Chapter 7

Sparse Recovery

In this section, we will take a look at the sparse recovery and sparse linear regression as applications of non-convex optimization. These are extremely well studied problems and find applications in several practical settings. This will be the first of four "application" sections where we apply non-convex optimization techniques to real-world problems.

7.1 Motivating Applications

We will take the following two running examples to motivate the sparse regression problem in two different settings.

Gene Expression Analysis The availability of DNA micro-array gene expression data makes it possible to identify genetic explanations for a wide range of phenotypical traits such as physiological properties or even disease progressions. In such data, we are given say, for n human test subjects participating in the study, the expression levels of a large number p of genes (encoded as a real vector $\mathbf{x}_i \in \mathbb{R}^p$), and the corresponding phenotypical trait $y_i \in \mathbb{R}$. Figure 7.1 depicts this for a hypothetical study on Type-I diabetes. For the sake of simplicity, we are considering cases where the phenotypical trait can be modeled as a real number – this real number may indicate the severity of a condition or the level of some other biological measurement. More expressive models exist in literature where the target phenotypical trait is itself represented as a vector [see for example, Jain and Tewari, 2015].

For the sake of simplicity, we assume that the phenotypical response is linearly linked to the gene expression levels i.e. for some $\mathbf{w}^* \in \mathbb{R}^p$, we have $y_i = \mathbf{x}_i^\top \mathbf{w}^* + \eta_i$ where η_i is some noise. The goal then is to use gene expression data to deduce an estimate for $\mathbf{w}^*$. Having access to the model $\mathbf{w}^*$ can be instrumental in discovering possible genetic bases for diseases, traits etc. Consequently, this problem has significant implications for understanding physiology and developing novel medical interventions to treat and prevent diseases/conditions.

However, the problem fails to reduce to a simple linear regression problem for two important reasons. Firstly, although the number of genes whose expression levels are being recorded is usually very large (running into several tens of thousands), the number of samples (test subjects) is usually not nearly as large, i.e. $n \ll p$. Traditional regression algorithms fall silent in such data-starved settings as they usually expect $n > p$. Secondly, and more importantly, we do not expect all genes being tracked to participate in realizing the phenotype. Indeed, the whole objective of this exercise is to identify a small set of genes which most prominently influence the given phenotype. Note that this implies that the vector $\mathbf{w}^*$ is very sparse. Traditional linear regression cannot guarantee the recovery of a sparse model.

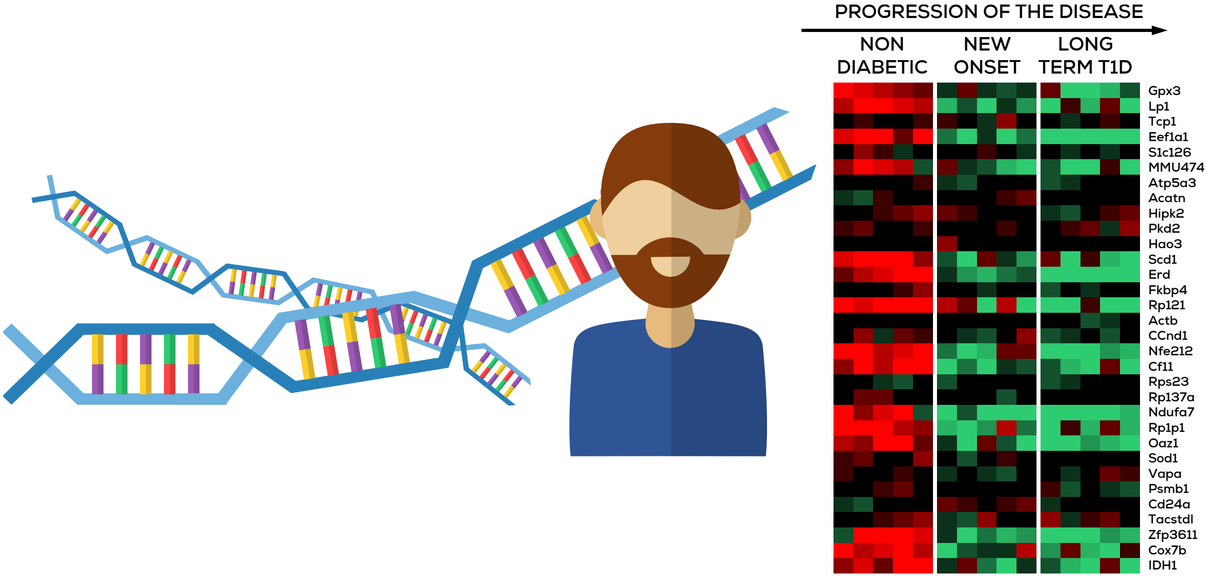

Figure 7.1: Gene expression analysis can help identify genetic bases for physiological conditions. The expression matrix on the right has 32 rows and 15 columns: each row represents one gene being tracked and each column represents one test subject. A bright red (green) shade in a cell indicates an elevated (depressed) expression level of the corresponding gene in the test subject with respect to a reference population. A black/dark shade indicates an expression level identical to the reference population. Notice that most genes do not participate in the progression of the disease in a significant manner. Moreover, the number of genes being tracked is much larger than the number of test subjects. This makes the problem of gene expression analysis, an ideal application for sparse recovery techniques. Please note that names of genes and expression levels in the figure are illustrative and do not correspond to actual experimental observations. Figure adapted from [Wilson et al., 2003].

Sparse Signal Transmission and Recovery The task of transmitting and acquiring signals is a key problem in engineering. In several application areas such as magnetic resonance imagery and radio communication, *linear* measurement techniques, for example, sampling, are commonly used to acquire a signal. The task then is to reconstruct the original signal from these measurements. For sake of simplicity, suppose we wish to sense/transmit signals represented as vectors in $\mathbb{R}^p$. For various reasons (conserving energy, protection against data corruption etc), we may want to not transmit the signal directly and instead, create a *sensing mechanism* wherein a signal $\mathbf{w} \in \mathbb{R}^p$ is encoded into a signal $\mathbf{y} \in \mathbb{R}^n$ and it is $\mathbf{y}$ that is transmitted. At the receiving end $\mathbf{y}$ must be decoded back into $\mathbf{w}$. A popular way of creating sensing mechanisms – also called *designs* – is to come up with a set of n linear functionals $\mathbf{x}_i : \mathbb{R}^p \to \mathbb{R}$ and for any signal $\mathbf{w} \in \mathbb{R}^p$, record the values $y_i = \mathbf{x}_i^\top \mathbf{w}$. If we denote $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^\top$ and $\mathbf{y} = [y_1, \ldots, y_n]^\top$, then $\mathbf{y} = X\mathbf{w}$ is transmitted. Note as a special case that if $n = p$ and $\mathbf{x}_i = \mathbf{e}_i$, then $X = I_{p \times p}$ and $\mathbf{y} = \mathbf{w}$, i.e. we transmit the original signal itself.

If p is very large then we naturally look for designs with $n \ll p$. However, elementary results in algebra dictate that the recovery of $\mathbf{w}$ from $\mathbf{y}$ cannot be guaranteed even if $n = p - 1$. There is irrecoverable loss of information and there could be (infinitely) many signals $\mathbf{w}$ all of which map to the same transmitted signal $\mathbf{y}$ making it impossible to recover the original signal uniquely. A result similar in spirit called the Shannon-Nyquist theorem holds for analog or continuous-time signals. Although this seems to spell doom for any efforts to perform *compressed* sensing and transmission, these negative results can actually be overcome by observing that in several useful settings, the signals we are interested in, are actually very sparse i.e. $\mathbf{w} \in \mathcal{B}_0(s) \subset \mathbb{R}^p$, $s \ll p$. This realization is critical since it allows the possibility of specialized design matrices to be used to transmit sparse signals in a highly compressed manner i.e. with $n \ll p$ but without any loss of information. However the recovery problem now requires a sparse vector to be recovered from the transmitted signal $\mathbf{y}$ and the design matrix X.

7.2 Problem Formulation

In both the examples considered above, we wish to recover a sparse linear model $\mathbf{w} \in \mathcal{B}_0(s) \subset \mathbb{R}^p$ that fits the data, i.e., $\mathbf{y}_i \approx \mathbf{x}_i^\top \mathbf{w}$, hence the name *sparse recovery.* In the gene analysis problem, the support of such a model $\mathbf{w}$ is valuable in revealing the identity of the genes involved in promoting the given phenotype/disease. Similarly, in the sparse signal recovery problem, $\mathbf{w}$ is the (sparse) signal itself.

This motivates the sparse linear regression problem. In the following, we shall use $\mathbf{x}_i \in \mathbb{R}^p$ to denote the features (gene expression levels/measurement functionals). Each feature will constitute a data point. There will be a *response* variable $y_i \in \mathbb{R}$ (phenotype response/measurement) associated with each data point. We will assume that the response variables are being generated using some underlying sparse *model* $\mathbf{w}^* \in \mathcal{B}_0(s)$ (gene influence pattern/sparse signal) as $y_i = \mathbf{x}_i^\top \mathbf{w}^* + \eta_i$ where η_i is some benign noise.

In both the gene expression analysis problem, as well as the sparse signal recovery problem, recovering $\mathbf{w}^*$ from the data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ then requires us to solve the following optimization problem: $\min\limits_{\mathbf{w} \in \mathbb{R}^p, \|\mathbf{w}\|_0 \leq s} \sum_{i=1}^n \left(y_i - \mathbf{x}_i^\top \mathbf{w}\right)^2$. Rewriting the above in more succinct notation gives us

$$\min_{\substack{\mathbf{w} \in \mathbb{R}^p \\ \|\mathbf{w}\|_0 \leq s}} \|\mathbf{y} - X\mathbf{w}\|_2^2, \tag{SP-REG}$$

where $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^\top$ and $\mathbf{y} = [y_1, \ldots, y_n]^\top$. It is common to model the additive noise as *white noise* i.e. $\eta_i \sim \mathcal{N}(0, \sigma^2)$ for some $\sigma > 0$. It should be noted that the sparse regression problem in ((SP-REG)) is an NP-hard problem [Natarajan, 1995].

7.3 Sparse Regression: Two Perspectives

Although we cast both the gene analysis and sparse signal recovery problems in the same framework of sparse linear regression, there are subtle but crucial differences between the two problem settings.

Notice that the problem in sparse signal recovery is to come up with both, a design matrix X, as well as a recovery algorithm $\mathcal{A} : \mathbb{R}^n \times \mathbb{R}^{n \times p} \to \mathbb{R}^p$ such that all sparse signals can be accurately recovered from the measurements, i.e. $\forall \mathbf{w} \in \mathcal{B}_0(s) \subset \mathbb{R}^p$, we have $\mathcal{A}(X\mathbf{w}, X) \approx \mathbf{w}$.

On the other hand, in the gene expression analysis task, we do not have as direct a control over the effective design matrix X. In this case, the role of the design matrix is played by the gene expression data of the n test subjects. Although we may choose which individuals we wish to include in our study, even this choice does not give us a direct handle on the properties of the design matrix. Our job here is restricted to coming up with an algorithm $\mathcal{A}$ which can, given the gene expression data for p genes in n test subjects, figure out a sparse set of s genes which collectively promote the given phenotype i.e. for any $\mathbf{w} \in \mathcal{B}_0(s) \subset \mathbb{R}^p$ and given $X \in \mathbb{R}^{n \times p}$, we desire $\mathcal{A}(X\mathbf{w}, X) \approx \mathbf{w}$.

The distinction between the two settings is now apparent: in the first setting, the design matrix is totally in our control. We may design it to have specific properties as required by the

Algorithm 8 Iterative Hard-thresholding (IHT)

```
Input: Data $X, \mathbf{y}$, step length $\eta$, projection sparsity level $k$
Output: A sparse model $\widehat{\mathbf{w}} \in \mathcal{B}_0(k)$
$\mathbf{w}^1 \leftarrow \mathbf{0}$
for $t = 1, 2, \ldots$ do
    $\mathbf{z}^{t+1} \leftarrow \mathbf{w}^t - \eta \cdot \frac{1}{n} X^\top (X\mathbf{w}^t - \mathbf{y})$
    $\mathbf{w}^{t+1} \leftarrow \Pi_{\mathcal{B}_0(k)}(\mathbf{z}^{t+1})$                //see § 3.1
end for
return $\mathbf{w}^t$
```

recovery algorithm. However, in the second case, the design matrix is mostly given to us. We have no fine control over its properties.

This will make an important difference in the algorithms that operate in these settings since algorithms for sparse signal recovery would be able to make very stringent assumptions regarding the design matrix since we ourselves create this matrix from scratch. However, for the same reason, algorithms working in *statistical learning* settings such as the gene expression analysis problem, would have to work with relaxed assumptions that can be expected to be satisfied by natural data. We will revisit this point later once we have introduced the reader to algorithms for performing sparse regression.

7.4 Sparse Recovery via Projected Gradient Descent

The formulation in ((SP-REG)) looks strikingly similar to the convex optimization problem ((CVX-OPT)) we analyzed in § 2 if we take $f(\mathbf{w}) = \|\mathbf{y} - X\mathbf{w}\|_2^2$ as the (convex) objective function and $\mathcal{C} = \mathcal{B}_0(s)$ as the (non-convex) constraint set. Given this, it is indeed tempting to adapt Algorithm 1 to solve this problem. The only difference would be that the projection step would now have to project onto a non-convex set $\mathcal{B}_0(s)$. However, as we have seen in § 3.1, this can be done efficiently. The resulting algorithm is a variant of the gPGD algorithm (Algorithm 2) that we studied in § 3 and is referred to as *Iterative Hard Thresholding* (IHT) in literature. The IHT algorithm is outlined in Algorithm 8.

It should come as no surprise that Algorithm 8 turns out to be extremely simple to implement, as well as extremely fast in execution, given that only gradient steps are required. Indeed, IHT is a method of choice for practitioners given its ease of use and speed. However, much less is clear about the recovery guarantees of this algorithm, especially since we have already stated that ((SP-REG)) is NP-hard to solve [Natarajan, 1995]. Note that since the problem involves non-convex constraints, Theorem 2.5 no longer applies. This seems to destroy all hope of proving a recovery guarantee, until one observes that the NP-hardness result does not preclude the possibility of solving this problem efficiently when there is special structure in the design matrix X.

Indeed if $X = I_{p \times p}$, it is trivial to recover *any* underlying sparse model $\mathbf{w}^*$ by simply returning $\mathbf{y}$. This toy case actually holds the key to efficient sparse recovery. Notice that when $X = I$, the design matrix is an *isometry* – it completely preserves the geometry of the space $\mathbb{R}^p$. However, one could argue that this is an uninteresting and expensive design with $n = p$. In a long line of illuminating results, which we shall now discuss, it was revealed that even if the design matrix is not a global isometry such as I but a *restricted isometry* that only preserves the geometry of sparse vectors, recovery is possible with $n \ll p$.

The observant reader would be wondering why are we not applying the gPGD analysis from § 3 directly here, and if notions similar to the RSC/RSS notions discussed there make sense here too. We request the reader to read on. We will find that not only do those notions extend here, but have beautiful interpretations. Moreover, instead of directly applying the gPGD analysis (Theorem 3.3), we will see a simpler convergence proof tailored to the sparse recovery problem which also gives a sharper result.

7.5 Restricted Isometry and Other Design Properties

As we observed previously, design matrices such as I_p which preserve the geometry of signals/models seem to allow for recovery of sparse signals. We now formalize this intuition further and develop specific conditions on the design matrix X which guarantee *universal recovery* i.e. for every $\mathbf{w} \in \mathcal{B}_0(s)$, it is possible to uniquely recover $\mathbf{w}$ from the measurements $X\mathbf{w}$.

It is easy to see that a design matrix that *identifies* sparse vectors cannot guarantee universal recovery. Suppose we have a design matrix $X \in \mathbb{R}^{n\times p}$ such that for some $\mathbf{w}_1, \mathbf{w}_2 \in \mathcal{B}_0(s)$ and $\mathbf{w}_1 \neq \mathbf{w}_2$, we get $\mathbf{y}_1 = \mathbf{y}_2$ where $\mathbf{y}_1 = X\mathbf{w}_1$ and $\mathbf{y}_2 = X\mathbf{w}_2$. In this case, it is information theoretically impossible to distinguish between $\mathbf{w}_1$ and $\mathbf{w}_2$ on the basis of measurements made using X i.e. using $\mathbf{y}_1$ (or $\mathbf{y}_2$). Consequently, this design matrix cannot be used for universal recovery since it produces measurements that confuse between sparse vectors. It can be seen[1] that such a design matrix will not identify just one pair of sparse vectors but an infinite set of pairs (indeed an entire subspace) of sparse vectors.

Thus, it is clear that the design matrix must preserve the geometry of the set of sparse vectors while projecting them from a high p-dimensional space to a low n-dimensional space. Recall that in sparse recovery settings, we usually have $n \ll p$. The *Nullspace Property* presented below, and the others thereafter, are formalizations of this intuition. For any subset of coordinates $S \subset [p]$, let us define the set $\mathcal{C}(S) := \left\{\mathbf{w} \in \mathbb{R}^p, \|\mathbf{w}_S\|_1 \geq \|\mathbf{w}_{\overline{S}}\|_1\right\}$. This is the (convex) set of points that place a majority of their weight on coordinates in the set S. Define $\mathcal{C}(k) := \bigcup_{S:|S|=k} \mathcal{C}(S)$ to be the (non-convex[2]) set of points that place a majority of their weight on some k coordinates. Note that $\mathcal{C}(k) \supset \mathcal{B}_0(k)$ since k-sparse vectors put *all* their weight on some k coordinates.

Definition 7.1 (Nullspace Property [Cohen et al., 2009])**.** *A matrix $X \in \mathbb{R}^{n\times p}$ is said to satisfy the null-space property of order k if ker$(X) \cap \mathcal{C}(k) = \{\mathbf{0}\}$, where ker$(X) = \{\mathbf{w} \in \mathbb{R}^p : X\mathbf{w} = \mathbf{0}\}$ is the kernel of the linear transformation induced by X (also called its null-space).*

If a design matrix satisfies this property, then vectors in its null-space are disallowed from concentrating a majority of their weight on any k coordinates. Clearly no k-sparse vector is present in the null-space either. If a design matrix has the null-space property of order $2s$, then it can never identify two s-sparse vectors[3] – something that we have already seen as essential to ensure global recovery. A strengthening of the Nullspace Property gives us the Restricted Eigenvalue Property.

Definition 7.2 (Restricted Eigenvalue Property [Raskutti et al., 2010])**.** *A matrix $X \in \mathbb{R}^{n\times p}$ is said to satisfy the restricted eigenvalue property of order k with constant α if for all $\mathbf{w} \in \mathcal{C}(k)$, we have $\frac{1}{n}\|X\mathbf{w}\|_2^2 \geq \alpha \cdot \|\mathbf{w}\|_2^2$.*

This means that not only are k-sparse vectors absent from the null-space, they actually retain a good fraction of their length after projection as well. This means that if $k = 2s$, then for any $\mathbf{w}_1, \mathbf{w}_2 \in \mathcal{B}_0(s)$, we have $\frac{1}{n}\|X(\mathbf{w}_1 - \mathbf{w}_2)\|_2^2 \geq \alpha \cdot \|\mathbf{w}_1 - \mathbf{w}_2\|_2^2$. Thus, the distance between *any two* sparse vectors never greatly diminished after projection. Such behavior is the hallmark

[1] See Exercise 7.1.

[2] See Exercise 7.2.

[3] See Exercise 7.3.

of an isometry, which preserves the geometry of vectors. The next property further explicates this and is, not surprisingly, called the Restricted Isometry Property.

Definition 7.3 (Restricted Isometry Property [Candès and Tao, 2005]). *A matrix $X \in \mathbb{R}^{n \times p}$ is said to satisfy the restricted isometry property (RIP) of order k with constant $\delta_k \in [0,1)$ if for all $\mathbf{w} \in \mathcal{B}_0(k)$, we have*

$$(1-\delta_k) \cdot \|\mathbf{w}\|_2^2 \leq \frac{1}{n}\|X\mathbf{w}\|_2^2 \leq (1+\delta_k) \cdot \|\mathbf{w}\|_2^2.$$

The above property is most widely used in analyzing sparse recovery and compressive sensing algorithms. However, it is a bit restrictive since it requires the distortion parameters to be of the kind $(1 \pm \delta)$ for $\delta \in [0,1)$. A generalization of this property that is especially useful in settings where the properties of the design matrix are not strictly controlled by us, such as the gene expression analysis problem, is the following notion of restricted strong convexity and smoothness.

Stronger condition would be smooth and strongly convex ||A(x-x*)|| ~ ||x-x*|| for all x (not just sparse). Basically means alpha and beta need to be close for all x.

Definition 7.4 (Restricted Strong Convexity/Smoothness Property [Jain et al., 2014, Jalali et al., 2011]). *A matrix $X \in \mathbb{R}^{n \times p}$ is said to satisfy the α-restricted strong convexity (RSC) property and the β-restricted smoothness (RSS) property of order k if for all $\mathbf{w} \in \mathcal{B}_0(k)$, we have*

$$\alpha \cdot \|\mathbf{w}\|_2^2 \leq \frac{1}{n}\|X\mathbf{w}\|_2^2 \leq \beta \cdot \|\mathbf{w}\|_2^2.$$

The only difference between the RIP and the RSC/RSS properties is that the former forces constants to be of the form $1 \pm \delta_k$ whereas the latter does not impose any such constraints. The reader will notice the similarities in the definition of restricted strong convexity and smoothness as given here and Definition 3.2 where we defined restricted strongly convexity and smoothness notions for general functions. The reader is invited to verify[4] that the two are indeed related.

Indeed, Definition 3.2 can be seen as a generalization of Definition 7.4 to general functions [Jain et al., 2014]. For twice differentiable functions, both definitions can be seen as placing restrictions on the (restricted) eigenvalues of the Hessian of the function.

It is a useful exercise to verify[5] that these properties fall in a hierarchy: RSC-RSS ⇒ REP ⇒ NSP for an appropriate setting of constants. We will next establish the main result of this section: if the design matrix satisfies the RIP condition with appropriate constants, then the IHT algorithm does indeed guarantee universal sparse recovery. Subsequently, we will give pointers to recent results that guarantee universal recovery in gene expression analysis-like settings.

7.6 Ensuring RIP and other Properties

Since properties such as RIP, RE and RSC are so crucial for guaranteed sparse recovery, it is important to study problem settings in which these are satisfied by actual data. A lot of research has gone into explicit construction of matrices that provably satisfy the RIP property.

Random Designs: The simplest of these results are the so-called random design constructions which guarantee that if the matrix is sampled from certain well behaved distributions, then it will satisfy the RIP property with high probability. For instance, the work of Baraniuk et al. [2008] shows the following result:

[4] See Exercise 7.4.

[5] See Exercise 7.5.

Theorem 7.1. *[Baraniuk et al., 2008, Theorem 5.2] Let $\mathcal{D}$ be a distribution over matrices in $\mathbb{R}^{n\times p}$ such that for any fixed $\mathbf{v} \in \mathbb{R}^p, \epsilon > 0$,*

$$\underset{X\sim\mathcal{D}^{n\times p}}{\mathbb{P}}\left[\left|\|X\mathbf{v}\|_2^2 - \|\mathbf{v}\|_2^2\right| > \epsilon \cdot \|\mathbf{v}\|_2^2\right] \leq 2\exp(-\Omega(n))$$

Then, for any $k < p/2$, matrices X generated from this distribution also satisfy the RIP property at order k with constant δ with probability at least $1 - 2\exp(-\Omega(n))$ whenever $n \geq \Omega\left(\frac{k}{\delta^2}\log\frac{p}{k}\right)$.

Thus, a distribution over matrices that, for every *fixed* vector, acts as an almost isometry with high probability, is also guaranteed to, with very high probability, generate matrices that act as a restricted isometry *simulataneously* over all sparse vectors. Such matrix distributions are easy to construct – one simply needs to sample each entry of the matrix independently according to one of the following distributions:

1. sample each entry from the Gaussian distribution $\mathcal{N}(0, 1/n)$.
2. set each entry to $\pm 1/\sqrt{n}$ with equal probability.
3. set each entry to 0 w.p. 2/3 and $\pm\sqrt{3/n}$ w.p. 1/6.

The work of Agarwal et al. [2012] shows that the RSC/RSS properties are satisfied whenever rows of the matrix X are drawn from a sub-Gaussian distribution over p-dimensional vectors with a non-singular covariance matrix. This result is useful since it shows that real-life data, which can often be modeled as vectors being drawn from sub-Gaussian distributions, will satisfy these properties with high probability. This is crucial for sparse recovery and other algorithms to be applicable to real life problems such as the gene-expression analysis problem.

If one can tolerate a slight blowup in the number of rows of the matrix X, then there exist better constructions with the added benefit of allowing fast matrix vector products. The initial work of Candès and Tao [2005] itself showed that selecting each row of a Fourier transform matrix independently with probability $\mathcal{O}\left(k\frac{\log^6 p}{p}\right)$ results in an RIP matrix with high probability. More recently, this was improved to $\mathcal{O}\left(k\log^2 k\frac{\log p}{p}\right)$ in the work of Haviv and Regev [2017]. A matrix-vector product of a k-sparse vector with such a matrix takes only $\mathcal{O}\left(k\log^2 p\right)$ time whereas a dense matrix filled with Gaussians would have taken up to $\mathcal{O}\left(k^2\log p\right)$ time. There exist more involved hashing-based constructions that simultaneously offer reduced sample complexity and fast matrix-vector multiplications [Nelson et al., 2014].

Deterministic Designs: There exist far fewer and far weaker results for deterministic constructions of RIP matrices. The initial results in this direction all involved constructing *incoherent* matrices. A matrix $X \in \mathbb{R}^{n\times p}$ with unit norm columns is said to be μ-incoherent if for all $i \neq j \in [p]$, $\langle X_i, X_j\rangle \leq \mu$. A μ-incoherent matrix always satisfies[6] RIP at order k with parameter $\delta = (k-1)\mu$.

Deterministic constructions of incoherent matrices with $\mu = \mathcal{O}\left(\frac{\log p}{\sqrt{n}\log n}\right)$ are well known since the work of Kashin [1975]. However, such constructions require $n = \widetilde{\Omega}\left(\frac{k^2\log^2 p}{\delta^2}\right)$ rows which is quadratically more than what random designs require. The first result to improve upon these constructions came from the work of Bourgain et al. [2011] which gave deterministic combinatorial constructions that assured the RIP property with $n = \widetilde{\mathcal{O}}\left(\frac{k^{(2-\epsilon)}}{\delta^2}\right)$ for some constant $\epsilon > 0$. However, till date, substantially better constructions are not known.

[6] See Exercise 7.6.

7.7 A Sparse Recovery Guarantee for IHT

We will now establish a convergence result for the IHT algorithm. Although the analysis for the gPGD algorithm (Theorem 3.3) can be adapted here, the following proof is much more tuned to the sparse recovery problem and offers a tighter analysis and several problem-specific insights.

Theorem 7.2. *Suppose $X \in \mathbb{R}^{n\times p}$ is a design matrix that satisfies the RIP property of order $3s$ with constant $\delta_{3s} < \frac{1}{2}$. Let $\mathbf{w}^* \in B_0(s) \subset \mathbb{R}^p$ be any arbitrary sparse vector and let $\mathbf{y} = X\mathbf{w}^*$. Then the IHT algorithm (Algorithm 8), when executed with a step length $\eta = 1$, and a projection sparsity level $k = s$, ensures $\left\|\mathbf{w}^t - \mathbf{w}^*\right\|_2 \leq \epsilon$ after at most $t = \mathcal{O}\left(\log \frac{\|\mathbf{w}^*\|_2}{\epsilon}\right)$ iterations of the algorithm.*

Proof. We start off with some notation. Let $S^* := \text{supp}(\mathbf{w}^*)$ and $S^t := \text{supp}(\mathbf{w}^t)$. Let $I^t := S^t \cup S^{t+1} \cup S^*$ denote the union of the supports of the two consecutive iterates and the optimal model. The reason behind defining this quantity is that we are assured that while analyzing this update step, the two *error vectors* $\mathbf{w}^t - \mathbf{w}^*$ and $\mathbf{w}^{t+1} - \mathbf{w}^*$, which will be the focal point of the analysis, have support within I^t. Note that $|I^t| \leq 3s$. Please refer to the notation section at the beginning of this monograph for the interpretation of the notation $\mathbf{x}_I$ and A_I for a vector $\mathbf{x}$, matrix A and set I.

With $\eta = 1$, we have (refer to Algorithm 8), $\mathbf{z}^{t+1} = \mathbf{w}^t - \frac{1}{n}X^\top(X\mathbf{w}^t - \mathbf{y})$. However, due to the (non-convex) projection step $\mathbf{w}^{t+1} = \Pi_{\mathcal{B}_0(k)}(\mathbf{z}^{t+1})$, applying projection property-O gives us

$$\left\|\mathbf{w}^{t+1} - \mathbf{z}^{t+1}\right\|_2^2 \leq \left\|\mathbf{w}^* - \mathbf{z}^{t+1}\right\|_2^2.$$

Note that none of the other projection properties are applicable here since the set of sparse vectors is a non-convex set. Now, by Pythagoras' theorem, for any vector $\mathbf{v} \in \mathbb{R}^p$, we have $\|\mathbf{v}\|_2^2 = \|\mathbf{v}_I\|_2^2 + \|\mathbf{v}_{\overline{I}}\|_2^2$ which gives us

$$\left\|\mathbf{w}_I^{t+1} - \mathbf{z}_I^{t+1}\right\|_2^2 + \left\|\mathbf{w}_{\overline{I}}^{t+1} - \mathbf{z}_{\overline{I}}^{t+1}\right\|_2^2 \leq \left\|\mathbf{w}_I^* - \mathbf{z}_I^{t+1}\right\|_2^2 + \left\|\mathbf{w}_{\overline{I}}^* - \mathbf{z}_{\overline{I}}^{t+1}\right\|_2^2$$

Now it is easy to see that $\mathbf{w}_{\overline{I}}^{t+1} = \mathbf{w}_{\overline{I}}^* = \mathbf{0}$. Hence we have

$$\left\|\mathbf{w}_I^{t+1} - \mathbf{z}_I^{t+1}\right\|_2 \leq \left\|\mathbf{w}_I^* - \mathbf{z}_I^{t+1}\right\|_2$$

Using the fact that $y = X\mathbf{w}^*$, and denoting $\overline{X} = \frac{1}{\sqrt{n}}X$ we get

$$\left\|\mathbf{w}_I^{t+1} - (\mathbf{w}_I^t - \overline{X}_I^\top\overline{X}(\mathbf{w}^t - \mathbf{w}^*))\right\|_2 \leq \left\|\mathbf{w}_I^* - (\mathbf{w}_I^t - \overline{X}_I^\top\overline{X}(\mathbf{w}^t - \mathbf{w}^*))\right\|_2$$

Adding and subtracting $\mathbf{w}^*$ from the expression inside the norm operator on the LHS, rearranging, and applying the triangle inequality for norms gives us

$$\left\|\mathbf{w}_I^{t+1} - \mathbf{w}_I^*\right\|_2 \leq 2\left\|(\mathbf{w}_I^t - \mathbf{w}_I^*) - \overline{X}_I^\top\overline{X}(\mathbf{w}^t - \mathbf{w}^*)\right\|_2$$

As $\mathbf{w}_I^t = \mathbf{w}^t, \mathbf{w}_I^{t+1} = \mathbf{w}^{t+1}$, observing that $X(\mathbf{w}^t - \mathbf{w}^*) = X_I(\mathbf{w}^t - \mathbf{w}^*)$ gives us

$$\begin{aligned}\left\|\mathbf{w}^{t+1} - \mathbf{w}^*\right\|_2 &\leq 2\left\|(I - \overline{X}_I^\top\overline{X}_I)(\mathbf{w}^t - \mathbf{w}^*)\right\|_2\\ &\leq 2\left(\left\|\mathbf{w}^t - \mathbf{w}^*\right\|_2 - \left\|\overline{X}_I^\top\overline{X}_I(\mathbf{w}^t - \mathbf{w}^*)\right\|_2\right)\\ &\leq 2\delta_{3s}\left\|\mathbf{w}^t - \mathbf{w}^*\right\|_2,\end{aligned}$$

which finishes the proof. The second inequality above follows due to the triangle inequality and the third inequality follows from the fact that RIP implies[7] that for any $|I| \leq 3s$, the smallest eigenvalue of the matrix $\overline{X}_I^\top \overline{X}_I$ is lower bounded by $(1 - \delta_{3s})$. □

We note that this result holds even if the hard thresholding level is set to $k > s$. It is easy to see that the condition $\delta_{3s} < \frac{1}{2}$ is equivalent to the *restricted condition number* (over $3s$-sparse vectors) of the corresponding sparse recovery problem being upper bounded by $\kappa_{3s} < 3$. Similar to Theorem 3.3, here also we require an upper bound on the restricted condition number of the problem. It is interesting to note that a direct application of Theorem 3.3 would have instead required $\delta_{2s} < \frac{1}{3}$ (or equivalently $\kappa_{2s} < 2$) which can be shown to be a harsher requirement than what we have achieved. Moreover, applying Theorem 3.3 would have also required us to set the step length to a specific quantity $\eta = \frac{1}{1+\delta_s}$ while executing the gPGD algorithm whereas while executing the IHT algorithm, we need only set $\eta = 1$.

An alternate proof of this result appears in the work of Garg and Khandekar [2009] which also requires the condition $\delta_{2s} < \frac{1}{3}$. The above result extends to a more general setting where there is additive noise in the model $\mathbf{y} = X\mathbf{w}^* + \boldsymbol{\eta}$. In this setting, it is known (see for example, [Jain et al., 2014, Theorem 3] or [Garg and Khandekar, 2009, Theorem 2.3]) that if the objective function in question (for the sparse recovery problem the objective function is the least squares objective) satisfies the (α, β) RSC/RSS properties at level $2s$, then the following is guaranteed for the output $\widehat{\mathbf{w}}$ of the IHT algorithm (assuming the algorithm is run for roughly $\mathcal{O}(\log n)$ iterations)

$$\left\|\widehat{\mathbf{w}} - \mathbf{w}^*\right\|_2 \leq \frac{3\sqrt{s}}{\alpha} \left\| \frac{X^\top \boldsymbol{\eta}}{n} \right\|_\infty$$

The consistency of the above solution can be verified in several interesting situations. For example, if the design matrix has normalized columns i.e. $\|X_i\|_2 \leq \sqrt{n}$ and the noise η_i is generated i.i.d. and independently of the design X from some Gaussian distribution $\mathcal{N}(0, \sigma^2)$, then the quantity $\left\| \frac{X^\top \boldsymbol{\eta}}{n} \right\|_\infty$ is of the order of $\sigma\sqrt{\frac{\log p}{n}}$ with high probability. In the above setting IHT guarantees with high probability

$$\left\|\widehat{\mathbf{w}} - \mathbf{w}^*\right\|_2 \leq \widetilde{\mathcal{O}}\left(\frac{\sigma}{\alpha}\sqrt{\frac{s \log p}{n}} \right),$$

i.e. $\left\|\widehat{\mathbf{w}} - \mathbf{w}^*\right\|_2 \to 0$ as $n \to \infty$, thus establishing consistency.

7.8 Other Popular Techniques for Sparse Recovery

The IHT method is a part of a larger class of *hard thresholding techniques*, which include algorithms such as Iterative Hard Thresholding (IHT) [Blumensath, 2011], Gradient Descent with Sparsification (GraDeS) [Garg and Khandekar, 2009], and Hard Thresholding Pursuit (HTP) [Foucart, 2011]. Apart from these gradient descent-style techniques, several other approaches have been developed for the sparse recovery problem over the years. Here we briefly survey them.

7.8.1 Pursuit Techniques

A popular non-convex optimization technique for sparse recovery and a few related optimization problems is that of discovering support elements iteratively. This technique is embodied in

[7] See Exercise 7.7.

pursuit-style algorithms. We warn the reader that the popular *Basis Pursuit* algorithm is actually a convex relaxation technique and not related to the other pursuit algorithms we discuss here. The terminology is a bit confusing but seems to be a matter of legacy.

The pursuit family of algorithms includes Orthogonal Matching Pursuit (OMP) [Tropp and Gilbert, 2007], Orthogonal Matching Pursuit with Replacement (OMPR) [Jain et al., 2011], Compressive Sampling Matching Pursuit (CoSaMP) [Needell and Tropp, 2008], and the Forward-backward (FoBa) algorithm [Zhang, 2011].

Pursuit methods work by gradually discovering the elements in the support of the true model vector $\mathbf{w}^*$. At every time step, these techniques add a new support element to an active support set (which is empty to begin with) and solve a traditional least-squares problem on the active support set. This least-squares problem has no sparsity constraints, and is hence a convex problem which can be solved easily.

The support set is then updated by adding a new support element. It is common to add the coordinate where the gradient of the objective function has the highest magnitude among coordinates not already in the support. FoBa-style techniques augment this method by having *backward* steps where support elements that were erroneously picked earlier are discarded when the error is detected.

Pursuit-style methods are, in general, applicable whenever the structure in the (non-convex) constraint set in question can be represented as a combination of a small number of *atoms*. Examples include sparse recovery, where the atoms are individual coordinates: every s-sparse vector is a linear combination of some s of these atoms.

Other examples include low-rank matrix recovery, which we will study in detail in § 8, where the atoms are rank-one matrices. The SVD theorem tells us that every r-rank matrix can indeed be expressed as a sum of r rank-one matrices. There exist works [Tewari et al., 2011] that give generic methods to perform sparse recovery in such structurally constrained settings.

7.8.2 Convex Relaxation Techniques for Sparse Recovery

Convex relaxation techniques have been extremely popular for the sparse recovery problem. In fact they formed the first line of attack on non-convex optimization problems starting with the seminal work of Candès and Tao [2005], Candès et al. [2006], Donoho [2006] that, for the first time, established polynomial time, globally convergent solutions for the compressive sensing problem.

A flurry of work then followed on relaxation-based techniques [Candès, 2008, Donoho et al., 2009, Foucart, 2010, Negahban et al., 2012] that vastly expanded the scope of the problems being studied, the techniques being applied, as well as their analyses. It is important to note that all methods, whether relaxation based or not, have to assume some design property such as NSP/REP/RIP/RSC-RSS that we discussed earlier, in order to give provable guarantees.

The relaxation approach converts non-convex problems to convex problems first before solving them. This approach, applied to the sparse regression problem, gives us the so-called LASSO problem which has been studied extensively. Consider the sparse recovery problem ((SP-REG)).

$$\min_{\substack{\mathbf{w}\in\mathbb{R}^p\\ \|\mathbf{w}\|_0\leq s}} \ \|\mathbf{y}-X\mathbf{w}\|_2^2 .$$

Non-convexity arises in the problem due to the non-convex constraint $\|\mathbf{w}\|_0 \leq s$ as the sparsity operator is not a valid norm. The relaxation approach fixes this problem by changing the constraint to use the L_1 norm instead i.e.

$$\min_{\substack{\mathbf{w}\in\mathbb{R}^p\\ \|\mathbf{w}\|_1\leq R}} \ \|\mathbf{y}-X\mathbf{w}\|_2^2 , \tag{LASSO-1}$$

or by using its regularized version instead

$$\min_{\mathbf{w}\in\mathbb{R}^p} \frac{1}{2n}\left\|\mathbf{y} - X\mathbf{w}\right\|_2^2 + \lambda_n \left\|\mathbf{w}\right\|_1. \qquad \text{(LASSO-2)}$$

The choice of the L_1 norm is motivated mainly by its convexity as well as formal results that assure us that the relaxation gap is small or non-existent. Both the above formulations ((LASSO-1)) and ((LASSO-2)) are indeed convex but include parameters such as R and λ_n that must be tuned properly to ensure proper convergence. Although the optimization problems ((LASSO-1)) and ((LASSO-2)) are vastly different from ((SP-REG)), a long line of beautiful results, starting from the seminal work of Candès and Tao [2005], Candès et al. [2006], Donoho [2006], showed that if the design matrix X satisfies RIP with appropriate constants, and if the parameters of the relaxations R and λ_n are appropriately tuned, then the solutions to the relaxations are indeed solutions to the original problems as well.

Below we state one such result from the recent text by Hastie et al. [2016]. We recommend this text to any reader looking for a well-curated compendium of techniques and results on the relaxation approach to several non-convex optimization problems arising in machine learning.

Theorem 7.3. *[Hastie et al., 2016, Theorem 11.1] Consider a sparse recovery problem* $\mathbf{y} = X\mathbf{w}^* + \boldsymbol{\eta}$ *where the model* $\mathbf{w}^*$ *is* s*-sparse and the design matrix* X *satisfies the restricted-eigenvalue condition (see Definition 7.2) of the order* s *with constant* α*, then the following hold*

1. *Any solution* $\widehat{\mathbf{w}}_1$ *to* ((LASSO-1)) *with* $R = \left\|\mathbf{w}^*\right\|_1$ *satisfies*

$$\left\|\widehat{\mathbf{w}}_1 - \mathbf{w}^*\right\|_2 \leq \frac{4}{\alpha}\sqrt{s}\left\|\frac{X^\top\boldsymbol{\eta}}{n}\right\|_\infty.$$

2. *Any solution* $\widehat{\mathbf{w}}_2$ *to* ((LASSO-2)) *with* $\lambda_n \geq 2\left\|X^\top\boldsymbol{\eta}/n\right\|_\infty$ *satisfies*

$$\left\|\widehat{\mathbf{w}}_1 - \mathbf{w}^*\right\|_2 \leq \frac{3}{\alpha}\sqrt{s}\lambda_n.$$

The reader can verify that the above bounds are competitive with the bounds for the IHT algorithm that we discussed previously. We refer the reader to [Hastie et al., 2016, Chapter 11] for more consistency results for the LASSO formulations.

7.8.3 Non-convex Regularization Techniques

Instead of performing a complete convex relaxation of the problem, there exist approaches that only partly relax the problem. A popular approach in this direction uses L_q regularization with $0 < q < 1$ [Chartrand, 2007, Foucart and Lai, 2009, Wang et al., 2011]. The resulting problem still remains non-convex but becomes a little well behaved in terms of having objective functions that are almost-everywhere differentiable. For instance, the following optimization problem may be used to solve the noiseless sparse recovery problem.

$$\begin{aligned} \min_{\mathbf{w}\in\mathbb{R}^p}\ & \left\|\mathbf{w}\right\|_q, \\ \text{s.t. } & \mathbf{y} = X\mathbf{w} \end{aligned}$$

For noisy settings, one may replace the constraint with a soft constraint such as $\left\|\mathbf{y} - X\mathbf{w}\right\|_2 \leq \epsilon$, or else move to an unconstrained version like LASSO with the L_1 norm replaced by the L_q norm.

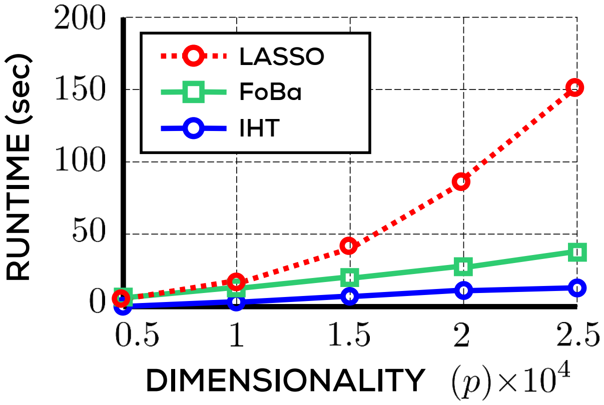

Figure 7.2: An empirical comparison of run-times offered by the LASSO, FoBA and IHT methods on sparse regression problems with varying dimensionality p. All problems enjoyed sparsity $s = 100$ and were offered $n = 2s \cdot \log p$ data points. IHT is clearly the most scalable of the methods followed by FoBa. The relaxation technique does not scale very well to high dimensions. Figure adapted from [Jain et al., 2014].

The choice of the regularization norm q is dictated by application and usually any value within a certain range within the interval $(0, 1)$ can be chosen.

There has been interest in characterizing both the global and the local optima of these optimization problems for their recovery properties [Chen and Gu, 2015]. In general, L_q regularized formulations, if solved exactly, can guarantee recovery under much weaker conditions than what LASSO formulations, and IHT require. For instance, the RIP condition that L_q-regularized formulations need in order to guarantee universal recovery can be as weak as $\delta_{2k+1} < 1$ [Chartrand, 2007]. This is very close to the requirement $\delta_{2k} < 1$ that must be made by any algorithm in order to ensure that the solution even be unique. However, solving these non-convex regularized problems at large scale itself remains challenging and an active area of research.

7.8.4 Empirical Comparison

To give the reader an appreciation of the empirical performance of the various methods we have discussed for sparse recovery, Figure 7.2 provides a comparison of some of these methods on a synthetic sparse regression problem. The graph plots the running time taken by the various methods to solve the same sparse linear regression problem with sparsity $s = 100$ but with dimensionalities increasing from $p = 5000$ to $p = 25000$. The graph indicates that non-convex optimization methods such as IHT and FoBa are far more scalable than relaxation-based methods. It should be noted that although pursuit-style techniques are scalable, they can become sluggish if the true support set size s is not very small since these techniques discover support elements one by one.

7.9 Extensions

In the preceding discussion, we studied the problem of sparse linear regression and the IHT technique to solve the problem. These basic results can be augmented and generalized in several ways. The work of Negahban et al. [2012] greatly expanded the scope of sparse recovery techniques beyond simple least-squares to the more general M-estimation problem. The work of Bhatia et al. [2015] offered solutions to the *robust* sparse regression problem where the responses may be corrupted by an adversary. We will explore the robust regression problem in more detail in § 9. We discuss a few more such extensions below.

7.9.1 Sparse Recovery in Ill-Conditioned Settings

As we discussed before, the bound on the RIP constant $\delta_{3s} < \frac{1}{2}$ as required by Theorem 7.2, effectively places a bound on the *restricted condition number* κ_{3s} of the design matrix. In our case the bound translates to $\kappa_{3s} = \frac{1+\delta_{3s}}{1-\delta_{3s}} < 3$. However, in cases such as the gene expression analysis problem where the design matrix is not totally under our control, the restricted condition number might be much larger than 3.

For instance, it can be shown that if the expression levels of two genes are highly correlated then this results in ill-conditioned design matrices. In such settings, it is much more appropriate to assume that the design matrix satisfies restricted strong convexity and smoothness (RSC/RSS) which allows us to work with design matrices with arbitrarily large condition numbers. It turns out that the IHT algorithm can be modified [see for example, Jain et al., 2014] to work in these ill-conditioned recovery settings.

Theorem 7.4. *Suppose $X \in \mathbb{R}^{n\times p}$ is a design matrix that satisfies the restricted strong convexity and smoothness property of order $2k+s$ with constants α_{2k+s} and β_{2k+s} respectively. Let $\mathbf{w}^* \in B_0(s) \subset \mathbb{R}^p$ be any arbitrary sparse vector and let $\mathbf{y} = X\mathbf{w}^*$. Then the IHT algorithm, when executed with a step length $\eta < \frac{2}{\beta_{2k+s}}$, and a projection sparsity level $k \geq 32\left(\frac{\beta_{2k+s}}{\alpha_{2k+s}}\right)^2 s$, ensures $\left\|\mathbf{w}^t - \mathbf{w}^*\right\|_2 \leq \epsilon$ after $t = \mathcal{O}\left(\frac{\beta_{2k+s}}{\alpha_{2k+s}} \log \frac{\|\mathbf{w}^*\|_2}{\epsilon}\right)$ iterations of the algorithm.*

Note that the above result does not place any restrictions on the condition number or the RSC/RSS constants of the problem. The result also mimics Theorem 3.3 in its dependence on the (restricted) condition number of the optimization problem i.e. $\kappa_{2k+s} = \frac{\beta_{2k+s}}{\alpha_{2k+s}}$. The proof of this result is a bit tedious, hence omitted.

7.9.2 Recovery from a Union of Subspaces

If we look closely, the set $\mathcal{B}_0(s)$ is simply a union of $\binom{p}{s}$ linear subspaces, each subspace encoding a specific sparsity pattern. It is natural to wonder whether the methods and analyses described above also hold when the vector to be recovered belongs to a general union of subspaces. More specifically, consider a family of linear subspaces $\mathcal{H}_1, \ldots, \mathcal{H}_L \subset \mathbb{R}^p$ and denote the union of these subspaces by $\mathcal{H} = \bigcup_{i=1}^L \mathcal{H}_i$. The restricted strong convexity and restricted strong smoothness conditions can be appropriately modified to suit this setting by requiring a design matrix $X : \mathbb{R}^p \to \mathbb{R}^n$ to satisfy, for every $\mathbf{w}_1, \mathbf{w}_2 \in \mathcal{H}$,

$$\alpha \cdot \|\mathbf{w}_1 - \mathbf{w}_2\|_2^2 \leq \|X(\mathbf{w}_1 - \mathbf{w}_2)\|_2^2 \leq L \cdot \|\mathbf{w}_1 - \mathbf{w}_2\|_2^2$$

It turns out that IHT, with an appropriately modified projection operator $\Pi_{\mathcal{H}}(\cdot)$, can ensure recovery of vectors that are guaranteed to reside in a small union of low-dimensional subspaces. Moreover, a linear rate of convergence, as we have seen for the IHT algorithm in the sparse regression case, can still be achieved. We refer the reader to the work of Blumensath [2011] for more details of this extension.

7.9.3 Dictionary Learning

A very useful extension of sparse recovery, or sparse *coding* emerges when we attempt to learn the design matrix as well. Thus, all we are given are observations $\mathbf{y}_1, \ldots, \mathbf{y}_m \in \mathbb{R}^n$ and we wish to learn a design matrix $X \in \mathbb{R}^{n\times p}$ such that the observations $\mathbf{y}_i$ can be represented as sparse combinations $\mathbf{w}_i \in \mathbb{R}^p$ of the columns of the design matrix i.e. $\mathbf{y}_i \approx X\mathbf{w}_i$ such that $\|\mathbf{w}_i\|_0 \leq s \ll p$. The problem has several applications in the fields of computer vision and signal processing and has seen a lot of interest in the recent past.

The alternating minimization technique where one alternates between estimating the design matrix and the sparse representations, is especially popular for this problem. Methods mostly differ in the exact implementation of these alternations. Some notable works in this area include [Agarwal et al., 2016, Arora et al., 2014, Gribonval et al., 2015, Spielman et al., 2012].

7.10 Exercises

Exercise 7.1. *Suppose a design matrix $X \in \mathbb{R}^{n \times p}$ satisfies $X\mathbf{w}_1 = X\mathbf{w}_2$ for some $\mathbf{w}_1 \neq \mathbf{w}_2 \in \mathbb{R}^p$. Then show that there exists an entire subspace $\mathcal{H} \subset \mathbb{R}^p$ such that for all $\mathbf{w}, \mathbf{w}' \in \mathcal{H}$, we have $X\mathbf{w} = X\mathbf{w}'$.*

Exercise 7.2. *Show that the set $\mathcal{C}(k) := \bigcup_{S:|S|=k} \mathcal{C}(S)$ is non-convex.*

Exercise 7.3. *Show that if a design matrix X satisfies the null-space property of order $2s$, then for any two distinct s-sparse vectors $\mathbf{v}^1, \mathbf{v}^2 \in \mathcal{B}_0(s)$, $\mathbf{v}^1 \neq \mathbf{v}^2$, it must be the case that $X\mathbf{v}^1 \neq X\mathbf{v}^2$.*

Exercise 7.4. *Show that the RSC/RSS notion introduced in Definition 7.4 is equivalent to the RSC/RSS notion in Definition 3.2 defined in § 3 for an appropriate choice of function and constraint sets.*

Exercise 7.5. *Show that RSC-RSS ⇒ REP ⇒ NSP i.e. a matrix that satisfies the RSC/RSS condition for some constants, must satisfy the REP condition for some constants which in turn must force it to satisfy the null-space property.*

Exercise 7.6. *Show that every μ-incoherent matrix satisfies the RIP property at order k with parameter $\delta = (k-1)\mu$.*

Exercise 7.7. *Suppose the matrix $X \in \mathbb{R}^{n \times p}$ satisfies RIP at order s with constant δ_s. Then show that for any set $I \subset [p], |I| \leq s$, the smallest eigenvalue of the matrix $X_I^\top X_I$ is lower bounded by $(1 - \delta_s)$.*

Exercise 7.8. *Show that the RIP constant is monotonic in its order i.e. if a matrix X satisfies RIP of order k with constant δ_k, then it also satisfies RIP for all orders $k' \leq k$ with $\delta_{k'} \leq \delta_k$.*

7.11 Bibliographic Notes

The literature on sparse recovery techniques is too vast for this note to cover. We have already covered several directions in §§ 7.8 and 7.9 and point the reader to references therein.