The Dissertation Committee for Ali Jalali
certifies that this is the approved version of the following dissertation:

Dirty Statistical Models

Committee:

Sujay Sanghavi, Supervisor

Constantine Caramanis

Inderjit Dhillon

Joydeep Ghosh

Pradeep Ravikumar

Dirty Statistical Models

by

Ali Jalali, B.S.; M.S.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2012

Dedicated to Soltän.

Acknowledgments

I wish to thank all professors, students and staff of The Wireless Networking and Communication Networking Group at UT Austin for their kind helps throughout my PhD. In particular, I would like to thank my adviser, Prof. Sujay Sanghavi, whose constant support was always encouraging me to learn more and without his dedication, this thesis was impossible. I would also like to express my gratitude to Prof. Pradeep Ravikumar for insightful discussions that helped me find my research path. My special thanks to my parents and little sister for their unconditional love and support during my hardest times. I am running out of words when it comes to thanking my friends, my family by choice, for being always with me in difficulties.

Dirty Statistical Models

Publication No. _____

Ali Jalali, Ph.D.
The University of Texas at Austin, 2012

Supervisor: Sujay Sanghavi

In fields across science and engineering, we are increasingly faced with problems where the number of variables or features we need to estimate is much larger than the number of observations. Under such high-dimensional scaling, for any hope of statistically consistent estimation, it becomes vital to leverage any potential structure in the problem such as sparsity, low-rank structure or block sparsity. However, data may deviate significantly from any one such statistical model. The motivation of this thesis is: can we simultaneously leverage more than one such statistical structural model, to obtain consistency in a larger number of problems, and with fewer samples, than can be obtained by single models? Our approach involves combining via simple linear superposition, a technique we term *dirty models*. The idea is very simple: while any one structure might not capture the data, a superposition of structural classes might. Dirty models thus searches for a parameter that can be *decomposed* into a number of simpler structures such as (a) sparse plus block-sparse, (b) sparse plus low-rank and (c) low-rank plus block-sparse. In this thesis, we propose dirty model based algorithms for different problems such as multi-task learning, graph clustering and time-series analysis with latent factors. We analyze these algorithms in terms of the number of observations we need to estimate the variables. These algorithms are based on convex optimization and sometimes they are relatively slow. We provide a class of low-complexity greedy algorithms that not only can solve these optimizations faster, but also guarantee the solution. Other than theoretical results, in each case, we provide experimental results to illustrate the power of dirty models.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In many applications such as pattern recognition, machine vision, bioinformatics, data mining, financial engineering, etc, we face parameter estimation problems where the number of observations is much less than the number of the dimension of the parameter. In such a high-dimensional regime, there is no hope for consistent estimation unless we restrict the parameter to have a certain structure model. In other words, the true parameter lives in a high-dimensional space, but the observed data has much lower intrinsic dimensions. The hope is that few observations suffice to recover the parameter in the low-dimensional sub-space. Common, and recently popular, structure models are

- **Sparsity**: This model suggests that most entries of the unknown parameter are zero and there are only few non-zero elements. This structure has been studied in many areas such as compressed sensing [13] and LASSO [131].
- **Block Sparse**: This model partitions the entries of the unknown parameter into a number of subsets (blocks) and suggests that within each block, either all entries are zero or most of them, maybe all, are non-zero [26, 111].
- **Low-Rank**: This model suggests that the parameter matrix has a low-dimensional row/column space and hence has much lower rank compare to its size [102, 116].
- **Sparse Markov Random Field**: This model suggests that the graphical model associated with a set of random variables is sparse [112, 113]; that is the joint probability distribution of a set of random variables can be factorized such that each factor is the joint probablity distribution of a subset of those random variables.

In each of these models, if we knew apriori the exact instance of the overall dimensional structure, inference is easy. The challenge however is finding the correct instance of the structure from a very large number of possibilities. For example, in sparse linear regression, if we know the location of non-zero entries, then we can set the other entries to zero and estimate only non-zero entries, say by solving a convex optimization problem. Similarly, in block sparse model, if we know which blocks are non-zero, or in low-rank model, if we know the singular vectors of the matrix, then we can estimate non-zero entries with very few observations. Hence, recovering the structure of a model is the critical task in high-dimensional setting. Once the structure of the model is recovered, estimation is a fairly easy job.

There is a hidden assumption behind choosing a certain model for the unknown parameter. In fact, we assume that the underlying true data structure falls into one of these models completely, i.e., the data is *clean*. Notice that when we use the term clean data, we do not necessarily mean that there is no niose. For example, if x is a vector, then $Q = xx^T + w$, with additive noise w, can be considered to have low-rank model (which is a clean model). However, in many real applications, we face more complex structures, where the underlying data structure does not fit any model solely. Instead, the data structure model can be expressed as the superposition of a number of simpler models, i.e., the data is *dirty*. For example, we can cosider the aformentioned matrix Q to be a superposition of low-rank and sparse models, assuming that the noise has only few non-zero elements. We call such a superposition of simple models a *dirty model*.

Our main idea in this thesis is to analyze dirty models for dirty data and try to recover the structure of each of these models. We have analyzed recovery of different dirty models via convex optimization and greedy algorithms in terms of their sample complexity, structure recovery guarantees and comparison to other clean models. As discussed before, once the structure is revealed, the estimation is easy because the unknown parameter will be restricted to a low-dimensional subspace.

1.1 Structure Recovery Techniques

In this section, we discuss two main structure recovery techniques and their properties. Instances of each of these techniques as well as relevant applications are presented in the consequent chapters of this thesis.

1.1.1 Convex Optimization

The use of convex optimization for structure recovery is very broad and popular. At a high level, given a set of observation D, we would like to estimate a target parameter θ^*. There are two ingredients for a consistent convex optimization based structure recovery algorithm as follows:

- **Loss Function:** We need a convex loss function $\mathcal{L}(\theta; D)$ that given our observation, it assigns a *penalty* to each parameter θ. This loss function must have θ^* as the asymptotically optimal point, i.e., $\mathbb{E}_D\left[\nabla \mathcal{L}(\theta^*; D)\right] = 0$. Moreover, the more curvature the loss function around θ^* has, the easier the estimation will be.

- **Regularizer:** In a high-dimensional setting, to get a consistent estimation, we assume that $\theta^* \in \mathcal{C}$ for some structure set $\mathcal{C}$. Examples of such a set are the set of 1-sparse vectors or the set of 1-rank matrices. We then construct a function $R_{\mathcal{C}}(\theta)$ to be the function that encourages to pick elements from the set $\mathcal{C}$ as opposed to an element from outside, i.e., $R_{\mathcal{C}}(\theta^*) \leq R_{\mathcal{C}}(\theta)$ for any $\theta \notin \mathcal{C}$. This function is often referred to as *regularizer*. Since the set $\mathcal{C}$ and consequently the regularizer $R_{\mathcal{C}}$ are often not-convex, we pick the convex envelope $\bar{R}_{\mathcal{C}}$ to ensure tractability.

Using these two ingredients, we solve the following convex program to get an estimate of θ^*.

$$\widehat{\theta} = \arg\min_{\theta} \ \mathcal{L}(\theta; D) + \lambda \bar{R}_{\mathcal{C}}(\theta).$$

The parameter λ can be chosen via techniques like cross-validation or bootstrapping. This class of algorithms are known to have two main issues:

- **Robustness:** A single observation can arbitrarily change $\widehat{\theta}$ and hence, the method is not robust with respect to outliers.
- **Flexibility:** If θ^* represents a multi-modal data, then $\underline{\theta}$ potentially represents an *average* mode which is not a representative of neither modes of the data.

Using our dirty model, we can fight against both issues. Imagine $\theta^* = \theta_1^* + \theta_2^*$ where θ_1^* and θ_2^* come from two structures $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively. To get robustness, we choose $\mathcal{C}_1$ to be the structure of our data and $\mathcal{C}_2$ to be the structure of the outliers. Similarly, to get flexibility, we choose $\mathcal{C}_1$ and $\mathcal{C}_2$ to represent different modes of the data. Subsequently, we solve the following convex optimization problem:

$$(\widehat{\theta}_1, \widehat{\theta}_2) = \arg\min_{\theta_1,\theta_2} \mathcal{L}(\theta_1 + \theta_2; D) + \lambda_1 \bar{R}_{\mathcal{C}_1}(\theta_1) + \lambda_2 \bar{R}_{\mathcal{C}_2}(\theta_2).$$

In the robust case, we output $\widehat{\theta} = \widehat{\theta}_1$ and in the flexible case, we output $\widehat{\theta} = \widehat{\theta}_1 + \widehat{\theta}_2$. In this thesis, we investigate some instances of this problem for both robustness (Chapters 3,4,5) and flexibility (Chapter 2) and show the advantages of this dirty model based algorithm over simple model counterparts.

One of the difficulties with robust dirty model analysis is to characterize different structures so that without any ambiguity they can be distinguished. Recall the example of low-rank and sparse models superposition and notice that a low-rank matrix can be also sparse or conversely, a sparse matrix can be low-rank. Thus, we need to not only reduce the size of the problem by imposing the structure, but also further restricting each structure to be consistently incoherent from each other to get robustness. Without this separation, the problem is not well-defined and characterization of the models is impossible. Motivated by this, we normally impose some conditions that uniquely partitions the space of parameters so that each non-zero element of the space belongs to only one structure.

1.1.2 Greedy Algorithms

Considering the original non-convex regularizer $R_{\mathcal{C}}(\cdot)$, the *quality* of the convex optimization method depends on the tightness of the convex relaxation $\bar{R}_{\mathcal{C}}(\cdot)$. Often to ensure tightness, convex optimization problems require strong assumptions such as irrepresentable condition that imposes strong restrictions on the loss function. Besides, convex optimization methods, although polynomial, are still computationally expensive. In this thesis, we show that greedy algorithms are good candidates to both reduce the computational complexity and relax strong assumptions while maintaining the same guarantees as convex optimization.

The main idea of greedy algorithms is basis pursuit and in particular orthogonal matching pursuit [135, 154] – that is iteratively find the next best "coordinate" in $\mathcal{C}$ and add it to the estimated structure. This simple forward greedy algorithm has been shown to perform as good as the convex optimization. However, it still requires the same strong assumptions. Like many forward algorithms, the beginning steps are far more important than the others and if the first steps are mistaken, then the algorithm performs poorly.

We introduce a forward-backward greedy algorithm to solve sparse + block-sparse dirty model based on the algorithms introduced in [68, 152]. We show this algorithm not only lowers the sample complexity, but also requires only mild assumption of restricted strong convexity [100]. This algorithm greedily picks the best coordinates in the forward step and then removes "bad" coordinates in the backward step. We show both theoretically and empirically outperforms the corresponding convex optimization method.

1.2 Studied Dirty Models

We studied different instances of two major classes of dirty models introduced in the next two sections. These dirty models include sparse+block-sparse and sparse+low-rank models.

1.2.1 Sparse + Block Sparse

Suppose we have multiple linear regression problems, i.e., multi-task learning problem [26]. Here, multiple tasks share some common structure such as sparsity, and estimating these tasks jointly by leveraging this common structure could be more statistically efficient. We have $r > 1$ response variables (tasks), and a common set of p features or covariates. The setting we focus on is where the response variables have *simultaneously sparse* structure: the index set of relevant features for each task is sparse; and there is a large overlap of these relevant features across the different regression problems. Such "simultaneous sparsity" arises in a variety of contexts [133]; indeed, most applications of sparse signal recovery in contexts ranging from graphical model learning, kernel learning, and function estimation have natural extensions to the simultaneous-sparse setting [7, 109, 111].

It is useful to represent the multiple regression parameters via a matrix, where each column corresponds to a task, and each row to a feature. Having simultaneous sparse structure then corresponds to the matrix being largely "block-sparse" – where each row is either all zero or mostly non-zero, and the number of non-zero rows is small. A lot of recent research in this setting has focused on ℓ_1/ℓ_q norm regularizations, for $q > 1$, that encourage the parameter matrix to have such block-sparse structure. Particular examples include results using the ℓ_1/ℓ_∞ norm [101, 136, 151], and the ℓ_1/ℓ_2 norm [89, 104].

Our method searches for a parameter matrix that can be *decomposed* into a row-sparse matrix (corresponding to the overlapping or shared features) and an elementwise sparse matrix (corresponding to the non-shared features). As we show both theoretically and empirically, with this simple fix we are able to leverage any extent of shared features, while allowing disparities in support and values of the parameters, so that we are *always* better than both the Lasso or block-sparse regularizers (at times remarkably so). In Chapter 2, we provide a convex optimization based algorithm and in Chapter 6, we provide a forward-backward greedy algorithm for the same problem.

1.2.2 Sparse + Low-Rank

In many applications such as principal component analysis, often times we want to recover a low-rank matrix and a sparse matrix from their sum. Additive large-magnitude noise potentially can change the rank of the matrix by much. In such a heavily noisy regime, PCA based approaches has poor performance. Thus, recovering the underlying low-rank structure in presence of the noise is a challenging job. We model this problem using dirty models and cast it as a convex optimization problem. We consider graph clustering and time-series analysis as applications of using this dirty model.

1.2.2.1 Graph Clustering

The problem we look at is: given an unweighted graph, partition the nodes, i.e., cluster, so as to minimize the sum of (a) number of edges between endpoints that are in different partitions, and (b) the number of missing edges between endpoints in the same partition. This is one particular non statistical application of sparse and low rank matrix separation. Given a clustering, we call edges of type (a) or (b) "disagreements"; we are thus interested in optimal clusterings – those that minimize the number of disagreements. This formulation has the advantage of not requiring an external parametric input of how many clusters there should be in the final solution; this is fully determined by the data at hand.

This problem, as formulated above, is exactly the same as the problem of correlation clustering, first proposed by Bansal, Blum and Chawla [10]. Specifically, edges that [10] labels as "+" are those that are "present in the graph" for us, and those labelled as "-" in [10] are those that are "missing". They consider the problem with +/- labels on the complete graph. [10] showed that finding the exact optimum is NP-complete; they then proceed to provide a constant-factor approximation algorithm for minimizing the number of disagreements.

We take an alternative approach to the problem; instead of looking for an approximation that holds for all problem inputs, we provide algorithms that either (a) yields the optimum (i.e. disagreement minimizing) clustering, or (b) generates a FAILURE flag, yielding no clustering. Our algorithm is

based on using recently developed matrix uncertainty principles [22, 31] - that (certain classes of) matrices cannot be simultaneously sparse and low-rank. For these matrices, we can *exactly* recover the sparse matrix ($\mathbf{B}$) and low-rank matrix ($\mathbf{K}$) given only their sum ($\mathbf{B}$+$\mathbf{K}$), using convex optimization. In the graph context, the low-rank matrix corresponds to the cliques that would be an ideal input corresponding to the optimal clustering, the composite matrix represents the given data, and the sparse matrix represents the disagreements. Existing results on sparse and low-rank matrix decompositions provide weak guarantees for the graph clustering case; we consider two types of relaxation based on nuclear norm in Chapter 3 and max-norm in Chapter 4 and provide much stronger guarantees.

1.2.2.2 Time-Series Analysis

Suppose we have a non-stationary random vector whose mean and variance evolves over the time and we observe some entries of this vector for a finite time. The evolution of the mean and variance of each entry depends on both observed and unobserved (latent) entries. The problem we would like to investigate is that whether or not it is possible to learn the evolution process of the mean and variance of the observed entries. We consider linear stochastic first-order evolution model $\dot{x}(t) = Ax(t) + \dot{B}(t)$, where x is the random vector, A is the evolution coefficients matrix and $B(t)$ is the standard Brownian motion noise, and learn the matrix A; the problem that is investigated for the case when there is no latent variable in [15].

We formulate this problem as a sparse plus low-rank dirty model of $A = S+L$, where, S captures the effect of observed variables on each other and L captures the effect of latent variables on observed entries. Here, we assume that few observed entries affect each observed entry and also each observed entry affects only few observed entries, and hence, S is sparse. Moreover, we show that if the number of latent variables is less than observed entries, then the matrix L is low-rank. Unlike most cases of studies in sparse and low-rank decomposition, here, the focus is to learn the structure of the sparse matrix S rather than the low-rank matrix L.

There are similarities between this problem and learning Gaussian graphical models with latent variables [28]. However, at least in [28] the focus has

been to recover the number of latent variables, which is the rank of L, and more importantly, in graphical model learning, the assumption is that the samples are independent and law of large numbers can be applied easily. We need to provide much more subtle analysis since because of the time evolution, our samples are highly correlated and hence, the usual concentration results (of independent variables) cannot be applied. In Chapter 5, we investigate this problem and provide guarantees of success for our method. Further, we apply our method to stock market prediction problem and observe significant improvements.