Fast and Scalable:

A Survey on Sparse Approximation Methods

Mehrdad Yaghoobi, Mike E. Davies

The University of Edinburgh, Technical Report
Tr.ED.IDCom.SMALL.12.09.n01

December 9, 2009

Abstract

Sparse approximation of signals, using a linear generative model, is the subject of this report. The aim of sparse approximation is to find an approximate representation of the given signal using few elementary functions. Using a minimal representation of the signal improves the performance of a wide range of signal processing applications. Denoising, coding, deblurring, sampling and inpainting are just some examples of these applications.

The sparse approximation problem is formulated as an optimization problem which is generally difficult to solve. It can not exactly be solved in a polynomial time. Fortunately, it is possible to relax it to other optimization problems which are (approximately) solvable. A wide range of optimization techniques have been used to solve such problems. These techniques scale differently with the size of problem. Here we are interested to the fast and scalable optimization methods.

Alternatively there exist other algorithms which find acceptable solutions, without directly minimizing any objective. It has been observed that using the sparse vectors, found by practical sparse approximation methods, improve the performance of mentioned applications, even though they are not the *sparsest* solutions. These algorithms have been considered as alternatives to the direct minimization techniques when they are simpler in implementation or guaranteed to provide sparse solutions.

The aim of this report is to investigate the potentially scalable sparse approximation methods. Although it is difficult to cover all such algorithms, most of the practical algorithms will be explored here. In this framework, the high computational or heavy memory demand methods can not efficiently be implemented on a large scale problem. The difficulties in the implementation of the sparse approximation methods in this setting will be highlighted here. Ideally we prefer to use the methods which computationally scale linearly or log-linearly, i.e. $\mathcal{O}(n \log(n))$. Unfortunately it is not possible for the sparse approximation methods. A brief complexity comparison between different greedy methods can be found in the last part of this report.

When we want to find a *sparser* solution, which can not be provided using a simple sparse approximation method and a structured dictionary, we should choose a parallelizable sparse approximation method. This means that the algorithm has to be implemented using a multicore processing unit. This is also discussed briefly, while a more comprehensive investigation is left for the future work.

Contents

Introduction

Sparse coding is a minimal representation of a given signal. This minimality can be measured as the sparsity of representation. One can classify the sparse coding problems as the sparse approximation or the sparse representation. A sparse representation is sometimes called an exact sparse representation. Such a sparse code can be found using an optimization problem, which we formulate in Chapter 1. This formulation includes a linear generative model, called a dictionary, which can present any given signal. To have the flexibility of choosing a sparse code, we need an overcomplete dictionary, i.e. the dictionary which has more elementary functions, called atoms, than a basis. The dictionary has to be selected such that it (approximately) provides a sparse representation. In a simple setting this can be done by combining some bases, e.g. Fourier basis, Wavelet, Curvelet.

A dictionary can be adapted to a given set of training data using a dictionary learning algorithm. The sparse approximation is an elementary part of the most dictionary learning problems. In this framework, we need to repeatedly find the sparse approximations of the given set of signals using a (partially) adapted dictionary. For a large scale dictionary adaptation, we need a scalable sparse approximation method. The sparse approximation formulations are thus extended to the sparse coding in the matrix form. This also provides an extra flexibility to find matrices with different sparsity patterns along the column or the row directions.

A common technique for solving the sparse coding problem often is to minimize an objective, subject to a constraint. The constraint can be removed when the coefficients have only to be admissible. Various optimization methods have been introduced to solve different formulations of the sparse approximation problem. The size of sparse coding problems is often such that some optimization techniques are not tractable. Although some linear or quadratic programming and stochastic sampling methods are tractable for small and medium size problems, they are too slow for the large problems. In contrast, the gradient descent based methods, which might be considered as slow for small problems, are good candidates for solving large scale sparse coding problems.

Direct optimization of the sparse coding problem is not the only approach to find sparse representations. It is sometimes preferred in practice to solve the optimization problem using a greedy method. These greedy methods gradually increase the selected support of the coefficient vector to reduce the approximation/representation errors. These methods are especially useful when the problem size is large such that applying other optimization methods are not tractable.

Chapter 2 briefly reviews popular sparse approximation algorithms. As these algorithms are numerous, it is difficult to completely cover them in a short technical report. It is thus preferred, firstly to classify different algorithms based on their approaches to the problem, then a brief explanation about the motivations and the applications of the approach are presented. The approaches, which have been used more often by the researchers, are explored in more detail.

Chapter 3 will explore the issues in scalability of the algorithms. When the computational complexity of an operation scales quadratically with the size of problem and the operation

has to be implemented at each iteration, total complexity of the algorithm does not allow a tractable implementation on a standard computer. On the other hand, the memory usage of the operator is also very important, if the operator needs to access the whole data at once. Such an operator is scalable if it is implementable using a set of smaller size operators. An example of such an operator is matrix-vector multiplication which can be implemented by breaking down the matrix to some disjoint set of columns and implementing each block separately to the corresponding part of the vector and some post operations. Such an operator can be implemented in parallel in a multicore/multiprocessor computer or in a Graphical Processing Unit (GPU), which accelerates the implementation of the operator.

An iterative algorithm is generally fast if each iteration is cheap and the algorithm converges quickly. Hence another important property of an algorithm, which has to be investigated, is its convergence. The convergence of only few methods have been analyzed analytically. Alternatively the convergence can be investigated empirically using a set of reference problems. Such comparative studies are explored in section 3.2.

Chapter 1

Sparse Coding Formulation

This chapter briefly explores different formulations for the sparse coding problem. These formulations are often presented using some optimization problems, which generally are not solvable, i.e. an exhaustive search is the only approach to solve such problems. Different relaxed formulations for such challenging problems will be presented in this chapter. As one aim of this technical report is to find a potentially scalable sparse approximation method for dictionary learning, the matrix form of the sparse approximation problem, which includes simultaneously sparse approximation as a special case, will also be presented here.

1.1 Sparse Coding Optimization Formulations

The aim of sparse coding is to represent a signal exactly or approximately by few coefficients. Let $\mathbf{D} \in \mathbb{R}^{d \times N}$, $\mathbf{y} \in \mathbb{R}^d$ and $\mathbf{x} \in \mathbb{R}^N$ be the generator matrix (or dictionary [1]), the signal and coefficient vectors respectively. The linear generative model is now formulated as,

$$\mathbf{y} = \mathbf{D}\mathbf{x}. \tag{1.1}$$

We assume that $\mathbf{D}$ is full rank ($\text{rank}(D) = \min(d, N)$). In this framework when $d = N$, the exact coefficient vector is uniquely found by the inverse operator of $\mathbf{D}$, $\mathbf{x} = \mathbf{D}^{-1}\mathbf{y}$. When the model is over-determined $d > N$, one can choose a full rank $\mathbf{D}_r \in \mathbb{R}^{d \times d}$, by using d rows of $\mathbf{D}$, and find $\mathbf{x}$ by using $\mathbf{D}_r^{-1}$. The underdetermined inverse problem ($d \leq N$), which is the main focus of this report, does not have a unique solution, which means that the number of equations are less than the number of unknown parameters. To resolve the ambiguity, different constraints have been proposed to impose prior information over the coefficients. The most well-known constraint is the minimum ℓ_2 norm, which has successfully been used for decades. It can be interpreted as imposing a Gaussian assumption on the probability density function (pdf) of coefficients, which is an optimal assumption for many applications. A minimum ℓ_2 norm representation can be calculated very fast using a *linear* operator. The inverse operator $\mathbf{D}^\dagger$ is called *pseudoinverse* and can be found by,

$$\mathbf{D}^\dagger = \mathbf{D}^T(\mathbf{D}\mathbf{D}^T)^{-1}. \tag{1.2}$$

An issue with using minimum ℓ_2 representation is that the coefficients are mostly non-zero. Although it is useful for certain applications, for example when we have erasure or noise [2] in the model, the minimum ℓ_2 overcomplete representation is not the optimal representation for a significant class of signal processing applications. Instead, one can use a sparsity penalty $\mathcal{J}(.)$ and find the sparsest representation. The signal representation is then formulated by,

$$\min_{\mathbf{x} \in \{\tau : \mathbf{y} = \mathbf{D}\tau\}} \mathcal{J}(\mathbf{x}). \tag{1.3}$$

In the ideal case, the operator $\mathcal{J}(.)$ counts the number of non-zero components. However the optimization problem (1.3) using such a sparsity measure, which is called ℓ_0, is an NP-hard problem, in general [3]. Finding the solution of these problems is computationally difficult, even in a medium size problem, and it can only in general be done using an exhaustive search. Another approach is to apply an optimization technique to reduce ℓ_0, subject to the constraint proposed in (1.3) [4, 5], which can only find a sparser representation than the initial solution. Alternatively a series of smoothed objectives, which converge to ℓ_0 in the limit, can be optimized iteratively [6]. In practice better local minima are yielded using smoothed objectives.

To find an acceptably sparse representation, on can use a relaxed sparsity measure, see for example [7] and [8] and references therein for the generalization of the sparsity measure. The relaxed sparsity measure is not necessarily smooth and is often fixed during sparse approximation. A common relaxed $\mathcal{J}(.)$ is $\ell_p^p(\mathbf{x}) := \sum_{1\leq i\leq N} |x_i|^p$, where x_i is the i^{th} element of $\mathbf{x}$ and $p \leq 1$. A special case, where $p = 1$, is particularly interesting since the problem (1.3) for $p = 1$ is convex and can be solved using different convex optimization methods. The global minimum[1] is then found using these optimization methods. Furthermore, the analysis of the optimization *methods* are easier using ℓ_1 sparsity measure. The sufficient conditions, under which the solutions of the sparse representation using ℓ_1 and ℓ_0 are equivalent, are investigated in [9, 10].

The set of K-sparse vectors are unbounded, which means it has infinitely large members in an ℓ_p norm space. By letting an upper bound on the set of K-sparse signals, here $\|\mathbf{x}\|_\infty < c$, the ℓ_1 penalty $\frac{1}{K}\|\mathbf{x}\|_1$ is the "convex envelope " [11] of the non-convex ℓ_0 [12], over the bounded K-sparse set. There is thus no better convex approximation for an ℓ_0 objective in this sense. Using a more accurate approximation for the objective, leads to a non-convex optimization problem. Various methods for optimizing such an objective have been introduced [13–16]. Although there is no easy way to exactly solve the sparse representation problem using this class of sparsity measures, in practice the sparse vectors found by these methods are sparser than ℓ_1 sparse representation. A slightly different sparsity measure is the logarithmic sparsity measure. It has some useful properties which facilitate its minimization.

$$\mathcal{J}_{log}(\mathbf{x}) = \sum_{1\leq i\leq N} \log x_i^2 \tag{1.4}$$

This is sometimes called *Gaussian entropy* [7, 14].

1.1.1 Sparse Approximation

The exact sparse coding problem introduced in (1.3) is for a noise-free model. In practice, it is often important to consider the noise effect in the model. The noise is often introduced as an extra additive term. The signal generative model is then presented by,

$$\mathbf{y} = \mathbf{D}\mathbf{x} + \mathbf{n}, \tag{1.5}$$

where $\mathbf{y}$, $\mathbf{x}$ and $\mathbf{D}$ are as before and $\mathbf{n}$ is the noise vector. Based on the distribution of noise in the model (1.5), one can define a measure on the signal space. When the noise has Gaussian or Laplace distribution, the expectation of the noise can empirically be calculated using ℓ_2 or ℓ_1 norms respectively. The ℓ_2 norm has often been used in the sparse coding problem, which will also be used here as the error measure. One can also assume the model mismatch as the noise in the proposed model in (1.5). In this framework, an underdetermined signal *approximation* can be formulated by,

$$\mathbf{x} \in \{\forall\theta : \|\mathbf{y} - \mathbf{D}\theta\|_2 \leq \epsilon\} \tag{1.6}$$

[1]Because the objective is not strictly convex, it could have non-unique solutions. Under a mild condition, which is often satisfied by the sparse representation settings, the solution is unique.

where ϵ is a constant. The problem is the same as (1.1), using $\epsilon = 0$. (1.6) is also an underdetermined system and the solution space has more than one element. By minimizing a strictly convex objective, e.g. $\ell_p : 1 < p$, over this convex set we can find the *unique* solution. The minimum ℓ_2 overcomplete approximation has been used for denoising, parameter estimation, system identification and classification. The minimization of the ℓ_2-norm over (1.6) can be solved analytically using the *regularized pseudo inverse* operator defined by,

$$\mathbf{D}^\dagger = \mathbf{D}^T(\mathbf{D}\mathbf{D}^T + \epsilon^2 I)^{-1}. \tag{1.7}$$

This operator is often preferred over (1.2) in practice, not only because it considers the noise effect, but also because it can solve the ambiguity caused by any singularity of $\mathbf{D}\mathbf{D}^T$, when $\mathbf{D}$ is rank deficient.

Like the noise-free model, the *linear* operator (1.7) generally finds a non-sparse solution. A sparsity measure can be minimized, with the constraint (1.6) to find a sparser approximation. Although ℓ_1 is not strictly-convex, it can be shown that the following optimization problem, called Basis Pursuit DeNoising (BPDN), has a unique solution,

$$\min_{\|\mathbf{y}-\mathbf{D}\mathbf{x}\|_2 \leq \epsilon} \|\mathbf{x}\|_1. \tag{1.8}$$

The dual representation of BPDN, called LASSO [17], is defined by,

$$\min_{\|\mathbf{x}\|_1 \leq \tau} \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2 \tag{1.9}$$

There is an injective mapping between ϵ and τ such that BPDN and LASSO have the same solutions. These problems are convex and can be solved exactly by using an appropriate convex optimization method. The solutions of BPDN and LASSO are sparse and denoised[2].

Sometimes it is useful to extend the problems (1.8) and (1.9) by using another sparsity measure. Although the problem is no longer convex, the local solutions, which can be found using some of the algorithms presented in Part 2, are often sparser.

(1.8) and (1.9) are constrained optimization problems. There are many effective optimization methods which can only be applied to the non-constrained problems. By using the Lagrangian multipliers method, we can generate an unconstrained problem. The optimization problem is now formulated by,

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1, \tag{1.10}$$

where λ is the Lagrangian multiplier. The sparsity of the approximation can be modified by changing λ. Although this optimization problem is not strictly-convex, it has a *unique* solution [20, 21, Proposition 3.1]. This can be proved by showing that the quadratic part is strictly-convex, the remaining part $\|\mathbf{x}\|_1$ is convex and the objective is unbounded when $\|\mathbf{x}\| \to \infty$ [22, Proposition 2.5.6]. The uniqueness of the solution is a necessary requirement for the Perfect (Exact) Recovery Problem [23]. It has been shown that the sparse representation of a signal is unique if the signal is sparse enough and the dictionary satisfies the Exact Recovery Condition (ERC)[3].

This change in definition significantly increases the number of algorithms that can be applied to solve the problem. For example most of the (sub-)gradient descent methods can now be applied to (1.10), see [24, 25]. Therefore (1.10) is the most desirable formulation for the sparse approximation problem.

[2]The optimality of the solutions using an orthogonal dictionary is guaranteed [18]. This framework has been used in the overcomplete setting. Promising results have been reported, for example, in [19].

[3]ERC of a set of indices Λ is defined by $ERC(\Lambda) := 1 - \max_{\omega \notin \Lambda} \|\mathbf{D}_\Lambda^\dagger \mathbf{d}_\omega\|_1$, where $\mathbf{D}_\Lambda$ is the matrix generated using the atoms indexed by Λ [21].

Similar to the sparse representation problem, one can generalize the sparse approximation problem by using a different $\mathcal{J}(.)$ as the sparsity measure. The generalized form of (1.8) is formulated by,

$$\min_{\|\mathbf{y}-\mathbf{D}\mathbf{x}\|_2 \leq \epsilon} \mathcal{J}(\mathbf{x}), \tag{1.11}$$

and the generalized form of (1.10) is formulated by,

$$\min_{\mathbf{x}} \|\mathbf{y}-\mathbf{D}\mathbf{x}\|_2^2 + \lambda \mathcal{J}(\mathbf{x}). \tag{1.12}$$

If $\mathcal{J}(.)$ is non-convex, e.g. $\ell_p : p < 1$, sparse approximation problems (1.11) and (1.12) have numerous local minima and the global solution can not easily be found, in general[4]. In practice, it is observed that (1.12) for $\ell_p : p < 1$ often converges faster and/or finds sparser solutions [14–16].

Debiasing:

The solutions of the problems (1.11) and (1.12), when $\mathcal{J}(.) \neq \ell_0$, are always biased [8]. It means there are better approximations with the same sparsity pattern. This can be compensated for by using a post processing step, called de-biasing. In this process the signal is orthogonally projected onto the space selected by the non-zero coefficients. Let $\mathbf{D}_I$ be the dictionary composed by using the selected atoms in the approximation. The orthogonal projection can be found using the linear operator pseudoinverse, which was defined in (1.2). Because $\mathbf{D}_I$ depends on the sparsity pattern, the calculation of $\mathbf{D}_I^\dagger$ can not be done a priori. This error is often reduced using a sparsity measure which is closer to ℓ_0. This is also another reason that ℓ_p and logarithmic sparsity measures are preferred to be used in some practical applications of the sparse approximations [26].

1.2 Sparse Matrix Coding

This section generalizes the sparse coding problem from the vector space to the matrix space. It is closely related to the simultaneous sparse approximation/representation [27], dictionary learning [28], sparse source separation [29] and structured dictionary learning [30–32]. Let $\mathbf{Y} \in \mathbb{R}^{d\times L}$, $\mathbf{X} \in \mathbb{R}^{N\times L}$ and $\mathbf{D} \in \mathbb{R}^{d\times N}$ be the signal matrix, the coefficient matrix and the dictionary, respectively. When $d < N$ and $\mathbf{D}$ is full-rank, the underdetermined linear generative model is defined by,

$$\mathbf{Y} = \mathbf{D}\mathbf{X}, \tag{1.13}$$

and the noisy linear generative model is also defined by,

$$\mathbf{Y} = \mathbf{D}\mathbf{X} + \mathbf{N}, \tag{1.14}$$

where $\mathbf{N} \in \mathbb{R}^{d\times L}$ is the noise (or the model mismatch) matrix. Given $\mathbf{Y}$ and $\mathbf{D}$, the solution spaces for the problems (1.13) and (1.14) are respectively defined as:

$$\Lambda_{exact} := \{\forall \mathbf{\Theta} : \mathbf{Y} = \mathbf{D}\mathbf{\Theta}\}, \tag{1.15}$$

and

$$\Lambda_{noisy} := \{\forall \mathbf{\Theta} : \|\mathbf{Y} - \mathbf{D}\mathbf{\Theta}\|_F \leq \epsilon\}, \tag{1.16}$$

[4]The term *in general* is used to note that under certain conditions (1.12) and (1.10) share the solution support. In this case, the solution support of (1.12) could obviously be found by solving (1.10). An extra step is necessary to find the coefficient magnitudes by solving a reduced order optimization problem, which has a unique solution.

where $\|.\|_F$ is the Frobenius norm[5] and $\epsilon \in \mathbb{R}^+$. These convex sets have more than one element each, as a result of underdetermination of the generating model. We can now impose extra constraints on the model to find desired solutions. Let the ℓ_p norm, for $p \geq 1$, be defined by,

$$\ell_p(\mathbf{\Theta}) = (\sum_{i,j} |\theta_{i,j}|^p)^{1/p}, \tag{1.17}$$

where $\theta_{i,j}$ is the (i,j) element of $\mathbf{\Theta}$. ℓ_p is a norm in the matrix space. $B_{\ell_p}(\gamma) = \{\mathbf{\Theta} : \ell_p(\mathbf{\Theta}) \leq \gamma\}$, called the ℓ_p ball, is thus closed and convex. Using a minimum ℓ_p constraint over Λ_{exact} and Λ_{noisy}, the cardinality of the solution sets are reduced to one, which is a similar result to the vector form of sparse approximation. A special case of this problem is when $p = 2$, where the solution can be found using the *linear* operator introduced in (1.2). (1.17) for a $p < 1$ generates a non-convex objective and (1.17) is no longer a norm. Similar to the vector space, $\ell_p^p(.) : p \leq 1$ generates a sparsity measure for the matrix vector space by the following formula,

$$\mathcal{J}_p(\Theta) = \sum_{i,j} |\theta_{i,j}|^p, \tag{1.18}$$

The sparse matrix representation or approximation is then defined by minimizing $\mathcal{J}(\mathbf{X}) = \mathcal{J}_p(\mathbf{X})$, such that $\mathbf{X}$ lies in Λ_{exact} or Λ_{noisy} respectively. The sparsity measure (1.18) is an element-wise operator. In next chapter, it will be shown how this separability facilitates sparse matrix coding.

A variation of sparse matrix approximation can also be formulated using a Lagrangian multiplier λ as follows,

$$\min_{\mathbf{X}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 + \lambda \mathcal{J}(\mathbf{X}). \tag{1.19}$$

An advantage of the formulation (1.19) over minimizing $\mathcal{J}(.)$ over Λ_{noisy}, is that when $\mathcal{J}(.)$ is a column-wise operator, e.g. (1.18), it can be minimized column by column, using a standard sparse approximation method.

1.2.1 Structured Sparsity

No sparsity pattern is proposed in the definition of $\mathcal{J}(.)$ in (1.18), which means that the value of $\mathcal{J}(.)$ does not change by relocating the non-zero elements. Such a pattern is often desirable when natural signals are sought. Simultaneous sparse, tree and harmonic structures are some examples of such sparsity patterns. In this framework a matrix $\mathbf{X}$ with a minimum number of non-zero columns is sought. The following definition for $\mathcal{J}(.)$ has been used for the simultaneous sparse coding [27, 33–35],

$$\mathcal{J}_{p,q}(\mathbf{\Theta}) := \sum_j (\sum_i |\theta_{i,j}|^q)^{\frac{p}{q}}, \tag{1.20}$$

where $0 < p \leq 1 \leq q$. By letting $q \geq 1$, $\mathcal{J}_{p,q}(\mathbf{\Theta}) = \sum_j (\|\theta_j\|_q)^p$. A minimum non-zero columns can be found by choosing $0 < p \leq 1$, which promotes the sparsity of $[\|\theta_j\|_q]_j$. Although it is possible to use any $q \geq 1$, particularly $q \to \infty$, it is preferred in practice to use $q \in \{1, 2\}$, which also provides noise robustness. **Note that the sparsity measure $\mathcal{J}_{p,q}(.)$ defined in (1.20) is not an element-wise operator, when $p \neq q$. Here the conventional sparse approximation methods can not directly be used for this problem.** The dictionary learning problem, using such a sparsity measure, has be presented, for example, in [28].

[5]Frobenius norm is the ℓ_2 norm of the matrix vector space and defined by $\|\mathbf{X}\|_F = |\langle \mathbf{X}, \mathbf{X} \rangle|^{1/2}$, where $\langle ., . \rangle$ is the inner-product of the matrix space which is defined by $\langle \mathbf{X}, \mathbf{Y} \rangle := \text{tr}\{\mathbf{X}^T\mathbf{Y}\}$

Chapter 2

Sparse Coding Algorithms

The sparse approximation methods are explored in the following section by starting with an overview on different approaches to the problem. Some greedy and gradient descent based methods are then introduced with an introductory presentation of the majorization minimization method (MM). This technique and the gradient projection method are the bases of most fast sparse approximation methods, which will be explored in the next sections in more details.

2.1 Sparse Approximation Methods

Sparse coding methods can be classified based on their approaches to the problem. Some of these classes are as follows,

1. *Greedy pursuit:* These methods start with a coarse approximation and gradually refine the approximation by changing the selected set of atoms and the magnitudes of the selected coefficients. These methods include Matching Pursuit (MP) [1], Orthogonal MP (OMP) [3,36] and their variations like Orthogonal Least Square (OLS) [37], also called Optimized OMP (OOMP) [38], Gradient Pursuit (GP) [5] and Stagewise OMP (StOMP) [39]. Slightly different methods in this class are the greedy methods for convex relaxed sparse approximation (1.10), called polytopes faces pursuit [40–42].

2. *Convex and non-convex optimization:* All methods that *directly* minimize the problems (1.8) and non-convex versions of that, (1.11) or (1.12). When the problem is convex, the sparse approximations can be found using standard linear and quadratic programming techniques [8, 11][1]. These methods are not very efficient for large-scale problems. Other optimization methods, like gradient descent based methods and regression methods, are often preferred for large scale problems, see for example [24, 43–45]. The objective that we want to minimize becomes non-convex using any non-convex sparsity measures. Finding the global minimum of such an objective is difficult in general. Some methods are proposed to find a local minimum of such optimization problems [13–16]. It has practically been shown that the local minimum is often sparser for the same approximation errors, which can justify the use of such methods [15].

3. *Based on Stochastic Modeling:* These methods are based on inducing some prior distributions onto the coefficient vectors, which promote sparsity of the representations. They are often based on the maximum a posteriori (MAP) framework, in which the Bayesian

[1] Some Matlab® implementations of such methods can be found in the following packages: 1- *Atomizer*, `http://sparselab.stanford.edu/atomizer`, 2- ℓ_1 *Magic*: `http://www.l1-magic.org`

inference has been used to calculate the posteriori distribution, see for example [46–48]. They can generally be classified in the class of non-convex optimization methods and have only aspire to finding the true MAP estimate.

4. *Exhaustive Search:* This method is only tractable when the size of problem is small or some prior information, for example about the support of coefficient vectors or the subspace in which the signal lies, is given. The complexity of the problem can be reduced using cutting-plane techniques [49].

Although this classification is neither rigid, while some methods might fit into more than one classes, nor complete, while some methods do not lie on any classes, it gives us a perspective of the sparse coding methods. Some the common convex/non-convex optimization methods and greedy methods have been explored in this chapter. The aim of this report is to review the methods which can be implemented on the large problems. The exhaustive search methods will not be reviewed in this report as they are hardly scalable, while the stochastic modeling based methods will only be considered as a subset of optimizing non-convex penalty functions.

2.2 Greedy Methods

Greedy methods are introduced to find an acceptable sparse approximation using an iterative scheme. In each iteration of the algorithm, some atoms are entered in to the support by choosing non-zero coefficients and the values of coefficients are updated, which is the forward step, and then some atoms might be deselected from the support, which is the backward step. In the simplest case one atom is added to the current support in the forward step and there is no backward step. This process can be extended by applying different forward and backward steps. The most famous method, called matching pursuit (MP) [1], is inspired by the greedy regression methods. Because MP is simple to implement and it is very fast (per iteration), it has been investigated in detail, see for example [50]. Some variations of MP are reviewed in [5], and their computational complexity are compared, which will be discussed below. A greedy method with a backward step is StOMP [39], which deselects the atoms with small contributions in signal representation at each backward iteration. Some of these methods are introduced in the following.

2.2.1 Matching Pursuit

MP was initially introduced to find time-frequency representations of the signals in [1] and was then found to be a very efficient sparse approximation method. The forward step of MP is to add one atom to the currently selected atoms. In a normalized dictionary $\mathbf{D}$, let $\{\alpha_i\}_{i\in[1,n]}$ be the selected atom indices and the signal $\mathbf{r}^{[n]} = \mathbf{y} - \sum_{i\in[1,n]} \mathbf{d}_{\alpha_i} x_{\alpha_i}$ be the residual of $\mathbf{y}$ in the nth iteration. The atom which has the maximum correlation, i.e. maximum inner-product with the residual signal at the nth iteration, is selected as the $n+1$th atom. The atom selection step can be formulated as,

$$\alpha_{n+1} = \arg\max_i \left| \left\langle \mathbf{d}_i, \mathbf{r}^{[n]} \right\rangle \right|, \tag{2.1}$$

and the corresponding coefficient is found by the following formula,

$$\mathbf{x}_{\alpha_{n+1}} = \left| \left\langle \mathbf{d}_{\alpha_{n+1}}, \mathbf{r}^{[n]} \right\rangle \right|. \tag{2.2}$$

There is no backward step in MP to cancel out the atoms. MP terminates after a certain number of iterations or when the residual error $\|\mathbf{r}^n\|_2^2$ becomes small ($< \epsilon : \epsilon \in \mathbb{R}^+$). An

issue with MP is that the algorithm might select an already selected atom, which makes the convergence of the algorithm slow. If the aim is to find a quantized approximation of the signal, the selected coefficient can be quantized at each iteration [51]. The quantization error might be compensated for by the following selected atoms, as long as the following selected atoms are non-orthogonal to the current atom.

Another issue with MP is that the coefficients do not provide the best approximation using the selected support. This can be compensated for by orthogonally projecting the signal onto the span of the support. It is the motivation for another greedy algorithm, which will be explored in the following subsection, called Orthogonal MP.

2.2.2 Orthogonal Matching Pursuit

Using the coefficient selection step (2.2), we can easily show that $\mathbf{d}_{\alpha_{n+1}} \perp \mathbf{r}^{[n+1]}$. This fact might not be true for all $\{\mathbf{d}_{\{\alpha_i\}_{i\in[1,n]}}\}$ and $\mathbf{r}^{[n+1]}$. Let $\mathbf{r}^{[n+1]} = \mathbf{r}_O^{[n+1]} + \sum_{i\in[1,n]} \mathbf{d}_{\alpha_i}\beta_i$ such that $\forall i \in [1, n+1] : \mathbf{d}_{\alpha_i} \perp \mathbf{r}_O^{[n+1]}$. In other words, $\{\beta_i\}_{i\in[1,n+1]}$ is found by projecting $\mathbf{r}^{[n+1]}$ onto $\text{span}\{\mathbf{d}_{\alpha_i}\}_{i\in[1,n+1]}$ and $\mathbf{r}_O^{[n+1]}$ is found by subtracting the projection. A relation between $\|\mathbf{r}^{[n+1]}\|_2^2$ and $\|\mathbf{r}_O^{[n+1]}\|_2^2$ can be found, using the orthogonality of $\mathbf{r}_O^{[n+1]}$ and $\sum_{i\in[1,n]} \mathbf{d}_{\alpha_i}\beta_i$, as follows,

$$\begin{aligned}\|\mathbf{r}^{[n+1]}\|_2^2 &= \|\mathbf{r}_O^{[n+1]} + \sum_{i\in[1,n]} \mathbf{d}_{\alpha_i}\beta_i\|_2^2 \\ &= \|\mathbf{r}_O^{[n+1]}\|_2^2 + \|\sum_{i\in[1,n]} \mathbf{d}_{\alpha_i}\beta_i\|_2^2, \\ \therefore \|\mathbf{r}_O^{[n+1]}\|_2^2 &\leq \|\mathbf{r}^{[n+1]}\|_2^2.\end{aligned} \tag{2.3}$$

This motivates using the projection to reduced the residual. Orthogonal MP has been introduced in [36] and [3] in such a framework. Although the projection step is computationally expensive, it can be implemented using QR and Cholesky matrix factorizations, see for example [50] and [5] for more details. However the projection operator is not really tractable for large scale problems. The gradient pursuit algorithm was introduced in [5] to relax the backward step and reduce the computational complexity of the algorithm. This algorithm is explored in the following subsection.

2.2.3 Gradient Pursuit

The extra step of OMP includes an orthogonal projection onto the span of the selected atoms. This projection can be done using the pseudoinverse operator which was defined in 1.2. A matrix inversion is needed to apply this operator, which is computationally expensive in a large scale problem. Although there are some more efficient ways to calculate the pseudoinverse of such matrices using their structures [5], an alternative can be to relax the coefficient adjustment step. Instead of fully projecting the residual onto the selected space, we can choose a new coefficient vector, with the same support, with less residual error. Let the residual error at the $n+1^{th}$ iteration be noted by $\mathbf{r}_R^{n+1}$. A new *relaxed* OMP would be relevant if the residual satisfies the following inequality,

$$\|\mathbf{r}_O^{[n+1]}\|_2^2 \leq \|\mathbf{r}_R^{[n+1]}\|_2^2 \leq \|\mathbf{r}^{[n+1]}\|_2^2. \tag{2.4}$$

In other words, the coefficient adjustment step is to reduce the following cost function, by changing $\{x_i\}_{i\in\mathcal{I}\bigcup\{\alpha_{n+1}\}}$, where $\mathcal{I}$ includes the indices of the first n selected atoms and $|\mathcal{I}| \leq n$,

$$\|\mathbf{y} - \sum_{i\in\mathcal{I}} \mathbf{d}_i x_i\|_2^2, \tag{2.5}$$

The minimizer of (2.5) is the projection onto $\text{span}\{\mathbf{d}_i\}_{i\in\mathcal{I}\bigcup\{\alpha_{n+1}\}}$, which can be found using the gradient descent or the conjugate gradient methods. The Gradient Pursuit method uses a certain number of iterations of these iterative algorithms [5], which are also guaranteed to satisfy (2.4).

2.2.4 Other Greedy Methods

The idea of pursuing a *good* sparse solution is interesting as it can be implemented relatively fast and the solutions, even though they are not the sparsest solutions, are sparse enough for some practical applications, e.g. coding, classification. As a result many variation on original MP algorithm have been proposed to improve the overall success of sparse approximation. OMP and GP have been reviewed earlier. The atom selection operation in MP, OMP and GP are identical, i.e. choosing the atom which is most correlated to the residual signal from the previous iteration. The orthogonal projection onto the span of selected atoms was introduced in OMP to reduce the residual error at each iterations, which is caused by non-orthogonality of the atoms. The issue is that the simple atom selection approach in OMP does not always provide the minimum residual error at each iteration (even after the orthogonal projection). A method called Orthogonal Least Square (OLS) [37] has been presented as an alternative approach for the atom selection step. The aim of OLS at each iteration is to choose the new atom such that the residual error, after projecting the signal onto the selected space, becomes minimum. In other words, the new atom is selected by minimizing the following problem at each iteration,

$$\alpha_{n+1} = \text{argmin}_{k\in\mathcal{K}\setminus\mathcal{I}} \min_{\{x_i\}_{i\in\mathcal{I}\bigcup\{k\}}} \|\mathbf{y} - \sum_{i\in\mathcal{I}\bigcup\{k\}} \mathbf{d}_i x_i\|_2^2, \tag{2.6}$$

where $\mathcal{K}$ is the index set of the dictionary and $\mathcal{I} = \{\alpha_k\}_{k\in[1,n]}$ is the index set of the first n selected atoms. Solving (2.6) can be simplified using (2.1) with a modified dictionary. The modified dictionary is generated at each iteration by projecting non-selected atoms, i.e. $\{\mathbf{d}_i\}_{i\in\mathcal{K}\setminus\mathcal{I}}$, onto $\mathcal{S}^c$, which is the complement space of $\mathcal{S} = \text{span}\{\mathbf{d}_i\}_{i\in\mathcal{I}}$ and re-normalizing the atoms[2]. Hence OLS includes the atom selection using modified dictionary and the projection onto the span of the selected atoms at each iterations. However this additional orthogonalization does not scale well and as such OLS has not been popular for large scale problems.

2.3 Relaxed Sparse Approximation Methods

The sparse approximation (1.12) is called "relaxed", when the sparsity measure $\mathcal{J}(.) \neq \|.\|_0$. The aim of relaxation is to make the objective function continuous and piecewise differentiable. The optimization of such problems are easier, as long as various (sub-) gradient methods can be used. If the relaxed objective is convex, the global minimum can be found using a gradient descent method. Although this is no longer true for the non-convex objective, sparser solutions can often be found by warm starting[3] and using a suitable step size at each update.

One large class of sparse approximation methods either explicitly or implicitly is based on an optimization technique called majorization minimization method. This framework helps to simplify a complex multivariable optimization problem to an iterative optimization of a set of single variable optimization problems, which can be optimized independently. This framework is explained in the next subsection, which is followed by introducing the sparse approximation methods based on this technique.

[2] If an atom is orthogonal to $\mathcal{S}^c$, we remove that atom from the modified dictionary.

[3] Initializing the algorithm with a point satisfying some conditions. Starting with the convex relaxed solution or another sparse solution are some examples of such a warm start.

2.3.1 Majorization Minimization Method

Optimization of a multivariable problem like (1.12) is challenging. A technique, called "Majorization Minimization Method" [52, 53], has been introduced to simplify such problems in an iterative framework. In the majorization method, the objective function is replaced by a surrogate objective function which majorizes it and can be easily minimized. Here, for scalability, we are particularly interested in surrogate functions in which the parameters are decoupled, so that the surrogate function can be minimized element-wise.

A function ψ majorizes ϕ when it satisfies the following conditions,

$$\begin{aligned} &\phi(\omega) \leq \psi(\omega, \xi), \ \forall \omega, \xi \in \Upsilon \\ &\phi(\omega) = \psi(\omega, \omega), \ \forall \omega \in \Upsilon, \end{aligned} \tag{2.7}$$

where Υ is the parameter space. The surrogate function has an additional parameter ξ. At each iteration we first choose this parameter as the current value of ω and find the optimal update for ω.

$$\omega_{new} = \arg\min_{\omega \in \Upsilon} \psi(\omega, \xi) \tag{2.8}$$

We then update ξ with ω_{new}. The algorithm continues until we find an accumulation point. In practice the algorithm is terminated when the distance between ω and ω_{new} is less than some threshold.

This iterative method can be viewed as a block-relaxed minimization of the joint objective $\psi(\omega, \xi)$ [52]. In one step, we find the minimum of ψ based on ω. In the next step we minimize the objective based on ξ.

$$\xi_{new} = \arg\min_{\xi \in \Upsilon} \psi(\omega, \xi) \tag{2.9}$$

In our formulation, minimization of $\psi(\omega, \xi)$ based on ξ is done using $\xi_{new} = \omega$ (due to the definition of majorization in (2.7)).

There are different ways to derive a surrogate function. Jensen's inequality and Taylor series have often been used for this purpose [54,55]. The Taylor series of a differentiable function $\phi(\omega)$ is,

$$\phi(\omega) = \phi(\xi) + d\phi(\xi)(\omega - \xi) + \frac{1}{2!} d^2\phi(\xi)(\omega - \xi)^2 + o(\omega^3). \tag{2.10}$$

When ϕ has a bounded curvature, i.e. $d^2\phi < c_s$ for a finite constant c_s, it is majorized by,

$$\phi(\omega) \leq \phi(\xi) + d\phi(\xi)(\omega - \xi) + \frac{c_s}{2}(\omega - \xi)^2, \forall \omega, \xi \in \Omega, \tag{2.11}$$

and we can define $\psi(\omega, \xi)$ (which satisfies (2.7)) as follows,

$$\psi(\omega, \xi) = \phi(\xi) + d\phi(\xi)(\omega - \xi) + \frac{c_s}{2}(\omega - \xi)^2. \tag{2.12}$$

Then, at each iteration, $\phi(\omega_{new}) \leq \psi(\omega_{new}, \omega) \leq \psi(\omega, \omega) = \phi(\omega)$, hence ϕ does not increase. Conditions for which such algorithms converge have been presented in [52] and [54].

In the next subsections some of the sparse approximation methods based on, derived from or related to the majorization minimization method will be explored. The surrogate function can be generated by a majorizing function for the quadratic term, the sparsity measure or both parts of (1.12). It demonstrates a possible wide range of sparse approximation methods, based on how the majorizing function is generated. If the surrogate objective is generated by majorizing the quadratic term of 1.12, the algorithm can be interpret as the gradient projection method, with a certain step size. The gradient projection is a well-known technique, which can

minimize a continuously differentiable constrained optimization problem. The current solution is updated in two consecutive steps in this method, moving in the gradient direction, followed by projection onto the admissible set, see for example [56, 57] for more details. This technique inspired many sparse approximation methods, which will be surveyed in the following.

2.3.2 Iterative Thresholding

A difficulty in multivariable optimization problem like (1.12) is the coupling effect. It means the problem can not separately be solved with respect to each parameter. The sparsity measure is often an element-wise operator[4]. By majorizing the quadratic term of (1.12) with an element-wise objective, based on the coefficients, the new objective can be minimized element-wise. This has been applied to the sparse approximation problem, and called iterative thresholding[5] [43,59,60]. The quadratic term of (1.12), $\|\mathbf{y}-\mathbf{D}\mathbf{x}\|_2^2$, has a bounded curvature and a majorizing objective can be found using Taylor series. By using (2.12), the majorizing objective for the quadratic is found as follows,

$$\begin{aligned} ||\mathbf{y}-\mathbf{D}\mathbf{x}||_2^2 &\leq ||\mathbf{y}-\mathbf{D}\mathbf{x}||_2^2 + c||\mathbf{x}-\mathbf{x}^\ddagger||_2^2 - ||\mathbf{D}\mathbf{x}-\mathbf{D}\mathbf{x}^\ddagger||_2^2 \\ &= ||\mathbf{y}-\mathbf{D}\mathbf{x}||_2^2 + \pi_{\mathbf{x}}(\mathbf{x},\mathbf{x}^\ddagger) \end{aligned} \tag{2.13}$$

where $\pi_{\mathbf{x}}(\mathbf{x},\mathbf{x}^\ddagger)$ is a function defined as follows,

$$\pi_{\mathbf{x}}(\mathbf{x},\mathbf{x}^\ddagger) := c||\mathbf{x}-\mathbf{x}^\ddagger||_2^2 - ||\mathbf{D}\mathbf{x}-\mathbf{D}\mathbf{x}^\ddagger||_2^2. \tag{2.14}$$

If $\|\mathbf{D}\| < c$, where $\|.\|$ is the spectral norm operator, $\pi_{\mathbf{x}}(.,.)$ is a *convex* function based on $\mathbf{x}$, with a minimum at $\mathbf{x} = \mathbf{x}^{[n]}$. Let $\phi(\mathbf{x})$ be the objective in (1.12). $\psi(\mathbf{x},\mathbf{x}^\ddagger) = \phi(\mathbf{x}) + \pi_{\mathbf{x}}(\mathbf{x},\mathbf{x}^\ddagger)$ satisfies the conditions (2.7). As mentioned in subsection 2.3.1, as long as the minimization of ϕ based on $\mathbf{x}^\ddagger$ is easily found by $\mathbf{x}^{\ddagger^*} = \mathbf{x}$, the alternating minimization can be done by minimizing ψ based on $\mathbf{x}$ and updating $\mathbf{x}^\ddagger$ by the current $\mathbf{x}^*$.

Although solving the decoupled problems is significantly easier than solving the original problem, only some of the sparsity measures $\mathcal{J}(.)$ allow the problem to be solved analytically. Among them we are interested in ℓ_1 and ℓ_0[6], which will be presented in the next subsections. Although for the sparsity measure $\ell_p : p < 1$, the decoupled problems can not be solved analytically, it can be solved using a gradient projection method, with a simple look up table to calculate the projection, to compare the results with the reweighting methods, which will be discussed in subsection 2.3.3.

2.3.2.1 ℓ_1 relaxed sparse approximation:

- **Non-adaptive gradient projection:** The sparse approximation in this setting was independently introduced in [58] and [43]. This method is a generalization of the algorithm introduced by Sardy *et al.* [62], for the block orthonormal and union of orthonormal dictionaries. The sparsity measure ℓ_1 is the sum of the absolute values of coefficients, $\|\mathbf{x}\|_1 = \sum_{i\in[1,N]} |x_i|$. Let the auxiliary parameter $\mathbf{x}^\ddagger$ be $\mathbf{x}^{[n]}$. $\psi(\mathbf{x},\mathbf{x}^{[n]})$ can now be reformulated as,

$$\psi(\mathbf{x},\mathbf{x}^{[n]}) \propto c\mathbf{x}^T\mathbf{x} - 2\mathbf{x}^T\left(\mathbf{D}^T\left(\mathbf{y}-\mathbf{D}\mathbf{x}^{[n]}\right) + c\mathbf{x}^{[n]}\right) + \lambda\|\mathbf{x}\|_1. \tag{2.15}$$

[4] The joint sparsity measure is a column-wise operator which can be used in the dictionary learning, where a minimum size dictionary is sought [31].

[5] It is also called sparse approximations using majorization minimization method or Expectation Minimization (EM) based sparse approximations [58].

[6] Although the sparse approximation using ℓ_0 is not classified as the relaxed problem, it can be solved using MM technique [61].

ψ is a convex function and its optimum can be found by the fact that the sub-gradient should include zero, $\mathbf{0} \in \partial\psi(\mathbf{x}, \mathbf{x}^{[n]})$, where the sub-gradient $\partial\psi(\mathbf{x}, \mathbf{x}^{[n]})$ can be found by,

$$\partial\psi(\mathbf{x}, \mathbf{x}^{[n]}) = 2c\mathbf{x} - 2\left(\mathbf{D}^T\left(\mathbf{y} - \mathbf{D}\mathbf{x}^{[n]}\right) + c\mathbf{x}^{[n]}\right) + \lambda\partial\|\mathbf{x}\|_1. \tag{2.16}$$

The optimal $\mathbf{x}^*$, which is the updated coefficients $\mathbf{x}^{[n+1]}$, can be found by applying the soft shrinkage operator $\mathcal{S}_\lambda$ [18] to the vector,

$$\mathbf{a} := \frac{1}{c}\left(\mathbf{D}^T\left(\mathbf{y} - \mathbf{D}\mathbf{x}^{[n]}\right) + c\mathbf{x}^{[n]}\right). \tag{2.17}$$

$\mathbf{a}$ is actually a scaled gradient of the quadratic term, which is sometimes called the Landweber [63] update [43]. soft shrinkage is a nonlinear operator defined by,

$$\{\mathbf{x}^{[n+1]}\}_i = \mathcal{S}_\lambda(\mathbf{a}) = \begin{cases} a_i - \lambda/2\ sign(a_i) & \lambda/2 < |a_i| \\ 0 & otherwise. \end{cases} \tag{2.18}$$

The convergence of the iterative method for ℓ_1 relaxed sparse approximation is shown in [43]. The convergence of the iterative thresholding algorithm is analyzed in [43] in a more general framework in which the dictionary is an operator, which can also be nonlinear and/or a continuous domain operator. In such a framework the convergence analysis is more difficult as the week convergence does not guarantee strong convergence of a sequence in the infinite dimensional space. It has been shown that iterative soft thresholding converges R-linearly [7] in [66]. Bredies *et al.* [66] also showed that the asymptotic convergence rate is of order $\mathcal{O}(n^{-1})$[8].

A slightly different approach to drive the iterative thresholding formulation for ℓ_1 minimization is using the proximal operator splitting technique [20]. In this framework $\frac{1}{2}\|\mathbf{D}\| < c$, which provides a larger step in the gradient direction at each coefficient update. Note that although ψ does not majorize ϕ, the iterative algorithm is guaranteed to converge to the minimizer of ϕ at the end. The larger step size, provided by choosing $\frac{1}{2}\|\mathbf{D}\| < c \leq \|\mathbf{D}\|$, accelerates convergence of the algorithm in practice.

- **Adaptive gradient projection:**

  It was mentioned that choosing a smaller c does not make the algorithm unstable, while practically accelerating its convergence. We can extend this idea by using even smaller c using a line search technique to guarantee that ϕ is reduce at each update [68, 69].

  A gradient projection based algorithm is another algorithm proposed by Figueiredo *et al.* in [24], called the Gradient Projection for Sparse Representations (GPSR). To simplify the problem and make the algorithm differentiable, they used a technique previously used in [8], called a parameter splitting. In this method, each parameter is split to two positive parameters[9]. Each pair of new parameters associates to an atom and its negative version. The dictionary size thus becomes double in the new framework. (1.12) now becomes a constrained optimization problem with a differentiable objective as follows,

$$\min_{\mathbf{x}, \bar{\mathbf{x}} \in \mathbb{R}_0^+} \|\mathbf{y} - \mathbf{D}\left(\mathbf{x} - \bar{\mathbf{x}}\right)\|_2^2 + \lambda\mathbf{1}^T\left(\mathbf{x} - \bar{\mathbf{x}}\right). \tag{2.19}$$

[7] Let $x^* = \lim_{n\to\infty}\{x^{[n]}\}$. It is said to converge to x^* at least with order $p \geq 1$, see for example [64], if there exists a constant c and a sequence $\{\epsilon_n\}$ such that $|x^{[n]} - x^*| < c\epsilon_n$ for all n and $\lim_{n\to\infty}\frac{\epsilon_{n+1}}{\epsilon_n^p} = \theta$ for $\theta \in (0, 1)$. A sequence is called to converge at least R-linearly if $p = 1$ [65]. A similar definition can be presented on convergence of a sequence of vectors in a normed space.

[8] See [67] for the definition of Big $\mathcal{O}$ and Small o.

[9] This framework, using the setting presented in [24], can only be used for the real value sparse approximation.

Algorithm 1 Fast Iterative Shrinkage/Thresholding Algorithm (FISTA)

initialization: $c > \|\mathbf{D}^T\mathbf{D}\|$, $\mathbf{x}^{[0]} = \mathbf{z}^{[1]}$, $t^{[1]} = 1$, K
for $k = 1$ **to** K **do**
  $\mathbf{x}^{[n]} = \mathcal{S}_\lambda(\mathbf{a}(\mathbf{z}^{[k]}))$
  $t^{[k+1]} = \frac{1+\sqrt{1+4{t^{[k]}}^2}}{2}$
  $\mathbf{z}^{[k+1]} = \mathbf{x}^{[k]} + \left(\frac{t^{[k+1]}-1}{t^{[k+1]}}\right)(\mathbf{x}^{[k]} - \mathbf{x}^{[k-1]})$
end for
output: $\mathbf{x}^{[K]}$

Figueiredo *et al.* proposed two different step sizes for the gradient projection method and proved the convergence of the final algorithm.

The non-linear operator $\mathcal{S}_\lambda$ is the projection onto an ℓ_1 ball. The radius of the ℓ_1 ball can be calculated after projection. To accelerate the convergence of the sparse approximation Daubechies *et al.* [70] suggested to adaptively change the radius of the ball, which is equivalent to use an adaptive λ. They also proved the convergence of the gradient projection method with this setting.

Although the *adaptive* gradient projection technique accelerates the convergence of the derived methods there does not exist analytical analysis about the convergence rate of all methods. A new technique, called the *"optimal first-order gradient method"*or Nesterov's method [71], can be used to adaptively change the gradient step size. The convergence rate of the new method is improved to the order $\mathcal{O}(n^{-2})$ [45, 72, 73]. The *optimality* of the method means that we can not get any better convergence rate using similar (first-order) gradient projection method. There exist different approaches to derive such optimal first order gradient projection algorithms. Here, the algorithm presented by Beck *et al.* in [45], which is called Fast Iterative Shrinkage/Thresholding Algorithm (FISTA), will be explored.

FISTA uses the values of two consequent iterations, i.e. $\mathbf{x}^{[n]}$ and $\mathbf{x}^{[n-1]}$, to find the new value $\mathbf{x}^{[n+1]}$. In other words, in this method there exists an extra series of parameters, named $\mathbf{z}^{[n]}$, which is generated using $\{\mathbf{x}^{[k]}\}_{k\in\{n-1,n\}}$. Let $\mathbf{a}(\mathbf{x}) := \frac{1}{c}\left(\mathbf{D}^T\left(\mathbf{y} - \mathbf{D}\mathbf{x}\right) + c\mathbf{x}\right)$ be the generalized functional of $\mathbf{a}$ in (2.17). In the modified (adaptive) iterative thresholding method, $\mathbf{z}^{[n]}$ is used, instead of $\mathbf{x}^{[n]}$, to find $\mathbf{x}^{[n+1]}$ which is summarized in Algorithm 1. The parameter $t^{[k]}$ change the effect of two previous iterations $\mathbf{x}^{[n-1]}$ in the algorithm. If we choose $t^{[k]} = 1$, FISTA is identical to the non-adaptive iterative thresholding. Let $\phi(\mathbf{x})$ and $\mathbf{x}^*$ be the objective and an optimal solution of the problem (1.12) respectively. Beck *et al.* showed in [45, Theorem 4.1] that for $k \leq 1$ the following inequality holds,

$$\phi(\mathbf{x}^{[n]}) - \phi(\mathbf{x}^*) \leq \frac{4c\|\mathbf{x}^{[0]} - \mathbf{x}^*\|^2}{(k+1)^2}, \tag{2.20}$$

which shows $\{\phi(\mathbf{x}^{[n]}) - \phi(\mathbf{x}^*)\} \simeq \mathcal{O}(1/k^2)$. They also introduced an adaptive method to adjust parameter c which can improve the convergence of the algorithm in practice, even though it does not change the order of convergence. The adaptive selection of c also is useful if the actual spectral norm of the dictionary is unknown.

Another fast method, which uses two recent iterations to find the next iteration, is TwIST [74]. Although TwIST algorithm shows promising results in the simulations, there is no analytical study on the convergence rate so far.

ℓ_0 sparse approximation The sparsity measure ℓ_0 counts the number of non-zero coefficients and can be reformulated as $\|\mathbf{x}\|_0 = \sum_{i\in[1,N]} f(x_i)$, where,

$$f(\alpha) := \begin{cases} 0 & \alpha = 0 \\ 1 & otherwise. \end{cases} \tag{2.21}$$

Let the auxiliary parameter $\mathbf{x}^{\ddagger}$ be $\mathbf{x}^{[n]}$ as before. The surrogate objective is reformulated as,

$$\psi(\mathbf{x}, \mathbf{x}^{[n]}) \propto c\mathbf{x}^T\mathbf{x} - 2\mathbf{x}^T\left(\mathbf{D}^T\left(\mathbf{y} - \mathbf{D}\mathbf{x}^{[n]}\right) + c\mathbf{x}^{[n]}\right) + \lambda \sum_{i\in[1,N]} f(x_i). \tag{2.22}$$

(2.22) is not convex and the sub-gradient method can not be used to minimize $\psi(\mathbf{x}, \mathbf{x}^{[n]})$. Instead we can decouple (2.22) to N optimization problems. The objective of the ith problem can be represented by,

$$\{\psi(\mathbf{x}, \mathbf{x}^{[n]})\}_i \propto cx_i^2 - 2x_i\ \{\mathbf{D}^T\left(\mathbf{y} - \mathbf{D}\mathbf{x}^{[n]}\right) + c\mathbf{x}^{[n]}\}_i + \lambda f(x_i) \tag{2.23}$$

(2.23) can be solved by letting x_i^* being zero or non-zero, followed by checking the validity of the solution. Let $\mathbf{a}$ be defined as in (2.17). x_i^* can be found using a nonlinear operator $\mathcal{H}_\lambda$, called hard shrinkage [18], as follows,

$$\{\mathbf{x}^{[n+1]}\}_i = \mathcal{H}_\lambda(\mathbf{a}) = \begin{cases} a_i & \sqrt{\lambda} < |a_i| \\ 0 & otherwise. \end{cases} \tag{2.24}$$

The convergence of the iterative hard thresholding (IHT) is proven in [61]. The algorithm can be modified to find a k-sparse approximation by replacing $\mathcal{H}$ with an orthogonal projection onto the space of k-sparse signals [59]. That keeps the k largest coefficients and set the others to zero. Although the algorithm has analytically been shown to have good performance in, for example, compressed sensing [75], its performance when the dictionary does not have the Restricted Isometry Property (RIP) is poor. Recently it has been shown that by choosing a larger gradient step, the IHT algorithm is much more successful in sparse approximation even when RIP is not satisfied [76].

2.3.3 Iterative Reweighting for Non-convex Objectives

It was shown in the previous subsection that the majorization minimization method can be used to replace the quadratic term with some decoupled terms to facilitate the minimization. This technique can also be used to replace the sparsity measure with an ℓ_1 or ℓ_2 norm. Because there exist efficient algorithms to solve such a regularized approximation problem, the ℓ_p sparse approximation can easily be solved, i.e. finding a local minimum when $p < 1$, by iteratively solving majorized problem. This technique has also been known as iterative reweighting technique in literature. Some of these methods will be explained in the following.

2.3.3.1 Iterative Reweighted ℓ_1

ℓ_p for $p < 1$ is concave in each orthant. It can be shown that any concave function is majorized by the tangent line [54], which can be used to generate a majorization function for the sparsity measure. If $\alpha \in \mathbb{R}^+$ and $\alpha_0 \in \mathbb{R}^+$, where α_0 is a fixed number, the following inequality holds,

$$\alpha^p \leq \alpha_0^p + p\alpha_0^{p-1}\alpha. \tag{2.25}$$

Note that such a majorizing function should be restricted to the corresponding orthant. One way is to use the absolute value operator to restrict the majorizing line to the orthant in which current coefficient vector $\mathbf{x}^{[n]}$ is located and symmetrically duplicating that line in other orthants as follows,

$$\sum_{i\in[1,N]} |x_i|^p \leq \sum_{i\in[1,N]} |x_i^{[n]}|^p + p \sum_{i\in[1,N]} |x_i^{[n]}|^{p-1}|x_i|. \tag{2.26}$$

When $\mathbf{x}_i^{[n]} \to 0$, the majorization function gets infinitely large, i.e. the original function is upperbounded by infinity. Note this is not a problem of MM but a characteristic of the cost function. In this case we let $\mathbf{x}_i^{[n]}$ stay at zero for the following iterations and reduce the problem size. An alternative is to use a modified sparse approximation $\ell_{p,\epsilon}$ with $0 < \epsilon \ll 1$ as follows,

$$\ell_{p,\epsilon}(\mathbf{x}) = \sum_{i\in[1,N]} (|x_i| + \epsilon)^p. \tag{2.27}$$

(2.27) is bounded on $x_i \in \mathbb{R}$, which solves the singularity at $x_i^{[n]} = 0$. The majorizing function can now be found as follows,

$$\ell_{p,\epsilon}(\mathbf{x}) \leq \ell_{p,\epsilon}(\mathbf{x}^{[n]}) + p \sum_{i\in[1,N]} (|x_i^{[n]}| + \epsilon)^{p-1}|x_i|. \tag{2.28}$$

By using such a majorization function for the sparsity measure we can find the surrogate objective as follows,

$$\psi(\mathbf{x}, \mathbf{x}^{[n]}) \propto \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 + \lambda \sum_{i\in[1,N]} |w_i x_i|, \tag{2.29}$$

which can be solved in a weighted pursuit framework [77]. Minimization of (2.29) would also be easier if we also majorize the quadratic part, and using the iterative thresholding scheme [78].

Iterative reweighted ℓ_1 has also been used for sparse representation with the ϵ-relaxed logarithmic sparsity measure $\sum_{i\in[1,N]} \log(|x_i| + \epsilon)$ in [15].

2.3.3.2 Iterative reweighting ℓ_2

The surrogate objective made using a weighted ℓ_1 penalty is a close approximation of the original objective, i.e. the approximation error is small. A problem in using such a majorizing function is that the simplified problem is still difficult to solve, which can be solved by another convex relaxed sparse approximation method. An alternative is to majorize with a weighted ℓ_2, see for example [13], which simplifies the problem to a quadratic optimization problem and lets us to solve it analytically. In this framework the algorithm is sometimes called Iterative Reweighting Least Square (IRLS), but it only refers to a sub-class of algorithms in this class.

If the quadratic majorizing function for $\ell_p : p < 1$ satisfies following conditions, the optimization problem becomes more tractable.

1. *Decoupled,* to make the optimization easier.
2. *Even,* to follow the original objective, which is even.
3. *Has the same tangent space at* $\mathbf{x}^{[n]}$*:* to majorize $\ell_p \,|_{\mathbf{x}^{[n]}}$

The quadratic function which satisfies these conditions can be presented as $\frac{p}{2}\sum_{i\in[1,N]} w_i x_i^2$, where w_i's are some weights which can be found by [14],

$$w_i = |\mathbf{x}_i^{[n]}|^{p-2}, \tag{2.30}$$

and by [16],

$$w_i = (|\mathbf{x}_i^{[n]}| + \epsilon)^{p-2}, \tag{2.31}$$

for the ϵ-relaxed ℓ_p. If $\mathbf{x}_i^{[n]} = 0$ in (2.30), we let it to being zero in the following iterations. The surrogate objective can be found as follows,

$$\psi(\mathbf{x}, \mathbf{x}^{[n]}) \propto \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 + \frac{p\lambda}{2} \sum_{i \in [1,N]} w_i x_i^2. \tag{2.32}$$

As we have only quadratic terms, the minimizer of the surrogate objective can be found by,

$$\mathbf{x}^{[n+1]} = \mathbf{W}^{-1}\mathbf{D}^T(\mathbf{D}\mathbf{W}^{-1}\mathbf{D}^T + \lambda\mathbf{I})^{-1}\mathbf{y}, \tag{2.33}$$

where $\mathbf{W} = \text{diag}(\{w_i\}_{i \in [1,N]})$. We need to invert a large matrix to calculate $\mathbf{x}^{[n+1]}$, which is not computationally possible for a large size problem. Similar to reweighted ℓ_1 approach, one can majorize the approximation error with a decoupled quadratic term and minimize the new majorizing function, which is equivalent to adaptively scaling each component of the Landweber update **a** (2.17), see [79, 80].

2.3.4 Other Sparse Approximation Methods

In the convex relaxed sparse approximation using iterative thresholding, it was mentioned that some non-differentiable and unconstrained optimization problems can be reformulated as the constrained differentiable problems (2.19). The new formulation is favorable to be solved using a quadratic programming and interior point method [8]. Recently the interior point method has also been used directly to solve ℓ_1 regularized sparse approximation problem [81]. Kim *et al.* [81] used the primal logarithmic barrier method to solve the following equivalent problem,

$$\min_{\mathbf{x}} \|\mathbf{D}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda \sum_{i \in \mathcal{I}} u_i, \text{ s.t. } \forall i \in \mathcal{I} - u_i \leq \mathbf{x}_i \leq u_i, \tag{2.34}$$

using a truncated Newton's method. The method uses an equality found using the dual form of (2.34) to simplify the problem and find an ϵ-suboptimal solution, where ϵ is the target duality gap. This technique can also be extended to the medium to large scale problems by solving the Newton system approximately.

Most of the sparse approximation methods based on the gradient projection technique converge very slowly if λ is small. However such a small λ is interesting when the approximation error has to be small. In this case one can adaptively change λ, by starting from a large value, and accelerate the gradient projection method, see [70]. The gradient projection technique can also be applied to solve the LASSO problem. See (1.9) [82] in which Van den Berg *et al.* showed a relation between Basis Pursuit (BP), LASSO and Basis Pursuit DeNoising (BPDN)[10] problems, i.e. by choosing correct parameters, the problems share the solutions. This fact helps us to solve these problems using a gradient projection method, if the relation between the parameters are known. A method for solving such problems, by iteratively turning them to LASSO problems with different τ, has been presented in [82].

As sparse approximation is formulated using an optimization problem, almost any optimization techniques can be applied to solve it. This chapter only covered common and potentially fast and scalable algorithms.

[10] Here, it is also called convex relaxed sparse approximation.

Chapter 3

Scalable Algorithms

This chapter reviews the comparative performance of previously mentioned algorithms. The issues in scalability of the algorithm, which cause computationally expensive operations, will be surveyed first. The algorithms that need such operations, e.g. matrix inversion, will then be highlighted. These algorithms are not scalable in the original form and need some modifications to moderate the complexity. As the aim of this report is to point out potentially scalable methods, such sparse approximation algorithms will be emphasized in subsection 3.2.

3.1 Scalability issues

This section reviews the operations which are not easily scalable. It also mentions how one can implemented some of them more efficiently by applying some modifications to the original operations.

3.1.1 Unstructured dictionaries

The linear generative model of the sparse signals, i.e. dictionary, can have a structure which allows a fast implementation. These dictionaries are often generated by subsampling an orthogonal basis in the ambient space as the row of the dictionary or the oversampling a particular parametric functional, e.g. Fourier functional, in the signal space. The dictionary is here called unstructured if the atoms are generated independently and there is no model for generating the atom. An example of such dictionaries is the typical learned dictionary, which is found using a dictionary learning algorithm, see for example [83] and references therein. These dictionaries are hardly implemented efficiently. The implementation of such dictionaries is $\mathcal{O}(dN)$, where $d \leq N$ and N are the signal and dictionary sizes respectively. In contrast, *some* dictionary structures allow us to implement the dictionary more efficiently, e.g. $\mathcal{O}(d\log(N))$ and $\mathcal{O}(N)$.

If the algorithm uses the $\mathbf{D}$ and $\mathbf{D}^*$ operators, each operation can be implemented block wise, with some extra operations. In this framework, the matrix and the operand vector are divided into k^2, $k > 1$, submatrices and k blocks respectively. Each dictionary submatrix is multiplied with the corresponding block of the operand. The dictionary-vector multiplication is then found by adding corresponding components of the submatrix multiplications. **Note:** this operation does not reduce the complexity of the dictionary implementation, but it lets us use a multicore processing and use the memory wisely.

3.1.2 Matrix inversion

Sometime we need to invert a matrix in sparse approximation methods, e.g. for projecting a signal onto a certain subspace. This operation is computationally expensive. It was mentioned in 3.1.1 that, if the dictionary is unstructured, the dictionary can be implemented in a parallel framework. Unfortunately the matrix inversion can not easily be implemented in a parallel setting[1]. If the matrix inversion can be approximated by other simpler matrix operations, the overall complexity of the algorithm reduces in practice. An example of such methods is introduced in [85], which can be used when the dictionary is approximately block-orthogonal and where we ignore the effect of the atoms with minimum energy in each block, in matrix inversion of the block.

3.1.3 Ill-conditioned dictionaries

When the condition number of a dictionary is high[2], some algorithms face a precision error. This happens particularly when it needs to calculate $(\mathbf{D}\mathbf{D}^T)^{-1}$.

Another problem with using such an ill-conditioned dictionary in iterative algorithms, e.g. iterative thresholding, is the slow convergence of the algorithms. Practical observations show that the adaptive iterative thresholding algorithms are faster, using such ill-conditioned dictionaries. Although this issue is general and not related to the size of problem, the issue would be more challenging in the large size problems. It should be noted however that when $\mathbf{D}$ is poorly conditioned, the quality of the sparse approximation is generally questionable.

3.1.4 Projection onto a set

It was shown that a large class of relaxed sparse approximation methods is based on the gradient projection technique. Although the projection onto an arbitrary set is not easy to find, such a projection is possible for the sets typically used. The ℓ_1 convex ball is the common set in the sparse approximation/representation context. Different methods have been presented to project a point onto this convex set [70,86]. An optimal projection might accelerate the iterative algorithm, as this operation is used at each iteration.

3.2 Comparative study

The greedy and gradient projection methods will be compared separately in this section. The reason is that these methods are structurally different and the comparison of these methods are not easy, as they seek different solutions. The greedy methods are introduced to find a reasonably sparse signal. In contrast gradient projection methods reduce an objective to find a fixed point[3]. In a special case where the objective is convex the solution is unique (with a mild condition). In this special case we can fairly compare the algorithms, as they find the same solution.

[1] By using block matrix inversion lemma [84], we can implement each matrix inversion by a series of smaller matrix inversions. Although this lemma allows us to invert large scale matrices, it increases the computation cost, as it needs consequent matrix inversions.

[2] The condition number of a matrix is the ratio of the largest to the smallest singular values, if ℓ_2 is the norm of the vector space.

[3] The fixed points of these algorithms are not necessarily local minima.

COMPARISON OF THE METHODS IN TERMS OF COMPUTATIONAL REQUIREMENTS IN ONE ITERATION.

Algo.	Computation Cost (flops)	Storage Cost (floating point numbers)
MP	$M + [\mathbf{\Phi} + N]$	$[\mathbf{\Phi} + M + 2n + N]$
OMP QR	$2Mn + 3M + [\mathbf{\Phi} + N]$	$Mn + 0.5n(n+1) + [\mathbf{\Phi} + M + 2n + N]$
OMP Cholesky	$2\mathbf{\Phi} + 3n^2 + 2M + [\mathbf{\Phi} + N]$	$0.5n(n+1) + [\mathbf{\Phi} + M + 2n + N]$
GP	$1\mathbf{\Phi} + n + 3M + [\mathbf{\Phi} + N]$	$M + [\mathbf{\Phi} + M + 2n + N]$
ACGP	$2\mathbf{\Phi} + 2n + 4M + [\mathbf{\Phi} + N]$	$M + [\mathbf{\Phi} + M + 2n + N]$

Table 3.1: Table I of the reference [5].

3.2.1 Greedy methods

It was mentioned in Section 2.2 that the computational complexity of the pursuit methods increases if they use extra operations, e.g. projection onto the solution space in OMP and OLS, and using modified Gram-Schmidt method to project atoms onto the complement of the selected atoms span. Although these operations are of order $\mathcal{O}(n^3)$, they are significantly more expensive than a simple matrix-matrix multiplication, which has the same order of complexity, when the matrices are unstructured. To overcome the extra complexity, one approach is to use a relaxed pursuit algorithm, e.g. GP. These methods only use some steps of the gradient or the conjugate-gradient descent. Although the new relaxed algorithm does not satisfy the nice features of the derived algorithm, e.g. OMP and OLS, the residual error decays faster in practice. The computational complexity of each iteration and the memory usage of different algorithms are presented in [5], which is shown here in Table 3.1. In this table, M, N and n are the dimension of signal and coefficient spaces and the iteration number respectively and Φ is the computational complexity of applying $\mathbf{D}$ or $\mathbf{D}^T$. Here *OMP QR*, *OMP Cholesky* and *ACGP* respectively stand for OMP using QR and Cholesky factorization and Approximate Conjugate Gradient Pursuit. It is clear from this table that the algorithms scale differently by scaling the problem size and the number of iterations. This emphasizes on the fact that the scalability of such greedy algorithms depends on the problem setting, i.e. the dictionary size, the sparsity of the approximation and the complexity of the dictionary/transposed-dictionary multiplications.

Although the comparative study of the computational cost and memory usage of the greedy methods are useful, such a study can not be fair as these methods are not proposed to do similar task and the sparse solutions are often different. With this respect a comparison between these methods is relevant if we only consider an application with a given setting, i.e. size of the problem. Therefore to investigate the scalability of the greedy methods, one approach is to consider a single application and compare the performance/complexity plots.

3.2.2 Relaxed sparse approximation methods

The analysis of the computational complexity of sparse approximation methods based on minimizing a *convex* objective is very important, as it has directly been used in many applications, e.g. Compressed Sensing and denoising, and indirectly been used in other sparse approximation methods, e.g. re-weighted ℓ_1, as an intermediate step. A comparative study with this setting is fair, as the minimizer of objective is unique. Hence we can compare the computational complexity and memory usage of each algorithm. In contrast, comparison of other non-convex relaxed sparse approximation methods are difficult, as they find different sparse solutions. The iterative re-weighted algorithms generally include, for example, K iterative solving of another convex problem. Although the computational cost is roughly K times greater, the solutions are sparser in practice, i.e. fewer non-zero coefficients. There is thus a compromise between the

algorithm complexity and the sparsity of solution.

Here the complexity of convex sparse approximation methods, base on minimizing (1.10), will be compared. Loris presented a fair approach to compare the complexity of these methods [87]. Let λ_{max} be the smallest value of λ that the solution of (1.10) is zero and $\bar{\mathbf{x}}$ and $\mathbf{x}^{(n)}$ respectively be the optimum solution and the coefficient vector at n^{th} iteration. In Figure 6 of [87], the normalized sparse approximation error, $\|\mathbf{x}^{(n)} - \bar{\mathbf{x}}\|/\|\bar{\mathbf{x}}\|$, is plotted versus different $\log_2 \lambda_{max}/\lambda$. To compare the convergence of different algorithms, Loris calculated the plots after certain simulation times, here $t = 6k\ sec, k \in [1, 10]$. A Gaussian random dictionary $\mathbf{D} \in \mathbb{R}^{1848\times 8192}$, which is clearly a well-conditioned dictionary, has been used in these simulations. Two non-adaptive iterative thresholding methods, see subsection 2.3.2.1, with $c > \|\mathbf{D}\|$ [43], and $c > \frac{1}{2}\|\mathbf{D}\|$ [20], and four adaptive iterative thresholding methods, accelerated projected gradient [70], Gradient Projection for Sparse Representation (GPSR) [24], ℓ_1-ls [81] and FISTA [45], were compared in [87]. An ideal plot with this setting is a plot which tends to the lower-right corner of the plot, which means the algorithm converges fast and the final error is low. It is shown that FISTA converges faster than other methods, but the final error is higher than some others, e.g. accelerated projected gradient and ℓ_1-ls, when the optimal solution is not very sparse.

If these simulations are repeated with a dictionary which is not well-conditioned, the algorithms converge slower. For such a dictionary, taken from a Geo-science inverse problem, the performance plots are shown in Figure 4 of [87]. As the algorithms converge slower, the plots are shown after running $t \in [1, 10]$ minutes of simulations. Again we observe that FISTA performs better than the others in this experiment.

These experiments only considered some fast ℓ_1 sparse approximation methods. Recently other algorithms based on parameter splitting and augmented Lagrange multipliers method [88, 89] have been introduced which are claimed to be faster than the introduced methods. Although the new algorithm, called SALSA, shows a promising performance in practice, there is no analytical study on the convergence and the rate of convergence so far.

Chapter 4

Conclusions

The sparse coding problem was formulated here by introducing some sparsity measures and the related optimization problems which should be minimized. In this framework, the solution space of an underdetermined linear system was constrained to the solutions with the minimal non-zero coefficients. The formulations were then extended to the matrix vector space, which facilitate the sparse coding of a set of signals or promote a structured sparsity pattern within the matrix. As the sparse approximation is one step of an alternating minimization algorithm for dictionary learning, these formulations also appear where the dictionary learning problem is formulated.

In the second chapter, the Greedy and Gradient (Projection) based sparse approximation methods were reviewed. In the first part, some greedy algorithms, mainly MP and its extensions, were explained and the relation between them were explored. The computational complexity of the algorithms increases if all coefficients have often to be updated. The complexity of OLS is very high as it has one orthogonal projection and one Modified Gram-Schmidt at each iteration. Therefore it would not be considered as a scalable algorithm.

The gradient projection and majorization minimization methods have often been used to solve relaxed sparse approximation problem, where it is changed to be more tractable or an approximate solution is sought. Most important derivations of these techniques for sparse approximation were explored here. As the computational complexity of each iteration is moderately low, these methods are good candidates for scalable problems, *if the algorithms also converge fast*. For ℓ_1 relaxed sparse approximation, the optimal first-order gradient methods have been shown to have a better upper bound on the convergence rate than the other (non-) adaptive gradient projection methods. On the other hand, although iterative reweighting algorithms can not guarantee to find the optimal solution of $\ell_p : p < 1$, they converge very fast into a basin of attraction. Further analytical or empirical investigation on the rate of convergences of these gradient projection based methods is necessary.

In chapter 3, the challenges in using sparse approximation method for a large scale problem were reviewed. These might make the sparse approximation of a scalable problem impossible or very slow. Some comparative studies have been discussed in Section 3.2. Although comparing the computation complexity and memory usage of the methods, which find different solutions, are not completely fair, an order comparison for greedy methods was presented here. This comparison would be helpful if we are interested to make a compromise between the computational complexity and memory usage of the algorithm and sparsity of the approximations. The observation is that the order of complexity of the algorithms are different using structured dictionaries, i.e. the dictionaries which can be implemented in $\mathcal{O}(n \log(n))$. Otherwise the dictionary multiplication is the most expensive part of the implementation and the computational complexity of MP, OMP and GP are roughly the same (per iteration). In practice when the

dictionary is fast, it has been observed that ACGP is faster than OMP, while it provides similar sparse solutions.

A more sophisticated comparison between ℓ_1 sparse approximation methods was explained here. Some of the fast iterative thresholding type methods were compared by exploring their convergence times and approximation errors using a well-conditioned and an ill-conditioned dictionary. Although we can not give a general statement about the algorithms, FISTA, which is an optimal first order gradient method, shows slightly better performance than the best of other methods.

This report only pointed out the issues in using sparse approximation for a large scale problem. A scalable algorithm in general should be able to use a parallel computation framework and use less memory. These make each iteration of the algorithm tractable in a large scale setup. It also needs to converge fast to reduce the total computation time. Therefore the algorithms with better rate of convergence, e.g. optimal first-order gradient methods, are good candidates as the scalable sparse approximation methods. The complexity of these algorithms further reduces if each iteration can be broken down into smaller independent blocks of computations, which allows parallel computation. A further research on parallelization of such algorithms and specifically its implementation on the Graphical Processing Units (GPU), which have advanced parallel structures, is necessary in order to practically implement such sparse approximation methods using a large scale setting.

Bibliography

[1] S. Mallat and Z. Zhang, "Matching pursuits with time frequency dictionaries," *IEEE Trans. on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[2] V.K. Goyal, J. Kovacevic, and J. A. Kelner, "Quantized frame expansions with erasures," *Applied and Computational Harmonic Analysis*, vol. 10, no. 3, pp. 203–233, 2001.

[3] G. Davis, S. Mallat, and M. Avellaneda, "Adaptive greedy approximations," *Constructive Approximation*, vol. 13, no. 1, pp. 57–98, 1997.

[4] N.G. Kingsbury and T. H. Reeves, "Iterative image coding with overcomplete complex wavelet transforms," in *Conference on Visual Communications and Image Processing*, 2003.

[5] T. Blumensath and M.E. Davies, "Gradient pursuits," *IEEE Trans. on Signal Processing*, vol. 56, no. 6, pp. 2370–2382, 2008.

[6] H. Mohimani, M. Babaie-Zadeh, and C. Jutten, "A fast approach for overcomplete sparse decomposition based on smoothed l0 norm," *IEEE Trans. on Signal Processing*, vol. 57, no. 1, pp. 289–301, Jan. 2009.

[7] K. Kreutz-Delgado, B. D. Rao, K. Engan, T.W. Lee, and T. J. Sejnowski, "Convex/schur-convex (CSC) log-priors and sparse coding," in *Joint Symposium on Neural Computation*, 1999.

[8] S.S. Chen, D.L. Donoho, and M.A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

[9] D. Donoho, "For most large underdetermined systems of linear equations, the minimal ℓ_1 -norm solution is also the sparsest solution," *Communications on Pure and Applied Mathematics*, vol. 59, no. 6, pp. 797–829, 2004.

[10] R. Gribonval and M. Nielsen, "Highly sparse representations from dictionaries are unique and independent of the sparseness measure," *Applied and Computational Harmonic Analysis*, vol. 22, no. 3, pp. 335–355, 2007.

[11] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[12] J. Dattorro, *Convex Optimization and Euclidean Distance Geometry*, Meboo Publishing, 2009, (v2009.06.18), Palo Alto, CA.

[13] I.F. Gorodnitsky and B.D. Rao, "Sparse signal reconstruction from limited data using focuss: a re-weighted minimum norm algorithm," *IEEE Trans. on Signal Processing*, vol. 45, no. 3, pp. 600–616, 1997.

[14] B.D. Rao and K. Kreutz-Delgado, "An affine scaling methodology for best basis selection," *IEEE Trans. on Signal Processing*, vol. 47, no. 1, pp. 187–200, 1999.

[15] E. J. Candes, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted l1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 877–905, 2008.

[16] I. Daubechies, R De Vore, M. Fornasier, and Sinan Gunturk C., "Iteratively re-weighted least squares minimization for sparse recovery," to appear in Comm. Pure Appl. Math., 2008.

[17] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *Journal of Royal Statistical Society Series B,*, vol. 58, pp. 267288, 1996.

[18] D.L. Donoho and J.M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425–455, 1994.

[19] A.K. Fletcher, S. Rangan, V.K. Goyal, and K. Ramchandran, "Denoising by sparse approximation: Error bounds based on rate-distortion theory," *Journal on Applied Signal Processing*, vol. 10, pp. 1–19, 2006.

[20] P.L. Combettes and V.R. Wajs, "Signal recovery by proximal forward-backward splitting," *SIAM Journal on Multiscale Modeling and Simulation*, vol. 4, pp. 1168–1200, 2005.

[21] J.A. Tropp, "Just relax: Convex programming methods for identifying sparse signals," *IEEE Trans. on Information Theory*, vol. 52, no. 3, pp. 1030–1051, 2006.

[22] C. Zalinescu, *Convex Analysis in General Vector Spaces*, World Scientific, River Edge, NJ, 2002.

[23] D.L. Donoho and X. Huo, "Uncertainty principles and ideal atomic decomposition," *IEEE Trans. on Information Theory*, vol. 47, no. 7, pp. 2845–2862, 2001.

[24] M.A.T. Figueiredo, R.D. Nowak, and S.J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE Journal of Selected Topics in Signal Processing: Special Issue on Convex Optimization Methods for Signal Processing*, vol. 1, no. 4, pp. 586–597, 2007.

[25] M. Elad, B. Matalon, and M. Zibulevsky, "Coordinate and subspace optimization methods for linear least squares with non-quadratic regularization," *Applied and Computational Harmonic Analysis*, vol. 23, no. 3, pp. 346–367, 2007.

[26] M.E. Davies and L. Daudet, "Sparse audio representations using the MCLT," *Signal Processing*, vol. 86, no. 3, pp. 457–470, 2006.

[27] J.A. Tropp, "Algorithms for simultaneous sparse approximation. part II: Convex relaxation," *Signal Processing*, vol. 86, no. 3, pp. 589–602, 2006.

[28] M. Yaghoobi, T. Blumensath, and M. Davies, "Dictionary learning for sparse approximations with the majorization method," *IEEE Trans. on Signal Processing*, vol. 57, no. 6, pp. 2178–2191, 2009.

[29] M. Zibulevsky and B. A. Pearlmutter, "Blind source separation by sparse decomposition in a signal dictionary," *Neural Computation*, vol. 13, no. 4, pp. 863–882, 2001.

[30] M. Yaghoobi and M. Davies, "Compressible dictionary learning for fast sparse approximation," in *IEEE Workshop on Statistical Signal Processing*, Aug. 31- Sept. 3 2009.

[31] M. Yaghoobi, T. Blumensath, and Davies M. E., "Parsimonious dictionary learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, April 2009, pp. 2869–2872.

[32] R. Rubinstein, M. Zibulevsky, and M. Elad, "Double sparsity: Learning sparse dictionaries for sparse signal approximation," To appear, 2009.

[33] S.F. Cotter, B.D. Rao, K. Engan, and K. Kreutz Delgado, "Sparse solutions to linear inverse problems with multiple measurement vectors," *IEEE Trans. on Signal Processing*, vol. 53, no. 7, pp. 2477–2488, 2005.

[34] J. Chen and X. Huo, "Theoretical results on sparse representations of multiple measurement vectors," *IEEE Trans. on Signal Processing*, vol. 54, no. 12, pp. 4634–4643, 2006.

[35] M. Fornasier and H. Rauhut, "Recovery algorithms for vector valued data with joint sparsity constraints," *SIAM Journal of Numerical Analysis*, vol. 46, no. 2, pp. 577–613, 2008.

[36] Y.C. Pati, R. Rezaiifar, and P.S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *Asilomar Conference on Signals, Systems and Computers*, 1993, pp. 40–44.

[37] S. Chen, S.A. Billings, and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *International Journal of Control*, vol. 50, no. 5, pp. 18731896, 1989.

[38] L. Rebollo-Neira and D. Lowe, "Optimized orthogonal matching pursuit approach," *IEEE Signal Processing Letters*, vol. 9, no. 4, pp. 137–140, Apr 2002.

[39] D. Donoho, Y Tsaig, I. Drori, and J. Starck, "Sparse solutions of underdetermined linear equations by stagewise orthogonal matching pursuit," Tech. Rep., Stanford University, 2006.

[40] D.L. Donoho, "Neighborly polytopes and sparse solutions of underdetermined linear equations.," Tech. Rep., Statistics Department, Stanford University, 2004.

[41] M. D. Plumbley, "Recovery of sparse representations by polytope faces pursuit," in *Independent Component Analysis and Blind Signal Separation*, 2006, pp. 206–313.

[42] M.D. Plumbley, "On polar polytopes and the recovery of sparse representations," *IEEE Trans. on Information Theory*, vol. 53, no. 9, pp. 3188–3195, Sept. 2007.

[43] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Comm. Pure Appl. Math*, vol. 57, pp. 1413–1541, 2004.

[44] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Annual of Statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[45] A. Beck and M. Teboulle, "A fast iterative "shrinkage-thresholding" algorithm for linear inverse problems," Tech. Rep., Technion - Israel Institute of Technology, 2008.

[46] M.S. Lewicki and T.J. Sejnowski, "Learning overcomplete representations," *Neural Comp*, vol. 12, no. 2, pp. 337–365, 2000.

[47] B.A. Olshausen and D.J. Field, "Sparse coding with an overcomplete basis set: a strategy employed by V1?," *Vision Research*, vol. 37, no. 23, pp. 3311–3325, 1997.

[48] D.P. Wipf and B.D. Rao, "Sparse bayesian learning for basis selection," *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2153–2164, August 2004.

[49] J.E. Tropp and S.J. Wright, "Computational methods for sparse solution of linear inverse problems," Tech. Rep. 2009-01, California Institute of Technology, March 2009.

[50] J.A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. on Information Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.

[51] C. De Vleeschouwer and A. Zakhor, "In-loop atom modulus quantization for matching pursuit and its application to video coding," *IEEE Trans. on Image Processing*, vol. 12, no. 10, pp. 1226– 1242, 2003.

[52] J. Leeuw, "Block-relaxation algorithms in statistics," in Information Systems and Data Analysis, ed. H.H. Bock, W. Lenski and M. M. Richter, Berlin: Springer-Verlag, pp. 308-325, 1994.

[53] K. Lange, D.R. Hunter, and I. Yang, "Optimization transfer using surrogate objective functions," *Journal of Computational and Graphical Statistics*, vol. 9, no. 1, pp. 1–20, 2000.

[54] K Lange, *Optimization*, Springer-Verlag, 2004.

[55] Z. Zhang, J.T. Kwok, and D.Y. Yeung, "Surrogate maximization/minimization algorithms and extensions," *Machine Learning*, vol. 69, no. 1, pp. 1–33, 2007.

[56] A.A. Goldstein, "Convex programming in hilbert space," *Bulletin of the American Mathematical Society*, vol. 70, no. 5, pp. 709–710, 1964.

[57] E.S. Levitin and B.T. Polyak, "Constrained minimization problems," *USSR Computational Mathematics and Mathematical Physics*, vol. 6, pp. 1–50, 1966.

[58] M.A.T. Figueiredo and R.D. Nowak, "An EM algorithm for wavelet-based image restoration," *IEEE Trans. on Image Processing*, vol. 12, no. 8, pp. 906–916, 2003.

[59] T. Blumensath and M.E. Davies, "Iterative thresholding for sparse approximations," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 629–654, 2008.

[60] M. Fornasier and H. Rauhut, "Iterative thresholding algorithms," *Applied and Computational Harmonic Analysis*, vol. 25, no. 2, pp. 187 – 208, 2008.

[61] T. Blumensath, M. Yaghoobi, and M.E. Davies, "Iterative hard thresholding and l0 regularisation," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, April 2007, vol. 3, pp. 877–880.

[62] S. Sardy, A.G. Bruce, and P. Tseng, "Block coordinate relaxation methods for nonparametric wavelet denoising," *Journal of Computational and Graphical Statistics*, vol. 9, no. 2, pp. 361–379, 2000.

[63] L. Landweber, "An iterative formula for Fredholm integral equations of the first kind," *Americam Journal of Mathematics*, vol. 73, pp. 615–624, 1951.

[64] E. Suli and D. Mayers, *An introduction to numerical analysis*, Cambridge University Press, 2003.

[65] F.A. Potra, "On Q-order and R-order of convergence," *Journal of Optimization Theory and Applications*, vol. 63, no. 3, pp. 415–431, 1989.

[66] K. Bredies and D. A. Lorenz, "Linear convergence of iterative soft-thresholding," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 813 – 837, 2008.

[67] R. L. Burden and J. Douglas Faires, *Numerical Analysis*, Brooks/Cole, 1997.

[68] M. Elad, "Why simple shrinkage is still relevant for redundant representations?," *IEEE Trans. on Information Theory*, , no. 12, pp. 5559–5569, 2006.

[69] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo, "Sparse reconstruction by separable approximation," *Signal Processing, IEEE Transactions on*, vol. 57, no. 7, pp. 2479–2493, July 2009.

[70] I. Daubechies, M. Fornasier, and I. Loris, "Accelerated projected gradient method for linear inverse problems with sparsity constraints," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 764 – 792, 2008.

[71] Y.E. Nesterov, "A method for solving the convex programming problem with convergence rate $\mathcal{O}(1/k^2)$," *Dokl. Akad. Nauk SSSR*, vol. 269, pp. 543–547, 1983.

[72] Y. Nesterov, "Gradient methods for minimizing composite objective function," Tech. Rep. 2007/76, CORE Discussion Paper, 2007.

[73] S. Becker, J. Bobin, and E. J. Candes, "Nesta: a fast and accurate first-order method for sparse recovery," submitted for publication, 2009.

[74] J.M. Bioucas-Dias and M.A.T. Figueiredo, "A new twist: Two-step iterative shrinkage/thresholding algorithms for image restoration," *IEEE Transactions on Image Processing*, vol. 16, no. 12, pp. 2992–3004, Dec. 2007.

[75] T. Blumensath and M.E. Davies, "Iterative hard thresholding for compressed sensing," Accepted for publication in Applied and Computational Harmonic Analysis, 2009.

[76] T. Blumensath and M.E. Davies, "Normalised iterative hard thresholding; guaranteed stability and performance," to appear in IEEE Journal of Selected Topics in Signal Processing.

[77] O. Divorra Escoda, L. Granai, and P. Vandergheynst, "On the use of a priori information for sparse signal approximations," *IEEE Trans. on Signal Processing*, vol. 54, no. 9, pp. 3468–3482, 2006.

[78] M.A.T. Figueiredo and R.D. Nowak, "A bound optimization approach to wavelet-based image deconvolution," in *International Conference on Image Processing (ICIP)*, 2005, pp. 782–785.

[79] T. Adeyemi and M.E. Davies, "Sparse representations of images using overcomplete complex wavelets," in *IEEE Workshop on Statistical Signal Processing (SSP)*, July 2005, pp. 805–809.

[80] M. Elad, B. Matalon, J. Shtok, and M. Zibulevsky, "A wide-angle view at iterated shrinkage algorithms," in *Proc. SPIE, Vol. 6701*, 2007.

[81] S.J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large-scale l1-regularized least squares," *IEEE Journal of Selected Topics in Signal Processing: Special Issue on Convex Optimization Methods for Signal Processing*, vol. 1, no. 4, pp. 606–617, 2007.

[82] E. van den Berg and M.P. Friedlander, "Probing the pareto frontier for basis pursuit solutions," *SIAM Journal on Scientific Computing*, vol. 31, no. 2, pp. 890–912, 2008.

[83] M. Yaghoobi, *Adaptive Sparse Coding and Dictionary Selection*, Ph.D. thesis, Institute for Digital Communications (IDCom), The University of Edinburgh, 2009.

[84] K.B. Petersen and M.S. Pedersen, *A Matrix Cookbook*, Novemver 14, 2008.

[85] B. Mailhe, R. Gribonval, F. Bimbot, and P. Vandergheynst, "A low complexity orthogonal matching pursuit for sparse signal approximation with shift-invariant dictionaries," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, April 2009.

[86] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the l1-ball for learning in high dimensions," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 272–279, Helsinki, Finland.

[87] I. Loris, "On the performance of algorithms for the minimization of l1-penalized functionals," *Inverse Problems*, vol. 25, no. 3, pp. 1–16, 2009, :035008.

[88] M. Afonso, J. Bioucas-Dias, and M. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," Submitted to the IEEE Transactions on Image Processing, Available at `http://arxiv.org/abs/0910.4887`, 2009.

[89] M. V. Afonso, J. Bioucas-Dias, and M. Figueiredo, "A fast algorithm for the constrained formulation of compressive image reconstruction and other linear inverse problems," Submitted, Available at `http://arxiv.org/abs/0909.3947v1`, 2009.