^{L-2} \ldots \rho W^1$ where $W^l x = A^l x + b^l$ and $A^l \in \mathbb{R}^{N_l \times N_{l-1}}$, $b^l \in \mathbb{R}^{N_l}$ are as follows: let $A^1_{1,1} = 0$, $A^1_{1,2} = 1/(\delta)$ and $A^1_{i,j} = 0$ otherwise, $A^l_{1,1} = 1$ for $l > 1$ and $A^l_{i,j} = 0$ otherwise, and $b^l = 0$ for every l. Clearly

$$W^1(x^{k,\delta}) = \begin{cases} e_1 \in \mathbb{R}^{N_1} \text{ if } k \text{ is even} \\ \mathbf{0} \in \mathbb{R}^{N_1} \text{ if } k \text{ is odd.} \end{cases} \quad \forall\, k \in \mathbb{N}.$$

Therefore as before we have $\phi_\delta(x) = f_a(x)$ for all $x \in \mathcal{T}_\delta^{\kappa_1} \cup \mathcal{C}_\delta^{\kappa_1}$, and we have thus shown (4.4.1).

We are left with the task of showing the existence of uncountably many $v \in \mathbb{R}^{N_0}$ such that

$$|\tilde{\phi}(v) - f_a(v)| \geq 1/2, \qquad \|v - x\|_\infty \leq \epsilon \text{ for some } x \in \mathcal{T}_\delta^{\kappa_1}.$$

For $\kappa_2 \in [1/4, 3/4]$, we define $v^k = (a(k+1-\kappa_2)^{-1}, 0, \ldots, 0)$. There are arbitrarily many κ_2 sufficiently close to κ_1 so that$\|v^k - x^{k,\delta}\|_\infty \leq \epsilon$, for every k such that $1 \leq k \leq K$. Hence this construction gives us uncountably many candidate v for (4.4.2).

Thus, it it suffices to show that for any $\kappa_2 \in [1/4, 3/4]$ there is an $v \in \mathcal{T}_0^{\kappa_2}$ where

$$T_0^{\kappa_2} := \{v^1, v^2, \ldots, v^K\} \tag{5.17.2}$$

such that $|\tilde{\phi}(v) - f(v)| \geq 1/2$. However, to show this we can argue exactly as in Claim III in the proof of Theorem 4.4.1. In particular, the reasoning there is independent of the initial choice of neural network. □

5.18 Proof of Theorem 4.4.3

Proof. We will use a similar argument as in the proof of Theorem 4.4.2. In particular, to generate the uncountable classification functions we set f_a as in Theorem 4.4.2 and for $\kappa \in [1/4, 3/4]$, we set $\mathcal{T}_1^\kappa = \mathcal{T}_0^\kappa$, where $\mathcal{T}_0^\kappa$ is defined in (5.17.2). Also, define

$\mathcal{T}_1^\kappa = \mathcal{T}_\delta^\kappa$ where $\mathcal{T}_\delta^\kappa$ is defined in (5.17.1), and $\delta < \epsilon$. We continue and define

$$\mathcal{C}_1^\kappa = \left\{x^{K+1}, \dots, x^{K+M}\right\}, \quad x^k = (a(k+1-\kappa_1)^{-1}, 0, \dots, 0)$$

and $\mathcal{C}_2^\kappa = \mathcal{C}_\delta^\kappa$ where $\mathcal{C}_\delta^\kappa$ is defined in (5.17.1). Note that it is clear from the definitions that by choosing δ small enough we have $\mathcal{T}_1 \subset \mathcal{B}_\epsilon^\infty(\mathcal{T}_2)$ and $\mathcal{C}_1 \subset \mathcal{B}_\epsilon^\infty(\mathcal{C}_2)$. It is clear that this gives an uncountable family of non-intersecting training and classification sets fulfilling the requirements of Theorem 4.4.3. To finish up the proof we argue similarly to the proof of Theorem 4.4.1 Claim III to deduce the existence of $v \in \mathcal{T}_1^\kappa$ and $w \in \mathcal{C}_1^\kappa$ with

$$|\tilde{\phi}_1(v) - f_a(v)| \geq 1/2, \qquad |\tilde{\phi}_1(w) - f_a(w)| \geq 1/2.$$

Moreover, by arguing exactly as in the proof of Theorem 4.4.1 Claim I it follows that

$$\tilde{\phi}_2(x) = f_a(x) \quad \forall x \in \mathcal{T}_2^\kappa \cup \mathcal{C}_2^\kappa,$$

which finishes the proof. □

5.19 Proof of Theorem 4.5.1 and Theorem 4.5.2

In this section we assume that $\epsilon = (\epsilon_1, \epsilon_2)$ with $\epsilon_1, \epsilon_2 \geq 0$ and at most one of them non-zero. To prove Theorem 4.5.1 and 4.5.2, we will define the matrices A_ϵ^{TV} and vectors y^{TV} by

$$A_\epsilon^{\mathrm{TV}} : \mathbb{R}^{4\times 4} \to \mathbb{R}^{3\times 4} \text{ by } (A_\epsilon^{\mathrm{TV}} x)_i = V_\epsilon(x)_i, \tag{5.19.1}$$

$$\begin{aligned} V_\epsilon &= \frac{1-\epsilon_1}{\sqrt{2}}\tilde{e}_{1,1} + \frac{1-\epsilon_2}{\sqrt{2}}\tilde{e}_{1,4} + \tilde{e}_{2,2} + \tilde{e}_{3,3} \in \mathbb{R}^{3\times 4}, \quad \tilde{e}_{i,j} = e_i \otimes e_j, \\ y^{\mathrm{TV}} &= \frac{1}{\sqrt{2}} e_1 \otimes \mathbf{1}_4 \in \mathbb{R}^{3\times 4}, \end{aligned} \tag{5.19.2}$$

We will now prove a result on the solution to the TV problem with these inputs.

Lemma 5.19.1. *Let A_ϵ^{TV} and y^{TV} be as above, and for a positive $r < 1/(2+\sqrt{2})$ let $\lambda = r^2$. Let Ξ_{BPTV} and Ξ_{DeblurTV} denote the problem functions of Basis Pursuit and Unconstrained Lasso with parameter λ respectively, with TV regularisation (either isotropic or anisotropic). Then*

$$\Xi_{\mathrm{BPTV}}((A_\epsilon^{\mathrm{TV}}, y^{\mathrm{TV}})) = \begin{cases} x^1 := e_4 \otimes \mathbf{1}_4 \in \mathbb{R}^{4\times 4} \text{ if } \epsilon_1 > 0 \\ x^2 := e_1 \otimes \mathbf{1}_4 \in \mathbb{R}^{4\times 4} \text{ if } \epsilon_2 > 0, \\ \{tx^1 + (1-t)x^2 \,|\, t \in [0,1]\} \text{ if } \epsilon_1 = \epsilon_2 \end{cases}$$

$$\Xi_{\mathrm{DeblurTV}}((A_\epsilon^{\mathrm{TV}}, y^{\mathrm{TV}}, \lambda)) = \begin{cases} \eta(\epsilon_1, \lambda) \otimes \mathbf{1}_4 \in \mathbb{R}^{4\times 4} \text{ if } \epsilon_1 > 0 \\ (P\eta(\epsilon_2, \lambda)) \otimes \mathbf{1}_4 \in \mathbb{R}^{4\times 4} \text{ if } \epsilon_2 > 0, \end{cases}$$

where $\eta(\epsilon_1, \lambda) = (\rho, \rho, \rho, 1 - \lambda - (1-\epsilon_1)\rho)^T$, $\rho = \rho(\epsilon_1) = \lambda(2-\epsilon_1)/4$ and P is the permutation operator reversing the elements. Moreover, when viewed as a map from $\mathbb{R}^{16}$ to $\mathbb{R}^{12}$, $\mathrm{cond}(A_\epsilon^{\mathrm{TV}}(A_\epsilon^{\mathrm{TV}})^*) = 2/(2 - 2\epsilon_1 + \epsilon_1^2)$.

Proof of Lemma 5.19.1. It will suffice to assume that $\epsilon_1 > 0$: the argument for $\epsilon_2 > 0$ is identical by symmetry. We start with Ξ_{BPTV}: it is easy to see that $\|x^1\|_{\mathrm{TV}} = 4$ and $A_\epsilon^{\mathrm{TV}} x^1 = y^{\mathrm{TV}}$. Suppose that $A_\epsilon^{\mathrm{TV}} \tilde{x} = y^{\mathrm{TV}}$ and that $\tilde{x} \neq x^1$. Since $A_\epsilon^{\mathrm{TV}} \tilde{x} = y^{\mathrm{TV}}$, we must have

$$\tilde{x} = e_1 \otimes (\tilde{x}_{1,1}, \tilde{x}_{1,2}, \tilde{x}_{1,3}, \tilde{x}_{1,4}) + e_4 \otimes (\tilde{x}_{4,1}, \tilde{x}_{4,2}, \tilde{x}_{4,3}, \tilde{x}_{4,4}) \in \mathbb{R}^{4\times 4}.$$

Consequently (with either the anisotropic or the isotropic TV norm), $\|\tilde{x}\|_{\mathrm{TV}} \geq \sum_{j=1}^4 (|\tilde{x}_{1,j}| + |\tilde{x}_{4,j}|)$. Additionally, $A_\epsilon^{\mathrm{TV}} \tilde{x} = y^{\mathrm{TV}}$ also implies that $(1-\epsilon_1)\tilde{x}_{1,j} + \tilde{x}_{4,j} = 1$ for $i = 1, \ldots, 4$. Therefore $|\tilde{x}_{1,j}| + |\tilde{x}_{4,j}| > 1$ unless $\tilde{x}_{1,j} = 0$ and $\tilde{x}_{4,j} = 1$ (conditions that would imply $\tilde{x} = x^1$) so that $\|\tilde{x}\|_{\mathrm{TV}} > \|x^1\|_{\mathrm{TV}}$. Therefore $\tilde{x} \notin \operatorname{argmin} \|x\|_{\mathrm{TV}}$ such that $A_\epsilon^{\mathrm{TV}} x = y^{\mathrm{TV}}$. For $\epsilon_1 = \epsilon_2 = 0$, the same argument as before shows that if $\tilde{x}$ is in $\Xi_{\mathrm{BPTV}}((A_\epsilon^{\mathrm{TV}}, y^{\mathrm{TV}}))$ then $\tilde{x}$ must be of the form

$$\tilde{x} = e_1 \otimes (\tilde{x}_{1,1}, \tilde{x}_{1,1}, \tilde{x}_{1,1}, \tilde{x}_{1,1}) + e_4 \otimes (\tilde{x}_{4,1}, \tilde{x}_{4,1}, \tilde{x}_{4,1}, \tilde{x}_{4,1}) \in \mathbb{R}^{4\times 4}.$$

All such vectors have TV norm greater than or equal to 4 with equality iff $\tilde{x}_{1,1} + \tilde{x}_{4,1} = 1$ and both $\tilde{x}_{1,1}$, $\tilde{x}_{4,1}$ are non-zero. Additionally such vectors all satisfy $A_\epsilon^{\mathrm{TV}} \tilde{x} = y^{\mathrm{TV}}$. This completes the proof for $\epsilon_1 = \epsilon_2 = 0$ and basis pursuit denoising.

The proof for Ξ_{DeblurTV} is considerably more involved. Letting $(\hat{x})_j$ denote the jth column of a matrix $\hat{x} \in \mathbb{R}^{m\times n}$ we define $\|(\hat{x})_j\|_{\mathrm{TV}} := \sum_{i=1}^{m-1} |\hat{x}_{i+1,j} - \hat{x}_{i,j}|$. We also define, for vectors $v \in \mathbb{R}^4$, $\Psi^\epsilon(v) := \lambda\|v\|_{\mathrm{TV}} + \|V_\epsilon v - (y^{\mathrm{TV}})_1\|_2^2$. Then, by the definition of A_ϵ^{TV} it follows that $\|A_\epsilon^{\mathrm{TV}} x - y^{\mathrm{TV}}\|_2^2 + \lambda\|x\|_{\mathrm{TV}}^* \geq \|V_\epsilon(x)_j - (y^{\mathrm{TV}})_j\|_2^2 + \sum_{i=1}^4 \lambda\|(x)_j\|_{\mathrm{TV}}$ with equality if and only if $(x)_j$ is constant in j. Since, by the definition, $(y^{\mathrm{TV}})_j$ is constant, we therefore have $\|A_\epsilon^{\mathrm{TV}} x - y^{\mathrm{TV}}\|_2^2 + \lambda\|x\|_{\mathrm{TV}}^* \geq 4 \min_{v\in\mathbb{R}^4} \Psi^\epsilon(v)$ with equality if and only if $(x)_j$ does not change with j and satisfies $(x)_j = \tilde{v}$ for some $\tilde{v} \in \operatorname{argmin}_{v\in\mathbb{R}^4} \Psi^\epsilon(v)$. We now examine the functional Ψ^ϵ. We will claim the following for $\epsilon_1 > 0$, where we use the standard $\wedge$, $\vee$ notation to denote min and max:

1. $|\tilde{v}_2 \wedge \tilde{v}_3|, |\tilde{v}_2 \vee \tilde{v}_3| \leq r$ and $(1-\epsilon_1)\tilde{v}_1 + \tilde{v}_4 \geq 1 - \sqrt{2}r$.
2. $\tilde{v}^1 \leq \tilde{v}_2 \wedge \tilde{v}_3 \leq \tilde{v}_2 \vee \tilde{v}_3 \leq \tilde{v}_4$.
3. $\tilde{v}^2 = \tilde{v}_2 \wedge \tilde{v}_3$ and $\tilde{v}^3 = \tilde{v}_2 \vee \tilde{v}_3$ so that $\tilde{v}_i$ is increasing for $i = 1, 2, \ldots, 4$
4. $\tilde{v}_1 = \tilde{v}_2 = \tilde{v}_3 = \lambda(2-\epsilon_1)/4$ and $\tilde{v}_4 = 1 - \lambda - (1-\epsilon_1)\tilde{v}_1$.

To see (1), note that $\Psi^\epsilon(\tilde{v}) \leq \Psi^\epsilon\left((0,0,0,1)^T\right) = r^2$. Therefore $((1-\epsilon_1)\tilde{v}_1 + \tilde{v}_4 - 1)^2/2 + \tilde{v}_2^2 + \tilde{v}_3^2 \leq r^2$ and the result follows. Next, we will show (2) by contradiction. Assume that $\tilde{v}_1 > \tilde{v}_2 \wedge \tilde{v}_3$. Let $\hat{v} = (\tilde{v}_2 \wedge \tilde{v}_3, \tilde{v}_2 \wedge \tilde{v}_3, \tilde{v}_2 \vee \tilde{v}_3, (1-\epsilon_1)\tilde{v}_1 + \tilde{v}_4 - (1-\epsilon_1)\tilde{v}_2 \wedge \tilde{v}_3)^T$. Then $(V_\epsilon \hat{v})_1 = (V_\epsilon \tilde{v})_1$. Moreover, $(V_\epsilon \hat{v})_2 = \tilde{v}_2 \wedge \tilde{v}_3$ and $(V_\epsilon \hat{v})_3 = \tilde{v}_2 \vee \tilde{v}_3$ so

$$\|V_\epsilon(\hat{v}) - (y^{\mathrm{TV}})_1\|_2^2 - \|V_\epsilon(\tilde{v}) - (y^{\mathrm{TV}})_1\|_2^2 = (\tilde{v}_2 \wedge \tilde{v}_3)^2 + (\tilde{v}_2 \vee \tilde{v}_3)^2 - (v_2^2 + v_3^2) = 0.$$

Observe also that either $\tilde{v}_2 \vee \tilde{v}_3 - \tilde{v}_2 \wedge \tilde{v}_3 = (v^2 - v^3)$ or $\tilde{v}_2 \vee \tilde{v}_3 - \tilde{v}_2 \wedge \tilde{v}_3 = (v^3 - v^2)$.

Thus $|\tilde{v}_2 \vee \tilde{v}_3 - \tilde{v}_2 \wedge \tilde{v}_3| = |v^2 - v^3|$ so

$$\begin{aligned}\frac{\Psi^\epsilon(\tilde{v}) - \Psi^\epsilon(\hat{v})}{\lambda} &= \|\tilde{v}\|_{\mathrm{TV}} - \|\hat{v}\|_{\mathrm{TV}} = |\tilde{v}_1 - \tilde{v}_2| + |\tilde{v}_2 - \tilde{v}_3| + |\tilde{v}_3 - \tilde{v}_4| \\ &- |(1-\epsilon_1)\tilde{v}_1 + \tilde{v}_4 - (1-\epsilon_1)\tilde{v}_2 \wedge \tilde{v}_3 - \tilde{v}_2 \vee \tilde{v}_3| - |\tilde{v}_2 \vee \tilde{v}_3 - \tilde{v}_2 \wedge \tilde{v}_3| \\ &= |\tilde{v}_1 - \tilde{v}_2| + |\tilde{v}_3 - \tilde{v}_4| - |(1-\epsilon)\tilde{v}_1 + \tilde{v}_4 - (1-\epsilon)\tilde{v}_2 \wedge \tilde{v}_3 - \tilde{v}_2 \vee \tilde{v}_3|.\end{aligned} \tag{5.19.3}$$

Note that by (1) it follows that $(1-\epsilon_1)\tilde{v}_1 + \tilde{v}_4 - (1-\epsilon_1)\tilde{v}_2 \wedge \tilde{v}_3 - \tilde{v}_2 \vee \tilde{v}_3 \geq 1 - \sqrt{2}r - (1-\epsilon_1)r - r > 0$ because $r \leq \frac{1}{\sqrt{2}+2}$. Thus, we can remove the modulus in the last part in the right hand side of the last equation in (5.19.3). Therefore

$$\begin{aligned}\frac{\Psi^\epsilon(\tilde{v}) - \Psi^\epsilon(\hat{v})}{\lambda} &= |\tilde{v}_1 - \tilde{v}_2| + |\tilde{v}_3 - \tilde{v}_4| - [(1-\epsilon_1)\tilde{v}_1 + \tilde{v}_4 - (1-\epsilon_1)\tilde{v}_2 \wedge \tilde{v}_3 - \tilde{v}_2 \vee \tilde{v}_3] \\ &= |\tilde{v}_1 - \tilde{v}_2| + |\tilde{v}_3 - \tilde{v}_4| - [\tilde{v}_1 - \tilde{v}_2 + \tilde{v}_4 - \tilde{v}_3 - \epsilon_1(\tilde{v}_1 - \tilde{v}_2 \wedge \tilde{v}_3)] \geq \epsilon_1(\tilde{v}_1 - \tilde{v}_2 \wedge \tilde{v}_3) > 0.\end{aligned}$$

where the final inequality follows by the assumption that $\tilde{v}^1 > \tilde{v}_2 \wedge \tilde{v}_3$. Thus, $\Psi^\epsilon(\hat{v}) < \Psi^\epsilon(\tilde{v})$, contradicting the minimality of $\Psi^\epsilon(\tilde{v})$.

We conclude that $\tilde{v}_1 \leq |\tilde{v}_2 \wedge \tilde{v}_3| \leq r$. But from the fact that $(1-\epsilon_1)\tilde{v}_1 + \tilde{v}_4 \geq 1 - \sqrt{2}r$, which was established in (1), we get $\tilde{v}_4 > 1 - (1+\sqrt{2})r$. Therefore $\tilde{v}_4 \geq r \geq |\tilde{v}_2 \vee \tilde{v}_3|$ (where the second inequality follows from 1. Result (2) follows.

Again, we argue by contradiction to show (3): if we do not have (3), $\tilde{v}_3 = \tilde{v}_2 \wedge \tilde{v}_3$ and $\tilde{v}_2 = \tilde{v}_2 \vee \tilde{v}_3$ (and $\tilde{v}_2 \vee \tilde{v}_3 > \tilde{v}_2 \wedge \tilde{v}_3$). This time we set $\hat{v} = (\tilde{v}_1, \tilde{v}_2 \wedge \tilde{v}_3, \tilde{v}_2 \vee \tilde{v}_3, \tilde{v}_4)$. Once again, we have $\|V_\epsilon \hat{v} - (y^{\mathrm{TV}})_1\|_2^2 = \|V_\epsilon \tilde{v} - (y^{\mathrm{TV}})_1\|_2^2$. Moreover, we see that (by (2))

$$\begin{aligned}\|\hat{v}\|_{\mathrm{TV}} &= |\tilde{v}_2 \wedge \tilde{v}_3 - \tilde{v}_1| + |\tilde{v}_2 \wedge \tilde{v}_3 - \tilde{v}_2 \vee \tilde{v}_3| + |\tilde{v}_2 \vee \tilde{v}_3 - \tilde{v}_4| \\ &= \tilde{v}_2 \wedge \tilde{v}_3 - \tilde{v}_1 + \tilde{v}_2 \vee \tilde{v}_3 - \tilde{v}_2 \wedge \tilde{v}_3 + \tilde{v}_4 - \tilde{v}_2 \vee \tilde{v}_3 = \tilde{v}_4 - \tilde{v}_1 \\ \|\tilde{v}\|_{\mathrm{TV}} &= |\tilde{v}_2 \vee \tilde{v}_3 - \tilde{v}_1| + |\tilde{v}_2 \wedge \tilde{v}_3 - \tilde{v}_2 \vee \tilde{v}_3| + |\tilde{v}_2 \wedge \tilde{v}_3 - \tilde{v}_4| \\ &= \tilde{v}_2 \vee \tilde{v}_3 - \tilde{v}_1 + \tilde{v}_2 \vee \tilde{v}_3 - \tilde{v}_2 \wedge \tilde{v}_3 + \tilde{v}_4 - \tilde{v}_2 \wedge \tilde{v}_3 = \tilde{v}_4 - \tilde{v}_1 + \tilde{v}_2 \vee \tilde{v}_3 - \tilde{v}_2 \wedge \tilde{v}_3.\end{aligned}$$

So $\|\tilde{v}\|_{\mathrm{TV}} > \|\hat{v}\|_{\mathrm{TV}}$, which is our required contradiction of the minimality of $\Psi^\epsilon(\tilde{v})$.

We can use (3) to get an expression for $\Psi^\epsilon(\tilde{v})$. Indeed,

$$\begin{aligned}\Psi^\epsilon(\tilde{v}) &= \frac{[(1-\epsilon_1)\tilde{v}_1 + \tilde{v}_4 - 1]^2}{2} + \tilde{v}_2^2 + \tilde{v}_3^2 + \lambda(\tilde{v}_4 - \tilde{v}_3 + \tilde{v}_3 - \tilde{v}_2 + \tilde{v}_2 - \tilde{v}_1) \\ &\geq \frac{[(1-\epsilon_1)\tilde{v}_1 + \tilde{v}_4 - 1]^2}{2} + 2\tilde{v}_1^2 + \lambda(\tilde{v}_4 - \tilde{v}_1).\end{aligned}$$

since $\tilde{v}_2, \tilde{v}_3 \geq \tilde{v}_1$, with equality iff $\tilde{v}_2 = \tilde{v}_3 = \tilde{v}_1$. Since this is a lower bound, we must have $\tilde{v}_2 = \tilde{v}_3 = \tilde{v}_1$. Therefore, as a function of $\tilde{v}_1$ and $\tilde{v}_4$, we must have (since $\tilde{v}$ is a minimizer)

$$\frac{\partial}{\partial_{\tilde{v}_1}}\Psi^\epsilon(\tilde{v}) = [(1-\epsilon_1)\tilde{v}_1 + \tilde{v}_4 - 1](1-\epsilon_1) + 4\tilde{v}_1 - \lambda = 0 \tag{5.19.4}$$

$$\frac{\partial}{\partial_{\tilde{v}_4}}\Psi^\epsilon(\tilde{v}) = (1-\epsilon_1)\tilde{v}_1 + \tilde{v}_4 - 1 + \lambda = 0 \tag{5.19.5}$$

Substituting (5.19.5) into (5.19.4) yields $\tilde{v}_1 = \lambda(2-\epsilon_1)/4$ and (5.19.5) gives the desired result for $\tilde{v}_4$. The corresponding result for $\epsilon_2 > 0$ is that $\tilde{v}_4 = \tilde{v}_3 = \tilde{v}_2 = \lambda(2-\epsilon_1)/4$ and $\tilde{v}_1 = 1 - \lambda - (1-\epsilon_1)\tilde{v}_1$, for which the proof techniques are identical. The major difference comes in the definition of $\hat{v}$ in part (2), which would now be defined as $\hat{v} = (\tilde{v}_1 + (1-\epsilon_2)\tilde{v}_2 - (1-\epsilon_2)\tilde{v}_2 \wedge \tilde{v}_3, \tilde{v}_2 \vee \tilde{v}_3, \tilde{v}_2 \wedge \tilde{v}_3, \tilde{v}_2 \wedge \tilde{v}_3)^T$, however, apart from that the proof is verbatim from the ϵ_1 case.

To see the final statement about $\text{cond}(A^{\text{TV}}_\epsilon (A^{\text{TV}}_\epsilon)^*)$, we note first that for $\epsilon_1 \geq 0$, $\epsilon_2 = 0$, the matrix V_ϵ has $V_\epsilon V_\epsilon^* = \text{diag}(\frac{(1-\epsilon_1)^2+1}{2}, 1, 1)$. Therefore $\|V_\epsilon V_\epsilon^*\|_2 = 1$ and $\|(V_\epsilon V_\epsilon^*)^{-1}\|_2 = 2/(2 - 2\epsilon_1 + \epsilon_1^2)$. When considered as a map from $\mathbb{R}^{16}$ to $\mathbb{R}^{12}$, we can write A^{TV}_ϵ as $V_\epsilon \oplus V_\epsilon \oplus V_\epsilon$. Thus $A^{\text{TV}}_\epsilon (A^{\text{TV}}_\epsilon)^* = V_\epsilon V_\epsilon^* \oplus V_\epsilon V_\epsilon^* \oplus V_\epsilon V_\epsilon^*$ and so $\|A^{\text{TV}}_\epsilon (A^{\text{TV}}_\epsilon)^*\|_2 = 1, \|(A^{\text{TV}}_\epsilon (A^{\text{TV}}_\epsilon)^*)^{-1}\|_2 = 2/(2 - 2\epsilon_1 + \epsilon_1^2)$. Thus $\text{cond}(A^{\text{TV}}_\epsilon) = 2/(2 - 2\epsilon_1 + \epsilon_1^2)$. □

Proof of Theorem 4.5.1. Our proof here is almost identical to that of Theorem 4.2.1 part (ii). We will use Proposition 2.9.1. Set

$$\begin{aligned} \iota^1_n &= (y^{\text{TV}}, A^{\text{TV}}_\epsilon) \quad \text{with } \epsilon = (4^{-n}, 0) \\ \iota^2_n &= (y^{\text{TV}}, A^{\text{TV}}_\epsilon) \quad \text{with } \epsilon = (0, 4^{-n}) \\ \iota^0 &= (y^{\text{TV}}, A^{\text{TV}}_\epsilon) \quad \text{with } \epsilon = (0, 0). \end{aligned}$$

By Lemma 5.19.1, we have $\Xi(\iota^1_n) = x^1, \Xi(\iota^2_n) = x^2$. Thus requirement (a) in Proposition 2.9.1 is satisfied with $S^1 = \{e_4 \otimes \mathbf{1}_4\}, S^2 = \{e_1 \otimes \mathbf{1}_4\}$ and $\kappa = 2\sqrt{2}$. It is also obvious (since $A^{\text{TV}}_\epsilon \to A^{\text{TV}}_{0,0}$ as $\epsilon \to (0,0)$) that Proposition 2.9.1, (b) is satisfied with $c_f = f(\iota^0)$. We immediately conclude by Proposition 2.9.1, part (iii) that $\epsilon^{\text{s}}_{\text{B}} \geq 2$ and that $\{\Xi, \Omega\}^{\Delta_1} \notin \Delta^G_1$.

It remains to show we show the results on the condition of elements $\iota \in \Omega$. Since $\text{cond}(A^{\text{TV}}_\epsilon (A^{\text{TV}}_\epsilon)^*) = 2/(2 - 2\epsilon_1 + \epsilon_1^2)$ and $2 - 2\epsilon_1 + \epsilon_1^2 = 1 + (\epsilon_1 - 1)^2$, we immediately obtain that $\text{cond}(A^{\text{TV}}_\epsilon (A^{\text{TV}}_\epsilon)^*) \leq 2$.

The result $\|\iota\| \leq 2$ follows from the fact that

$$\|A^{\text{TV}}_\epsilon\|_2 = \sqrt{\|A^{\text{TV}}_\epsilon (A^{\text{TV}}_\epsilon)^*\|_2} \leq \sqrt{\text{cond}(A^{\text{TV}}_\epsilon (A^{\text{TV}}_\epsilon)^*)} = \sqrt{2}$$

and that $\|y^{\text{TV}}\|_2 \leq 2$.

We now need to check the feasibility condition number. Let $\iota \in \Omega$. Then $\iota = (y^{\text{TV}}, A^{\text{TV}}_\epsilon)$ for some ϵ. Suppose that $(\hat{y}, \hat{A}) \in \tilde{\Omega}_\infty$ is such that $(\hat{y} + y^{\text{TV}}, A^{\text{TV}}_\epsilon + \hat{A})$ is not feasible for the BPTV problem. Then there is no x such that $(A^{\text{TV}}_\epsilon + \hat{A})x = \hat{y} + y^{\text{TV}}$. In particular, the rank of $A^{\text{TV}}_\epsilon + \hat{A}$ when viewed as a map from $\mathbb{R}^{16} \to \mathbb{R}^{12}$ must be strictly less than 12, since otherwise $A^{\text{TV}}_\epsilon + \hat{A}$ is surjective. We will argue that for this to happen, $\hat{A}$ must have a norm at least $1/\sqrt{2}$.

Note that the active set $\mathcal{A}(\Omega)$ is exactly the entries with coordinates $S := \{(1,1), (1,4), (2,2), (3,3)\}$ for each of the matrices V_ϵ. For $\text{rank}(A^{\text{TV}}_\epsilon + \hat{A}) < 12$, we must therefore have $\text{rank}(V_\epsilon + \hat{V}_\epsilon) < 3$ where $\hat{V}_\epsilon$ is 0 except on S. Thus, we require at least one of the following:

$$(V_\epsilon + \hat{V}_\epsilon)_{1,1} = (V_\epsilon + \hat{V}_\epsilon)_{1,4} = 0 \quad \text{or } (V_\epsilon + \hat{V}_\epsilon)_{2,2} = 0 \quad \text{or } (V_\epsilon + \hat{V}_\epsilon)_{3,3} = 0$$

But $(V_\epsilon)_{2,2} = (V_\epsilon)_{3,3} = 1$ and $(V_\epsilon)_{1,1} \vee (V_\epsilon)_{1,4} = 1/\sqrt{2}$. Therefore $\|\hat{V}_\epsilon\|_{\max} \geq 1/\sqrt{2}$. Thus since $\|\cdot\|_2 \geq \|\cdot\|_{\max}$, $\rho(A_\epsilon^{\mathrm{TV}}, y^{\mathrm{TV}}) > 1/\sqrt{2}$. Finally since both $\|A_\epsilon^{\mathrm{TV}}\|_2, \|y^{\mathrm{TV}}\| \leq 2$, we must have $C_{\mathrm{FP}}(\iota) \leq 2\sqrt{2}$. □

The proof of Theorem 4.5.2 is similar.

Proof of Theorem 4.5.2. As before, our aim will be to use Proposition 2.9.1. We pick ι^0, ι_n^1 and ι_n^2 as in the proof of Theorem 4.5.1 and choose $\Omega = \cup_{n=1}^\infty \{\iota_n^1\} \cup_{n=1}^\infty \{\iota_n^2\} \cup \{\iota^0\}$. The argument that Proposition 2.9.1, (b) is satisfied is verbatim as in Theorem 4.5.1.

The proof that Proposition 2.9.1, (a) is satisfied is somewhat more involved Set

$$S^1 = \{v^1 \otimes \mathbf{1}_4 \,|\, v^1 = (a, a, a, b) \text{ with } a \in [0, \lambda/2], b \in [1 - 3\lambda/2, 1]\}$$
$$S^2 = \{v^2 \otimes \mathbf{1}_4 \,|\, v^2 = (b, a, a, a) \text{ with } a \in [0, \lambda/2], b \in [1 - 3\lambda/2, 1]\}.$$

We have that $d_{\mathcal{M}}(x^1, x^2)$ with $x^1 \in S^1, x^2 \in S^2$ is bounded below by $\min S$, where $S \subseteq \mathbb{R}$ is defined by

$$S := \{\|w \otimes \mathbf{1}_4\|_2 \,|\, w = (b - a, 0, 0, a - b) \text{ with } a \in [0, \lambda/2], b \in [1 - 3\lambda/2, 1]\}.$$

If $a \in [0, \lambda/2]$ and $b \in [1 - 3\lambda/2, 1]$ then $b - a \in [1 - 2\lambda, 1]$. Noting that $1 - 2\lambda > 0$ since $\lambda < (2 + \sqrt{2})^{-2}$, we get that $|1 - 2\lambda| \leq 1$. Thus $\min S \geq \|(1 - 2\lambda, 0, 0, 1 - 2\lambda) \otimes \mathbf{1}_4\|_2 = \sqrt{8}(1 - 2\lambda)$. We set $\kappa = \sqrt{8}(1 - 2\lambda)$.

By Lemma 5.19.1, we have $\Xi(\iota_n^1) = \{\eta(4^{-n}, \lambda) \otimes \mathbf{1}_4\}$ and $\Xi(\iota_n^2) = \{P\eta(4^{-n}, \lambda) \otimes \mathbf{1}_4\}$. If $\rho = \lambda(2 - 4^{-n})$ it is easy to see that $\rho \in [0, \lambda/2)$. Therefore $1 - \lambda - (1 - 4^{-n})\rho \geq 1 - \lambda - \lambda/2 = 1 - 3\lambda/2$. Moreover, $1 - \lambda - (1 - 4^{-n})\rho$ is clearly bounded above by 1. Therefore $\eta(\epsilon_1, \lambda) \otimes \mathbf{1}_4 \in S^1$, $(P\eta(\epsilon_2, \lambda)) \in S^2$. Proposition 2.9.1, (a) follows.

We now need to check each of the condition numbers. This process is identical to the one in the proof of Theorem 4.5.1 noting that C_{FP} does not apply for a lasso problem.

We have now checked that the set Ω satisfies all the required bounds on the condition numbers and that Proposition 2.9.1, (a) and Proposition 2.9.1, (b) hold. Theorem 4.5.2 follows. □

Chapter 6

Conclusions and future work

The main results of this thesis have shown that a general collection of mathematical problems used in the mathematics of information cannot be solved with a computational device. In many cases, this is unfortunately still the case with some restrictions on the condition number. By using a general algorithm, we have shown these results for a more general computational model than just a turing machine - indeed, all our results hold for BSS machines as well.

The natural question of how 'bad' this non-computability is lead to the novel definitions of breakdown epsilons, both in a deterministic and probabilistic setting. Small breakdown epsilons may mean that a non-computable problem is amenable to a solution that is somehow 'good enough'.

To demonstrate this, we introduced the RIP and nullspace property in levels, a new collection of conditions that guarantee that the method of compressed sensing will work well (assuming that the method can then be implemented on a computer). We showed that these concepts are more relevant to real world examples than the standard RIP and nullspace properties.

Although it is possible to produce an input set with the nullspace property/ nullspace property in levels so that basis pursuit denoising will be non-computable, the breakdown epsilon for such an input set was shown to be small. This goes some way to explaining the success of compressed sensing - if a matrix vector pair is sufficiently nice so as to exhibit the RIP/nullspace property in levels (which happens with high probability if the sampling pattern is chosen in a sensible random way) then non-computability will only affect the output in a way that is difficult to visualise with the human eye. Therefore although the approach fails in the traditional sense of computability, basis pursuit denoising is successful in practice.

Our results have shown that both image processing and statistical estimation are non-computable. In the case of statistical estimation, we were able to establish that the dual certificate criteria yields a positive result on the computability of the method. Establishing the relevance of this criteria to real world examples is an important future step.

Finally, for neural networks we observed that the computational problem of training a neural network and using it on a classification set is non-computable. Since we showed

this result using a general algorithm, this result applied even if the algorithm was given an oracle to avoid local minima for the non-convex training problem. We termed this the 'first paradox of neural networks': that in general, training a neural network is impossible.

Understanding neural networks also required an adaptation to the SCI hierarchy and general algorithms. We introduced the new concept of randomised general algorithms and showed that similar issues occur even if the algorithm is permitted to use a very general form of randomness. Intriguingly, we also showed that if the time taken to compute a randomised general algorithm is allowed to be non-measurable then the definition permits the existence of a randomised general algorithm which can solve LPO.

We also observed some important phenomena for neural networks: firstly, it is possible to train a neural network on a training set and get the right result on both the training set and an arbitrarily large classification set. However, there exist perturbations of the training set for which the neural network gives the wrong result.

This raises questions about the notion of 'success' in neural networks - the neural network was able to classify both the training set and classification set, but it is in some sense unstable. It is then pertinent to ask whether or not this approach was indeed successful.

The second phenomena that we observed was that there exists pairs of training sets which are arbitrarily close together so that no trained neural network can get the correct answer on the first training set but it is possible to get the right answer by training on the second training set. This shows that training neural networks can be extremely sensitive to perturbations.

In some sense these examples for neural networks were not isolated in that we showed each of these results for uncountably many examples of training sets and classification functions. This of course leads to the natural question of identifying which training sets and classification sets are somehow problematic.

As a final result, combining the first phenomena on success with the universal approximation theorem yields the second paradox of neural networks. Namely, the trained neural network is successful on both the training set and arbitrarily large classification set. However, this trained neural network exhibits an instability. On the other hand, the universal approximation theorem guarantees the existence of a stable neural network that also succeeds on the training set and the classification set. In some sense, the training process will identify the 'wrong' neural network', even if the training succeeds at finding the neural network that minimises the cost function.

We will now discuss some of the open questions that naturally follow from these results. This is done in two separate sections - firstly, we discuss 'general open questions' that are broad and may require new definitions. We then discuss 'specific open questions' that are more technical in nature and may result as a consequence of a particular lemma or proposition proven in this thesis.

6.1 General open questions

1. Of course, owing to the breadth of the field, it is impossible to cover every single technique used in the mathematics of information. We have discussed some broadly successful methods in this thesis, but there are others out there and it is natural to ask if they too are non-computable and if a small breakdown epsilon or a restricted input set is the reason why they might be useful in practice. Since we have chosen some very standard convex optimisation problems to demonstrate the non-computability issues, it is surely the case that our examples are not unique in exhibiting non-computability. We conjecture that these issues will be observed for the vast majority of techniques used in the mathematics of information.

2. On the one hand, these non-computability results are unfortunate. It would have naturally been beneficial if the celebrated methods of the field were computable for all possible choices of input data. On the other hand, this presents an opportunity for mathematicians. Namely, understanding exactly when the methods studied in this thesis can be computed and when they can't will require substantial mathematical analysis and potentially a variety of new techniques and algorithms. A large part of this question has been answered for basis pursuit and basis pursuit denoising. In that case, it was observed that sparsity and the RIP in levels have implications on both the method of basis pursuit and the computability of the method. We conjecture that a similar phenomena will be observed for the other areas discussed in this thesis - that the same structure that guarantees the method will be successful will also guarantee positive results on the computability of the method.

3. As mentioned in Section 4.5, it is important to note that the results in this thesis imply that comparisons between two different methods using empirical data are not valid unless there is some understanding of how closely the underlying algorithms were able to perform the methods. This will require considerable further work and suggests that in future, additional analysis will be required when future algorithms are constructed to demonstrate their robustness.

4. Standard models of complexity theory (for example, consideration of the classes $\mathcal{P}$ and $\mathcal{NP}$) assume that the problem is itself computable. However, we have demonstrated here that there are non-computable problems which may be computable up to a certain tolerance. This will require a new theory of complexity - the seemingly oxymoronic idea of a 'complexity theory for non-computable problems'. We are currently undertaking research in this area.

5. As a related issue, at the time of writing the idea of computer aided proofs is gaining significant traction within mathematics. Indeed, a proof of the Kepler conjecture that used computational assistance has just been published. It is possible that future results will require the use of a computer applied to non-computable problems to establish a result in pure mathematics. Again, this

seems like a contradiction in terms, but the breakdown epsilons may permit some leeway here if the mathematical argument is robust to slight errors in the computational approach.

6. The results on neural networks warrant substantial further thought. In particular, positive results need to establish more than just the existence of a neural network which will succeed. Such a result would also need to explain how *training* the neural network will produce an acceptable output (assuming that there is an oracle that can train correctly). In addition, there would need to be a counterpart result showing that the training process is computable with the given training set. This will likely require considerable resources and effort.

7. In the previous question, the word 'success' was used without consideration of the meaning of 'success' in the context of a neural network. However, the example of a neural network that succeeds on both the training and classification set but fails on uncountably many small perturbations shows that it may not be sufficient to declare a network successful if it works on the training set and is validated on a classification set. A precise, mathematical definition of success is required to properly understand why neural networks are so powerful in practice. This may in fact depend on the real-world application that the neural network is being used to help with. For example, a neural network that is being used to improve the quality of casual photography may be held to a very different standard of success compared to one that is being used in a self driving car.

6.2 Specific open questions

1. Throughout this thesis, there are a variety of breakdown epsilons presented. It is important to know if these results are optimal, or if it is possible to improve upon the results presented here either by means of a new technique or by pushing the existing proof techniques further.

2. The results here are typically done in the real input case. However, Fourier matrices (which are very important in compressed sensing) are complex. Analysing these inputs will require some further work - some results are already proven in the case of complex inputs (like Proposition 5.3.1), whereas others are only applicable for real inputs (like Lemma 5.7.1).

3. Theorem 4.2.15 and Theorem 4.3.5 are both specific examples of Proposition 5.11.2. This proposition constructs a family of matrices (depending on parameters α and γ) which satisfy the nullspace property but are somehow 'bad' inputs to a basis pursuit denoising or unconstrained lasso algorithm. It should be possible to use this proposition to prove a more general result than the specific choices of α and γ parameters used in the two main theorems.

4. Proposition 5.11.2 is only stated for the nullspace property and so far we have been unable to show a similar result for the RIP. Thus the question of whether or

not RIP matrices are non-computable with basis pursuit denoising is still open. A negative result may require very careful selection of the parameters α and γ or perhaps an entirely new collection of 'bad' matrices whereas a positive result will require something akin to a proof that the RIP implies the success of compressed sensing whilst avoiding an argument that goes through the nullspace property akin to the one presented in It is not obvious whether or not we should expect a positive result.

5. Theorem 4.2.2 is a very specific example of a matrix vector pair with infinite RCC condition number. However, Lemma 5.7.1 gives examples of a much broader class of matrix vector pairs that have condition number. It is therefore possible to give a variety of examples for which the condition is infinity. A different example from just a Hadamard or a Bernoulli matrix is given in Proposition 5.7.2 where it is shown that under certain conditions the well studied Hadamard to Haar matrices also have infinite condition number. There should be a number of further extensions to these results that are available using Lemma 5.7.1.

6. Is it possible to expand the analysis done in [83] and 3.2.2 to show that a wider variety of randomly sampled matrices exhibit the RIP_L with a high probability? As in Section 3.2.2, we believe this can be achieved using results from for example [74].

7. Can one improve on the number of measurements required in Theorems 3.2.10 & 3.2.11 (perhaps by reducing log factors) or else show that the bounds in (3.2.4) & (3.2.5) are optimal?

8. If instead of taking the measurement locations Ω at random using a multilevel subsampling scheme, for which deterministic Ω, matrices M and sparsity patterns $(\mathbf{s}, \mathbf{M})$ does $P_\Omega M$ satisfy the RIP_L with a sufficiently small constant $\delta_{s,M}$ for Theorem 3.3.4 to apply?

Bibliography

[1] B. Adcock, A. Bastounis, A. C. Hansen, and B. Roman. On fundamentals of models and sampling in compressed sensing. *preprint*, 2015.

[2] B. Adcock, A. Hansen, B. Roman, and G. Teschke. Chapter four - generalized sampling: Stable reconstructions, inverse problems and compressed sensing over the continuum. volume 182 of *Advances in Imaging and Electron Physics*, pages 187 – 279. Elsevier, 2014.

[3] B. Adcock and A. C. Hansen. Generalized sampling and infinite-dimensional compressed sensing. *Found. Comp. Math.*, 2016.

[4] B. Adcock, A. C. Hansen, C. Poon, and B. Roman. Breaking the coherence barrier: A new theory for compressed sensing. *Forum of Mathematics, Sigma*, 1-84, 2017.

[5] B. Adcock, A. C. Hansen, and B. Roman. A note on compressed sensing of structured sparse wavelet coefficients from subsampled fourier measurements. *IEEE Signal Processing Letters*, 23(5):732–736, May 2016.

[6] C. D. Aliprantis and K. C. Border. *Infinite dimensional analysis*. Springer, Berlin, third edition, 2006. A hitchhiker's guide.

[7] J. Andersson and J. Strömberg. On the theorem of uniform recovery of random sampling matrices. *IEEE Trans. Inform. Theory*, 60(3):1700–1710, 2014.

[8] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hedge. Model-based compressive sensing. *IEEE Trans. Inform. Theory*, 56(4):1982–2001, 2010.

[9] A. Bastounis and A. C. Hansen. On the absence of uniform recovery in many real-world applications of compressed sensing and the restricted isometry property and nullspace property in levels. *SIAM Journal on Imaging Sciences*, 10(1):335–371, 2017.

[10] A. Bastounis, A. C. Hansen, and A. Ben. From global to local: Getting more from compressed sensing. *SIAM News*, October:5,7, 2017.

[11] S. Becker, J. Bobin, and E. J. Candès. Nesta: A fast and accurate first-order method for sparse recovery. *SIAM J. Img. Sci.*, 4(1):1–39, Jan. 2011.

[12] J. Ben-Artzi, A. C. Hansen, O. Nevanlinna, and M. Seidel. New barriers in complexity theory: On the solvability complexity index and the towers of algorithms. *Comptes Rendus Mathematique*, 353(10):931 – 936, 2015.

[13] J. Ben-Artzi, A. C. Hansen, O. Nevanlinna, and M. Seidel. On the Solvability Complexity Index Hierarchy and towers of algorithms. *Preprint*, 2017.

[14] A. Ben-Tal and A. S. Nemirovski. *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2001.

[15] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 417–424, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[16] J. Bigot, C. Boyer, and P. Weiss. An analysis of block sampling strategies in compressed sensing. *Arxiv*, 1305.4446, 2013.

[17] L. Blum, M. Shub, and S. Smale. On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines. *Bull. Amer. Math. Soc. (N.S.)*, 21(1):1–46, 1989.

[18] H. Boche, R. Calderbank, G. Kutyniok, and J. Vybíral. *Compressed Sensing and its Applications*. Springer, 2015.

[19] L. Bottou. Stochastic learning. In O. Bousquet and U. von Luxburg, editors, *Advanced Lectures on Machine Learning*, Lecture Notes in Artificial Intelligence, LNAI 3176, pages 146–168. Springer Verlag, Berlin, 2004.

[20] A. Bourrier, M. Davies, T. Peleg, P. Perez, and R. Gribonval. Fundamental performance limits for ideal decoders in high-dimensional linear inverse problems. *IEEE Trans. Inform. Theory*, 60(12):7928–7946, Dec 2014.

[21] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[22] D. Bridges and F. Richman. *Varieties of Constructive Mathematics*. London Mathematical Society Lecture Note Series. Cambridge University Press, 1987.

[23] P. Bürgisser and F. Cucker. On a problem posed by Steve Smale. *Ann. of Math. (2)*, 174(3):1785–1836, 2011.

[24] P. Bürgisser and F. Cucker. *Condition : the geometry of numerical algorithms*. Grundlehren der mathematischen Wissenschaften. Springer, Berlin, Heidelberg, New York, 2013.

[25] T. T. Cai, L. Wang, and G. Xu. New bounds for restricted isometry constants. *IEEE Trans. Inform. Theory*, 56(9):4388–4394, 2010.

[26] T. T. Cai, L. Wang, and G. Xu. Shifting inequality and recovery of sparse signals. *IEEE Trans. Signal Process.*, 58(3, part 1):1300–1308, 2010.

[27] T. T. Cai and A. Zhang. Sharp RIP bound for sparse signal and low-rank matrix recovery. *Appl. Comput. Harmon. Anal.*, 35(1):74–93, 2013.

[28] T. T. Cai and A. Zhang. Sparse representation of a polytope and recovery of sparse signals and low-rank matrices. *IEEE Transactions on Information Theory*, 60(1):122–132, 2014.

[29] E. J. Candès. An introduction to compressive sensing. *IEEE Signal Process. Mag.*, 25(2):21–30, 2008.

[30] E. J. Candès. The restricted isometry property and its implications for compressed sensing. *C. R. Math. Acad. Sci. Paris*, 346(9-10):589–592, 2008.

[31] E. J. Candès and D. Donoho. New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities. *Comm. Pure Appl. Math.*, 57(2):219–266, 2004.

[32] E. J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inform. Theory*, 52(2):489–509, 2006.

[33] E. J. Candès and T. Tao. Decoding by linear programming. *IEEE Trans. Inform. Theory*, 51(12):4203–4215, 2005.

[34] E. J. Candès and T. Tao. Near-optimal signal recovery from random projections: universal encoding strategies? *IEEE Trans. Inform. Theory*, 52(12):5406–5425, 2006.

[35] W. R. Carson, M. Chen, M. R. D. Rodrigues, R. Calderbank, and L. Carin. Communications-inspired projection design with application to compressive sensing. *SIAM J. Imaging Sci.*, 5(4):1185–1212, 2012.

[36] V. Caselles, A. Chambolle, D. Cremers, M. Novaga, and T. Pock. *An introduction to Total Variation for Image Analysis*, volume 9, pages 263–340. 2010.

[37] A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20(1):89–97, Jan 2004.

[38] T. F. Chan and J. Shen. Mathematical models for local nontexture inpaintings. *SIAM J. Appl. Math*, 62:1019–1043, 2002.

[39] N. Chauffert, P. Ciuciu, J. Kahn, and P. Weiss. Variable density sampling with continuous sampling trajectories. *SIAM J. Imaging Sci.*, 2014.

[40] K. Choi, J. Wang, L. Zhu, T.-S. Suh, S. Boyd, and L. Xing. Compressed sensing based cone-beam computed tomography reconstruction with a first-order methoda). *Medical Physics*, 37(9):5113–5125, 2010.

[41] A. Cohen, W. Dahmen, and R. Devore. Compressed sensing and best k-term approximation. *J. Amer. Math. Soc*, pages 211–231, 2009.

[42] F. Cucker. A theory of complexity, condition, and roundoff. *Forum of Mathematics, Sigma*, 3, 002 2015.

[43] G. Dantzig. *Linear programming and extensions.* Rand Corporation Research Study. Princeton Univ. Press, Princeton, NJ, 1963.

[44] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Comm. Pure Appl. Math.*, 41(7):909–996, 1988.

[45] I. Daubechies. *Ten Lectures on Wavelets.* Society for Industrial and Applied Mathematics, 1992.

[46] I. Daubechies. Big data's mathematical mysteries. *Quanta Magazine*, 2015.

[47] M. A. Davenport, M. F. Duarte, Y. C. Eldar, and G. Kutyniok. Introduction to compressed sensing. In *Compressed Sensing: Theory and Applications.* Cambridge University Press, 2011.

[48] M. E. Davies and R. Gribonval. Restricted isometry constants where ℓ^p sparse recovery can fail for $0 \ll p \leq 1$. *IEEE Transactions on Information Theory*, 55(5):2203–2214, May 2009.

[49] M. N. Do and M. Vetterli. The contourlet transform: An efficient directional multiresolution image representation. *IEEE Trans. Image Proc.*, 14(12):2091–2106, 2005.

[50] D. L. Donoho and J. Tanner. Counting faces of randomly-projected polytopes when the projection radically lowers dimension. *J. Amer. Math. Soc.*, 22(1):1–53, 2009.

[51] P. Doyle and C. McMullen. Solving the quintic by iteration. *Acta Math.*, 163(3-4):151–180, 1989.

[52] M. Elad. Deep, deep trouble. *SIAM News*, May:12, 2017.

[53] Y. C. Eldar and G. Kutyniok, editors. *Compressed Sensing: Theory and Applications.* Cambridge University Press, 2012.

[54] S. Esedolu and S. J. Osher. Decomposition of images by the anisotropic rudin-osher-fatemi model. *Communications on Pure and Applied Mathematics*, 57(12):1609–1626, 2004.

[55] H. Fawzi, J. Gouveia, P. A. Parrilo, R. Z. Robinson, and R. R. Thomas. Positive semidefinite rank. *Mathematical Programming*, 153(1):133–177, Oct 2015.

[56] H. Fawzi, J. Saunderson, and P. A. Parrilo. Equivariant semidefinite lifts and sum-of-squares hierarchies. *SIAM Journal on Optimization*, 25(4):2212–2243, 2015.

[57] S. Feferman. Penrose's godelian argument. *Psyche*, 2:21–32, 1996.

[58] M. Fornasier and H. Rauhut. Compressive sensing. In *Handbook of Mathematical Methods in Imaging*, pages 187–228. Springer, 2011.

[59] S. Foucart. A note on guaranteed sparse recovery via ℓ_1-minimization. *Appl. Comput. Harmon. Anal.*, 29(1):97–103, 2010.

[60] S. Foucart and H. Rauhut. *A mathematical introduction to compressive sensing*. Applied and Numerical Harmonic Analysis. Birkhäuser/Springer, New York, 2013.

[61] M. D. Guay, W. Czaja, M. A. Aronova, and R. D. Leapman. Compressed sensing electron tomography for determining biological structure. *Scientific Reports*, 6:27614 EP –, 06 2016.

[62] M. Guerquin-Kern, M. Häberlin, K. P. Pruessmann, and M. Unser. A fast wavelet-based reconstruction method for Magnetic Resonance Imaging. *IEEE Trans. Med. Imaging*, 30(9):1649–1660, 2011.

[63] M. Guerquin-Kern, L. Lejeune, K. P. Pruessmann, and M. Unser. Realistic analytical phantoms for parallel Magnetic Resonance Imaging. *IEEE Trans. Med. Imaging*, 31(3):626–636, 2012.

[64] A. C. Hansen. On the solvability complexity index, the n-pseudospectrum and approximations of spectra of operators. *J. Amer. Math. Soc.*, 24(1):81–124, 2011.

[65] S. Hashemi, S. Beheshti, P. R. Gill, N. S. Paul, and R. S. C. Cobbold. Accelerated compressed sensing based ct image reconstruction. *Computational and Mathematical Methods in Medicine*, 2015:161797, 2015.

[66] T. Hastie, R. Tibshirani, and M. Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations (Chapman & Hall/CRC Monographs on Statistics & Applied Probability)*. Chapman and Hall/CRC, May 2015.

[67] L. He and L. Carin. Exploiting structure in wavelet-based Bayesian compressive sensing. *IEEE Trans. Signal Process.*, 57(9):3488–3497, 2009.

[68] L. He, H. Chen, and L. Carin. Tree-structured compressive sensing with variational Bayesian analysis. *IEEE Signal Process. Letters*, 17(3):233–236, 2010.

[69] M. A. Herman and T. Strohmer. High-resolution radar via compressed sensing. *IEEE Trans. Signal Process.*, 57(6):2275–2284, 2009.

[70] D. Hilbert. *Principles of Mathematical Logic*. Ams Chelsea, 1950.

[71] D. J. Holland, M. J. Bostock, L. F. Gladden, and D. Nietlispach. Fast multidimensional nmr spectroscopy using compressed sensing. *Angewandte Chemie International Edition*, 50(29):6548–6551, 2011.

[72] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In *2009 IEEE 12th International Conference on Computer Vision*, pages 2146–2153, Sept 2009.

[73] A. Jones, A. Tamtögl, I. Calvo-Almazán, and A. Hansen. Continuous compressed sensing for surface dynamical processes with helium atom scattering. *Scientific Reports*, 6:27776 EP –, 06 2016.

[74] A. D. Jones, B. Adcock, and A. C. Hansen. On asymptotic incoherence and its implications for compressed sensing of inverse problems. *IEEE Trans. Information Theory*, 62(2):1020–1037, 2016.

[75] R. Kannan and C. L. Monma. *On the Computational Complexity of Integer Programming Problems*, pages 161–172. Springer Berlin Heidelberg, Berlin, Heidelberg, 1978.

[76] M. A. Khajehnejad, W. Xu, A. S. Avestimehr, and B. Hassibi. Weighted l1 minimization for sparse recovery with prior information. In *Proceedings of the 2009 IEEE International Conference on Symposium on Information Theory - Volume 1*, ISIT'09, pages 483–487, Piscataway, NJ, USA, 2009. IEEE Press.

[77] K. Ko. *Complexity theory of real functions*. Progress in Theoretical Computer Science. Birkhäuser Boston, Inc., Boston, MA, 1991.

[78] F. Krahmer and R. Ward. Stable and robust sampling strategies for compressive imaging. *IEEE Trans. Image Proc.*, 23(2):612–622, 2014.

[79] G. Kutyniok, J. Lemvig, and W.-Q. Lim. Compactly supported shearlets. In M. Neamtu and L. Schumaker, editors, *Approximation Theory XIII: San Antonio 2010*, volume 13 of *Springer Proceedings in Mathematics*, pages 163–186. Springer New York, 2012.

[80] E. Y. Lam and J. W. Goodman. Iterative statistical approach to blind image deconvolution. *J. Opt. Soc. Am. A*, 17(7):1177–1184, Jul 2000.

[81] R. Leary, Z. Saghi, P. A. Midgley, and D. J. Holland. Compressed sensing electron tomography. *Ultramicroscopy*, 131:70 – 91, 2013.

[82] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015. Insight.

[83] C. Li and B. Adcock. Compressed sensing with local structure: Uniform recovery guarantees for the sparsity in levels class. *Appl. Comput. Harmon. Anal.*, 2017.

[84] Y. Lou, T. Zeng, S. Osher, and J. Xin. A weighted difference of anisotropic and isotropic total variation model for image processing. *SIAM J. Imaging Sciences*, 8(3):1798–1823, 2015.

[85] A. K. Louis, P. Maaß, and A. Rieder. *Wavelets*. Teubner Studienbücher Mathematik. [Teubner Mathematical Textbooks]. B. G. Teubner, Stuttgart, second edition, 1998. Theorie und Anwendungen. [Theory and applications].

[86] M. Lustig, D. L. Donoho, and J. M. Pauly. Sparse MRI: the application of compressed sensing for rapid MRI imaging. *Magn. Reson. Imaging*, 58(6):1182–1195, 2007.

[87] A. Maheshwari and M. Smid. *Introduction to Theory of Computation*. 2012.

[88] MATLAB. *version 9.1.0 (R2016b)*. The MathWorks Inc., Natick, Massachusetts, 2016.

[89] J. D. McEwen and Y. Wiaux. Compressed sensing for radio interferometric imaging: Review and future direction. In *2011 18th IEEE International Conference on Image Processing*, pages 1313–1316, Sept 2011.

[90] C. McMullen. Families of rational maps and iterative root-finding algorithms. *Ann. of Math. (2)*, 125(3):467–493, 1987.

[91] C. McMullen. Braiding of the attractor and the failure of iterative algorithms. *Invent. Math.*, 91(2):259–272, 1988.

[92] M. Minsky and S. Papert. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, Cambridge, MA, USA, 1969.

[93] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pages 807–814, USA, 2010. Omnipress.

[94] Y. Nesterov. *Introductory lectures on convex optimization : a basic course*. Applied optimization. Kluwer Academic Publ., Boston, Dordrecht, London, 2004.

[95] A. Pinkus. Approximation theory of the mlp model in neural networks. *Acta Numerica*, 8:143195, 1999.

[96] C. Poon. On the role of total variation in compressed sensing. *SIAM Journal on Imaging Sciences*, 8(1):682–720, 2015.

[97] G. Puy, M. E. Davies, and R. Gribonval. Linear embeddings of low-dimensional subsets of a hilbert space to R^m. *EUSIPCO - 23rd European Signal Processing Conference*, 2015.

[98] G. Puy, J. P. Marques, R. Gruetter, J. P. Thiran, D. V. D. Ville, P. Vandergheynst, and Y. Wiaux. Spread spectrum magnetic resonance imaging. *IEEE Transactions on Medical Imaging*, 31(3):586–598, March 2012.

[99] G. Puy, P. Vandergheynst, R. Gribonval, and Y. Wiaux. Universal and efficient compressed sensing by spread spectrum and application to realistic fourier imaging techniques. *EURASIP Journal on Advances in Signal Processing*, 2012(1):1–13, 2012.

[100] G. Puy, P. Vandergheynst, and Y. Wiaux. On variable density compressive sampling. *IEEE Signal Process. Letters*, 18:595–598, 2011.

[101] H. Rauhut and R. Ward. Interpolation via weighted minimization. *Appl. Comput. Harmon. Anal.*, (to appear).

[102] J. Renegar. Incorporating condition measures into the complexity theory of linear programming. *SIAM Journal on Optimization*, 5(3):506–524, 1995.

[103] J. Renegar. Linear programming, complexity theory and elementary functional analysis. *Mathematical Programming*, 70(1):279–351, 1995.

[104] B. Roman, B. Adcock, and A. C. Hansen. On asymptotic structure in compressed sensing. *Arxiv*, 1406.4178, 2015.

[105] J. Romberg. Imaging via compressive sampling. *IEEE Signal Process. Mag.*, 25(2):14–20, 2008.

[106] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1-4):259–268, Nov. 1992.

[107] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Neurocomputing: Foundations of research. chapter Learning Representations by Back-propagating Errors, pages 696–699. MIT Press, Cambridge, MA, USA, 1988.

[108] L. A. Shepp and B. F. Logan. The fourier reconstruction of a head section. *IEEE Transactions on Nuclear Science*, 21(3):21–43, June 1974.

[109] M. Shub and S. Smale. Complexity of Bézout's theorem. I. Geometric aspects. *J. Amer. Math. Soc.*, 6(2):459–501, 1993.

[110] M. Shub and S. Smale. On the intractability of Hilbert's Nullstellensatz and an algebraic version of $NP \neq P$? *Duke Math. J.*, 81(1):47–54, 1995.

[111] S. Smale. The fundamental theorem of algebra and complexity theory. *Bull. Amer. Math. Soc. (N.S.)*, 4(1):1–36, 1981.

[112] S. Smale. Complexity theory and numerical analysis. In *Acta numerica, 1997*, volume 6 of *Acta Numer.*, pages 523–551. Cambridge Univ. Press, Cambridge, 1997.

[113] J. C. Spall. *Introduction to Stochastic Search and Optimization.* John Wiley & Sons, Inc., New York, NY, USA, 1 edition, 2003.

[114] V. Studer, J. Bobin, M. Chahid, H. Moussavi, E. Candès, and M. Dahan. Compressive fluorescence microscopy for biological and hyperspectral imaging. *Proc Natl Acad Sci USA*, 109(26):1679–1687, 2011.

[115] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

[116] R. J. Tibshirani. The lasso problem and uniqueness. *Electron. J. Statist.*, 7:1456–1490, 2013.

[117] A. M. Tillmann and M. E. Pfetsch. The computational complexity of the restricted isometry property, the nullspace property, and related concepts in compressed sensing. *IEEE Trans. Inform. Theory*, 60(2):1248–1259, 2014.

[118] Q. Tran-Dinh and V. Cevher. A primal-dual algorithmic framework for constrained convex minimization. *ArXiv*, 1406.5403v2, 2014.

[119] Y. Traonmilin and R. Gribonval. Stable recovery of low-dimensional cones in hilbert spaces: One rip to rule them all. *Arxiv*, 1510.00504, 2015.

[120] A. M. Turing. On Computable Numbers, with an Application to the Entscheidungsproblem. *Proc. London Math. Soc.*, S2-42(1):230.

[121] E. van den Berg and M. P. Friedlander. SPGL1: A solver for large-scale sparse reconstruction, June 2007. http://www.cs.ubc.ca/labs/scl/spgl1.

[122] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Rev.*, 38(1):49–95, Mar. 1996.

[123] M. J. Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using l1-constrained quadratic programming (lasso). *IEEE Trans. Inf. Theor.*, 55(5):2183–2202, May 2009.

[124] L. Wang, D. Carlson, M. R. D. Rodrigues, D. Wilcox, R. Calderbank, and L. Carin. Designed measurements for vector count data. In *Advances in Neural Information Processing Systems*, pages 1142–1150, 2013.

[125] Q. Wang, M. Zenge, H. E. Cetingul, E. Mueller, and M. S. Nadar. Novel sampling strategies for sparse mr image reconstruction. *Proc. Int. Soc. Mag. Res. in Med.*, (22), 2014.

[126] K. Weihrauch. *Computable analysis*. Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag, Berlin, 2000. An introduction.

[127] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, ukasz Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016.

[128] M. Ziegler and V. Brattka. Computability in linear algebra. *Theor. Comput. Sci.*, 326(1-3):187–211, Oct. 2004.

Appendix A

Appendix

A.1 BPDN stopping criteria

Proposition A.1.1. *Fix $\epsilon > 0$ and let (x_n) be a sequence generated by a primal convergent algorithm for BP_δ, where $\delta > 0$. Assume that $\|b\| > \delta$. Define the sequences $(p_n)_{n=1}^{\infty}$ and $(G_n)_{n=1}^{\infty}$ by*

$$p_n = \begin{cases} \frac{Ax_n - b}{\|A^*(Ax_n - b)\|_\infty} & \text{if } A^*(Ax_n - b) \neq 0 \text{ or } \|Ax_n - b\|_2 > \epsilon + \delta \\ 0 & \text{otherwise} \end{cases}$$

and $G_n = \|x_n\|_1 + \min_{m \leq n} \left(\langle b, p_m \rangle + \delta \|p_m\|_2 \right)$. Then

$$0 \leq \|x_n\|_1 - BP_\delta \leq G_n. \tag{A.1.1}$$

Furthermore, $\lim_{n \to \infty} G_n = 0$.

To prove this theorem, we also need to prove Lemma 5.11.1.

Proof of Lemma 5.11.1. The statement that $0 \in M^{\mathrm{BP}}(y, A, \delta)$ if and only if $\|y\|_2 \leq \delta$ is obvious from the fact that 0 has minimal ℓ^1 norm and the fact that $\|A0 - y\|_2 = \|y\|_2$. To see that if $0 \notin M^{\mathrm{BP}}(y, A, \delta)$ then for any $x \in M^{\mathrm{BP}}(y, A, \delta)$ we have $\|Ax - y\|_2 = \delta$, suppose otherwise. Our aim will be to construct a vector $\tilde{x}$ such that $\|\tilde{x}\|_1 < \|x\|_1$, yet $\|A\tilde{x} - y\|_2 \leq \delta$, contradicting the statement that x is a minimiser.

Let $\|Ax - y\|_2 = \epsilon < \delta$. We consider $\tilde{x} = (\|y\|_2 - \delta)x/(\|y\|_2 - \epsilon)$. By the previous statement, $\|y\| > \delta$ and by definition, $0 \leq \epsilon < \delta$. Thus $(\|y\|_2 - \delta)/(\|y\|_2 - \epsilon) < 1$. Furthermore, $x \neq 0$ so $\|\tilde{x}\|_1 < \|x\|_1$. It remains to prove that $\|A\tilde{x} - y\| \leq \delta$. We have

by a simple calculation that

$$\begin{aligned}\left\|\frac{\|y\|_2-\delta}{\|y\|_2-\epsilon}Ax-y\right\|_2 &= \left(\frac{\|y\|_2-\delta}{\|y\|_2-\epsilon}\right)\left\|Ax-y\frac{\|y\|_2-\epsilon}{\|y\|_2-\delta}\right\|_2\\ &= \left(\frac{\|y\|_2-\delta}{\|y\|_2-\epsilon}\right)\left\|Ax-y\left(1+\frac{\delta-\epsilon}{\|y\|_2-\delta}\right)\right\|_2\\ &\leq \left(\frac{\|y\|_2-\delta}{\|y\|_2-\epsilon}\right)\left[\|Ax-y\|_2+\frac{(\delta-\epsilon)\|y\|_2}{\|y\|_2-\delta}\right]\\ &= \left(\frac{1}{\|y\|_2-\epsilon}\right)[\epsilon(\|y\|_2-\delta)+(\delta-\epsilon)\|y\|_2]=\delta.\end{aligned}$$

The result follows. □

We state without proof the following theorem (see [21, 60] and similar discussions in [14, 94]) which is itself a corollary of the Fenchel Duality Theorem.

Theorem A.1.2. *Let $A \in \mathbb{R}^{m\times N}$. Suppose that F and G are convex functions with $F : \mathbb{R}^m \to (-\infty,\infty]$, $G : \mathbb{R}^N \to (-\infty,\infty)$. If there is an $x \in \mathbb{R}^N$ with $Ax \in dom(F)$ and there are $\tilde{x}$, $\tilde{\xi}$ with $\tilde{x} \in \operatorname{argmin}_{x\in\mathbb{R}^N}(F(Ax)+G(x))$ and $\tilde{\xi} \in \operatorname{argmax}_{\xi\in\mathbb{R}^m}(-F^*(\xi)-G^*(-A^*\xi))$ then*

$$F(A\tilde{x})+G(\tilde{x}) = -F^*(\tilde{\xi})-G^*(-A^*\tilde{\xi}).$$

Moreover, $\tilde{x},\tilde{\xi}$ solves

$$\min_{x\in\mathbb{R}^N}\max_{\xi\in\mathbb{R}^m}\langle Ax,\xi\rangle+G(x)-F^*(\xi) = \max_{\xi\in\mathbb{R}^m}\min_{x\in\mathbb{R}^N}\langle Ax,\xi\rangle+G(x)-F^*(\xi)$$

Proof. The sequence (x_n) is bounded in the 1-norm, so it has a convergent subsequence (x_{n_k}) with $\lim_{k\to\infty} x_{n_k} = x^*$ for some vector x^*. Since (x_{n_k}) satisfies (4.2.1), we see that x^* is an optimal point for BP_δ. Therefore by Lemma 5.11.1, $\lim_{k\to\infty}\|Ax_{n_k}-b\|_2 = \|Ax^*-b\|_2 = \delta > 0$. We claim that for large enough k, $\|A^*(Ax_{n_k}-b)\|_\infty$ is bounded away from zero. By continuity, it will suffice to show that $A^*(Ax^*-b)\neq 0$. Indeed, with $f(x) := \|Ax-b\|_2^2$, we note that f is convex and see that $2A^*(Ax^*-b) = (\nabla f)(x^*)$. Thus $A^*(Ax^*-b) = 0$ if and only if x^* is a global minimum for f (because for any convex function f, $f'(c) = 0$ implies that $f(c)$ is a global minimum for f). But by the definition of a primal convergent algorithm there is an x^0 with $f(x^0) < \delta = f(x^*)$, contradicting the minimality of $f(x^*)$.

Note that the basis pursuit problem can be written as

$$\arg\min_{x\in\mathbb{R}^m}\chi_{\|\cdot-b\|_2\leq\delta}(Ax)+\|x\|_1 \text{ where } \chi_K(v) = \begin{cases}0 \text{ if } v\in K\\ \infty \text{ otherwise}\end{cases}$$

Our aim will be to apply Theorem A.1.2 with $F = \chi_{\|\cdot-b\|_2\leq\delta}$, $G = \|\cdot\|_1$. Because norms are convex, it is clear that F and G are also convex. Since $\|Ax^0-b\| < \delta$, there is an $x = x^0$ with $x \in \mathbb{R}^N$ and $Ax \in \operatorname{dom}(F)$. Moreover, $G : \mathbb{R}^N \to (-\infty,\infty)$. The

existence of an x^* is obvious. To see the existence of ρ^*, first note that

$$F^*(\rho) = \sup_{\rho' \in \mathbb{R}^m} \langle \rho, \rho' \rangle - F(\rho') = \langle \rho, b \rangle + \sup\{\langle \rho, \rho' - b \rangle \mid \|\rho' - b\| \leq \delta\} = \langle \rho, b \rangle + \delta \|\rho\|_2.$$

By splitting into two cases, one where $\|\rho\|_\infty > 1$, and one where $\|\rho\|_\infty \leq 1$ (using Holder's inequality), we see that $G^*(\rho) = \chi_{\|\cdot\|_\infty \leq 1}(\rho)$. Thus $-F^*(\rho) - G^*(-A^*\rho) = -\delta\|\rho\|_2 - \chi_{\|\cdot\|_\infty \leq 1}(-A^*\rho)$ and so

$$\max_{\rho \in \mathbb{R}^m} -F^*(\rho) - G^*(-A^*\rho) = \max\{-\delta\|\rho\|_2 - \langle \rho, b \rangle \mid \|A^*\rho\|_\infty \leq 1\}$$

Let ρ^n be such that $\|A^*\rho^n\|_\infty \leq 1$ and $\lim_{n\to\infty} -\delta\|\rho^n\|_2 - \langle \rho^n, b \rangle = \max_{\rho \in \mathbb{R}^m} -F^*(\rho) - G^*(-A^*\rho)$. Since $\max_{\rho \in \mathbb{R}^m} -F^*(\rho) - G^*(-A^*\rho) \geq -F^*(0) - G^*(-A^*0) = 0$, we must have that for n sufficiently large, $-\delta\|\rho^n\|_2 - \langle \rho^n, b \rangle \geq -1$. Because $\|Ax^0 - b\|_2 < \delta$, we can write $b = \eta + Ax^0$ where $\|\eta\|_2 < \delta$. Thus $-\langle \rho^n, b \rangle = -\langle A^*\rho^n, x^0 \rangle - \langle \rho^n, \eta \rangle < \|x^0\|_1 + \|\eta\|_2\|\rho^n\|_2$, because $\|A^*\rho^n\|_\infty \leq 1$. Therefore

$$0 \leq -\delta\|\rho^n\|_2 - \langle \rho^n, b \rangle + 1 \leq \|x^0\|_1 + 1 - (\delta - \|\eta\|_2)\|\rho^n\|_2$$

Thus $\|\rho^n\|_2$ must eventually be bounded. Therefore (for some $C > 0$)

$$\max\{-\delta\|\rho\|_2 - \langle \rho, b \rangle \mid \|A^*\rho\|_\infty \leq 1\} = \max\{-\delta\|\rho\|_2 - \langle \rho, b \rangle \mid \|A^*\rho\|_\infty \leq 1, \|\rho\|_2 \leq C\}$$

and since the maximization on the right hand side is a maximization problem of a continuous function taken over a closed, bounded set, the maximum is attained at some ρ^*.

We can therefore apply Theorem A.1.2. We have

$$\|x^*\|_1 = \operatorname{argmax}\{-\delta\|\rho\|_2 - \langle \rho, b \rangle \mid \|A^*\rho\|_\infty \leq 1\} = -\delta\|\rho^*\|_2 - \langle \rho^*, b \rangle \tag{A.1.2}$$

Since

$$\|A^*p_m\|_\infty = \begin{cases} 1 \text{ if } A^*(Ax_m - b) \neq 0 \text{ or } \|Ax_n - b\|_2 > \epsilon + \delta \\ 0 \text{ otherwise} \end{cases}$$

we have that p_m is feasible for the dual problem and thus

$$\|x^*\|_1 \geq \max_{m \leq n} -\delta\|p_m\|_2 - \langle p_m, b \rangle = -G_n + \|x_n\|_1$$

from which we immediately get (A.1.1).

It remains to show that $G_n \to 0$. Since eventually both $\|A^*(Ax_{n_k} - b)\|_\infty \neq 0$ and $\|Ax_n - b\|_2 \leq \epsilon + \delta$, we have $\lim_{k\to\infty} p_{n_k} = \frac{Ax^* - b}{\|A^*(Ax^* - b)\|_\infty}$, and we denote this limit by

p^*. We claim that $p^* = \rho^*$. If this claim holds then

$$\begin{aligned}-\delta\|\rho^*\|_2 - \langle \rho^*, b\rangle = -\delta\|p^*\|_2 - \langle p^*, b\rangle &= \liminf_{k\to\infty} -\delta\|p_{n_k}\|_2 - \langle p_{n_k}, b\rangle \\ &\leq \liminf_{n\to\infty} \max_{m\leq n} -\delta\|p_m\|_2 - \langle p_m, b\rangle \\ &\leq -\delta\|\rho^*\|_2 - \langle \rho^*, b\rangle\end{aligned}$$

where we have used the continuity of norms and inner products and the maximality of $-\delta\|\rho^*\|_2 - \langle \rho^*, b\rangle$. Since $\liminf_{n\to\infty} \max_{m\leq n} -\delta\|p_m\|_2 - \langle p_m, b\rangle = \liminf -G_n + \|x_n\|_1$, we see that $\liminf -G_n + \|x_n\|_1 = -\delta\|\rho^*\|_2 - \langle \rho^*, b\rangle$. By (A.1.2), the right hand side is $\|x^*\|_1$ and by continuity the left hand side is $\|x^*\|_1 + \liminf -G_n$. Therefore $-\limsup G_n = \liminf -G_n = 0$ and because $G_n \geq 0$ by (A.1.1), the result will follow.

It will therefore suffice to show that $p^* = \rho^*$. The final part of Theorem A.1.2 tells us that if $L(x, \xi) = \langle Ax, \xi\rangle + \|x\|_1 - \delta\|\xi\|_2 - \langle \xi, b\rangle$ then

$$L(x^*, \rho^*) = \min_{x\in\mathbb{R}^N} \max_{\xi\in\mathbb{R}^m} L(x, \xi) = \max_{\xi\in\mathbb{R}^m} \min_{x\in\mathbb{R}^N} L(x, \xi).$$

We therefore obtain the following saddle point property: that for all $x \in \mathbb{R}^N$, $\xi \in \mathbb{R}^m$, we have $L(x^*, \xi) \leq L(x^*, \rho^*) \leq L(x, \rho^*)$. Thus $0 \in \partial L(\cdot, \rho^*)(x^*), \partial L(x^*, \cdot)(\rho^*)$ where ∂ denotes the subdifferential. The x derivative yields $0 \in \partial\|\cdot\|_1(x^*) + A^*\rho^*$. In particular, since $v \in \partial\|\cdot\|_1(x^*)$ implies that $|v_{\mathrm{supp}(x^*)}| = |\mathrm{sgn}(x^*_{\mathrm{supp}(x^*)})| = 1$, $|v_{\mathrm{supp}(x^*)^c}| \in [0, 1]$ and x^* is non-zero, we must have $\|v\|_\infty = 1$. Hence $\|A^*\rho^*\|_\infty = 1$. In particular, $\rho^* \neq 0$ and so $L(x^*, \cdot)$ is differentiable at ρ^*. This derivative must be 0 and so $-\delta\frac{\rho^*}{\|\rho^*\|_2} - b + Ax^* = 0$. Thus $\rho^* = c(Ax^* - b)$ for some $c \geq 0$. But $\|A^*\rho^*\|_\infty = 1$ so $c = \|A^*(Ax^* - b)\|_\infty$ and so $\rho^* = \frac{Ax^*-b}{\|A^*(Ax^*-b)\|_\infty} = p^*$. □

A.2 Lasso stopping criteria

We will now prove the two results on when a lasso minimiser from section 5.15.

Proof of Lemma 5.15.2. We start by assuming that $\|A^*b\|_\infty \leq \frac{\lambda}{2}$. Set $-\lambda\rho = 2A^*(A0 - b)$. Then $\|\rho\|_\infty \leq 1$, so ρ is in the subdifferential of $\|\cdot\|_1$ at 0. Therefore $2A^*(A0 - b) + \lambda\rho = 0$ where $\rho \in \delta(\|\cdot\|_1)(0)$, and so 0 is indeed a minimiser. Conversely, if $0 \in \mathrm{argmin}_{x\in\mathbb{R}^N} \|Ax - y\|^2 + \lambda\|x\|_1$ then there is a ρ in the subdifferential of $\|\cdot\|_1$ at 0 with $2A^*(A0 - b) + \lambda\rho = 0$. However, such a ρ must have $\|\rho\|_\infty \leq 1$ and thus $\|2A^*(-b)\|_\infty \leq \lambda\|\rho\|_\infty \leq \lambda$. Thus $\|A^*b\|_\infty \leq \lambda/2$. □

Proof of Lemma 5.15.3. Since the proof is very similar to the proof of Lemma A.1.1, we only give the details. Firstly, we note that lasso is equivalent to finding $\mathrm{argmin}_{x\in\mathbb{R}^N} F(Ax) + G(x)$ where $F(x) = \|\cdot - b\|_2^2, G(x) = \lambda\|x\|_1$. It is clear that $A0 \in \mathrm{dom}(F)$ and that the optimal value of lasso is attained at some x^* (which by potentially passing to a convergent subsequence as in the proof of Lemma A.1.1 we set to be $x^* = \lim_{n\to\infty} x_n$). It is also easy to see that the convex conjugates of F and G are given by $F^*(\xi) = \|\xi\|_2^2/4 + \langle \xi, b\rangle$ and $G^*(\xi) = \chi_{\|\cdot\|_\infty\leq\lambda}(\xi)$. Since $F^*(\xi) \to \infty$ as $\|\xi\| \to$

∞, the maximum of $-F^*(\xi)-G^*(A^*\xi)$ is attained. Thus Theorem A.1.2 applies and we get $\|Ax^*-b\|_2^2+\lambda\|x^*\|_1 = \max_{\xi\in\mathbb{R}^m} -F^*(\xi)-G^*(A^*(\xi) \geq -G(n)+\|Ax_n-b\|_2^2+\lambda\|x_n\|_1$.

All that remains is to see that $G_n \to 0$ as $n \to \infty$. As before, it will suffice to show that $\lim_{n\to\infty} p_n = \xi^*$ after passing to a subsequence. To see this, note that at any optimal points (x^*, ξ^*) for both lasso and its dual, we must have that the subdifferential in the ξ direction of $L(x^*, \xi^*) = \langle Ax, \xi\rangle + \lambda\|x\|_1 - \|\xi\|_2^2/4$ at (x, ξ^*) is 0. In particular $Ax^* - \xi^*/2 - b = 0$ and thus $2(Ax^* - b) = \xi^*$. Moreover, by 5.15.2, $x^* \neq 0$. Taking subdifferentails of the lasso objective function yields $2A^*(Ax^* - b) + \lambda\text{sgn}(x) = 0$. Since x is non-zero, $\|\text{sgn}(x)\|_\infty = 1$ and so $\|A^*(Ax^* - b)\|_\infty = \lambda/2$. Thus $\lim_{n\to\infty} p_n = 2(Ax^* - b)$ and so p_n converges to ξ^*. $\square$