Feature Selection under Multicollinearity
&
Causal Inference on Time Series

A THESIS
SUBMITTED FOR THE DEGREE OF
Master of Science (Engineering)
IN THE
Faculty of Engineering

BY

Indranil Bhattacharya

Computer Science and Automation
Indian Institute of Science
Bangalore – 560 012 (INDIA)

November, 2017

DEDICATED TO

My Parents

Thank you for being there with me through thick and thin.

Acknowledgements

There are so many people to thank for helping me, guiding me, supporting me during the last three years of my life here at IISc. I will try my best to acknowledge all of them without being my usual long winded self.

I am very grateful to GOD, The ALMIGHTY, for without His graces and blessings this would not have been possible. I am extremely indebted to my family, especially my mother for her unwavering belief in my abilities that kept me going even under tougher times.

I would like to express my gratitude to my research advisor Dr. Arnab Bhattacharyya for his constant guidance, encouragement and patience for the last three years. He has always been my source of inspiration. He gave me the freedom of taking up whatever courses I felt were aligned to my interests, and at the same time he also provided me with all sorts of technical as well as non-technical support to keep me focused on my research. Not only these, he also encouraged me, gave me enough opportunities to collaborate with researchers both from academia as well as the industry. I have learned a lot from him during these years and I will always remain grateful to him. He has a great approach to research and life in general and I hope to emulate it in near future.

I would like to thank Dr. Praveen Pankajakshan from Shell Technology Center, Bangalore for providing indispensable advice, knowledge, guidance and support on the research projects that I have worked upon in collaboration with Shell. Thank you Praveen for all the planned and unplanned discussions, hangouts which have benefited me immensely. It was a pleasure working with you.

I would like to thank our departmental chairman Prof. Jayant Haritsa and our divisional chairman Prof. Y Narahari for their constant support and encouragement. They have always been very kind to me. I would like to thank everybody in CSA office for taking care of the administrative tasks smoothly and making my stay here hassle free. My heartfelt thanks to the MHRD for supporting me financially throughout my Masters' program. I am extremely indebted to Indian Institute of Science, Bangalore and the Department of Computer Science and Automation especially for all the facilities it has provided me.

I would also like to thank my current lab mates Suprovat, Saravanan, Anurita, Ashish, Arshed, Naganand and my ex-lab mates Kiran, Chetan and Amit with whom I have had frequent discussions on several research problems. They have helped me understand a lot about research and seeing them working productively have always motivated me to work harder and improve the quality of my research.

I would like to thank all my dear friends Arpita, Suvam, Sambaran, Aniket da, Abhiruk da, Sayantan da, Abhijat, Rupam, Suman, Divya, Kuntal, Rintu and Anubhav for making IISc a memorable and eventful episode in the journey of my life.

Last but not the least, I would also like to thank all of them who knowingly and unknowingly supported me, helped me and made my life at IISc memorable.

Publications based on this Thesis

1. Praveen Pankajakshan, Suchismita Sanyal, Onno E. de Noord, Indranil Bhattacharya, Arnab Bhattacharyya, Umesh Waghmare. *Machine Learning and Statistical Analysis for Materials Science : Stability Analysis, Fingerprint Descriptors and Chemical Insights*
   - Submitted to The Journal of Chemistry of Materials, 2017

2. Indranil Bhattacharya, Arnab Bhattacharyya, Praveen Pankajakshan. *Temporal Lag estimation and Granger Causality on time series*
   - Submitted to KDD, 2017 (Research track)

Abstract

In this work, we study and extend algorithms for *Sparse Regression* and *Causal Inference* problems. Both the problems are fundamental in the area of Data Science.

The goal of regression problem is to find out the "best" relationship between an output variable and input variables, given samples of the input and output values. We consider *sparse regression* under a high-dimensional *linear* model with *strongly correlated* variables, situations which cannot be handled well using many existing model selection algorithms. We study the performance of the popular feature selection algorithms such as LASSO, Elastic Net, BoLasso, Clustered Lasso as well as Projected Gradient Descent algorithms under this setting in terms of their running time, stability and consistency in recovering the true support. We also propose a new feature selection algorithm, BoPGD, which cluster the features first based on their sample correlation and do subsequent sparse estimation using a bootstrapped variant of the projected gradient descent method with projection on the non-convex L_0 ball. We attempt to characterize the efficiency and consistency of our algorithm by performing a host of experiments on both synthetic and real world datasets.

Discovering *causal* relationships, beyond mere *correlation*, is widely recognized as a fundamental problem. The Causal Inference problems use observations to infer the underlying causal structure of the data generating process. The input to these problems is either a multivariate time series or i.i.d sequences and the output is a *Feature Causal Graph* where the nodes correspond to the variables and edges capture the direction of causality. For high dimensional datasets, determining the causal relationships becomes a challenging task because of the *curse of dimensionality.* Graphical modeling of temporal data based on the concept of "Granger Causality" has gained much attention in this context. The blend of Granger methods along with model selection techniques, such as LASSO, enables efficient discovery of a "sparse" subset of causal variables in high dimensional settings. However, these temporal causal methods use an input parameter, L, the *maximum time lag.* This parameter is the maximum gap in time between the occurrence of the output phenomenon and the causal input stimulus. However, in many situations of interest, the maximum time lag is not known, and indeed, finding the range of causal effects is an important problem. In this work, we propose and evaluate a data-driven and computationally efficient method for Granger causality inference in the Vector Auto Regressive (VAR) model without foreknowledge of the maximum time lag. We present two algorithms *Lasso Granger++* and *Group Lasso Granger++* which not only constructs the hypothesis feature causal graph, but also simultaneously estimates a value of maxlag ($\hat{L}$) for each variable by balancing the trade-off between “goodness of fit” and “model complexity”.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction to Feature Selection

High dimensional regression and sparse estimation is used in many real-world applications nowadays. In situations where the number of parameters (features) handily outnumbers the number of observations (samples), the problem of consistent estimation becomes harder due to (near) non-identifiability. Also when some (or all) of the features themselves exhibit strong empirical correlation (near linear dependence), estimating the right subset of features in terms of their regressive influence, becomes highly inconsistent. For example, in Genomics, the group of genes sharing the same biological pathway [SDC03] have strong linear dependence. Same is true in genome-wide association studies where SNPs are strongly correlated within segments of the DNA sequence [Bal06]. In our current work, we focus on the simple yet useful *high dimensional linear model.*

In Machine Learning, the *Least Square Linear Regression* problem models the "best" *linear* relationship between the output (also called response/target) variable and the input variables (also called features/predictors/covariates), given samples containing both. Formally, given a response vector $y \in \mathbb{R}^n$, a design matrix $X \in \mathbb{R}^{n \times p}$ of predictor variables, a true underlying coefficient vector $\beta_0 \in \mathbb{R}^p$ and an $n \times 1$ error/noise vector ζ, the linear model can be expressed as follows :

$$y = X\beta_0 + \zeta \tag{1.1}$$

The objective of *Ordinary Least Square Linear Regression (OLS)* is to find out a value of β which minimizes the empirical risk function $f(\beta) = \frac{1}{n}\sum\limits_{i=1}^{n} \ell(y_i, \langle x_i, \beta\rangle)$, where $\ell(y_i, \langle x_i, \beta\rangle) = (y_i - x_i^T\beta)^2$ is the usual least squared loss function.

For high-dimensional data, *model interpretation* becomes a challenging task. That is why *Feature Selection* methods are often employed to obtain a *sparse* yet most *significant* subset of features that affect the output. These methods improve the model prediction accuracy by dropping off irrelevant, noisy features from the model. Although this makes the estimator biased, but at the same time it also reduces the variance in estimation substantially making the mean squared error of estimation sufficiently low. In literature, this problem of least square linear regression with feature selection is popularly known as the *Sparse Linear Regression* problem which alongside minimizing the empirical risk also outputs a sparse coefficient vector $\hat{\beta}_{SPL}$ as shown in the following optimization problem :

$$\hat{\beta}_{SPL} \in \underset{\beta\in\mathbb{R}^p:\|\beta\|_0\leq s}{\arg\min} \frac{1}{2n}\|y - X\beta\|_2^2 \tag{1.2}$$

The non-zero entries of β form its *support*, denoted by $S = \{j : \beta(j) \neq 0, j = \{1, 2, ..., p\}\}$. However, optimal estimation of problem 1.2 is infeasible because of *NP-hardness* results due to *non-convexity* constraint of the L_0 norm operator.

An important line of research has been developed to address this problem by relaxing the non-convex L_0 constraint with appropriate convex counterparts (since convex problems are easy and global minima is achievable). A class of *Iterative Shrinkage Thresholding Algorithms* (ISTA) [BT09] have been developed which does convex-relaxation of 1.2, the most popular among them being LASSO [Tib96] which does least square regression with regularization by the ℓ_1 norm. Lasso precisely solves for a β s.t :

$$\hat{\beta}_{LASSO} \in \underset{\beta\in\mathbb{R}^p}{\arg\min} \frac{1}{2n}\|y - X\beta\|_2^2 + \lambda\|\beta\|_1 \tag{1.3}$$

where $\lambda \geq 0$ is a tuning parameter (called regularization/penalization factor). Depending on the value of λ, the Lasso estimate will have many coefficients of $\hat{\beta}_{LASSO}$ set to zero, because of the nature of ℓ_1 penalty. The features corresponding to the support of $\hat{\beta}_{LASSO}$ is regarded as the *active set* of predictors which captures the maximum information in the linear relationship between y and X. Lasso is a popular choice when it comes to feature selection due to its model consistency behavior such as neighborhood stability [MB06], irrepresentable condition [ZY06] assuming various conditions on the design matrix X. However, there exists situations when *strong correlation* (or near linear dependence) among a few or all predictors bring *instability* in Lasso estimate i.e. Lasso tends to select different variables from the group of correlated variables, even if some or all of them belong to the true active set S_0. A couple of Lasso variants viz. Relaxed Lasso [Mei07], Adaptive Lasso [Zou06] and a few other convex-relaxation based methods such as the elastic-net [ZH05], clustered Lasso [She08], OSCAR [BR08], BoLasso [Bac08], Cluster Representative Lasso (CRL) and Cluster Group Lasso (CGL) [BRvdGZ13] have been proposed to address this problem. However, some of them have slower convergence rates since the optimization problems that they end up solving is non-smooth. Also none of the above methods except CRL and CGL, *explicitly* take into account the correlation structure among the variables and thus still exhibit difficulties when groups of variables are nearly linearly dependent. Although the CRL and CGL methods are extremely fast, yet they have a major drawback. In the process of avoiding false negatives i.e. to avoid not selecting an active variable from the true support S_0, these methods often overestimate S_0 and have to pay a price for this by an increase in the number of false positives.

Instead, the methods of choice for most practical applications are actually the Projected Gradient Descent (PGD) (also called Iterative Hard Thresholding (IHT)) methods. These methods directly project the gradient descent update onto the underlying non-convex feasible set. This projection is not the usual prox-operator like that of any convex prox function, but hard-thresholding which can be performed very efficiently for several interesting structures such as sparsity, low rank etc. The work of Jain et. al. [JTK14] have shown that these PGD/IHT methods are able to achieve global convergence when the risk function satisfies Restricted Strong Convexity (RSC) and Restricted Strong Smoothness (RSS) properties with suitable choices of Lipschitz's constants as long as the projection is performed onto sets of sparsity $s \gg s^*$, s^* being the size of the optimal support (equation 1.2).

In this context, we have proposed a new feature selection algorithm called the ***B**ootstrap* ***P**rojected* ***G**radient* ***D**escent* (BoPGD). It falls under the IHT/PGD regime, but in practice it is faster than PGD and other L_1 and greedy counterparts. Also it takes into account the inter-feature correlation, and cluster features based on their sample correlation. Subsequently it performs sparse estimation using a bootstrapped-variant of the PGD/IHT method with appropriately chosen sparsity parameter. We attempt to characterize the efficiency, consistency and scalability of our algorithm and showcase that it is comparatively faster than others, by performing a host of experiments on both synthetic and real world datasets. This sheds light for under what kind of models and scenarios, our four-step approach, - (i) Clustering the variables first based on sample correlation, (ii) Subsequent sparse estimation using PGD on the set of clusters, (iii) Using Bootstrapping [ET94] to eliminate the selection of noisy clusters, and in turn nuisance features too, and, finally, (iv) Another sparse estimation using PGD on the reduced set of features to extract the active set, is beneficial.

1.2 Introduction to Causality

Discovering *causal relationships* among multiple natural or artificial processes has been there since the dawn of human scientific history. With the advancements in technology, we are now living in an era of “Big Data” where massive amounts of time series data have become available for analysis and mining. For example, today, we have access to terabytes of micro-array time series data which records the gene-expression levels under different treatments over time; we can easily retrieve petabytes of climate and meteorological data containing measurements of several climate enforcing agents such as temperature, solar radiation, green-house gases (viz. CO_2, CH_4) concentration in the atmosphere etc.; sensor data, IOT data monitoring the functionality of complex natural and artificial processes are everywhere now; and exabytes of social media (e.g facebook, twitter) contents are generated rapidly over time on the Internet. Yet extracting the *causal structure* from the data itself is quite challenging since most of the data points reside in a very high dimensional space. How to develop efficient and scalable learning algorithms to uncover the temporal dependency structures between time series and reveal insights from data has become a key problem in machine learning and data mining community today. It is, however, well understood that mere *statistical correlation* does not imply *causation* [Mat00, Pea09], but under some specific conditions, it might be possible to derive causality from correlations in the observed data [Arn99]. Since our present work is focused entirely on causal inference of time series data, henceforth, unless otherwise mentioned, all the causal relationships we mention are on temporal lagged variables only.

One of the earliest works in quantifying the causal relationship amongst temporal variables was introduced in the field of econometrics by the Nobel laureate Clive Granger (1969). The notion he proposed is now popularly known as the *Granger Causality* [Gra69, Gra80]. Although there has been extensive debates on the validity of this causal model, yet it is generally believed to capture a significant empirical aspect of causality, especially when working with time series data. In particular, it has very little to say about situations where there is a hidden confounding variable causally influencing two observed variables, or when there is an indirect chain of causal structures. We will not be addressing such issues in our current work.

Granger Causality was initially introduced for a pair of variables, and the question of how to extend the same for multivariate time series data was not clearly addressed. Later, the frameworks of Bayesian networks [Hec98, Mee97] and the related feature causal networks [MS06] have been introduced as suitable tools for addressing this problem. Many algorithms, such as Auto-correlation, Cross-correlation [BJRL15], Randomization tests [OE94], statistical tests [SGS00] have been proposed to recover the temporal structures among multiple variables but they are not so computationally efficient under the higher dimensional setting. Recently, there has been a growing interest in combining the notion of Granger Causality with model selection techniques such as LASSO [Tib96], Group-LASSO [ALA07, LLNM$^+$09, LALR09] for extracting the temporal causal structure from high dimensional data. One of the major advantages of using LASSO is its statistical consistency. It has been proven [MB06] that the probability of Lasso falsely including any of the non-neighboring variables of a given node into its neighborhood estimate vanishes exponentially fast, even if the number of non-neighboring variables may grow very rapidly with the number of observations.

Most of the methods for mining Granger Causality on time series data use the Vector Autoregressive model (VAR) [Lüt11] with a *fixed* value of *maximum time lag* or *maxlag* (also called the *model order* in different context), denoted by L. Since they quintessentially mean the same thing, we will use them interchangeably throughout this discussion. This parameter *maxlag* is the maximum time in past to look back to when regressing for present and subsequently future values of the target time series variable as a function of all other causal variables in the system. Incorporating a "good" estimate of L requires sufficient domain knowledge of the underlying physical system being modeled since it captures the maximum gap in time between the occurrence of the output phenomenon and the causal input stimulus. Both over estimate and under estimate of L impacts the Mean Squared Error (MSE) of estimation. Moreover, when L is sufficiently large there is a huge blow-up in the effective number of features and any algorithm estimating L accurately will have to deal with the space and time complexity issues efficiently. Although techniques such as Generalized Cross-Validation [K$^+$95, BA03, GHW79], autocorrelation function (ACF) and partial auto-correlation (PACF) function [Wei94] exist to address this issue, but in most context they are either inapplicable (for eg. if the data does not exhibit any periodicity) or extremely inefficient when applied to high volume of time series data. Lütkepohl Helmut, in his book titled "New introduction to multiple time series analysis" [Lüt05] discusses a sequence of several *Statistical Hypothesis Testing* schemes such as log-likelihood ratio test, F-Statistic test etc. to determine the "best" estimate of model order L, in a VAR(L) process. All of these hypothesis testing schemes are very effective in deducing a "good" estimate of model order when the number of variables is small, but they do not scale well with the dimensionality of the data since they do not encourage any form of sparsity constraint in their estimation framework.

Our current work precisely focuses on this problem of "Granger Causal Inference" on *multivariate*, *high-dimensional* time series data modeled as a vector autoregressive process but with *unknown model order*. We propose a semi-automated way of estimating this *maxlag* parameter which "best" fits the Granger Causal model to the given time series data with high accuracy.

Our method concurrently estimates the "best" (in terms of *"goodness of fit"*) value of L, and also outputs the coefficients of the causal variables in the VAR process using standard model selection mechanism. The latter is used to reconstruct the hypothesis feature causal graph. Our approach is purely data-driven and so prior domain knowledge of the underlying physical system is not mandatory. Importantly, it is scalable over high dimensions, both in terms of space and time complexity, whereas a brute force search for the same would simply not be feasible even for moderately large datasets. We attempt to characterize the performance of our method by conducting a host of experiments on synthetic as well as on real-world time series datasets.

1.3 Organization

The entire thesis is broadly divided into two parts based on the two problems defined before. The first part is focused entirely on Feature Selection and the second part on Granger Causality. We have already started chapter 1 with an introduction to both these well-studied areas of Feature Selection and Granger Causality on time series data. We have also addressed the motivation behind our work along with the background survey. We briefly mentioned our contribution too and the results we state further. We describe more about the two problems in the future chapters.

The section on Feature Selection is organized as follows :

In chapter 2, we formally define the problem along with its preliminaries. We begin with defining what is called High dimensional Sparse Estimation. We then introduce the classical yet useful linear model and accordingly the notion of Sparse Linear Regression. We also provide some definitions and notations that we will refer later during the course of this discussion.

Chapter 3 covers a brief survey of some of the existing and most significant methods of Feature Selection. We describe all the different categories of feature selection techniques and the algorithms that belong to these categories along with their merits and demerits.

In chapter 4, we introduce our new feature selection algorithm ***Bo**otstrap* ***P**rojected* ***G**radient* ***D**escent* (BoPGD). We provide both intuition and detailed insights into the four key steps of our algorithm.

In chapter 5, we present the results of our experimental evaluation on a host of synthetic as well as public datasets demonstrating the accuracy, efficiency and scalability of our algorithm.

The section on Causal Inference is organized as follows :

In chapter 6, we formally define the problem along with its preliminaries. We begin with defining what do we mean by a Feature Causal Network and the notion of time lag. We then introduce the vector autoregressive model (VAR) which is the framework of our causal model. Then we state the key notions of “Granger Causality” and introduce the *Linear Granger Causality test* both on bi-variate and multi-variate data. We end this chapter highlighting the data generation process and the evaluation criteria for measuring graph similarity.

Chapter 7 briefly describes some of the existing and most significant methods for modeling Granger Causality on time series data. All of these methods use VAR model with a fixed value of the model order known a priori based on domain knowledge or intuition of the underlying data generating process.

In chapter 8, we introduce our algorithms *Lasso Granger++* and *Group Lasso Granger++* which are extensions of the Lasso Granger and Group Lasso Granger frameworks for modeling Granger Causality on vector autoregressive time lagged data. We again emphasize that we do not fix the model order (maxlag) beforehand. We also describe the model selection criterion used for selecting the “best estimate” of L.

In chapter 9, we present the results of our experimental evaluation on a host of synthetic as well as public datasets on time series demonstrating the accuracy, efficiency and scalability of our algorithm.

The final chapter 10 concludes our work on both the areas and mentions some of the future work directions too.

Part I

Feature Selection under Multicollinearity

Chapter 2

Preliminaries

In this section we formally introduce Feature Selection and describe the key notions and notations used. We also provide some basic definitions and concepts that are aligned to the problem. We shall refer them later in future chapters.

2.1 High dimensional Sparse Estimation

Given a design matrix of data points $X = [x_1, \ldots, x_n]^T$, where each $x_i \in \mathbb{R}^d$, and a target vector $y = [y_1, \ldots, y_n]$, where each $y_i \in \mathbb{R}$, the goal is to compute an s^*-sparse coefficient vector θ^* s.t.,

$$\theta^* = \underset{\theta \in \mathbb{R}^d : \|\theta\|_0 \leq s^*}{\arg\min} f(\theta) \tag{2.1}$$

Typically, f can be thought of as the empirical risk function i.e. $f(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \langle x_i, \theta \rangle)$[1] and ℓ is some standard loss function (e.g. least squared loss). $\|\theta\|_0$ is the L_0 norm function which denotes the number of non-zero entries in θ. It is easy to see that the L_0 norm is a non-convex function of θ. The set of non-zero entries of θ form the *support* denoted by $S = \{j : \theta_j \neq 0, j = \{1, 2, ..., d\}\}$, then $\|\theta\|_0 =\mid S_0 \mid$. It is important to note that optimal estimation is infeasible since the problem is NP-hard due to the non-convex constraint of $\|\theta\|_0$.

2.2 High dimensional Linear Model

We consider the simple, yet useful high-dimensional linear model. Given a response vector $y \in \mathbb{R}^n$, a design matrix $X \in \mathbb{R}^{n \times d}$ of predictor variables, a true underlying coefficient vector

[1] $\langle x_i, \theta \rangle$ denotes the dot product of the vectors x_i and θ

$\theta_0 \in \mathbb{R}^d$ and an $n \times 1$ label error/noise vector ζ, the linear model can be expressed as follows :

$$y = X\theta_0 + \zeta \tag{2.2}$$

In high-dimensional settings d and n are very large, and in some scenarios $d \gg n$. The noise elements ζ_i are usually modeled as independent draws from $\mathcal{N}(0, \sigma^2)$.

2.3 Ordinary Least Square Regression (OLS)

Under the High dimensional Linear Model (2.2), when the loss function ℓ is the *least squared loss*, then the empirical risk $f(\theta) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \langle x_i, \theta\rangle)^2 = \frac{1}{n}\|y - X\theta\|_2^2$. The optimization problem of OLS is as follows :

$$\theta_{ols} = \underset{\theta \in \mathbb{R}^d}{\arg\min}\ \frac{1}{n}\|y - X\theta\|_2^2 \tag{2.3}$$

This problem is convex in the parameter θ and has a closed form solution given by :

$$\theta_{ols} = (X^T X)^{-1} X^T y \tag{2.4}$$

However, if X is not full rank, the matrix $X^T X$ is non-invertible making the OLS estimate non-unique. Also the solution is non-sparse (if the original model itself is not sparse) since it does not enforce any sparsity constraint in the optimization problem.

2.4 Sparse Linear Regression

Under the same assumptions of OLS (2.3) and High dimensional Sparse Estimation (2.1), the optimization problem of Sparse Linear Regression is formulated as follows :

$$\theta^* = \underset{\theta \in \mathbb{R}^d : \|\theta\|_0 \le s^*}{\arg\min}\ \frac{1}{n}\|y - X\theta\|_2^2 \tag{2.5}$$

It is well-known that this problem cannot be optimally solved in the worst case due to NP-hardness results [GJY11]. However, the work of Jain et. al. [JTK14] showed that if f is differentiable and satisfies RSC and RSS properties (defined next), then global convergence is guaranteed for PGD/IHT class of algorithms.

2.5 Restricted Strong Convexity (RSC)

A differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is said to satisfy Restricted Strong Convexity (RSC) at sparsity level s with strong convexity constraint α_s if $\exists\ s_1,\ s_2$ s.t $s = s_1 + s_2$ and $\forall\ \theta_1, \theta_2$ s.t. $\|\theta_1\|_0 \leq s_1$ and $\|\theta_2\|_0 \leq s_2$, the following holds :

$$f(\theta_1) - f(\theta_2) \geq \langle \theta_1 - \theta_2, \nabla_\theta f(\theta_2) \rangle + \frac{\alpha_s}{2} \|\theta_1 - \theta_2\|_2^2 \tag{2.6}$$

2.6 Restricted Strong Smoothness (RSS)

A differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is said to satisfy Restricted Strong Smoothness (RSS) at sparsity level s with strong convexity constraint L_s if $\exists\ s_1,\ s_2$ s.t $s = s_1 + s_2$ and $\forall\ \theta_1, \theta_2$ s.t. $\|\theta_1\|_0 \leq s_1$ and $\|\theta_2\|_0 \leq s_2$, the following holds :

$$f(\theta_1) - f(\theta_2) \leq \langle \theta_1 - \theta_2, \nabla_\theta f(\theta_2) \rangle + \frac{L_s}{2} \|\theta_1 - \theta_2\|_2^2 \tag{2.7}$$

It is easy to see that, when f is the empirical risk function with the usual least squared loss denoted by $f = \frac{1}{2n} \|y - X\theta\|_2^2$, the values of α_s and L_s are the smallest and the largest eigenvalues respectively of the Hessian matrix $X^T X$.

2.7 Why Feature Selection ?

There are a couple of reasons why least squares estimate (2.3) is not satisfactory. They are as follows :

- **Prediction Accuracy :** The least squares estimate has low bias (mostly zero) but large variance, in fact the variance of estimation grows linearly with the dimension d. Prediction accuracy is reflected though the Mean Squared Error (MSE). MSE has two components - the square of the bias of estimator, and, the variance of the estimator. OLS has zero bias and high variance and therefore MSE is relatively high. MSE can be improved by shrinking or setting some coefficients to zero. By doing this, we sacrifice a little bit of bias to reduce the variance, and thereby improving the overall prediction accuracy.

- **Model Interpretation :** When the number of predictors is large enough, we would often like to determine a smaller subset of features that exhibit the "strongest effects" in determining the output. Driven by Occam's razor's principle of parsimony, the idea is to get rid of the smaller details in order to get the "big picture".

2.8 Challenges in Feature Selection

There are a couple of issues that makes the problem of Feature Selection quite challenging. These are as follows :

- **Curse of dimensionality :** When the number of features (d) is very large, the choice of selecting the "right" subset determining the output also becomes a challenging task. The estimation algorithm that chooses the "best" subset has to explore all 2^d possibilities which is huge when d is large.

- **Big data settings :** The number of observations n is also large and sometimes d overshoots n, making the problem even harder, nearly non-identifiable.

- **Multi-collinear features :** In most of the real world applications, many or all of the features are highly correlated with each other. The design matrix is not full rank and is therefore non-invertible. This makes the estimate non-unique and inconsistent making the problem harder.

- **Noisy settings :** Also most of the real world application datasets suffer from noisy observations and missing data syndrome. A lot of pre-processing and data imputation techniques are needed to clean the data, since the subset selection procedure is sensitive to these noises.

Chapter 3

Existing Methods of Feature Selection

In this chapter, we describe some of the existing and most popular approaches to Feature Selection with linear regression. This fall under the general heading *Model Selection.* All of the feature selection methods can be grouped broadly into four categories. They are :

- Classical (Discrete) methods
- Shrinkage methods
- Methods of Derived Input Directions
- Iterative Hard Thresholding methods

Feature selection retains only the "most prominent" subset of predictors and discards the rest. Least Squares Regression is then used subsequently to estimate the coefficients of the input features that are retained. We now discuss the different strategies for choosing the "right" subset.

3.1 Classical Methods

This class of subset selection techniques are discrete, combinatorial approach based - where a variable is either retained or discarded from the active set. The three main techniques under this scheme are :

- Best Subset Selection
- Forward-Stepwise and Backward-Stepwise Selection
- Forward-Stagewise Regression

Best Subset Selection : Best Subset Selection finds for each $k \in \{0, 1, \ldots, d\}$, the subset of size k which gives the smallest Residual Sum of Squares (RSS) given by

$$RSS(\theta) = \sum_{i=1}^{n} (y_i - \theta_0 - \sum_{j=1}^{d} x_{ij}\theta_j)^2 \tag{3.1}$$

Note that all the subsets need not be nested in this approach. For example, the best subset of size 2 need not include the variable from the best subset of size 1. The Leaps and Bounds procedure due to Furnival and Wilson [FW74] is an efficient algorithmic implementation of this type which is feasible usually when $d \leq 40$. The question of how to choose k, the best subset size, involves the trade-off between bias and variance. A number of model selection criteria such as AIC [Aka98], BIC [S+78], Cross-Validation, are typically used to choose the smallest k-sized model which minimizes an estimate of the expected prediction error.

Forward and Backward Stepwise Selection : Both Forward-Stepwise and Backward-Stepwise Selection are *greedy* algorithms producing a nested sequence of models indexed by the size parameter k. Instead of searching though all possible subsets, Forward-Stepwise Selection starts with the intercept term (θ_0), and then sequentially adds into the model the predictor that improves the "goodness of fit" the most. Although it might produce sub-optimal solution compared to best subset method, it is computationally tractable for large d even when $d \gg n$. Also it has lower variance in estimation compared to the best subset method. On the other hand, Backward-Stepwise Regression starts with the full model and then sequentially drops-off the predictor that has the least impact (in terms of Z-score) on the fit. Backward selection, however, can only be used when $n > d$.

Forward-Stagewise Regression : Forward Stagewise Regression starts like forward stepwise selection, with an intercept equal to $\bar{y}$ and centered predictors with coefficients initially set to all zeros. At each step, it identifies the variable most correlated with the current residual. It then computes the OLS estimate of the residual on the chosen variable and then adds it to the current coefficient for that variable. This is continued till none of the variables have correlation with the residuals. Forward-Stagewise regression can take much more than d steps to reach the least squares fit and bring down all the correlations below the desired threshold. Because of this "slow-fitting" nature it is often discarded in high-dimensional settings.

Classical Subset Selection methods, by retaining a subset of features and discarding the rest, produces a model that is *interpretable* and sometimes have *lower prediction error* than the full model. However, since it is a *discrete* process i.e., a variable is either retained or discarded, it often exhibits *high variance* which impacts the mean squared error of prediction.

3.2 Shrinkage Methods

Unlike Classical discrete methods, this class of methods are more continuous and therefore does not have much variance. These methods are called shrinkage methods since they reduce the regression coefficients iteratively by imposing a penalty on their size. They relax the non-convex L_0 constraint in Sparse Linear Regression (2.4) with appropriate convex counterparts such as the L_2 and L_1 norms, thereby making the objective function jointly convex in the parameter θ. There are a variety of algorithms under this scheme. We list down the most important ones and describe them briefly.

- Ridge Regression
- LASSO and its variants
- Elastic Net
- BoLasso
- OSCAR
- Cluster Representative Lasso (CRL) and Cluster Group Lasso (CGL)

Ridge Regression : Ridge Regression (also known as the Tikhonov/L_2 regularization) minimizes a penalized residual sum of squares i.e. alongside minimizing the empirical risk it also shrinks the regression coefficients by penalizing their L_2 norm as shown in the following optimization problem :

$$\hat{\theta}_{ridge} = \arg\min_{\theta \in \mathbb{R}^{d+1}} \frac{1}{2n} \sum_{i=1}^{n} \big(y_i - \theta_0 - \sum_{j=1}^{d} x_{ij}\theta_j\big)^2 + \lambda \sum_{j=1}^{d} \theta_j^2 \tag{3.2}$$

Note that θ_0 is the intercept term which is *not* penalized. λ is the regularization/penalization parameter which controls the amount of shrinkage in the regression coefficients and in turn the *degrees of freedom* of the fit. If λ is large, the shrinkage will be high and most of the coefficients of θ will be driven equally towards zero because of the nature of L_2-penalty. Whereas if $\lambda = 0$, this reduces to the OLS estimate (2.3). λ is usually chosen via cross-validation. The ridge solutions are not equivalent under scaling of the inputs, and so one normally standardizes the inputs before applying ridge. Henceforth, assuming X and y have been centered, the ridge equation 3.2 can be re-written in vectorized form as follows :

$$\hat{\theta}_{ridge} = \underset{\theta \in \mathbb{R}^d}{\arg\min} \frac{1}{2n} \|y - X\theta\|^2 + \lambda \|\theta\|_2^2 \tag{3.3}$$

Note that this is convex in θ and therefore has a unique minima given by :

$$\hat{\theta}_{ridge} = (X^T X + \lambda \mathbf{I})^{-1} X^T y \tag{3.4}$$

where $\mathbf{I}$ is the $d \times d$ Identity matrix. When columns of X are correlated with each other, the matrix $X^T X$ is *not* full rank, hence non-invertible. But adding a non-zero λ value to the diagonal of $X^T X$ makes the latter invertible under those circumstances. Hoerl and Kennard [HK70] showed that although the ridge estimate is biased, the variance is controlled by shrinkage of the coefficients where the shrinkage is dependent on the eigenvalue along the corresponding basis vector. However, if the original model is not-sparse, ridge cannot induce sparsity because of the nature of L_2 penalty, and therefore no automatic feature selection can be performed.

LASSO and its variants : The immediate (closest) convex relaxation of Sparse Linear Regression (2.4) is the *Least Absolute Selection and Shrinkage Operator*, popularly known as the LASSO. The lasso estimate [Tib96] is defined as follows :

$$\hat{\theta}_{lasso} = \underset{\theta \in \mathbb{R}^{d+1}}{\arg\min} \frac{1}{2n} \sum_{i=1}^{n} \Big(y_i - \theta_0 - \sum_{j=1}^{d} x_{ij}\theta_j\Big)^2 + \lambda \sum_{j=1}^{d} \mid \theta_j \mid \tag{3.5}$$

where all the terms are same as that in ridge regression (3.2) with the only difference in the penalization of the coefficient vector. Lasso shrinks the absolute value of the coefficients which makes the solution non-linear in the y_i's and also there is no closed form solution of $\hat{\theta}_{lasso}$ unlike ridge because of non-differentiability at zero. However, efficient implementations of Lasso exists (e.g. LARS-Lasso implementation [EHJ+04]) which computes the entire path of solutions as λ is varied, with the same computational cost as for ridge regression. Just as in ridge, the problem can be written in a vectorized form after standardizing the predictors and ignoring the intercept as follows :

$$\hat{\theta}_{lasso} = \underset{\theta \in \mathbb{R}^d}{\arg\min} \frac{1}{2n} \|y - X\theta\|^2 + \lambda \|\theta\|_1 \tag{3.6}$$

Note that $\|\theta\|_1$ is the L_1 norm of θ which geometrically corresponds to a rhomboid (with many corners, flat faces and edges) in high dimensions. This increases the possibility of many of the estimated parameters to be zero when intersected with the elliptical contour region of the residual sum of squares. Depending on the value of λ, the regularization/penalization factor, the lasso estimate will have many of the coefficients set to zero. The features corresponding to the non-zero coefficients, also called the *support*, form the *active set* of predictors denoted by $\hat{S}_{lasso} = \{j : \hat{\theta}_{lasso}(j) \neq 0, j = \{1, 2, ..., d\}\}$, which captures the maximum information in the linear relationship between y and X.

Lasso is a popular choice when it comes to feature selection due to its model consistency behavior such as neighborhood stability [MB06], irrepresentable condition [ZY06] under various conditions of the design matrix X. However, there exists situations when *strong correlation* (or near linear dependence) among a few or all predictors bring *instability* in Lasso estimate i.e. Lasso tends to select different variables from the group of correlated variables, even if some or all of them belong to the true active set. Related works of Meinshausen and Yu [MY09], Zhao and Yu [ZY06], Donoho [DS06], on the model consistency behavior of Lasso (as n and d grows), states that under certain assumptions on X and if the true model is sparse, Lasso identifies the correct predictors with high probability. For e.g. : If X satisfies the following condition, then Lasso estimate is always consistent[1].

$$\max_{j \in S^c} \left\| x_j^T X_S (X_S^T X_S)^{-1} \right\|_1 \leq 1 - \epsilon \quad \text{for some} \quad \epsilon \in (0, 1] \tag{3.7}$$

where S is the true support of X, S^c are the features with true coefficients set to 0. This says that the "good" variables S should not be highly correlated with the nuisance variables S^c for consistent estimation. A couple of Lasso variants viz. Relaxed Lasso, Adaptive Lasso have been proposed to address this problem.

The *Relaxed Lasso* [Mei07] uses a two-step approach for reducing the bias - In the first step it uses Lasso on the entire set of predictors X to select the set of non-zero predictors, and then in the second step it applies lasso again but this time only on the selected set of predictors from the first step. Since the variables in the second step have "less competition" from the noisy variables, the coefficients will be shrunken less and the estimation will be more or less consistent. Another variant, called the *Adaptive Lasso* [Zou06] uses a weighted L_1 penalty instead of the uniform L_1 penalty as shown in the following optimization problem :

$$\hat{\theta}_{lasso} = \underset{\theta \in \mathbb{R}^{d+1}}{\arg\min} \frac{1}{2n} \sum_{i=1}^{n} \left(y_i - \theta_0 - \sum_{j=1}^{d} x_{ij}\theta_j\right)^2 + \lambda \sum_{j=1}^{d} w_j \mid \theta_j \mid \tag{3.8}$$

[1] Statistical Consistency means as the sample size grows, the estimates converge to the true values.

where $w_j = 1/\mid \hat{\theta}_j \mid^{\nu}$ is the data-dependent weight, $\hat{\theta}_j$ is the OLS estimate and $\nu > 0$. As the sample size grows, the weights for zero-coefficient predictors get inflated to ∞, whereas the weights for non-zero coefficient predictors converge to a finite constant. The adaptive lasso yields consistent estimates of the parameters while retaining the convexity property of the lasso.

Elastic Net : The Elastic Net [ZH05] is a regularized regression method which encompasses the best of both lasso and ridge regression methods. The penalty term is a linear combination of the L_1 and L_2 penalties as shown in the following optimization problem :

$$\hat{\theta}_{elasNet} = \underset{\theta \in \mathbb{R}^{d+1}}{\arg\min} \frac{1}{2n} \sum_{i=1}^{n} \left(y_i - \theta_0 - \sum_{j=1}^{d} x_{ij}\theta_j\right)^2 + \lambda \sum_{j=1}^{d} (\alpha\theta_j^2 + (1-\alpha) \mid \theta_j \mid) \tag{3.9}$$

Note that in extreme cases for example, when $\alpha = 0$ it reduces to LASSO and when $\alpha = 1$ it reduces to ridge. Like Lasso and ridge, after centering y and standardizing X, the above optimization problem can be written in a vector form as :

$$\hat{\theta}_{elasNet} = \underset{\theta \in \mathbb{R}^{d}}{\arg\min} \frac{1}{2n} \|y - X\theta\|^2 + \lambda(\alpha \|\theta\|_2^2 + (1-\alpha) \|\theta\|_1) \tag{3.10}$$

where λ is the usual regularization parameter and $0 \leq \alpha \leq 1$ is the parameter governing the mix of the L_1 and L_2 penalties respectively. Both these parameters are usually determined by cross-validation. Elastic Net shrinks together the coefficients of correlated predictors like the ridge which is governed by the L_2-part of the mixed penalty, and then selects variables by inducing sparsity on the coefficients of these "averaged features" like the lasso which is governed by the L_1-part of the penalty.

BoLasso : If the true model is sparse, under low correlation settings of the input features, it is proven that Lasso indeed recovers the sparsity pattern in terms of regular, sign and pattern consistency. However, in presence of strong correlation among the relevant and irrelevant variables, Lasso estimation becomes inconsistent. While there are adaptive weighted versions of Lasso (as discussed before), the work of Francis Bach [Bac08] sheds light on the model consistency behavior of Lasso through the *bootstrap*. He showed that when the decay of the regularization parameter λ is proportional to $\frac{1}{\sqrt{n}}$ (where n is the number of observations), Lasso will select all the relevant variables with probability tending to one exponentially fast in the number of samples, while it selects all the other irrelevant variables too with strictly positive probability. If several datasets generated from the same distribution were available, then running lasso individually on each of them and then taking the intersection of their supports would ensure the selection of all the relevant variables always, while the irrelevant (noisy) variables, which get selected randomly, would be eliminated as a result of the intersection. However, in practice only one dataset is given, but re-sampling methods such as the *bootstrap* [ET94] can mimic the availability of several datasets. Thus this procedure of feature selection is known as the *BoLasso* (**Bo**otstrap-enhanced **L**east **A**bsolute **S**election and **S**hrinkage **O**perator). The pseudo-code of Bolasso algorithm is presented in **Algorithm** 1.

Algorithm 1 BoLasso

```
Input : Data $(X, y) \in \mathbb{R}^{n\times(d+1)}$, number of bootstrap replicates $m$
for k = 1 to m do
    Generate bootstrap samples $(X^k, y^k) \in \mathbb{R}^{n\times(d+1)}$
    Compute Lasso estimate $\hat{\theta}^k$ from $(X^k, y^k)$
    Calculate the support $S_k = \{j : \hat{\theta}^k_j \neq 0\}$
end for
Compute $S = \bigcap_{k=1}^{m} S_k$
Finally, compute $\hat{\theta}_S$ from $(X_S, y)$
```

OSCAR : In situations where groups of predictors are highly correlated with each other supervised clustering of features that form meaningful groups can be beneficial both from prediction accuracy and model interpretation point of view. The OSCAR (Octagonal Shrinkage and Clustering Algorithm for Regression) method due to Bondell et. al. [BR08] performs efficient variable selection with multi-collinear predictors. It does so via a novel penalization method which simultaneously eliminates noisy, irrelevant variables and performs supervised clustering on the active variables. Assuming the response (y) have been centered and the predictors (X) have been standardized, the optimization problem of OSCAR is expressed as follows :

$$\begin{gathered}\hat{\theta}_{oscar} = \arg\min_{\theta\in\mathbb{R}^d} \sum_{i=1}^{n} \big(y_i - \sum_{j=1}^{d} x_{ij}\theta_j\big)^2 \\ \text{subject to} \\ \sum_{j=1}^{d} \mid \theta_j \mid + c \sum_{j<k} \max(\mid \theta_j \mid, \mid \theta_k \mid) \leq t\end{gathered} \tag{3.11}$$

where $c \geq 0$ is a tuning parameter controlling the relative weighting of the L_1 norm and the pairwise L_∞ norm. Similarly $t > 0$ is also a tuning parameter controlling the magnitude of the two norms. Note that the L_1 norm encourages sparsity, whereas the pairwise L_∞ norm encourages equality of coefficients. Also the pairwise L_∞ norm allows construction of multiple groups of varying sizes, instead of a single clustered group which would have been obtained using an L_∞ norm over the entire coefficient vector. Note that setting $c = 0$ in equation 3.11 yields the LASSO problem which gives only sparsity and no clustering, whereas when $c \to \infty$ the constraint region becomes a square with only clustering and no variable selection. However under all settings of c, the constraint region remains convex. Geometrically, OSCAR constructs a new type of penalty (constraint) region that is octagonal in shape which guarantees both sparsity and equality for coefficients of correlated predictors having similar relationship with the output. The exact equality of coefficients obtained via this penalty is the key idea behind creation of the grouped predictors as in the supervised clustering technique they use. OSCAR **do not** explicitly consider the correlation structure among the predictors in their optimization problem. However, Bondell et. al. proved that the OSCAR solution implicitly constructs this grouping of highly correlated predictors. For a given set of choices of the tuning parameters, OSCAR can be cast as a quadratic programming (QP) problem with $\mathcal{O}(d^2)$ parameters and $\mathcal{O}(d^2)$ many linear constraints which can be solved using a quadratic programming algorithm SQOPT [GMS08].

Cluster Representative Lasso and Cluster Group Lasso : These methods are applied in a high dimensional linear model with strongly correlated set of predictors. The objective like before is estimating the *true active support*, denoted by S_0. Under this situation, as we already know, Lasso solution is statistically inconsistent i.e Lasso tends to select only one variable from the group of correlated (or nearly linearly dependent) variables even if many or all of them belong to the active set S_0. A couple of other convex-relaxation based methods such as the elastic-net, the adaptive lasso, OSCAR have been proposed to address this problem, but they do not *explicitly* take the correlation structure among the variables into account and thus still exhibit difficulties when groups of variables are almost linearly dependent.

Bühlmann et. al. [BRvdGZ13] proposed a new technique which takes care of the correlation structure before doing sparse estimation. They primarily propose to cluster the variables first and then do subsequent sparse estimation using Lasso on the matrix of cluster representatives or using Group Lasso based on the structure from the clusters. Regarding the clustering step, they proposed a novel bottom-up agglomerative clustering algorithm based on canonical correlation among the variables, since this reflects the notion of linear dependence the most. They proved that it finds the "finest" clustering which preserves the criterion that intra-group correlation should be high and inter-group correlation should be lower than a threshold. Also it is statistically consistent. In the next step, based on the group compatibility constant of X, they either use Group Lasso [YL06] on the set of clusters, or ordinary Lasso [Tib96] on the matrix formed by taking a representative element from each cluster.

Let the design matrix X be grouped into q clusters after the first step, denoted by $G_1, G_2, \ldots, G_q$. Now, the two methods proposed for down-selecting the clusters are as follows :

- Cluster Representative Lasso (CRL)
- Cluster Group Lasso (CGL)

Cluster Representative Lasso (CRL) : For each cluster, a representative is constructed which is usually the centroid of the cluster denoted by :

$$\bar{X}^{(r)} = \frac{1}{\mid G_r \mid} \sum_{j \in G_r} X^{(j)}, \quad r = 1, 2, \ldots, q$$

where $X^{(j)}$ denotes the j^{th} column of X, $\bar{X}$ is an $n \times q$ matrix (called the cluster representative matrix) whose r^{th} column is given by $\bar{X}^{(r)}$. The Lasso optimization problem on this cluster representative matrix is as follows :

$$\hat{\theta}_{CRL} = \underset{\theta \in \mathbb{R}^d}{\arg\min} \frac{1}{2n} \left\|y - \bar{X}\theta\right\|^2 + \lambda_{CRL} \left\|\theta\right\|_1 \tag{3.12}$$

The selected clusters are given by $\hat{S}_{cls,CRL} = \{r : \hat{\theta}_{CRL}(r) \neq 0, r = 1, 2, \ldots, q\}$ and then the selected variables are obtained by taking the union of all the variables from the selected clusters as $\hat{S}_{CRL} = \cup_{r \in \hat{S}_{cls,CRL}} G_r$.

Cluster Group Lasso (CGL) : Another obvious way of selecting the clusters is via group lasso. The clusters themselves form a natural grouping of the coefficient vector, denoted by $\theta = (\theta_{G_1}, \ldots, \theta_{G_q})^T$, where $\theta_{G_r} = (\{\theta_j : j \in G_r\})^T$. The optimization problem for cluster group lasso is given by :

$$\hat{\theta}_{CGL} = \underset{\theta \in \mathbb{R}^d}{\arg\min} \frac{1}{2n} \left\|y - \bar{X}\theta\right\|^2 + \frac{\lambda_{CGL}}{\sqrt{n}} \sum_{r=1}^{q} w_r \left\|X^{(G_r)}\theta_{G_r}\right\|_2 \tag{3.13}$$

where $w_r = \sqrt{\mid G_r \mid}$ (usually) is the group multiplier accounting for the variable group-size. The group penalty is not the usual penalty $(\lambda \sum_{r=1}^{q} w_r \left\|\theta_{G_r}\right\|_2)$, but a different one, termed as the "group-wise prediction penalty" which is much more appropriate when $X^{(G_r)}$ exhibits strong correlation. It is well known that the group lasso enjoys a group selection property where either the entire group is selected as a whole ($\hat{\theta}_{CGL,G_r} \neq 0$) or none of them (all zero vector i.e. $\hat{\theta}_{CGL,G_r} = 0$). The selected clusters are given by $\hat{S}_{cls,CGL} = \{r : \hat{\theta}_{CGL,G_r} \neq 0, r = 1, 2, \ldots, q\}$

and then the selected variables are obtained by taking the union of all the variables from the selected clusters as $\hat{S}_{CGL} = \cup_{r \in \hat{S}_{cls,CGL}} G_r = \{j : \hat{\theta}_{CGL,j} \neq 0, j = 1, 2, \ldots, d\}$.

These methods are very efficient in support recovery, their recall is very high. Also they are extremely fast and scalable under the high-dimensional setting. However, to avoid false negatives (i.e. to avoid not selecting an active variable from S_0), they often over-estimate the predicted support $\hat{S}$ which impacts the precision and hence the F_1-score as well.

3.3 Derived Input Directional Methods

This class of methods can also be used in those scenarios where a large number of input features are highly correlated with each other. These are called methods of Derived Input Directions because they consider a small number of linear combinations Z_m, $m = 1, 2, \ldots, M$ of the original inputs X_j's, $j = \{1, 2, \ldots, d\}$, and then use these derived Z_m's in place of original X_j's as input features in regression. We discuss two methods under this scheme -

- Principal Components Regression (PCR)
- Partial Least Squares (PLS)

They differ in how the linear combinations are constructed.

Principal Components Regression (PCR) : Given a set of n tuples $S_1, S_2, \ldots, S_n$ where $S_i = (X_i, y_i) \in \mathbb{R}^{d+1}$, PCR constructs the derived input columns $z_m = Xv_m$, and then regresses y on $z_1, z_2, \ldots, z_M$ for some $M \leq d$. Since the z_m's are orthogonal to each other, the output can be expressed as follows :

$$\hat{y}_M^{pcr} = \bar{y} + \sum_{m=1}^{M} \hat{\theta}_m z_m \tag{3.14}$$

where $\hat{\theta}_m = \langle z_m, y \rangle / \langle z_m, z_m \rangle$. Since PCR depends on the scaling of the inputs, so typically the input is standardized before using PCR. It is easy to see when $M = d$, we would get back the usual least squares estimate. Principal Components Regression is very similar to Ridge Regression (which shrinks the coefficients of the principal components depending on the corresponding eigenvalue), except that it discards the $d - M$ smallest eigenvalue components. The choice of M is crucial here since it governs the prediction error and model complexity. Usually Cross-Validation technique is employed to determine a “good” estimate of M.

Partial Least Squares (PLS) : Like Principal Components Regression, PLS also constructs a set of linear combinations of the input features for regression, but unlike PCR, it uses y along with X to do so. We begin here also by standardizing each x_j, $j = \{1, 2, \ldots, d\}$ to have mean zero and unit variance. Partial Least Squares procedure begins by computing the similarity of each feature x_j with the output y, denoted by $\phi_{1j} = \langle x_j, y \rangle$ for each j. The first PLS direction is obtained subsequently as follows :

$$z_1 = \sum_{j=1}^{d} \phi_{1j} x_j \tag{3.15}$$

Thus in the construction of each z_m, the inputs are weighted by the strength of their univariate effect on y. The output y is regressed on z_1 to get the coefficient $\hat{\theta}_1$, and then $x_1, x_2, \ldots x_d$ are orthogonalized w.r.t z_1. The above procedure is repeated until $M \leq d$ prominent directions have been obtained. M is again a hyper-parameter which is chosen usually by means of cross-validation. Note that the Partial Least Squares seeks directions that have *high variance* and *strong correlation* with the *response*, whereas, the Principal Components Regression captures those directions which have *high variance* only.

3.4 Iterative Hard Thresholding (IHT) Methods

This class of methods are also known as Projected Gradient Descent (PGD) methods. They employ empirical risk minimization along with hard L_0 constraints. Efficient estimation, however, is faced with feasibility issues due to NP-hardness results [GJY11]. Interestingly, some recent works [NYWR09, LYF14, JJR11, SSSZ10] have shown that these hardness results can be avoided by assuming some conditions on the loss function f being minimized such as Restricted Strong Convexity (RSC) and Restricted Strong Smoothness (RSS) (2.5, 2.6). But these methods are either convex-relaxation based methods, which typically suffer from slow convergence rates as they solve non-smooth optimization problems, or greedy methods, which are slow in non-negligible sparsity setting due to their incremental approach of adding/removing elements individually from the support.

Instead, PGD/IHT methods are the methods of choice for most practical situations since they offer the fastest and most scalable solutions. These methods directly project the gradient descent update onto the underlying non-convex feasible set. This projection is not the usual prox-operator for any convex prox function, but hard thresholding, which can be performed very efficiently for several interesting structures viz. sparsity, low-rank etc. Several algorithms based on PGD/IHT-style methods have been proposed for sparse recovery such as Iterative Hard Thresholding (IHT) [BD09], Hard Thresholding Pursuit (HTP) [Fou11], GraDeS [GK09], CoSaMP [NT09], Subspace Pursuit (SP) [DM09]. But the analysis of these methods have been restricted to Restricted Isometry Property (RIP) or Incoherence Property settings which usually do not fit in to high-dimensional statistical estimation problems. However, recent work due to Jain et. al. [JTK14] have been able to analyze this PGD/IHT-style algorithms for high-dimensional statistical estimation problems. Their convergence results hold for any arbitrarily differentiable, possibly non-convex functions as long as they satisfy RSC/RSS properties. The general framework of a PGD/IHT algorithm is shown in **Algorithm** 2. The projection operator

Algorithm 2 Iterative Hard-Thresholding method (**PGD**)

```
Input : Function $f$ with gradient oracle, sparsity level $s$, step-size $\eta$
$\theta^0 = 0$, $t = 0$
while not converged do
    $\theta^{t+1} = \theta^t - \eta\nabla_\theta f(\theta^t)$                ▷ Gradient Descent Step
    $\theta^{t+1} = P_s(\theta^{t+1})$                ▷ Projection/Hard Thresholding Step
    $t = t + 1$
end while
Output : $\theta^t$
```

$P_s(\theta)$ can be implemented efficiently by projecting θ onto the set of s-sparse vectors by selecting the s largest elements of θ in magnitude. The analysis due to [JTK14] considers the RSC/RSS properties of f to derive geometric convergence rates for the IHT procedure (**Algorithm** 2) as shown in theorem 3.1.

Theorem 3.1 *Let f satisfies RSC and RSS properties with parameters L and α. Let **Algorithm** 2 be invoked with f, $s \geq 32(\frac{L}{\alpha})^2 s^*$ and $\eta = \frac{2}{3L}$. Then, the τ-th iterate of the algorithm, for $\tau = \mathcal{O}(\frac{L}{\alpha}\log\big(\frac{f(\theta^0)}{\epsilon}\big))$ satisfies*

$$f(\theta^\tau) - f(\theta^*) \leq \epsilon$$

where $\theta^* = \underset{\theta:\|\theta\|_0 \leq s^*}{\arg\min} f(\theta)$

This shows that PGD/IHT methods which obeys RSC/RSS constraints achieve global convergence as long as they employ projection onto sets of sparsity $s \gg s^*$. These results also hold under badly-conditioned sparse recovery problems. Last but not the least, these methods are orders of magnitude faster than the L_1 and greedy methods stated before.

Our algorithm, **BoPGD**, described in the next chapter, is largely based on this PGD/IHT scheme.

Chapter 4

Our Method

We propose a new feature selection algorithm which is consistent, scalable and efficient especially under the high-dimensional multi-collinear setting. Since it is principally based on Projected Gradient Descent (Iterative Hard Thresholding) based methods powered by Bootstrap enhanced supervised selection of correlated predictors, we call it the **BoPGD**, **Bo**otstrapped (**Bo**otstrap-enhanced) **P**rojected **G**radient **D**escent.

4.1 Bootstrapped Projected Gradient Descent (BoPGD)

Given samples $S_i = (X_i, y_i) \in \mathbb{R}^{d+1}$, under the linear model $y_i = \langle \theta, X_i \rangle + \zeta_i$, where $\zeta_i \sim \mathcal{N}(0, \sigma^2)$ is the i.i.d Gaussian label noise and the empirical risk function is the average least squared loss i.e. $f(\theta; S_{1:n}) = \frac{1}{2n} \|y - X\theta\|_2^2$. Suppose the samples $X_{1:n}$ are drawn i.i.d from a Sub-Gaussian distribution with the covariance matrix Σ with $\Sigma_{jj} \leq 1 \ \forall j \in \{1, \ldots, d\}$, then using results proved by Jain et. al. [JTK14], f satisfies RSC and RSS properties w.h.p. With these assumptions in hand, the objective of BoPGD is the following :

- Take a high-dimensional linear model with possibly multi-collinear features, and produce a smaller-dimensional linear model retaining only the “most significant” subset
- The model must have a “good” fit to the given data
- The model must be far from being complex and should be easily interpretable
- The sparse estimation routine must be stable and statistically consistent especially under the multi-collinear setting
- Lastly, the algorithm must be efficient, fast and scalable under high dimensions

There are four key steps in BoPGD. They are listed below in the order in which they are executed :

- Clustering covariates based on empirical correlation
- Supervised selection of clusters using PGD
- Eliminate irrelevant clusters using Bootstrapping
- Sparse estimation on the reduced set of features using PGD

1. Clustering covariates based on empirical correlation : Consider an index set $\{1, 2, \ldots, d\}$ for the covariates in X under the linear model 2.2. Let $X^{(G)} = \{X^{(j)} : j \in G\}$ denote the group of variables (features) forming the cluster $G \subseteq \{1, 2, \ldots, d\}$. The goal of this clustering step is to find a partition of features $\{1, 2, \ldots, d\}$ into disjoint clusters $\mathcal{G} = \{G_1, G_2, \ldots, G_q\}$ with their union $\bigcup_{r=1}^{q} G_r = \{1, 2, \ldots, d\}$ and intersection $G_r \cap G_l = \phi$ $(\forall r \neq l)$. The partition $\mathcal{G}$ should satisfy the following criteria :

- The intra-group correlation should be high
- The correlation between the groups should be low

The objective of this clustering step is to eliminate the root cause of inconsistency in feature selection - the linear dependence among predictors which causes identifiability problems in sparse estimation (e.g. Lasso). We use the standard agglomerative hierarchical clustering based on sample correlation to cluster the predictors. The pairwise distance between the predictors is computed and stored in a Dissimilarity matrix D with entries $D(i,j) = 1- \mid \hat{\rho}(X^{(i)}, X^{(j)}) \mid$, where $\hat{\rho}(X^{(i)}, X^{(j)})$ denotes the sample Pearson correlation between features $X^{(i)}$ and $X^{(j)}$. We choose complete linkage as the dissimilarity measure between clusters. The cut-off for determining the number of clusters is governed by a threshold value which reflects the notion of similarity between the variables. For example, a cut-off value of 0.3 denote a horizontal cut in the dendrogram such that all the clusters below this line have absolute correlation strength of 0.7 and more. As shown in figure 4.1 this cutoff value resulted in the formation of 10 clusters viz. $cls = \big\{1, 2, 3, 4, \{5, 7, 8\}, \{9, 10\}, 6, 11, 12, 13\big\}$ where intra-cluster correlation ≥ 0.7 and inter-cluster correlation is less. Alternatively, one can proceed in an agglomerative way, recording the new value of the linkage function in every iteration and then use the partition $\hat{b}$ corresponding to the iteration for which the gap in function values is maximum i.e. $\hat{b} = \arg\max_{b}(h_{b+1} - h_b)$, and use that to form the clusters.

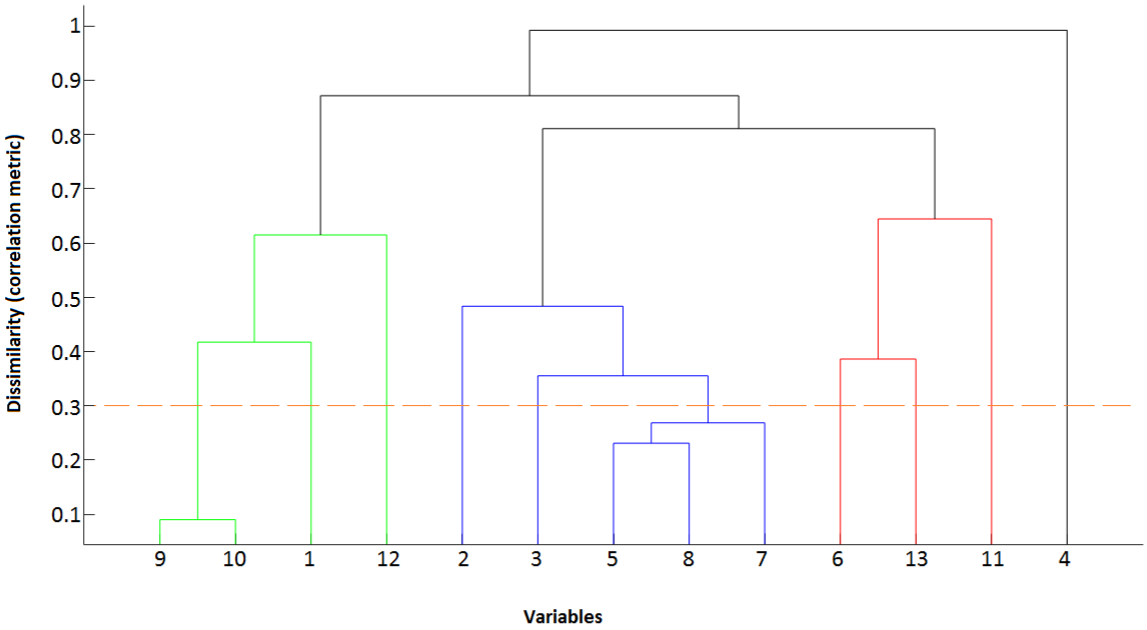


Figure 4.1: Dendrogram based on absolute correlation among variables from a real data

Consistency Result : Consider the $n \times d$ design matrix $X_{1:n} \sim \mathcal{N}(0, \Sigma)$ generated as per our model assumption (stated before) and $\Sigma_{jj} = 1 \; \forall j$, then due to Bühlmann and Geer [BVDG11], it is well-known that

$$\max_{j,k} \mid \hat{\Sigma}_{j,k} - \Sigma_{j,k} \mid = \mathcal{O}\Big(\sqrt{\frac{\log d}{n}}\Big) \tag{4.1}$$

where $\hat{\Sigma}$ is the sample (empirical) covariance matrix. Also the following condition ensures that the true clusters are in some sense "tight" and "separated" from each other.

$$\begin{aligned} &\min\{\mid \hat{\Sigma}_{j,k} \mid : j, k \in G_r \; (j \neq k), \; r = 1, 2, \ldots, q\} \\ > &\max\{\mid \hat{\Sigma}_{j,k} \mid : j \in G_r, k \in G_\ell, \; r, \ell = 1, 2, \ldots, q \; (r \neq \ell)\} \end{aligned} \tag{4.2}$$

Under the assumptions 4.1 and 4.2, the hierarchical clustering algorithm using complete linkage dissimilarity metric, will consistently produce the true clusters if $\frac{\log d}{n} \to 0$ as d increases. In simple words, the tighter the correlation within clusters and weaker the correlation across clusters, the better we can estimate the true underlying grouping of the variables.

2. Supervised selection of clusters using PGD : Let the design matrix X be grouped into q clusters after the first step, denoted by $G_1, G_2, \ldots, G_q$. We construct the cluster representative matrix by taking a representative feature from each cluster. Like CRL (3.12), the representative for each cluster is the centroid denoted by :

$$\bar{X}^{(r)} = \frac{1}{\mid G_r \mid} \sum_{j \in G_r} X^{(j)}, \quad r = 1, 2, \ldots, q$$

where $X^{(j)}$ denotes the j^{th} column of X. $\bar{X}$, an $n \times q$ matrix is the cluster representative matrix whose r^{th} column is given by $\bar{X}^{(r)}$, the representative feature for the r^{th} cluster. Unlike CRL, which uses Lasso for selection of clusters in the next step, we use Iterative Hard Thresholding (IHT/PGD) algorithm on the cluster representative matrix. The PGD algorithm (2) is controlled by two parameters viz. step-size α and sparsity value s. α is fixed to a very small value (usually 0.001) and the sparsity (s) is varied all the way from 1 to q, the number of clusters formed in the previous step. For each value of s, we compute the supporting cluster representatives which correspond to their respective clusters. Then we take the union of the features present in those supporting clusters, fit an OLS estimate on those selected predictors and record the average test error on the cross-validation data. Finally, we estimate the best sparsity value to be either the value at which the average error is minimum or the *smallest* s (if it exists) such that the error lies within a multiplicative ϵ-bound of the minimum. This procedure is summarized in Algorithm 3.

Algorithm 3 Supervised selection of clusters using IHT

```
Input : Cluster representative matrix $\bar{X}$, response vector $y$, Clusters $G_1, G_2, \ldots G_q$
for $s = 1$ to $q$ do                                ▷ Different choices of sparsity values
    $\theta_s = \mathbf{PGD}(\bar{X}, y, s, \alpha)$              ▷ PGD is the Algorithm 2 described before
    $S_{cls,s} = \{j : \theta_s(j) \neq 0\}$
    $S_s = \cup_{r \in S_{cls,s}} G_r$
    $cv\_err(s) = \mathbf{OLS}(X_{S_s}, y)$         ▷ OLS is ordinary least squares regression procedure
end for
$min\_err = \min_s cv\_err(s)$ and $s^* = \arg\min_s cv\_err(s)$
Find the smallest $\hat{s}$ : $cv\_err(\hat{s}) \leq (1 + \epsilon) * min\_err$
Output : $\hat{s}$
```

3. Eliminate irrelevant clusters using Bootstrapping : The previous step guarantees selection of the "true" clusters with high probability. Additionally it also selects a few noisy clusters with some positive probability. Now if several datasets generated from the same distribution were available, then running PGD on the cluster representative matrix constructed individually on each of them and then taking the intersection of their supports would ensure the selection of all the relevant clusters, while the irrelevant (noisy) ones, which get selected randomly, would be eliminated as a result of the intersection. However, in practice only one dataset is given, hence, like BoLasso, we use re-sampling methods such as the *bootstrap* [ET94] to mimic the availability of several datasets. The idea is to create m replicas of the n data points (X, y) i.e. $(X^{(i)}, y^{(i)})$ for $i = 1, 2, \dots, m$ where each replica has n points. These n pairs $(X_j^{(i)}, y_j^{(i)})$, $j = 1, 2, \dots, n$ are sampled uniformly at random (u.a.r) with replacement from the n original pairs in (X, y). The sampling of these mn pairs of observations is independent i.e. we sample n pairs u.a.r with replacement from (X, y), and then given (X, y), the m replicas are sampled i.i.d from the distribution of (X, y). For each of these m bootstrapped replicas, we run Projected Gradient Descent on the cluster representative matrix[1] (constructed on them individually), and then select the top $\hat{s}$ clusters from each of them. Notice here we do not vary the sparsity selection parameter, instead use the one which we have already estimated in the previous step. Then, we use a voting scheme to take a hard (or soft) intersection of the supports from each of the m replicas to finally select the "active" clusters. The union of features present in these selected clusters $(\hat{S}_{0,b})$ form an estimated upper bound on the "true support" S_0. Algorithm 4 summarizes this step.

Algorithm 4 Bootstrap enhanced selection of clusters

```
1: **Input :** Data $(X, y) \in \mathbb{R}^{n \times (d+1)}$, number of bootstrap replicates $m$, Clusters $G_1, G_2, \dots G_q$,
   sparsity parameter $\hat{s}$
2: **for** k = 1 to m **do**
3:     Generate bootstrap samples $(X^k, y^k) \in \mathbb{R}^{n \times (d+1)}$
4:     Generate the cluster representative matrix $\bar{X}^{(k)}$
5:     $\hat{\theta}^k = \mathbf{PGD}(\bar{X}^{(k)}, y^{(k)}, \hat{s})$
6:     $\hat{S}_k = \{j : \hat{\theta}^k_j \neq 0\}$
7: **end for**
8: Compute $\hat{S} = \bigcap_{k=1}^{m} \hat{S}_k$
9: **Output :** $\hat{S}_{0,b} = \bigcup_{r \in \hat{S}} G_r$
```

4. Sparse estimation on the reduced set of features using PGD : The features present in $\hat{S}_{0,b}$ contains the "active" set of features and therefore recall is almost always 1 here. But since it is designed as selecting clusters of correlated features instead of individual features, it often overestimates the predicted support by including the correlated variables as well which do not belong to the "true" support. In this step, we perform another sparse estimation using Projected Gradient Descent (Iterative Hard Thresholding) on this reduced-dimensional feature space $(X_{\hat{S}_{0,b}})$ to eliminate these noisy but correlated features from the support. Since the

[1] We re-use the prior knowledge of clusters from step 1

"true" variables in $X_{\hat{S}_{0,b}}$ have "lesser competition" from the noisy variables, the coefficients will be shrunken less and the estimation will be more or less consistent. The step-size α is fixed to a very small value (usually 0.0001) and the sparsity parameter (s) is varied all the way from 1 to $\mid \hat{S}_{0,b} \mid$, the cardinality of the reduced-dimensional feature space. For each value of s, we estimate the support, take the union of the features present in the support, fit an OLS estimate on them and record the average test error on the cross-validation data. We select the best sparsity value $s^{\dagger}$ to be either the value at which the average error is minimum or the *smallest* s (if it exists) such that the error lies within a multiplicative ϵ-bound of the minimum. Finally, our estimated support $\hat{S}_0$ is simply the set of non-zero features at the selected sparsity value $s^{\dagger}$. This procedure is summarized in Algorithm 5. The entire **BoPGD** algorithm (combining these four steps) is now presented as a whole in **Algorithm 6**.

Algorithm 5 Sparse Estimation using PGD

```
Input : Data $(X, y) \in \mathbb{R}^{n \times (d+1)}$, $\hat{S}_{0,b}$ - an upper bound on support
for $s = 1$ to $\mid \hat{S}_{0,b} \mid$ do                    ▷ Different choices of sparsity values
    $\hat{\theta}_s = \textbf{PGD}(X, y, s, \alpha)$
    $\hat{S}_s = \{j : \hat{\theta}_s(j) \neq 0\}$
    $cv\_err(s) = \textbf{OLS}(X_{\hat{S}_s}, y)$
end for
$min\_err = \min_s cv\_err(s)$
Find the smallest $s^{\dagger}$ : $cv\_err(s^{\dagger}) \leq (1 + \epsilon) * min\_err$
Compute the predicted Support $\hat{S}_0 = \{j : \hat{\theta}_{s^{\dagger}}(j) \neq 0\}$
Output : $\hat{S}_0$
```

4.2 Complexity Analysis

Let $X \in \mathbb{R}^{n \times d}$ be the design matrix, n being the number of samples and d being the dimension of each data point, the time complexity of Bootstrapped Projected Gradient Descent can be analyzed as follows :

- The hierarchical clustering step deals with all d^2 pairwise correlation based distances and keeps on merging all the variables in increasing order of this value. This can be done efficiently in $\mathcal{O}(d^2 \log d)$ time.

- The running time of Projected Gradient Descent is upper bounded by $\mathcal{O}(d^2 n + d \log d)$ and since this is repeated for m bootstrap replicates, the time complexity of the PGD step along with bootstrapping is precisely $\mathcal{O}(m(d^2 n + d \log d))$.

Therefore, the overall time complexity of BoPGD is $\mathcal{O}(m(d^2 n + d \log d) + d^2 \log d)$.

Algorithm 6 Bootstrapped Projected Gradient Descent (BoPGD)

```
**Input :** Data $(X, y) \in \mathbb{R}^{n\times(d+1)}$, number of bootstrap replicates $m$
Cluster features from $X$ using agglomerative hierarchical clustering
Compute $q$ clusters $G_1, G_2, \dots G_q$
Generate the Cluster Representative Matrix, $\bar{X}$
**for** $s = 1$ to $q$ **do**                    ▷ Different choices of sparsity values in $2^{nd}$ step
    $\theta_s = \mathbf{PGD}(\bar{X}, y, s, \alpha)$
    $S_{cls,s} = \{j : \theta_s(j) \neq 0\}$
    $S_s = \cup_{r \in S_{cls,s}} G_r$
    $cv\_err(s) = \mathbf{OLS}(X_{S_s}, y)$
**end for**
$min\_err = \min_s cv\_err(s)$
Find the *smallest* $\hat{s}$ : $cv\_err(\hat{s}) \leq (1+\epsilon) * min\_err$
**for** k = 1 to m **do**                    ▷ $m$ bootstrap replications of $(X, y)$
    Generate bootstrap samples $(X^k, y^k) \in \mathbb{R}^{n\times(d+1)}$
    Generate the $k^{th}$ Cluster Representative Matrix $\bar{X}^{(k)}$
    $\hat{\theta}^k = \mathbf{PGD}(\bar{X}^{(k)}, y^{(k)}, \hat{s})$
    $\hat{S}_k = \{j : \hat{\theta}^k_j \neq 0\}$
**end for**
Compute $\hat{S} = \bigcap_{k=1}^{m} \hat{S}_k$
Compute $\hat{S}_{0,b} = \bigcup_{r \in \hat{S}} G_r$
**for** $s = 1$ to $\mid \hat{S}_{0,b} \mid$ **do**                    ▷ Different choices of sparsity values in $4^{th}$ step
    $\hat{\theta}_s = \mathbf{PGD}(X, y, s, \alpha)$
    $\hat{S}_s = \{j : \hat{\theta}_s(j) \neq 0\}$
    $cv\_err(s) = \mathbf{OLS}(X_{\hat{S}_s}, y)$
**end for**
$min\_err = \min_s cv\_err(s)$
Find the *smallest* $s^\dagger$ : $cv\_err(s^\dagger) \leq (1+\epsilon) * min\_err$
Compute the predicted Support $\hat{S}_0 = \{j : \hat{\theta}_{s^\dagger}(j) \neq 0\}$
Calculate the error on test data : $test\_err = \mathbf{OLS}(X^{test}_{\hat{S}_0}, y^{test})$
**Output :** $\hat{S}_0$ and $test\_err$
```

Chapter 5

Experiments and Results

In this chapter, we present the results of our experimental evaluation on a host of synthetic and real-world datasets. First we present the findings for synthetic experiments where the ground truth i.e. the "true" support is known. We compare our method, BoPGD against some of the existing and most competitive ones in terms of support recovery, mean squared prediction error and other evaluation criteria as described in section 5.1. We supplement our synthetic experiments with more experiments on some real-world public datasets from the UCI[1] repository.

5.1 Evaluation Criteria

Let $S_0 = \{j : \theta_0(j) \neq 0 : j \in [d]\}$ denote the "true" support and $\hat{S} = \{j : \hat{\theta}(j) \neq 0 : j \in [d]\}$ denote the support "estimated" (*recovered*) by any feature selection algorithm. We use the standard measures of Precision (P), Recall (R) and F_1 score to evaluate the performance of an algorithm in the context of variable screening and Mean Squared Error (MSE) on test data as a predictive measure of "model fitting". The metrics are defined as follows :

$$P = \frac{\mid \hat{S} \cap S_0 \mid}{\mid \hat{S} \mid} \quad R = \frac{\mid \hat{S} \cap S_0 \mid}{\mid S_0 \mid} \quad F_1 = \frac{2PR}{P+R} \tag{5.1}$$

The models are fitted on the training data and we compare the test set Mean Squared Error (MSE) denoted by $MSE = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}[(Y_{test}^i - \hat{Y}_{test}^i)^2]$, where Y_{test}^i and $\hat{Y}_{test}^i$ denote the "true" value and the "estimated" value of the response variable respectively on the test data set.

[1] https://archive.ics.uci.edu/ml/index.html

5.2 Synthetic Data

In this section, we present the consistency and efficiency of our method in comparison with other benchmarked feature selection methods viz. Lasso, Elastic-Net, BoLasso, CRL and Ordinary Projected Gradient Descent (PGD). We simulated data from a high-dimensional linear model (2.2) with a fixed design matrix X, and the noise variable $\zeta \sim \mathcal{N}(0, \sigma^2 I)$ with $\sigma = 0.1$ fixed. The number of samples (n) is set to 1000 and the dimension (d) of each data point is varied within the interval $[10, 1000]$ across all our experiments. We generate the design matrix X from a multivariate Gaussian distribution $\mathcal{N}_d(0, \Sigma)$ where the covariance matrix Σ has a "block-diagonal" form as described in [BRvdGZ13]. Σ is partitioned into Q blocks such that for each block the features within the block are highly correlated $(\rho \geq 0.85)$ and the correlation across the blocks is very low (≤ 0.15). A sample correlation heat map of Σ is shown in figure 5.1. Note that there are $Q = 5$ blocks (each marked *red*) and the correlation coefficient ρ within each block is ≥ 0.8 and the inter-block correlation is ≤ 0.2.

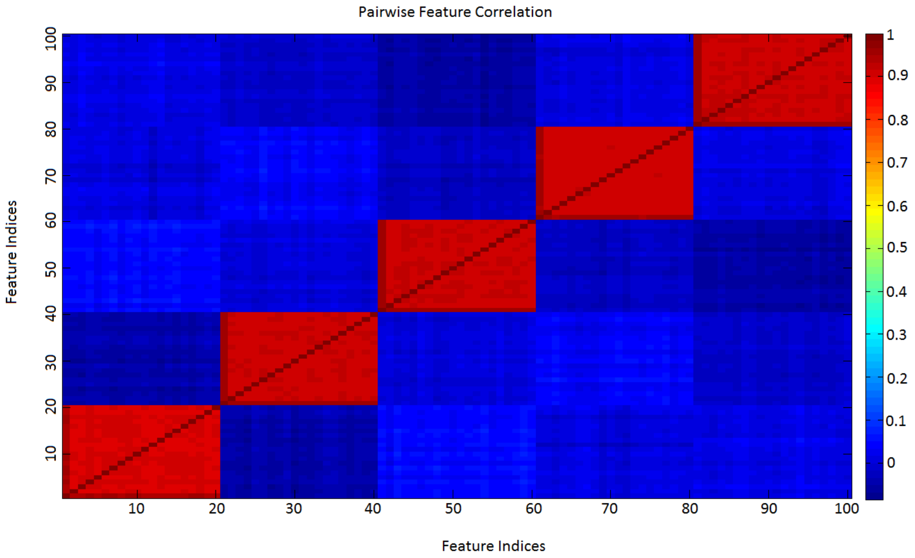


Figure 5.1: Correlation Heat map of features simulated from $\mathcal{N}_d(0, \Sigma)$: $d = 100$, $Q = 5$

The "true" support (S_0) is randomly chosen from $[1, d]$ across all the experiments and the "true" coefficient (weight) vector θ_0 corresponding to S_0 is also chosen uniformly at random between $[-0.5, 0.5]$ throughout all our experiments. For each of the methods we choose a suitable grid of values for the tuning parameters and we select the "best" among them using 5-fold Cross Validation. The reported results are averages taken over 25 independent simulations. The source code is available at `https://github.com/MessianNil/BoPGD`.

Synthetic Experiment 1 [Single block support] : Consider the above block diagonal model for X with $n = 1000$, $d = 100$, $Q = 5$ which implies each block has 20 features. The "true" support (S_0) size is fixed to 10 i.e $\mid S_0 \mid = 10$ and is chosen randomly from the first block Q_1. Thus in this setup, all the active variables are concentrated in a single block. They are chosen randomly (in each simulation run) to be one half of the block size, but they are strongly correlated with the other half of noisy variables too. We compare our method with some of the existing ones such as Lasso, Elastic-Net, BoLasso, Cluster Representative Lasso (CRL) and Iterative Hard Thresholding/Projected Gradient Descent (PGD). We ignore OSCAR from the comparison here since it is extremely slow for high values of d and n. The results are summarized in table 5.1. From table 5.1, it is clear that BoPGD is superior in comparison to other methods if we look at the F_1 score which balances the true positive rate and the false positive rate of support prediction. The mean squared error is more or less same for all the methods. The CRL has recall = 1 since it can easily cluster all the features into 5 groups using average linkage measure of distance as is shown in the dendrogram plot (figure 5.2).

Methods	**P**	**R**	**F$_1$**	**MSE**
Lasso	0.505	0.992	0.668	0.007
Elastic Net	0.464	0.996	0.631	0.006
BoLasso	0.756	0.996	0.859	0.005
CRL	0.500	1.000	0.667	0.005
PGD	0.913	0.988	0.947	0.006
BoPGD	1.000	0.996	0.998	0.006

Table 5.1: Results on Synthetic Experiment 1 (averaged across 25 simulations)

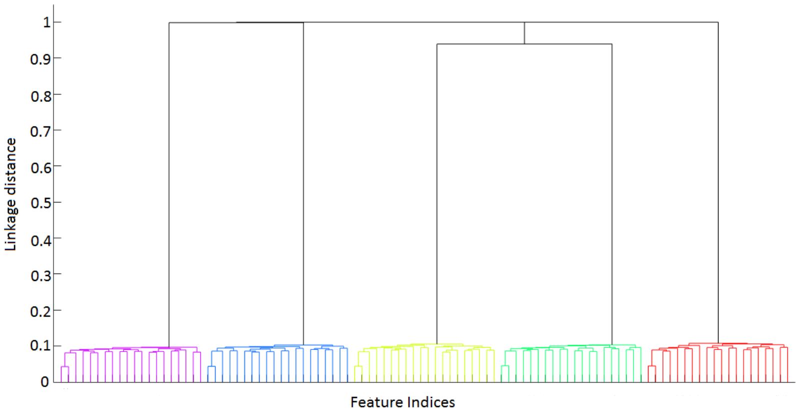


Figure 5.2: Dendrogram used in hierarchical clustering step with $Q = 5$ clusters

Each cluster is labeled with a unique color. Note that BoPGD also recovers the exact set of clusters as CRL does. If it had not done a subsequent sparse estimation (step 4 of BoPGD), it would also have recall = 1 but at the cost of low precision and F_1 score since some of the noisy, irrelevant variables correlated with the "active" ones would have gotten selected as well.

Synthetic Experiment 2(a) [Multi block support] : Consider the same block diagonal model for simulating X with $d = 120$ and $Q = 6$ blocks. The "true" support S_0 comprises of 30 features chosen uniformly at random (in each simulation) from the first three blocks Q_1, Q_2, Q_3. That is why this set up is called "Multi block support" since the active variables are spread out across multiple blocks. For each block, half of it contains the active variables and the other half comprises of nuisance variables strongly correlated with the active ones. The clustering step of CRL and BoPGD consistently selects the first three clusters across all iterations. The results of our experimental evaluation is presented in table 5.2. Here both PGD and BoLasso are comparatively better than BoPGD when the evaluation metrics are recall and F_1 score.

Methods	**P**	**R**	**F$_1$**	**MSE**
Lasso	0.459	0.997	0.628	0.014
Elastic Net	0.386	0.997	0.555	0.010
BoLasso	0.992	0.965	0.978	0.007
CRL	0.500	1.000	0.667	0.008
PGD	0.966	0.961	0.964	0.007
BoPGD	0.998	0.931	0.963	0.007

Table 5.2: Results on Synthetic Experiment 2(a) (averaged across 25 simulations)

Synthetic Experiment 2(b) [Multi block support] : Consider the same set up with certain changes in parameters. $X \sim \mathcal{N}_d(0, \Sigma)$ with $d = 100$ and Σ is partitioned into $Q = 5$ blocks. The "true" support S_0 comprises of 50 features chosen uniformly at random (in each simulation) from all the 5 blocks $\{Q_1, \ldots, Q_5\}$. That is why this set up is called "Multi block support" since the active variables are spread out across all the blocks. For each block, half of it contains the active variables and the other half comprises of nuisance variables strongly correlated with the active ones. The clustering step of CRL and BoPGD consistently selects all the clusters across all iterations. Hence, the active variables in this setup has much more "competition" with the non-active ones. The results of our experimental evaluation is presented in table 5.3.

Synthetic Experiment 3 : The purpose of this experiment is to demonstrate the efficiency (in terms of Precision, Recall and F_1 score) of BoPGD in comparison with the other algorithms as the dimension of the data is increased. In this setup, dimension (d) of each data point is

Methods	P	R	F_1	MSE
Lasso	0.710	0.941	0.809	0.013
Elastic Net	0.652	0.979	0.783	0.016
BoLasso	0.934	0.921	0.927	0.032
CRL	0.500	1.000	0.667	0.009
PGD	0.998	0.862	0.925	0.021
BoPGD	0.978	0.900	0.938	0.034

Table 5.3: Results on Synthetic Experiment 2(b) (averaged across 25 simulations)

varied all the way from 10 to as large as 890. The size of the true support is $d/2$ for each d and it is chosen uniformly at random from $[1, d]$. The data generation process of X is the same block diagonal model as before and the block size is adjusted so that the number of blocks Q is held constant ($Q = 5$) for all values of d. The results are presented in figure 5.3.

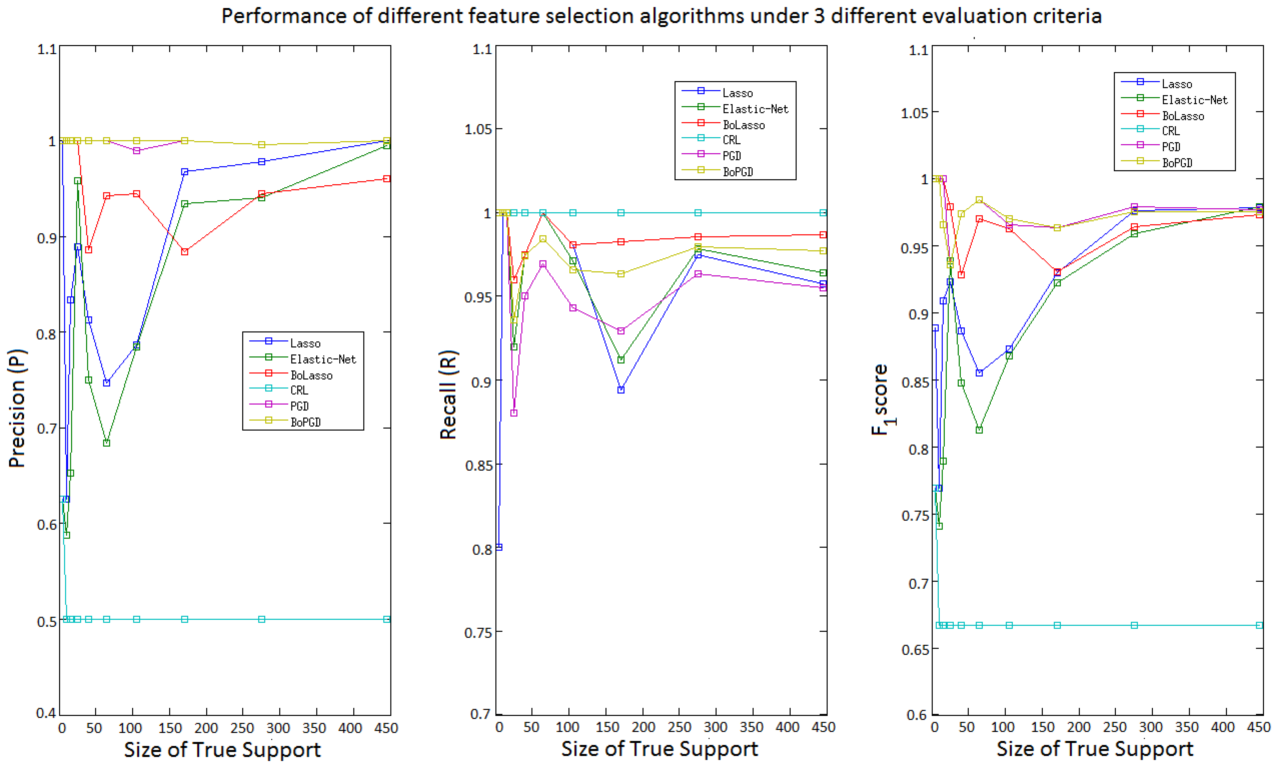


Figure 5.3: Performance of different algorithms as dimension (d) is varied

In each subplot, the blue line represents Lasso, green is Elastic Net, red is BoLasso, cyan represents Cluster Representative Lasso, magenta is PGD and yellow indicates BoPGD. The abscissa for each plot denotes the size of the true support ($\mid S_0 \mid$). For the first subplot on the left, the ordinate denotes the precision (P) for each algorithm, similarly for the next 2 subplots the ordinate denotes the Recall (R) and F_1 score respectively. As you can see from figure 5.3, both PGD and BoPGD are better than others in terms of P and F_1 score, while CRL is best when the evaluation criteria is Recall (true positive rate). Also BoPGD has a better recall than PGD as d increases. The performance of Lasso and Elastic Net fluctuates a lot as d changes. Thus if we look at the overall measure - F_1 score, which balances both true positive rate and false positive rate, then BoPGD is definitely a superior choice.

5.3 Real World Data

We present our findings on a few real-world public datasets collected from the UC Irvine Machine Learning Repository [Lic13].

Experiment 1 : We use the Abalone dataset[1] from the UCI repository. Abalones are basically sea snails. The age of an abalone can be determined by counting the number of layers in its shell. However, it is a time-consuming and cumbersome process : It involves cutting the shell through the cone, staining it, and counting the number of rings through a microscope. This UCI dataset contains 4177 measurements of 9 physical features of abalone. The goal is to predict the age of abalone from the listed descriptors. The attributes along with their brief description is given in table 5.4. The attribute 9 (Rings) is the target variable. Features 2 though 8 are strongly correlated ($\rho \geq 0.75$) as shown in figure 5.4. Here also we compare BoPGD with the other algorithms as before. The support estimated ($\hat{S}$) by BoPGD includes features $\{1, 3, 4, 5, 6, 7, 8\}$ with a test set prediction error of 2.232. Since here the ground truth is unknown, the only way to compare our method is to look up the results involving this dataset in literature. The results

Index	Name	Data type	Description
1	Sex	Nominal	M (male), F (female) and I (infant)
2	Length	Real number	Longest shell measurement (in mm)
3	Diameter	Real number	Perpendicular to length (in mm)
4	Height	Real number	With meat in shell (in mm)
5	Whole weight	Real number	Whole abalone weight (in grams)
6	Shucked weight	Real number	Weight of meat (in grams)
7	Viscera weight	Real number	Gut weight (after bleeding) (in grams)
8	Shell weight	Real number	Weight after being dried (in grams)
9	Rings	Integer	+1.5 gives the age in years

Table 5.4: Description of Abalone data

[1] https://archive.ics.uci.edu/ml/datasets/Abalone

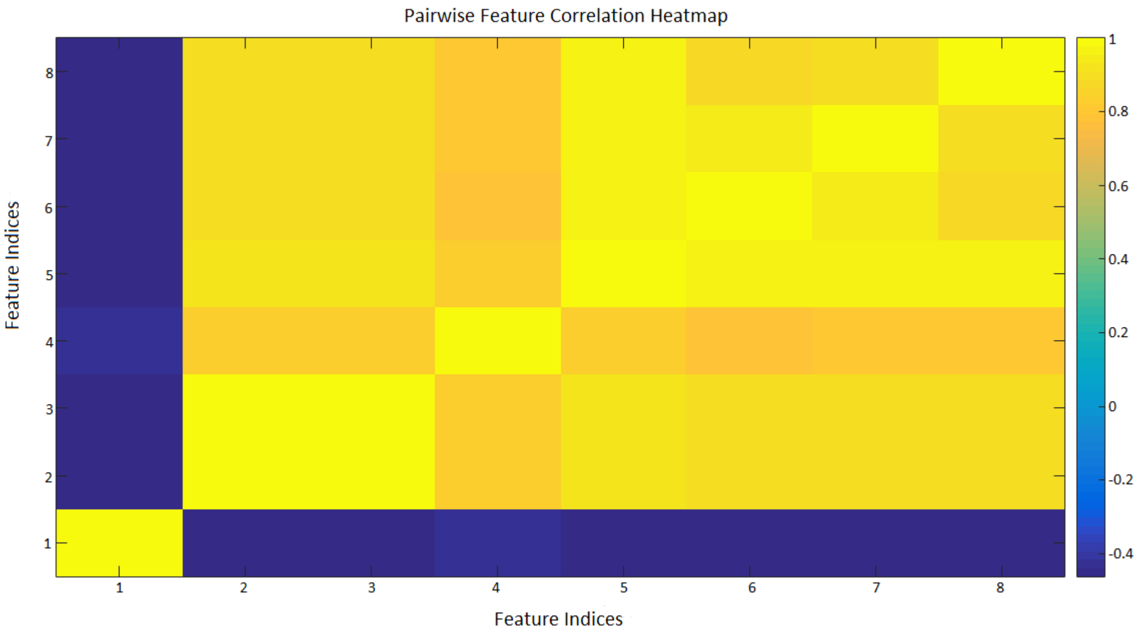


Figure 5.4: Correlation heat map of predictors from Abalone data

involving these standard feature selection methods are presented in table 5.5. Both CRL and BoPGD consistently cluster features $\{2, 3, 4, 5, 6, 7, 8\}$ together. CRL then applies Lasso while BoPGD applies projected gradient descent on the cluster representative matrix respectively. Additionally using the sparse estimation step BoPGD is able to get rid of feature 2 from $\hat{S}$. Lasso shows inconsistency in selecting features from this strongly correlated set, for e.g. it often estimates features $2, 5$ in $\hat{S}$ and sometimes it doesn't. Also it shows sign inconsistency with feature 8. Since Elastic Net simply averages the coefficients of the correlated features in some sense, it usually picks all the features in its estimated support. BoLasso, however picks the correct subset of features but shows sign inconsistency only with feature 8. The performance of PGD and BoPGD here is similar in terms of support recovery.

Methods	**Estimated Support ($\hat{S}$)**	**MSE on test data**
Lasso	$\{1, 2^*, 3, 4, 5^*, 6, 8\}$	2.425
Elastic Net	$\{1, 2, 3, 4, 5, 6, 7^*, 8\}$	2.244
BoLasso	$\{1, 3, 4, 5, 6, 7, 8\}$	2.221
CRL	$\{1, 2, 3, 4, 5, 6, 7, 8\}$	2.742
PGD (IHT)	$\{1, 3, 4, 5, 6, 7, 8\}$	2.476
BoPGD	$\{1, 3, 4, 5, 6, 7, 8\}$	2.232

Table 5.5: Results on Abalone data

Experiment 2 : We use the AutoMPG dataset[1] from the UCI repository. Flashback to the 1970s, when cars were big, heavy and used lots of gas. The Auto MPG sample data set is a collection of 398 automobile records from the year 1970 to 1982. This data concerns city-cycle fuel consumption in miles per gallon (MPG), to be predicted in terms of 3 multivalued, discrete and 5 continuous, real-valued attributes. The attributes along with their brief description is given in table 5.6. The attribute 1 (MPG) is the response variable. We purposefully ignore feature 9 (car name) from the data. Features $\{2, 3, 4, 5\}$ are strongly correlated ($\rho \geq 0.84$). The support estimated along with their normalized coefficients (except CRL whose coefficients are reported in the original feature scale) of all the algorithms is shown in table 5.7. Since here the ground truth is unknown, we verified with the existing results reported on this dataset and the most predictive ones are : $\{$(5) Weight, (3) Displacement, (4) Horsepower, (7) Model Year, (8) Origin$\}$. Also feature 5 has a negative correlation with MPG. PGD and BoPGD predict the same set of features, retaining the appropriate correlation, but both miss out on feature 4 (horsepower), possibly due to small number of instances. Lasso never selects feature 3 in

Index	**Name**	**Data type**	**Description**
1	MPG	Continuous (Real)	Miles per gallon
2	Cylinders	Multi-valued discrete	Number of gas cylinders
3	Displacement	Continuous (Real)	Engine size
4	Horsepower	Continuous (Real)	Power of the car
5	Weight	Continuous (Real)	Weight of the car
6	Acceleration	Continuous (Real)	Speed
7	Model Year	Multi-valued discrete	Year of manufacture
8	Origin	Multi-valued discrete	Source of manufacture
9	Car Name	String	Unique car name for each instance

Table 5.6: Description of AutoMPG data

Ftr. index	**Lasso**	**Elastic Net**	**BoLasso**	**CRL**	**OSCAR**	**PGD**	**BoPGD**
2	0	-0.502	-0.716*	-0.648	-0.102	0	0
3	0	-0.004	0.282*	1.890	-0.102	0.271	0.206
4	-0.002	-0.029	0	-0.971	-0.102	0	0
5	-0.006	-0.003	0.011	-5.285	-0.102*	-0.731	-0.661
6	0	0	0.082*	0	0	0	0
7	0.530	0.580	-0.007*	2.771	0.102*	0.353	0.349
8	0.387	0.974	0.160	1.174	0.061	0.163	0.166

Table 5.7: Results on AutoMPG data

[1] https://archive.ics.uci.edu/ml/datasets/Auto+MPG

its support. Elastic Net averages the coefficients of the correlated features in some sense and picks all the correlated features in its estimated support. BoLasso is inconsistent in support recovery. The $*$ marked coefficients indicate they are irregular in $\hat{S}$. Both CRL and BoPGD cluster features $\{2, 3, 4, 5\}$ together but since the representative feature from this cluster is dominant as estimated by Lasso on the cluster representative matrix, hence, all the features from this cluster are selected by CRL. Since the number of samples and also the dimension of each sample is less in this dataset, we also included OSCAR for comparison. As expected, the performance of OSCAR here is better than others. It reports the correct subset of features and also cluster the strongly correlated features into one group most of the time. The features belonging to a group have the same absolute value for coefficients. However, in some iterations (under different training set), it mis-clusters feature 5 and 7.

Part II

Causal Inference on Time Series

Chapter 6

Preliminaries

In this section we formally introduce the problem and describe the key concepts and notions used for inferring *Granger Causality* on time series data. We start with some basic definitions and concepts that are essential to understand the problem and finally we will formulate the causal inference problem by putting everything together.

6.1 Feature Causal Graph

Given a set of features $V = \{x_1, x_2, ..., x_P\}$, where each x_i is a time series, we construct a directed graph over the set of features [1], called the *feature causal graph*, $G = (V, E)$. A *feature vector* at some particular point in time t is the P-tuple $(x_1^t, x_2^t, \ldots, x_P^t)$ of the features. This P should not be confused with the model order (or maxlag p) which we have mentioned in the introduction chapter. From here on, we will denote the model order with L and the number of time series variables using P. Nodes in graph G correspond to the features and edges capture causality. An edge $e \in E$, directed from x_i to x_j is labeled with a natural number ℓ, called the *time lag* (*or temporal lag*), which captures a *causal* relationship of the form $x_i^{t-\ell} \rightarrow x_j^t$ $\forall t > \ell$ i.e. the current value of x_j is affected by the value of x_i ℓ-steps back, and this makes x_i a causal variable for x_j. Each causal variable can have different time lags at which they influence the observations of another variable. For example, if say x_1 is causally affected by k variables $x_2, \ldots, x_{k+1}$ with corresponding time lags $\ell_2, \ldots, \ell_{k+1}$, the ℓ_i's in principle could be any arbitrary Natural number. The semantics of a *feature causal graph* [MS06] is same as that of a *Bayesian network* [Hec98, Mee97] with the additional premise that an edge necessarily entails causation in the former.

[1] Note that we are referring to each time series variable as a feature. This should be obvious once we formulate our problem as feature selection especially using Group Lasso.

6.2 Vector Autoregressive Model

Vector Auto regression (VAR) is an econometric model used to capture the linear inter dependencies among multiple temporal variables. VAR models generalize the univariate autoregressive (AR) model by allowing for more than one evolving variable. In VAR, each variable has an equation which explains its evolution as a linear combination of its own time-lagged values and the lagged values of other model variables as well.

A VAR model describes the evolution of a set of P variables (often called endogenous variables) over a sample time period $t = 1, \ldots, T$ as a *linear* function of only their past values. Let X^t denote the $P \times 1$ vector of all the features at time t, thus the i^{th} element X_i^t denotes the observation at time t of the i^{th} variable. An L-th order VAR process, denoted VAR(L), is

$$X^t = C + A_1 X^{(t-1)} + A_2 X^{(t-2)} + \cdots + A_L X^{(t-L)} + R^t \tag{6.1}$$

where $X^{(t-\ell)}$ containing the ℓ-period back observations of all the model variables is called the ℓ^{th} *lag* of X, C is a $P \times 1$ vector of constants (intercept terms), A_ℓ is a time-invariant $P \times P$ coefficient matrix, and R^t is again a $P \times 1$ vector of error (residual) terms satisfying the following assumptions :

1. Every error term has *zero* mean i.e. $\mathbb{E}[R^t] = 0$.
2. The covariance matrix $Q = \mathbb{E}[(R^t)(R^t)^T]$ of error terms is positive semi-definite.
3. There is no correlation across time i.e. $\mathbb{E}[(R^t)^T(R^{(t-k)})] = 0$, $\forall\ 0<k<t$.

The formulation 6.1 is a notational shorthand for multiple Linear regression formulation, one for each variable.

6.3 Granger Causality

A notion of causality that is highly relevant to the present context of temporal causal modeling is the *Granger Causality* [Gra69, Gra80]. Introduced long back in the field of econometrics by the Nobel laureate Clive Granger (1969), it is now one of the most popular approaches to quantify causal relationships among time series data. It is based on two major principles :

1. The cause happens prior to the effect
2. The cause makes unique changes in the effect

A time series X is said to "Granger Cause" another time series Y, denoted by $X \rightarrow Y$ if and only if regressing with past values of both X and Y is *statistically more significant* that doing so with past values of Y only. More formally, given two stationary time series $X = \{x^t\}_{t \in \mathbb{Z}}$ and $Y = \{y^t\}_{t \in \mathbb{Z}}$, let's consider the following two information sets :

1. $\mathbb{I}^*(t)$: the set of all information in the universe upto time t, and,
2. $\mathbb{I}^*_{-X}(t)$: the set of all information in the universe excluding X upto time t

Under the two principles of Granger Causality, the conditional distribution of the future values of Y given $\mathbb{I}^*(t)$ and $\mathbb{I}^*_{-X}(t)$ should differ. Then X is said to "Granger Cause" Y if $\mathbb{P}(Y^{(t+1)} \in S \mid \mathbb{I}^*(t)) \neq \mathbb{P}(Y^{(t+1)} \in S \mid \mathbb{I}^*_{-X}(t))$ for some measurable set $S \subseteq \mathbb{R}$ and $\forall t \in \mathbb{Z}$. Note that the original definition of Granger Causality is very general and does not assume anything about the underlying data generative model. For modeling the distributions of multivariate data, *Linear models* are mostly used since they are simple, robust and have strong empirical performance on most of the practical applications. As a result Vector Autoregressive models have turned out to be one of the most dominant approaches for modeling Granger Causality on time series data.

6.4 Testing Granger Causality

Linear Granger Causality test was initially introduced for a pair of variables only. Later it was extended for multivariate data which we will cover in the next subsection.

6.4.1 Linear Granger Causality Test for Bivariate data

Let $X = \{x^t\}_{t=1}^T$ and $Y = \{y^t\}_{t=1}^T$ be two time series variables of length T. The vectors (in bold) $\mathbf{x}_t = [x^{(t-1)}, x^{(t-2)}, \ldots, x^{(t-L)}]$ and $\mathbf{y}_t = [y^{(t-1)}, y^{(t-2)}, ..., y^{(t-L)}]$ denote the history (i.e. all L time-lagged values) of X and Y respectively up to time t, where L is the maximum time lag (the model order) of the VAR process. The *Linear Granger Causality test* to ascertain whether X *"Granger Causes"* Y is conducted as follows :

(a) First two different VAR models are fit to the data as follows :

$$y^t \approx \langle \alpha, \mathbf{y}_t \rangle + \langle \beta, \mathbf{x}_t \rangle \tag{6.2}$$

$$y^t \approx \langle \gamma, \mathbf{y}_t \rangle \tag{6.3}$$

where $\alpha = [\alpha_1, \ldots, \alpha_L]$ and $\gamma = [\gamma_1, \ldots, \gamma_L]$ are two different coefficient vectors of $\mathbf{y}_t$ and $\beta = [\beta_1, \ldots, \beta_L]$ that of $\mathbf{x}_t$, and $\langle a, b \rangle = a^T b$ is the usual notation for *dot* (inner) product of two vectors a and b.

(b) Then, any standard joint statistical significance test (viz. F-Statistic test, χ^2-test or some other statistical significance test) is conducted to obtain a p-value along with the residual error which helps to determine whether model 6.2 is a "better fit" than model 6.3 with significant statistical advantage. When the first model outperforms the second, it is concluded that X *"Granger causes"* Y. Similarly, it can be ascertained whether Y *"Granger causes"* X as well.

6.4.2 Linear Granger Causality Test for Multivariate data

Given multivariate time series $X_i = \{x_i^t\}_{t=1}^T$, $\forall i = \{1, 2, \ldots, P\}$, where P is the number of time series variables and T is the length of each time series. Consider X_i to be the target variable, so our objective is to find which of the time series variables $X_1, \ldots, X_P$ *"Granger causes"* X_i. A VAR model of order L is fit to X_i, $\forall t = L + 1$ to T as follows :

$$x_i^t = \sum_{j=1}^{P} \langle \beta_j^i, x_j^{(t,L)} \rangle + \epsilon_i^t \tag{6.4}$$

where $x_j^{(t,L)} = [x_j^{(t-1)}, \ldots, x_j^{(t-L)}]$ is the history of X_j up to time t, $\beta_j^i = [\beta_j^i(1), \ldots, \beta_j^i(L)]$ is the coefficient vector modeling the effect of time series X_j on X_i, L is the maximum time lag, and ϵ_i^t is independent additive white noise. Note $\langle \beta_j^i, x_j^{(t,L)} \rangle$ denote the dot (inner) product of the vectors $x_j^{(t,L)}$ and β_j^i, and therefore equation 6.4 can be re-written as :

$$x_i^t = \sum_{j=1}^{P} \sum_{\ell=1}^{L} \beta_j^i(\ell) x_j^{(t-\ell)} + \epsilon_i^t \tag{6.5}$$

We can determine that time series X_j *"Granger causes"* X_i, if *at least one* value in β_j^i is *non-zero* again by some statistical significance test [MPS08]. If we run the VAR model for every feature $X_i : i \in [P]$, we can find the causal variables for every feature and construct the output feature causal graph. In chapter 7 we will mention some methods which use these tests extensively.

6.5 Data Generation Process

In this section, firstly we describe the time series data generation process and then precisely formulate the modeling problem whose objective is to characterize the causal relationships among the temporal features.

Given a feature causal graph $G = (V, E)$ over a set of P features $V = \{x_1, x_2, ..., x_P\}$, where each vertex x_i is a time series and directed edges capture the direction of causal influences, we associate a stochastic process that generates time series data with respect to this graph. We start by fixing a *predefined window size* L such that L [1] is at least the time lag corresponding to each edge in the graph G. This ensures that L is greater than or equal to the maximum time lag in the network. Now given this graphical model over the temporal variables, the stochastic process starts by initializing a sequence of L feature vectors for time points $t = 0, \ldots, L-1$. At each step henceforth, it generates the next feature vector according to the conditional probability distribution $P\left(\{x_i^L\}|\{x_j^t\}_{j=1,2,\ldots,P,t=0,1,\ldots,L-1}\right)$ on the graphical model where the variables x_j^t for $t = 0, 1, \ldots L-1$ are as initialized in the last L steps. This is the *"Unit Causal Graph"* generation method as stated in [CG06]. The conditional probabilistic model could in principle be any arbitrary statistical model [CG06, HSK06], but, in the current work, we assume that the both the conditional probabilities and the initial distribution of time series data are linear combinations of Gaussians [RG99]. Under these assumptions, it is easy to see that the stochastic model associated with the causal feature graph is equivalent to the VAR model.

The goal of a causal modeling algorithm is to infer the underlying causal structure, given as input the time series data generated by its associated stochastic process. The performance of a causal modeling algorithm can be measured purely in terms of similarity between the hypothesis (output) causal graph and the original graph that gave rise to the time series data. The details about the evaluation criteria is described in the next section.

6.6 Evaluation Criteria

In this section, we briefly describe the evaluation criteria to quantify the similarity between the hypothesis causal graph and the original graph. We use the metrics of Precision, Recall and F_1 measure to the problem of predicting a 0 or 1 entry in the adjacency matrix representation of the graph [SSGS06]. Note that for any pair of features x_i and x_j, there are two entries in the adjacency matrix A, $A(i,j)$ and $A(j,i)$. The entry $A(i,j) = 1$ implies that there is a directed edge from x_j to x_i in the feature causal graph, which further means that x_j causally influences x_i. A bi-directional edge having both $A(i,j) = A(j,i) = 1$, corresponds to causality in both directions i.e $x_i \rightarrow x_j$ as well as $x_j \rightarrow x_i$. Given this formulation, precision and recall are well defined. For example, predicting a bi-directional edge between x_i and x_j when there is actually a directed edge only from $x_i \rightarrow x_j$, would entail one correct prediction and one error. So in this case precision (P) would be 0.5, recall (R) is 1, and therefore F_1 score 0.67.

Let A denote the adjacency matrix of the source (original) feature causal graph, and $\hat{A}$ denote the same for the hypothesis (output) graph, then the expressions for *precision* (P),

[1] Note that we are abusing the notation L for maxlag, since we want to establish the natural connection between the model order in a VAR process and the window size here.

recall (R) are as follows :

$$P = \frac{|\{(i,j) \in V \times V : \hat{A}(i,j) = A(i,j)\}|}{|\{(i,j) \in V \times V : \hat{A}(i,j) = 1\}|} \tag{6.6}$$

$$R = \frac{|\{(i,j) \in V \times V : \hat{A}(i,j) = A(i,j)\}|}{|\{(i,j) \in V \times V : A(i,j) = 1\}|} \tag{6.7}$$

and the F_1 score which tries to balance the overall quality of prediction is the harmonic mean of P and R expressed as :

$$F_1 = \frac{2PR}{P + R} \tag{6.8}$$

Chapter 7

Existing Methods of Granger Causality

In this chapter we discuss some of the existing and most popular approaches for modeling Granger Causality on multivariate time series. All of these methods use the VAR model with a *fixed* value of the model order L known a priori based on domain knowledge or intuition of the underlying data generating process. We will end this chapter with a brief description of some existing techniques for estimating the model order of a VAR process when the latter is not known.

7.1 Exhaustive Graphical Granger Method

The most trivial way of applying Granger Causality on multivariate time series data is to simply conduct the *Linear Granger causality test* 6.4.1 for every pair of features in order to determine the presence/absence and the orientation (if present) of the corresponding edge(s) in the output feature causal graph. The pseudo code of this method is as follows :

```
procedure EXHAUSTIVE_GRANGER(X,T,P,L)
    Initialize $X^{Lagged} \leftarrow Lag(X,T,L)$
    Initialize $G = (V,E) \leftarrow CompleteFeatureGraph(X)$
    for each edge $e = (x,y) \in E$ do
        if $LGtest(x,y,X^{Lagged})$ == 'Yes' and $LGtest(y,x,X^{Lagged})$ == 'No' then
            Orient edge $e = (x,y)$ as $x \rightarrow y$ in $G$
        else if $LGtest(x,y,X^{Lagged})$ == 'No' and $LGtest(y,x,X^{Lagged})$ == 'Yes' then
            Orient edge $e = (x,y)$ as $y \rightarrow x$ in $G$
        else if $LGtest(x,y,X^{Lagged})$ == 'Yes' and $LGtest(y,x,X^{Lagged})$ == 'Yes' then
            Place a bi-directional edge $x \rightarrow y$ and $y \rightarrow x$ in $G$
        else
```

```
            Do not place an edge $e$ between $x$ and $y$
        end if
    end for
    Return $G$
end procedure
```

The procedure takes as input a multivariate time series data X with P features each being a time series of length T. The function $Lag(X, T, L)$ creates a L-time lagged version of the data as described in the previous chapter in section 6.5, L being the *pre-defined* max time lag in the VAR process. G is a complete graph over the set of P features and the routine $LGtest(x, y, X^{Lagged})$ (as already described in 6.4.1) returns 'Yes/No' inferring whether x "Granger Causes" y.

7.2 Vector Autoregressive Granger Method

This is the most natural and intuitive method of estimating Granger Causality on multivariate time lagged data. It is the underlying stochastic model of data generation given as input the feature causal graph. We have already covered the VAR model in section 6.2. Here also we fix a lag L and fit a VAR model of order L to each feature as described in equation 6.5. It is easy to see that if we take all the entries of a variable say x_i from $t = L + 1$ to T and write them in a vector form of length $T - L$ and similarly the r.h.s of the same equation (6.5) as a matrix X^{Lagged} of past L values of all the features, then this equation can also be written as :

$$Y = X^{Lagged}\beta + E \tag{7.1}$$

where the t^{th} entry of Y is x_i^t and the corresponding row of the matrix X^{Lagged} is a row vector $[[x_1^{t-1}, \ldots, x_1^{t-L}], \ldots, [x_P^{t-1}, \ldots, x_P^{t-L}]]$ of size PL and E is the vector of white noise.

This reduces to the classical Ordinary Least Squares (OLS) Regression formulation which has a closed form solution. The *non-zero* entries of the coefficient vector β is used to determine the subset of features causally affecting the target feature (x_i in this case). The above process is repeated for all the P features in the model and the OLS estimate of the coefficient vectors are used to construct the output feature causal graph.

7.3 Lasso Granger Method

The Exhaustive Granger method does not address the issue of combinatorial explosion in the computational sense. It conducts *Linear Granger Causality Test* (6.4.1) $\mathcal{O}(P^2)$ times where P is the number of features. This is expensive and hence prohibitive for large values of P (high-dimensional setting). Also the statistical significance tests are carried out pairwise sequentially ignoring the possible interactions between all the other variables. Furthermore, if the true feature causal graph is *sparse*, the VAR Granger method cannot retrieve the sparsity pattern in neighborhood selection. The Lasso Granger method [VSSBLC$^+$05, ALA07, FSGM$^+$07] addresses these issues. It identifies the correct subset of features on which the feature in question is conditionally dependent, given the fact that the best regressor (in *least squared* sense) for that variable has non-zero coefficients only for the same corresponding variables in the neighborhood.

The LASSO algorithm [Tib96] is one of the most widely used procedure for automatic variable selection (also called feature selection, model selection). It solves the following optimization problem :

$$\hat{\beta} = \arg\min \frac{1}{n} \sum_{(x,y)\in\mathbb{S}} (y - \beta^T x)^2 + \lambda \left\|\beta\right\|_1 \tag{7.2}$$

where (x, y) are i.i.d draws from the input sample $\mathbb{S}$, n is the number of samples in $\mathbb{S}$, and λ is a constant (usually called the regularizer) to be determined. The first term minimizes the empirical risk and the second term induces *sparsity* in the coefficient vector β due to the nature of L_1-norm penalty. Note if $\lambda = 0$, this problem reduces to OLS. LASSO is a convex problem making it possible to achieve global minima using an efficient "Least Angle Regression Shrinkage" (or LARS) procedure [EHJ$^+$04] which in turn uses the coordinate descent algorithm to incrementally update the weights, one variable at a time.

The Lasso Granger method applies this LASSO type formulation to the VAR model for each x_i, $i = \{1, 2, ..., P\}$ to obtain a sparse estimate of the coefficient vector. This is the optimization problem of Lasso Granger :

$$\min_{\beta} \sum_{t=L+1}^{T} \left(x_i^t - \sum_{j=1}^{P} \langle \beta_j^i, x_j^{(t,L)} \rangle\right)^2 + \lambda \left\|\beta\right\|_1 \tag{7.3}$$

where T is the length of each time series, λ is the regularization parameter enforced to obtain a sparse β and the rest of the variables are same as defined in section 6.4.2. Following the same arguments and notations as demonstrated in section 7.2, equation 7.3 can be written in a vector-matrix form as :

$$\min_{\beta} \left\|Y - X^{Lagged}\beta\right\|^2 + \lambda \left\|\beta\right\|_1 \tag{7.4}$$

where Y is a vector of length $T - L$, the t^{th} entry of Y being x_i^t and the corresponding row of the matrix X^{Lagged} is a row vector $[[x_1^{t-1}, \ldots, x_1^{t-L}], \ldots, [x_P^{t-1}, \ldots, x_P^{t-L}]]$ of size PL. Let us emphasize the fact once again that the maxlag parameter L is fixed a priori. Like the VAR Granger method (7.2), the above process is repeated for all the P features in the model and the *non-zero* entries of the coefficient vector β are used to determine the subset of features causally affecting the target feature and hence construct the output feature causal graph.

7.4 Group Lasso Granger Method

The above method 7.3, although computationally efficient, has neglected one important aspect of the problem - the natural *group structure* existing among the temporal lagged variables imposed by the respective time series they belong to. The Group Lasso Granger method [LALR09] overcomes these limitations by applying a regression method suited for high-dimensional data, and also leveraging the group structure among the temporal lagged variables according to the time series they belong to. It employs a technique called *Group Lasso* which alongside minimizing the empirical risk also performs variable selection by penalizing the intra-group and inter-group variable selection differently.

Consider a partitioning of an entire set of predictors $\{x_1, \ldots, x_P\}$ into J groups, the *Group Lasso* procedure due to [YL06] solves the following optimization problem :

$$\hat{\beta}_{group} = \arg\min_{\beta} \|Y - X\beta\|^2 + \lambda \sum_{j=1}^{J} \sqrt{\rho_j} \left\|\beta_{\mathbb{G}_j}\right\|_2 \tag{7.5}$$

where Y is the $N \times 1$ vector ($N :$ the number of samples) of responses, X is the $N \times P$ matrix of predictors, and $\beta_{\mathbb{G}_j} = \{\beta_k : k \in \mathbb{G}_j\}$ where $\mathbb{G}_j$ denotes the set of group indices and ρ_j accounts for varying group size. Notice that using L_2-norm to penalize within a group ensures similar shrinkage of coefficient values for all intra-group predictors and the L_1-penalty on top of this group structure performs group selection.

Extending this notion to time series data, the group lasso formulation [LALR09] is :

$$\min_{\beta} \left\|Y - X^{Lagged}\beta\right\|^2 + \lambda \sum_{j=1}^{P} \left\|\beta_{\mathbb{G}_j}\right\|_2 \tag{7.6}$$

where Y is a vector of length $T - L$, the t^{th} entry of Y being x_i^t and the corresponding row of the matrix X^{Lagged} is a row vector $[[x_1^{t-1}, \ldots, x_1^{t-L}], \ldots, [x_P^{t-1}, \ldots, x_P^{t-L}]]$ of size PL. Note that there are P groups here and since the groups are of equal size, $\rho_j = 1$, $\forall j = \{1, \ldots, P\}$. Here also the maxlag parameter L is fixed beforehand. The above method is repeated for each feature in the model and an edge $x_j \rightarrow x_i$ is placed in the output feature causal graph if and only if the coefficients $\beta_{\mathbb{G}_j}$ corresponding to x_j is non-zero implying that feature x_j is selected as a group by Group Lasso.

7.5 Consistency of Lasso Granger

Besides computational efficiency, the other major advantage of using LASSO for learning granger causal structures over graphical models is its consistency. It has been proven [MB06, ALA07] - the probability that LASSO falsely includes any of the non-neighboring variables of a given node into its neighborhood estimate is exponentially small in the number of observations, even under situations where the number of neighboring variables grow rapidly with the number of observations.

Let p be the number of features, a be an arbitrary node in the true graph $G = (V, E)$, ne_a be the set of neighbors of a in G, $\hat{ne}_a$ be the estimated neighborhood of a and let θ^{a,ne_a} be the coefficient vector of the optimal linear regressor for a. To keep things consistent, we have used the same notations as that in [MB06]. Also we make a few assumptions to prove consistency of neighborhood selection with Lasso.

Assumption 1 : [High-dimensionality] The number of variables (p) is allowed to grow as the number of observations (n) raised to an arbitrarily high power i.e $\exists$ a $\gamma > 0$ such that

$$p = \mathcal{O}(n^{\gamma}) \quad for \quad n \to \infty$$

Assumption 2 : [Non-singularity] A regularity assumption about the covariance matrix generating the data i.e for all $a \in V$ and $n \in \mathbb{N}$, $Var(a) = 1$ and $\exists\ v^2 > 0$ so that

$$Var(a \mid V \setminus \{a\}) \geq v^2$$

Assumption 3 : [Sparsity] This assumption entails a restriction on the size of the neighborhood of a variable. There exists some $0 \leq \kappa \leq 1$ such that

$$\max_a \mid ne_a \mid = \mathcal{O}(n^{\kappa}) \quad for \quad n \to \infty$$

and there exists some $\nu < \infty$ so that

$$\left\| \theta^{a, ne_b \setminus \{a\}} \right\|_1 \leq \nu \quad \forall (a, b) \in E$$

Assumption 4 : [Magnitude of partial correlations] There exists a constant $\delta > 0$ and some

$\zeta > \kappa$ with κ as already stated in Assumption 3, so that for every $(a, b) \in E$

$$| \pi_{ab} | \geq \delta n^{-(1-\zeta)/2}$$

where π_{ab} is the partial correlation between a and b after having eliminated the linear effects from all the remaining variables [Lau96].

For all $(a, b) \in E$, the *neighborhood stability* of a w.r.t b is defined as follows :

$$S_a(b) := \sum_{k \in ne_a} sign(\theta_k^{a,ne_a}) \theta_k^{b,ne_a}$$

Assumption 5 : [Neighborhood Stability] There exists some $\delta < 1$ such that for all $(a, b) \in E$

$$| S_a(b) | < \infty$$

Then due to [MB06] we have the following two theorems :

Theorem 7.1 *Under the assumptions 1-5, and let the penalty parameter for sample size n satisfy $\lambda_n \sim dn^{-(1-\epsilon)/2}$ with some $\kappa < \epsilon < \zeta$ and $d > 0$, there exists some $c > 0$ such that $\forall a \in V$*

$$P(\hat{ne}_a \subseteq ne_a) = 1 - \mathcal{O}(exp(-cn^{\epsilon})) \quad for \quad n \to \infty$$

Theorem 7.2 *Under the assumptions of Theorem 7.1, for $\lambda = \lambda_n$ and some $c > 0$ such that $\forall a \in V$*

$$P(ne_a \subseteq \hat{ne}_a) = 1 - \mathcal{O}(exp(-cn^{\epsilon})) \quad for \quad n \to \infty$$

These theorems can be directly applied to derive a corollary on the consistency of the Lasso Granger method.

Corollary : Suppose that a true feature causal graph G and its associated stochastic graphical model M gives rise to the time series data. If the assumptions in Theorem 7.1 and 7.2 are satisfied by M, then the graph $\hat{G}$ output by Lasso Granger method, taking the time series data as input, will be consistent with the true graph G with probability approaching 1, as n and p tend to ∞.

The proof of this corollary along with a comparative analysis of some of the above methods is presented in [ALA07].

7.6 Existing methods on VAR order selection

In this section, we will take a small digression from Granger Causality methods and discuss some of the existing techniques for determining the model order of a VAR process when it is *not known* beforehand. In practice, the model order L is usually unknown. This is relevant to our problem which we discuss in detail in future chapters.

Let us consider the VAR model 6.1 once again.

$$X^t = C + A_1 X^{(t-1)} + A_2 X^{(t-2)} + \cdots + A_L X^{(t-L)} + R^t \tag{7.7}$$

If this is a correct representation of the process X^t, then the same is true for the following model with $A_{L+1} = 0$.

$$X^t = C + A_1 X^{(t-1)} + A_2 X^{(t-2)} + \cdots + A_L X^{(t-L)} + A_{L+1} X^{(t-L-1)} + R^t \tag{7.8}$$

Thus if X^t is a VAR(L) process, it is also a VAR($L+1$) process. To remove this ambiguity, it is practical to assign a *unique number* to L and call it the order of a VAR process. Therefore, X^t is a VAR process of order L, if L is the smallest possible number such that $A_L \neq 0$ and $A_i = 0 \ \forall i > L$. It is not hard to see that fitting a VAR($L+i$) : $i > 0$ process to a VAR(L) process has inferior mean squared error (MSE) as illustrated vividly in [Lüt05].

Here is a testing scheme for determining the model order of a VAR process. Assume M to be a known upper bound for the VAR order, then the following sequence of Null and Alternative Hypotheses tests are conducted using the Likelihood Ratio (LR) statistic :

$$\begin{aligned}
H_0^1 : A_M = 0 &\qquad vs \qquad H_1^1 : A_M \neq 0 \\
H_0^2 : A_{M-1} = 0 &\qquad vs \qquad H_1^2 : A_{M-1} \neq 0 \mid A_M = 0 \\
\vdots \qquad &\qquad\qquad\qquad \vdots \\
H_0^i : A_{M-i+1} = 0 &\qquad vs \qquad H_1^i : A_{M-i+1} \neq 0 \mid A_M = \cdots = A_{M-i+2} = 0 \\
\vdots \qquad &\qquad\qquad\qquad \vdots \\
H_0^M : A_1 = 0 &\qquad vs \qquad H_1^M : A_1 \neq 0 \mid A_M = \cdots = A_2 = 0
\end{aligned}$$

Each null hypothesis is tested conditioned on all the previous ones being true. The procedure terminates when one of the null hypotheses is rejected i.e if H_0^i is rejected, then $\hat{L} = M - i + 1$ is chosen as the estimate of the model order of the VAR process. The LR statistic for testing the i^{th} hypothesis is

$$\lambda_{LR}(i) = T\big[\ln |\,\tilde{Q_R}(M-i)\,| - \ln |\,\tilde{Q_R}(M-i+1)\,|\,\big] \tag{7.9}$$

where $Q_R = \mathbb{E}(RR^T)$ is the covariance matrix of residuals R, $\tilde{Q_R}$ is the maximum likelihood estimator (MLE) of Q_R when a VAR(M) model is fitted to a time series of length T. The choice of significance levels for this statistical significance tests is tricky and in general they reflect the approximate probabilities of Type-I errors only. A detailed description of this scheme and a few others is covered in [Lüt05]. These hypotheses testing schemes based on statistical significance tests require the strong assumption of a *single*, *universal* model order L that is applicable for all the variables in the model. Although this reduces the computational complexity of model selection, it unnecessarily constraints the linear relationship among the predictors to same lag order which in reality might not be true.

Gredenhoff and Karlsson [GK99] instead suggested a technique called *asymmetric VAR* which allows each feature to have a different lag order in its own vector auto regression. Hsiao [Hsi81] discusses about one such asymmetric procedure which starts from a univariate auto regressive (AR) model and sequentially adds the time lagged version of other variables based on Akaike's "Final Prediction Error" criterion (FPE) [Aka69]. There is a more general method due to [Kea01] which estimates all L^P possible VAR models and choose the best using Akaike Information Criterion (AIC) [Aka98] or Schwarz Information Criteria (BIC) [S$^+$78]. Ding and Karlsson [Din14] also introduced a Bayesian framework for variable lag order selection using Markov Chain Monte Carlo (MCMC) techniques with various priors.

But all of the above methods for lag order selection are computationally intensive both in terms of space and time complexity and therefore none of them can be employed for estimation of parameters in the high dimension setting. Also they are not designed to consider sparsity constraint in their estimation framework. So shrinkage based approaches, (using Lasso, Group Lasso), which impose sparsity on the parameter space to make estimation tractable, have become more popular.

Chapter 8

Our Method

This chapter highlights our contribution to this rich area of causal inference. Let us restate the problem once again. Given a *multivariate*, possibly *high-dimensional* time series data, our objective is to infer the underlying “causal structure” using the notion of “Granger Causality”. We model the time series as a vector autoregressive (VAR) process but with *unknown model order (L)*. We present two algorithms here which not only constructs the hypothesis feature causal graph, but also simultaneously estimates a value of maxlag ($\hat{L}$) by balancing the trade-off between “goodness of fit” and “model complexity”.

8.1 Our concurrent estimation method

As we have already pointed out before, all the existing methods for mining “Granger Causality” on time series data fit a VAR model, with the parameter maxlag L fixed a priori. We propose a method which concurrently estimates the “Granger coefficients” in the VAR model for each time series variable and the “best” value of maxlag which minimizes the MSE subject to the constraint that the model should not be over-parameterized. The maxlag (L) values for each time series variable could in principle be different and arbitrarily large. For example, consider the following VAR equations :

$$x(t) = a_1 * x(t-1) + a_2 * y(t-2) + \epsilon_1(t) \tag{8.1}$$

$$y(t) = b_1 * y(t-10) + \epsilon_2(t) \tag{8.2}$$

x is being causally affected by itself at lag = 1 and by y at lag = 2 time steps back. Similarly every current value of y is affected by its own value $l = 10$ time steps back in the past. Our method reports the maximum lag for each feature (e.g $L = 2$ and $L = 10$ for x and y resp.) as the “best” estimate of model order in its own multivariate VAR model. We have already highlighted the efficiency and consistency of shrinkage based methods such as Lasso Granger and Group Lasso Granger in extracting the underlying causal structure especially when the data is high dimensional. Our method, described next, is based on the same framework with the only difference that we do not fix a value of maxlag L beforehand. The general approach for both of our algorithms *Lasso Granger++* (8.1.1) and *Group Lasso Granger++* (8.1.2) is same except for the Group effect considered in the second one.

8.1.1 Lasso Granger++

Let $\mathbb{S}$ be a time series data on P variables $\{x_1, \dots, x_P\}$ where each x_i, $\forall i \in \{1, \dots, P\}$ is a time series of length T. We assume an upper bound, M on the maxlag value for each feature. Note M could be as large as $T-2$. We start with an initial estimate of L, denoted as L_0 to some positive integer ℓ. We recommend initializing $L_0 = \ell = 1$, although it is not mandatory to do so [1]. We build a multivariate VAR model of order ℓ for the target variable (let it be x_i) and use Lasso Granger method (7.3) to solve it and obtain a *sparse* coefficient vector β^0. Therefore, the first optimization problem we solve is as follows :

$$\min_{\beta} \quad \frac{1}{2n} \sum_{t=\ell+1}^{T} \left(x_i^t - \sum_{j=1}^{P} \langle \beta_j^i, x_j^{(t,\ell)} \rangle \right)^2 + \lambda \left\|\beta\right\|_1 \tag{8.3}$$

which is equivalent to :

$$\beta^0 = \arg\min_{\beta} \quad \frac{1}{2n} \left\|Y - X^{Lagged}\beta\right\|^2 + \lambda \left\|\beta\right\|_1 \tag{8.4}$$

where $n = T - \ell$ denotes the number of samples (observations), Y is a (column) vector of length n and X^{Lagged} is a $n \times P\ell$ matrix. The t^{th} entry of Y is x_i^t and the corresponding row of X^{Lagged} is a row vector $[[x_1^{t-1}, \dots, x_1^{t-\ell}], \dots, [x_P^{t-1}, \dots, x_P^{t-\ell}]]$ of size $P\ell$. λ is the usual regularization parameter and the rest of the notations are same as described before in section 7.3. Each non-zero entry of the coefficient vector β^0 correspond to some time lagged version of a feature from $\{x_1, \dots, x_P\}$, with lags from $\{1, \dots, \ell\}$. Let us call these non zero entries the support of β^0, i.e. $supp(\beta^0) = \{i : \beta_i^0 \neq 0\}$.

In the next step, we increment our maxlag value by ℓ and thus our new estimate for L is $L_1 = 2\ell$. But now while regressing for the target variable (e.g x_i in this case) from $t = 2\ell + 1$ to T, we **do not** take into account the entire past 2ℓ values of all the features. Instead, we do the following :

[1] We will come back to this issue of initializing L_0 to an integer greater than 1, later in this chapter.

(a) We consider all the feature values from the relatively older past i.e. from time $(t-\ell-1)$ to $(t-2\ell)$, and,
(b) From the recent past i.e. time $(t-1)$ to $(t-\ell)$ we consider only those time lagged values of features present in the support of β^0 obtained previously when maxlag was ℓ.

Since we have already seen the recent history of all features (when our maxlag was ℓ), we have the significant values in the support and the remaining $P\ell - \mid supp(\beta^0) \mid$ variables have no influence in determining the present value of x_i. But the older past is still unexplored and so we consider time lagged values of all the features for that time period. This is the intuition behind our adopting the strategy of selective feature pruning. Therefore, our new optimization problem becomes :

$$\beta^1 = \arg\min_{\beta} \quad \frac{1}{2n} \left\| Y - X^{Lagged}\beta \right\|^2 + \lambda \left\|\beta\right\|_1 \tag{8.5}$$

where, the number of samples is $n = T - 2\ell$, $Y = [x_i^{2\ell+1}, \dots, x_i^T]^T$ is a vector of size $n \times 1$ and X^{Lagged} is a $n \times (P\ell + k_0)$ matrix, $k_0 = \mid supp(\beta^0) \mid$. The t^{th} row of X^{Lagged} is the (row) vector $\big[[\tilde{x}_i^j : i \in supp(\beta^0), j \in \{t-1, \dots, t-\ell\}], [x_1^{t-l-1}, \dots, x_1^{t-2\ell}], \dots, [x_P^{t-l-1}, \dots, x_P^{t-2\ell}]\big]$ of size $P\ell + k_0$. The optimization routine returns β^1 whose support ($supp(\beta^1) = \{i : \beta_i^1 \neq 0\}$) corresponds to some time lagged version of features from $\{x_1, \dots, x_P\}$, and lags from $\{1, \dots, 2\ell\}$.

The above procedure is repeated for all subsequent guesses of L until the first k such that $L_{k-1} = k\ell > M$. Finally, we use any of the following measures equivalently for selecting the "best" estimate of L :

1. Akaike Information Criterion (AIC) [Aka98]

2. AIC with bias correction (AICc) [HT89]

3. Mean Squared Error (MSE) ϵ-convergence

Let $\hat{L}$ denote our estimate of maximum time lag parameter L for feature x_i. We choose $\hat{L}$ to be the *smallest* value for which the AIC (or AICc) is within multiplicative ϵ-bound of the *minimum* (ϵ is small, usually 0.01). If such a value does not exist, the Lag corresponding to the *minimum* AIC (or AICc) value is chosen. This is also asymptotically equivalent to choosing the *smallest* value from our lag estimates $\{L_0, L_1, \dots, L_{k-2}\}$ for which the MSE converges to within some ϵ-bound of the *minimum* MSE (ϵ is small, usually 0.01). The support of β corresponding to $\hat{L}$ is used to extract the features causally affecting x_i along with their coefficients (causal strengths) in the VAR model of order $\hat{L}$. The above procedure can be invoked for each feature x_i, $i \in [P]$ independently to determine the output feature causal graph.

Next, we present the pseudo code of *Lasso Granger++* [7], but before that let us familiarize ourselves with some notations and sub-routines that the algorithm uses.

- $\mathbf{u}$ is a vector, i^{th} element is denoted by $\mathbf{u}(i)$
- All entries of $\mathbf{u}$ are denoted by $\mathbf{u}(:)$
- Entries from i^{th} to j^{th} index is denoted by $\mathbf{u}(i:j)$
- $\mathbf{X}$ is a matrix, $(i,j)^{th}$ entry is $\mathbf{X}(i,j)$
- i^{th} column of $\mathbf{X}$ is denoted by $\mathbf{X}(:,i)$
- i^{th} row of $\mathbf{X}$ is denoted by $\mathbf{X}(i,:)$
- *vectorize* : reshapes a $P \times L$ matrix column wise to a row vector of length $P * L$
- If $\mathbf{u}$ and $\mathbf{v}$ are row vectors of length m and n resp., [$\mathbf{u}$,$\mathbf{v}$] appended one after the other is also a row vector of length $m+n$

The algorithm [7] takes as input the following :

- $\mathbf{S} = [x_1, x_2, ..., x_P]$: a $T \times P$ matrix. Each $x_i \in \mathbb{R}^T$ is a time series.
- An upper bound, M, for maxlag. M could be as high as $T-2$.
- An initial estimate of L which we denote by ℓ.

The algorithm [7] outputs :

- A set of features $\{x_j : j \subseteq [P]\}$ which causally influence feature x_i along with their corresponding weights.
- $\hat{L}$, the "best" estimate of L in the VAR model of x_i.

8.1.2 Group Lasso Granger++

The above method 8.1.1, although computationally efficient, ignores the natural *group structure* existing among the temporal lagged variables imposed by the respective time series they belong to. The existence of any non-zero element in the coefficient vector β^i_j is interpreted as "x_j *Granger Causing* x_i". The penalization term, instead of sparsifying the entire vector β individually, should shrink β^i_j as a group. The Group Lasso Granger method [LALR09] overcomes this

Algorithm 7 : Lasso Granger++

```
Initialize $L = \ell$
$suppCols = \phi$
$k = 0$
while $L \leq M$ do
    $\mathbf{X} = \phi$    ▷ $\mathbf{X}$ is a matrix of size $(T - L) \times (P\ell + k)$ , where $k \geq 0$
    $\mathbf{y} = \phi$    ▷ $\mathbf{y}$ is a (column) vector of length $(T - L)$
    if $L == \ell$ then    ▷ Current lag is $\ell$, fit a VAR model as stated in eq. 8.4
        for $t = L + 1$ to $T$ do
            $\mathbf{y}(t - L) = x_i(t)$
            $temp = \mathbf{S}((t - L : t - 1), :)$
            $\mathbf{X}(t - L, :) = vectorize(temp)$
        end for
    else    ▷ Current lag is $> \ell$, fit a VAR model as stated in eq. 8.5
        for $t = L + 1$ to $T$ do
            $\mathbf{y}(t - L) = x_i(t)$
            $pL = L - \ell$
            $u = suppCols(t - pL, :)$
            $temp = \mathbf{S}((t - L : t - pL - 1), :)$
            $v = vectorize(temp)$
            $\mathbf{X}(t - L, :) = [u, v]$
        end for
    end if
    $\beta = LASSO(\mathbf{X}, \mathbf{y})$
    $supp = \{j : \beta(j) \neq 0\}$
    $suppCols = \mathbf{X}(:, supp)$
    $k =| supp |$
    Extract the features corresponding to suppCols and store them
    $L = L + \ell$
end while
Use $\epsilon$-convergence criterion for minimum MSE (or minimum AIC/AICc) to infer $\hat{L}$
Return the set of features corresponding to suppCols of $\hat{L}$
```

* *LASSO* [Tib96] outputs a *sparse* coefficient vector β given X and y such that $y \approx X * \beta$

limitation by leveraging the group structure among the variables according to the time series they belong to. It employs *Group Lasso* optimization [YL06] which alongside minimizing the empirical risk also performs variable selection by penalizing the intra-group and inter-group variable selection differently.

In *Group Lasso Granger++*, our approach stays the same as *Lasso Granger++* except with one change - the optimization routine. Following the same notations from 8.1.1, our first group lasso formulation, when $L = \ell$, is :

$$\beta_G^0 = \arg\min_{\beta} \quad \frac{1}{2} \left\| Y - X^{Lagged} \beta \right\|^2 + \lambda \sum_{j=1}^{P} \sqrt{\rho_j} \left\| \beta_{\mathbb{G}_j} \right\|_2 \tag{8.6}$$

where Y and X^{Lagged} are same as that defined in equation 8.4. $\beta_{\mathbb{G}_j} = \{\beta_k : k \in \mathbb{G}_j\}$ where $\mathbb{G}_j$ denotes the set of group indices and ρ_j accounts for varying group size. Note that there are P groups here and they are of equal size (ℓ) initially.

Similarly the next optimization routine with $L = 2\ell$ is :

$$\beta_G^1 = \arg\min_{\beta} \quad \frac{1}{2} \left\| Y - X^{Lagged} \beta \right\|^2 + \lambda \sum_{j=1}^{P} \sqrt{\rho_j} \left\| \beta_{\mathbb{G}_j} \right\|_2 \tag{8.7}$$

where, the number of samples is $n = T - 2\ell$, $Y = [x_i^{2\ell+1}, \ldots, x_i^T]^T$ is a vector of size $n \times 1$ and X^{Lagged} is a $n \times (P\ell + k_0)$ matrix, $k_0 = \mid supp(\beta_G^0) \mid$. The t^{th} row of X^{Lagged} is the (row) vector $\left[[\tilde{x}_i^j : i \in supp(\beta_G^0), j \in \{t-1, \ldots, t-\ell\}], [x_1^{t-l-1}, \ldots, x_1^{t-2\ell}], \ldots, [x_P^{t-l-1}, \ldots, x_P^{t-2\ell}] \right]$ of size $P\ell + k_0$. Here ρ_j might be different for each group depending on the subset of groups chosen in the previous iteration and is usually set to be the dimension of each group. It is important to observe that the construction of column indices for each group is crucial for all lag estimates greater than L_0. Every row of X^{Lagged} for all $L_i : i > 0$ has two components - (a) time lagged feature values from older past, and, (b) selective feature values from recent past. So the assignment of column indices to every group from both these parts must be taken care of accordingly.

The rest of the algorithm is same as *Lasso Granger++* and we use the same measure(s) to select $\hat{L}$ from $\{L_0, L_1, \ldots, L_{k-2}\}$ where k is the number of guesses for maxlag. Like *Lasso Granger++*, we can invoke the *Group Lasso Granger++* procedure independently for each feature x_i, $i \in [P]$ to determine the output feature causal graph.

Now we present the pseudo code of *Group Lasso Granger++* [8]. We extend the notations and sub-routines used in algorithm 7 and introduce one new notation too.

- $\mathbf{G}$ is vector containing the group indices of all features. For e.g - if there are 10 columns, and they are grouped as $G1$: Columns $1, 2$, $G2$: Columns $3, 4$ and $G3$: Columns 5 though 10. Then $\mathbf{G} = [1, 1, 2, 2, 3, 3, 3, 3, 3, 3]$.

Few crucial aspects :

1. The choice of the regularization parameter λ plays in key role in both Lasso and Group Lasso routines discussed above. We tune λ by varying it in different step sizes within an interval $[0.001, 20]$ and for every choice of lag L_i, we choose that λ for which the AIC (or AICc in case of small samples) is *minimum*. Notice that very high value of λ will result in under-fit since it will over sparsify the model and very small value of λ will unnecessarily over-parameterize the model and nullify the sparsity of the coefficient vector β. Since AIC (or AICc) is the most widely used "Model Selection" criterion and in our case there is in fact a direct correspondence between a choice of λ and the degree of freedom captured by AIC, we appeal to it's use. Later in section 8.3 we will describe more about our model selection criteria.

2. The initial value of maxlag L_0 can be any positive integer greater than 1. But to get the "best" estimate for maxlag, we can adopt the following two pass approach. For example, let $L_0 = 5$. So, in the first pass, our subsequent guesses for L are $L_1 = 10$, $L_2 = 15$, $L_3 = 20$ and so on. Let the maxlag value $\hat{L}$ chosen as per our selection criteria be say 55. We know the true maxlag is either equal to or less than 55. To verify that, we run a second pass of our algorithm but now with the upper bound $M = 55$ and $L_0 = 1$. This will refine our estimate $\hat{L}$ if possible.

8.2 Complexity Analysis

We have already stated our motivation behind choosing a value of maxlag which jointly minimizes the empirical risk and model complexity. With the Least Angle Regression (LARS) [EHJ+04] implementation of LASSO, the time complexity of Lasso Granger, for a given choice of maxlag L, is $\mathcal{O}(n(PL)^2)$, where $n = T - L$ is the number of samples. If instead of doing feature pruning at each estimate, we adopt the brute force approach of simply incrementing L in steps of ℓ and consider the L time-lagged values of all the P features, then the problem will become intractable soon. When the maxlag estimate is $L_{k-1} = k\ell$, the effective number of features in the regression is precisely $PL_{k-1} = Pk\ell$. Under the high dimensional setting (i.e. P is large) and when the "true" maxlag itself is large enough, the product PL_{k-1} will be large.

Algorithm 8 : Group Lasso Granger++

```
Initialize $L = \ell$
$suppCols = \phi$
$k = 0$
while $L \leq M$ do
    $\mathbf{X} = \phi$        ▷ $\mathbf{X}$ is a matrix of size $(T - L) \times (P\ell + k)$ , where $k \geq 0$
    $\mathbf{y} = \phi$        ▷ $\mathbf{y}$ is a (column) vector of length $(T - L)$
    if $L == \ell$ then        ▷ Current lag is $\ell$, fit a VAR model as stated in eq. 8.6
        for $t = L + 1$ to $T$ do
            $\mathbf{y}(t - L) = x_i(t)$
            $temp = \mathbf{S}((t - L : t - 1), :)$
            $\mathbf{X}(t - L, :) = vectorize(temp)$
            $\mathbf{G} \leftarrow$ Get Group Index for all the $P\ell$ features
        end for
    else        ▷ Current lag is $> \ell$, fit a VAR model as stated in eq. 8.7
        for $t = L + 1$ to $T$ do
            $\mathbf{y}(t - L) = x_i(t)$
            $pL = L - \ell$
            $u = suppCols(t - pL, :)$
            $temp = \mathbf{S}((t - L : t - pL - 1), :)$
            $v = vectorize(temp)$
            $\mathbf{X}(t - L, :) = [u, v]$
            $\mathbf{G} \leftarrow$ Get Group Index for the new $P\ell$ and the old $k \geq 0$ features
        end for
    end if
    $\beta = GroupLasso(\mathbf{X}, \mathbf{y}, \mathbf{G})$
    $supp = \{j : \beta(j) \neq 0\}$
    $suppCols = \mathbf{X}(:, supp)$
    $k =\mid supp \mid$
    Extract the features corresponding to $suppCols$ and store them
    $L = L + \ell$
end while
Use $\epsilon$-convergence criterion for minimum MSE (or minimum AIC/AICc) to infer $\hat{L}$
Return the set of features corresponding to $suppCols$ of $\hat{L}$
```

* *GroupLasso* [YL06] : Shooting algorithm for the group lasso in the penalized form

Therefore, the time complexity (τ_b) of this brute force approach (without the feature pruning) will be :

$$\tau_b = \mathcal{O}\Bigg(\sum_{j=1}^{\frac{T-2}{\ell}} n_j (Pj\ell)^2\Bigg), \quad where \quad n_j = T - j\ell \tag{8.8}$$

$$= \mathcal{O}\Bigg(\sum_{j=1}^{\frac{T-2}{\ell}} (T - j\ell) P^2 j^2 \ell^2\Bigg) \tag{8.9}$$

which simplifies to $\mathcal{O}(\frac{P^2 T^4}{\ell})$. Also since the effective number of features in the regression is a measure of the memory space taken by the algorithm, the space complexity of this brute force approach will be $\mathcal{O}\big(\sum_{j=1}^{\frac{T-2}{\ell}} Pj\ell\big)$ which simplifies to $\mathcal{O}(\frac{PT^2}{\ell})$.

But with our method of selective feature pruning, we are able to bring down both the time and space complexity dramatically. With the initial lag estimate $L_0 = \ell$, the running time of Lasso Granger is $\mathcal{O}((T - \ell)(P\ell + s_0)^2)$, with $s_0 = 0$. When the lag estimate is $L_1 = 2\ell$, the running time is $\mathcal{O}((T - 2\ell)(P\ell + s_1)^2)$, where $s_1 =\mid supp(\beta^0) \mid$ is the cardinality of the support of the coefficient vector from previous estimate of L i.e. L_0. Note that $0 \leq s_1 \ll P\ell$ because of the sparsity inducing property of the L_1 norm operator. Thus when the maxlag estimate is L_{j-1}, the running time of Lasso Granger is $\mathcal{O}((T - j\ell)(P\ell + s_{j-1})^2)$, where $s_{j-1} =\mid supp(\beta^{j-2}) \mid$ is the cardinality of the support of the coefficient vector from previous estimate L_{j-2}. Therefore, the overall time complexity (τ) of *Lasso Granger++* is :

$$\tau = \mathcal{O}\Bigg(\sum_{j=1}^{\frac{T-2}{\ell}} (T - j\ell)(P\ell + s_{j-1})^2\Bigg), \quad where \quad 0 \leq s_{j-1} \ll P(j-1)\ell \tag{8.10}$$

$$= \mathcal{O}\Bigg(P^2 T^2 \ell + \max_j \big(\frac{T^2 s_{j-1}^2}{\ell} + PT^2 s_{j-1}\big)\Bigg) \tag{8.11}$$

$$= \mathcal{O}\Bigg(T^2 \max\big(P^2\ell, \frac{s^2}{\ell}, Ps\big)\Bigg), \quad where \quad s = \max_j s_{j-1} \tag{8.12}$$

It is easy to see that each term in the r.h.s of τ (equation 8.10) is lesser than that of τ_b (equation 8.9) and therefore τ as a whole is much smaller than τ_b.

Similarly, the space complexity (κ) of *Lasso Granger++* is :

$$\kappa = \mathcal{O}\Bigg(\sum_{j=1}^{\frac{T-2}{\ell}} (P\ell + s_{j-1})\Bigg), \quad where \quad 0 \leq s_{j-1} \ll P(j-1)\ell \tag{8.13}$$

$$= \mathcal{O}\big(PT + \sum_{j=1}^{\frac{T-2}{\ell}} s_{j-1}\big) \tag{8.14}$$

$$= \mathcal{O}\Bigg(T \max\big(P, \frac{s}{\ell}\big)\Bigg), \quad where \quad s = \max_{j} s_{j-1} \tag{8.15}$$

which is significantly smaller than $\frac{PT^2}{\ell}$. Thus the effective number of features **does not** grow linearly with L and this is what makes our algorithm efficient as well as scalable both in terms of space and time complexity. We demonstrate the space and time complexity savings with our experiments on synthetic datasets.

8.3 Model Selection Criteria

In this section, we describe the different error measures and/or information criteria we use for tuning the regularization parameter λ in both Lasso and Group Lasso routines and also for choosing the "best" estimate for maxlag.

8.3.1 Mean Squared Error

We use the ϵ-convergence Mean Squared Error (MSE) criterion for choosing the best value of maxlag estimate from all our k guesses $\{L_0, \ldots, L_{k-1}\}$. We fix a small value for ϵ (usually 0.001) and choose the smallest value of L from $\{L_0, \ldots, L_{k-1}\}$ for which the corresponding MSE is within $1+\epsilon$ multiplicative bound of the minimum MSE across all the k estimates. MSE is a standard risk function[1] measuring the the quality of an estimator. It is always non-negative, and values closer to zero are better. Consider Y to be the vector of n observed values corresponding to the inputs to the function which generated the predictions and $\hat{Y}$ be the corresponding vector of predictions, then the MSE of the predictor can be estimated by :

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 = \frac{1}{n}\left\|Y - \hat{Y}\right\|^2 \tag{8.16}$$

In our case, the expression for MSE for a particular estimate L_{j-1} is :

$$MSE = \frac{1}{n}\left\|Y - X^{Lagged}\beta\right\|^2 \tag{8.17}$$

where $n = T - j\ell$ denotes the number of samples, Y is a vector of length n containing the target variable and X^{Lagged} is a $n \times (P\ell + s_{j-1})$ matrix, where $s_{j-1} =\mid supp(\beta^{j-2}) \mid \geq 0$ is the cardinality of the support of the coefficient vector from previous estimate L_{j-2}. This MSE

[1] Refer to the wikipedia article on MSE : https://en.wikipedia.org/wiki/Mean_squared_error

ϵ-convergence criterion asymptotically implies choosing a value of L which also corresponds to the "elbow point" in the AIC error curve, the point at which the AIC score (explained later) is minimum. This is true in all those cases where the "true" maxlag is larger than one. We will demonstrate this later with our synthetic experiments.

8.3.2 Akaike Information Criterion

The *Akaike information criterion* (AIC) is a measure of the relative quality of statistical models for a given set of data. Based on the foundations of information theory and parameter estimation, AIC provides a framework that deals with the trade-off between the "goodness of fit" and the "complexity" of the statistical model, both of which are determined from the data. It is one of the most widely used *Model Selection* tools.

Let us consider a statistical model M for some observed data X. Let $\hat{L}$ be the maximum value of the likelihood function for M i.e $\hat{L} = p(X \mid \hat{\theta}, M)$ and k be the number of estimated parameters in the model, then the AIC[1] value of the model M is defined as :

$$AIC(M) = -2 \ln \hat{L} + 2k \tag{8.18}$$

Given a set of candidate models for the data, the preferred model is the one with the *minimum* AIC value. AIC rewards goodness of fit as assessed by the log likelihood function. But it also includes a penalty that is an increasing function of the number of estimated parameters. This is because increasing the number of parameters in the model almost always improves the calculated goodness of the fit but it unnecessarily over parameterizes the model making it complex. The penalty term discourages over-fitting and balances this out.

Let us consider a VAR model of order L being fit to the target variable $X_i = \{x_i^t\}_{t=1}^T$.

$$x_i^t = \sum_{j=1}^{P} \langle \beta_j^i, x_j^{(t,L)} \rangle + \epsilon_i^t \tag{8.19}$$

where $x_j^{(t,L)} = [x_j^{(t-1)}, \ldots, x_j^{(t-L)}]$ is the history of X_j up to time t, $\beta_j^i = [\beta_j^i(1), \ldots, \beta_j^i(L)]$ is the coefficient vector modeling the effect of time series X_j on X_i and ϵ_i is independent additive white noise. Going one level up, the equation 8.19 can be re-written as follows :

$$x_i^t = \langle \beta_{(1:P)}^i, x_{(1:P)}^{(t,L)} \rangle + \epsilon_i^t \tag{8.20}$$

[1] Refer to the wikipedia article on AIC : https://en.wikipedia.org/wiki/Akaike_information_criterion

where $x^{(t,L)}_{(1:P)} = [x^{(t,L)}_1, \ldots, x^{(t,L)}_P]^T = [[x^{(t-1)}_1, \ldots, x^{(t-L)}_1], \ldots, [x^{(t-1)}_P, \ldots, x^{(t-L)}_P]]^T$ and $\beta^i_{(1:P)} = [\beta^i_1, \ldots, \beta^i_P]^T = [[\beta^i_1(1), \ldots, \beta^i_1(L)], \ldots, [\beta^i_P(1), \ldots, \beta^i_P(L)]]^T$ are both vectors of length PL and $\langle \beta^i_{(1:P)}, x^{(t,L)}_{(1:P)} \rangle$ denotes their dot product. If we assume that the residuals ϵ^t_i are distributed according to independent, identical Normal distributions with zero mean and variance $= \sigma^2$, then it implies that, given the entire time observation set for x_i, the distribution of x^t_i is also Gaussian with mean $\mu = \langle \beta^i_{(1:P)}, x^{(t,L)}_{(1:P)} \rangle$ and variance equals σ^2 i.e

$$Pr(x^t_i \mid X, \beta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(- \frac{1}{2\sigma^2}\big(x^t_i - \langle \beta^i_{(1:P)}, x^{(t,L)}_{(1:P)} \rangle\big)^2 \right) \tag{8.21}$$

The log likelihood function $\ln(\hat{L}(\beta))$ under this assumption evaluates to :

$$\ln(\hat{L}(\beta)) = -\frac{n}{2}\ln(2\pi) - n\ln\sigma - \frac{1}{2\sigma^2}\sum_{t=L+1}^{T}\big(x^t_i - \langle \beta^i_{(1:P)}, x^{(t,L)}_{(1:P)} \rangle\big)^2 \tag{8.22}$$

where $n = T - L$ is the number of samples. Therefore, the expression for AIC with k parameters is as follows :

$$AIC = -2\ln(\hat{L}(\beta)) + 2k, \quad where \quad k = 1 + \mid supp(\beta^i_{(1:P)}) \mid \tag{8.23}$$

The penalty term k includes the standard deviation σ of the noise process and the degrees of freedom in the VAR model of order L. Let $\hat{\sigma}^2$ denote the maximum likelihood estimate of σ^2, then differentiating the expression 8.23 w.r.t σ, we get $\hat{\sigma}^2 = \frac{RSS}{n}$, where RSS is the residual sum of squares error expressed as $RSS = \sum_{t=L+1}^{T}\big(x^t_i - \langle \beta^i_{(1:P)}, x^{(t,L)}_{(1:P)} \rangle\big)^2$. Therefore the expression for AIC simplifies to :

$$AIC = n\ln\big(\frac{RSS}{n}\big) + n(\ln(2\pi) + 1) \tag{8.24}$$

8.3.3 Corrected Akaike Information Criterion

In settings where the number of samples n is small and k is comparatively large, $2k$ provides a much smaller adjustment for bias, making AIC a substantially negatively biased estimator. Using AIC, when n is not many times larger than k^2, increases the probability of selecting models that have too many parameters, i.e. of over-fitting. The Corrected Akaike Information Criterion (AICc) [HT89] accounts for this by incorporating a greater penalty for extra parameters. The usual rule of thumb for applying AICc instead of AIC is typically when $n/k < 40$. The expression for AICc is as follows :

$$\begin{aligned} AICc &= -2\ln(\hat{L}(\beta)) + \frac{2kn}{n-k-1} \quad (8.25) \\ &= AIC + \frac{2k(k+1)}{n-k-1} \quad (8.26) \end{aligned}$$

where n denotes the number of samples, and k captures the variance of the estimator (same as stated in equation 8.23). Note that when $n \gg k$, $n - k - 1 \approx n$ and therefore equation 8.25 reduces to equation 8.23, thereby proving that AICc converges to AIC as n gets large.

Chapter 9

Experiments and Results

In this chapter, we present the results of our experimental evaluation on a host of synthetic and real-world datasets. First we present the findings for synthetic experiments where the ground truth is known. We compare the true maxlag for each time series variable in the original model with that of the values predicted by our method - Lasso Granger++ and Group Lasso Granger++. We also compare the similarity between the original feature causal graph and the hypothesis graph that our method outputs, in terms of the evaluation criteria already mentioned in section 6.6. We supplement our synthetic experiments with more experiments on real-world data which includes the gene expression data of the human cancer cell (HeLa S3) cycle and the climate data containing records of several climate enforcing agents across North America.

9.1 Synthetic Data

In this section, we present the consistency and efficiency of our method assuming a *causally sufficient system* [BL13] i.e. a system where no common cause of any two observed variables in the system is left out unobserved. We use VAR model throughout as the generative model for synthetic temporal data and the data generation process we employ is same as already described in section 6.5. The source code for all of our synthetic experiments is available at `https://github.com/MessianNil/GrangerCausality`.

Synthetic Experiment 1 : We start off with the 3 basic building structures in a Granger Causal Network viz. the Co-parent structure, the Collider structure and the Chain structure. Consider three time series variables X, Y and Z which causally affect each other through different path delays (lags) labeled as τ_1 and τ_2 as shown in figure 9.1. In the Co-parent structure (a) Z is the common cause for X and Y, in the Collider structure (b) Z is causally affected by both X and Y, and, the Chain structure (c) where X is the indirect cause of Y

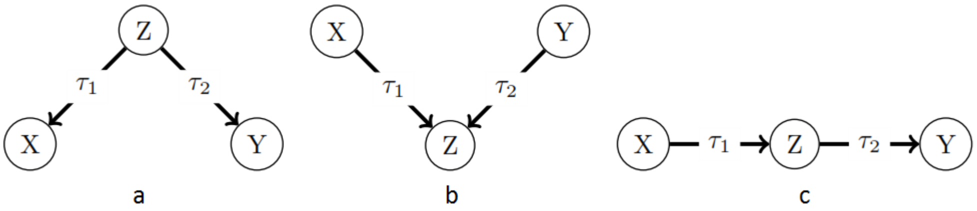


Figure 9.1: Three basic structures in a Granger Causal Network

through Z. The initial distribution of X, Y, Z are Gaussian with mean=0 and variance=1 and the lag values along the edges τ_1, τ_2 are integers chosen uniformly at random from the interval $[1, 10]$. The VAR equations for each of the three cases are as follows :

$$\textbf{(a) Co-Parent structure} \begin{cases} x(t) &= a * z(t-\tau_1) + \eta_1(t) \\ y(t) &= b * z(t-\tau_2) + \eta_2(t) \\ z(t) &= \eta_3(t) \end{cases}$$

$$\textbf{(b) Collider structure} \begin{cases} x(t) &= \eta_1(t) \\ y(t) &= \eta_2(t) \\ z(t) &= a * x(t-\tau_1) + b * y(t-\tau_2) + \eta_3(t) \end{cases}$$

$$\textbf{(c) Chain structure} \begin{cases} x(t) &= \eta_1(t) \\ y(t) &= b * z(t-\tau_2) + \eta_2(t) \\ z(t) &= a * x(t-\tau_1) + \eta_3(t) \end{cases}$$

The coefficients of the VAR model (a, b) are drawn from the uniform distribution $U(0, 1)$ and the additive independent white noise processes η_i's are simulated as $\mathcal{N}(0, \sigma^2)$, where $\sigma = 0.3$. The results of running our method on these three models is summarized in table 9.1. We also included for comparison the standard Lasso Granger and Group Lasso Granger methods with L known and fixed beforehand. While the evaluation scores of standard Lasso Granger method are comparatively lower than our method, the scores of standard Group Lasso Granger is better at least for the collider and chain networks as it should be. The relatively lower values of precision (P) is due to the instances where the choice of lags τ_1 and τ_2 results in violation of the *m-separation criterion* as explained in [BL13]. But our maxlag estimation is highly accurate as shown in the Lag prediction accuracy column. For the Co-parent structure, the maxlag value reported for X and Y are τ_1 and τ_2 respectively. Similarly, for the Collider structure, the maxlag value reported for Z is $\max(\tau_1, \tau_2)$ in almost all of the simulations and

Lasso Granger++				
Structure	P	R	F_1	Lag prediction accuracy
Co-Parent	0.824	1.000	0.884	1.000
Collider	0.764	0.950	0.808	0.967
Chain	0.867	1.000	0.914	1.000
Group Lasso Granger++				
Structure	P	R	F_1	Lag prediction accuracy
Co-Parent	0.884	1.000	0.927	1.000
Collider	0.791	0.950	0.831	1.000
Chain	0.867	1.000	0.914	1.000
Standard Lasso Granger (with fixed L)				
Structure	P	R	F_1	Lag prediction accuracy
Co-Parent	0.727	1.000	0.808	N.A
Collider	0.757	0.950	0.824	N.A
Chain	0.768	1.000	0.854	N.A
Standard Group Lasso Granger (with fixed L)				
Structure	P	R	F_1	Lag prediction accuracy
Co-Parent	0.701	1.000	0.808	N.A
Collider	0.801	0.950	0.841	N.A
Chain	0.934	1.000	0.960	N.A

Table 9.1: Results on Synthetic Experiment 1 (averaged across 10 simulations)

so on for the Chain structure.

Synthetic Experiment 2 : We consider the 3-variable VAR model used in [DCB06]. This model gives rise to spurious causality due to smaller time lags. The VAR equations are :

$$\begin{aligned} x(t) &= 0.8 * x(t-1) - 0.5 * x(t-2) + 0.4 * z(t-1) + \eta_1(t) \\ y(t) &= 0.9 * y(t-1) - 0.8 * y(t-2) + \eta_2(t) \\ z(t) &= 0.5 * z(t-1) - 0.2 * z(t-2) + 0.5 * y(t-1) + \eta_3(t) \end{aligned}$$

where the initial distributions of x, y and z are Gaussian with zero mean and unit variance. The noise variables η_i's are distributed as $\mathcal{N}(0, \sigma^2)$ with $\sigma = 0.3$. Figure 9.2(a) is the ground truth representation of the feature causal graph and parts (b) and (c) highlights the spurious edges in red. The spurious edge from Y to X is detected when the maxlag (model order) is 2, because of causal influence of Y on Z at lag 1 and that of Z to X again at lag 1. Similarly when the model order is 1, the spurious edge from Z to Y is inferred because of the causal edge from y_{t-2} to z_{t-1} as shown in figure 9.3. The performance of our algorithms as compared to the standard (with maxlag known and fixed beforehand) Lasso Granger and Group Lasso

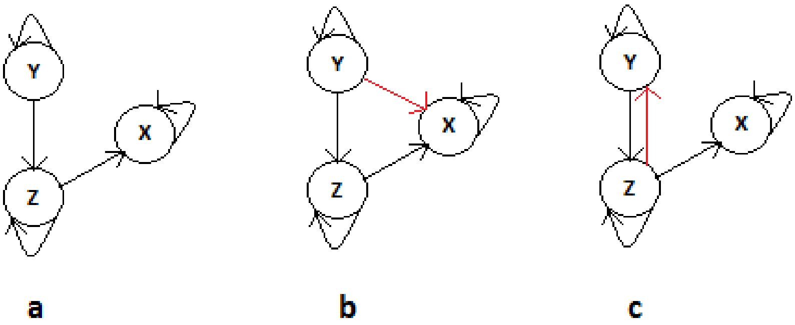


Figure 9.2: Ground Truth and Spurious edges

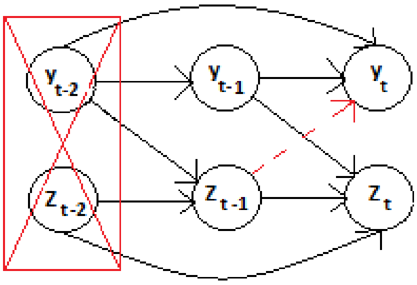


Figure 9.3: Spurious causality when model order is 1 (not high enough)

Granger methods is summarized in table 9.2. In Lasso Granger++, since there is no group-wise penalization, the spurious edges are more frequent there resulting in relatively lower values of precision (P) and F_1 score, but the same is not true in case of Group Lasso Granger++. The maxlag prediction accuracy here is 100% i.e. $\hat{L}_x = \hat{L}_y = \hat{L}_z = 2$ for both the algorithms. Here also the evaluation scores of the standard method is comparatively lower than our method, although the performance of Group Lasso Granger is more or less same as that of ours. Also note that if the lag value specified is incorrect in the standard methods, then they give very wrong inferences.

Methods	P	R	F_1	Lag prediction accuracy
Lasso Granger++	0.675	1.000	0.803	1.000
Group Lasso Granger++	0.921	1.000	0.956	1.000
Standard Lasso Granger (L fixed)	0.648	1.000	0.784	N.A
Standard Group Lasso Granger (L fixed)	0.915	1.000	0.955	N.A

Table 9.2: Results on Synthetic Experiment 2 (averaged across 10 simulations)

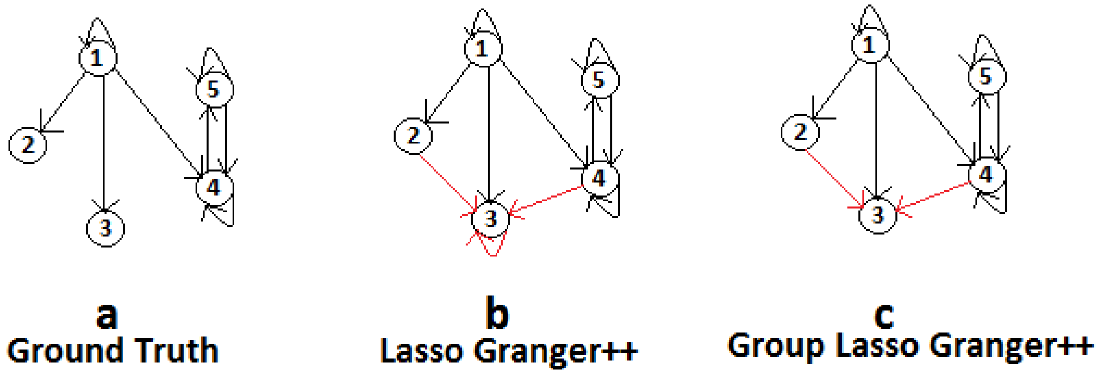


Figure 9.4: Ground Truth and the Hypothesis Causal Graphs inferred by our method

Synthetic Experiment 3 : Consider another five-variable VAR model mentioned in [Set10]. The VAR equations are :

$$\begin{aligned}
x_1(t) &= 0.95 * \sqrt{2} * x_1(t-1) - 0.9025 * x_1(t-2) + w_1(t) \\
x_2(t) &= 0.5 * x_1(t-2) + w_2(t) \\
x_3(t) &= -0.4 * x_1(t-3) + w_3(t) \\
x_4(t) &= -0.5 * x_1(t-2) + 0.25 * \sqrt{2} * x_4(t-1) + 0.25 * \sqrt{2} * x_5(t-1) + w_4(t) \\
x_5(t) &= -0.25 * \sqrt{2} * x_4(t-1) + 0.25 * \sqrt{2} * x_5(t-1) + w_5(t)
\end{aligned}$$

The initial distribution of x_i's are simulated as $\mathcal{N}(0,1)$ and the w_i's are independent Gaussian white noise processes with zero mean and variance σ^2 with $\sigma = 0.3$. Figure 9.4 shows the true feature causal graph, and the output causal graphs inferred by our algorithms Lasso Granger++ and Group Lasso Granger++, for this model. The results of our experimental evaluation on this model as compared to the existing methods is summarized in table 9.3. The maxlag prediction accuracy is 100% again for our algorithms i.e. $\hat{L}_{x_1} = \hat{L}_{x_2} = \hat{L}_{x_4} = 2$, $\hat{L}_{x_3} = 3$ and $\hat{L}_{x_5} = 1$. In figure 9.5, the abscissa shows different guesses of maxlag made by our algorithm and the ordinate is the corresponding AIC score. Parts (a) through (e) are the error curves for

Methods	P	R	F_1	Lag prediction accuracy
Lasso Granger++	0.727	1.000	0.842	1.000
Group Lasso Granger++	0.800	1.000	0.888	1.000
Standard Lasso Granger (L fixed)	0.701	1.000	0.821	N.A
Standard Group Lasso Granger (L fixed)	0.803	1.000	0.889	N.A

Table 9.3: Results on Synthetic Experiment 3 (averaged across 10 simulations)

Group Lasso Granger++ only for each of the time series variables x_1 through x_5. We select $\hat{L}$ for a feature to be the *smallest* value of L for which the AIC is minimum. The selection of $\hat{L}$ is highlighted in figure 9.5 with blue asterisk.

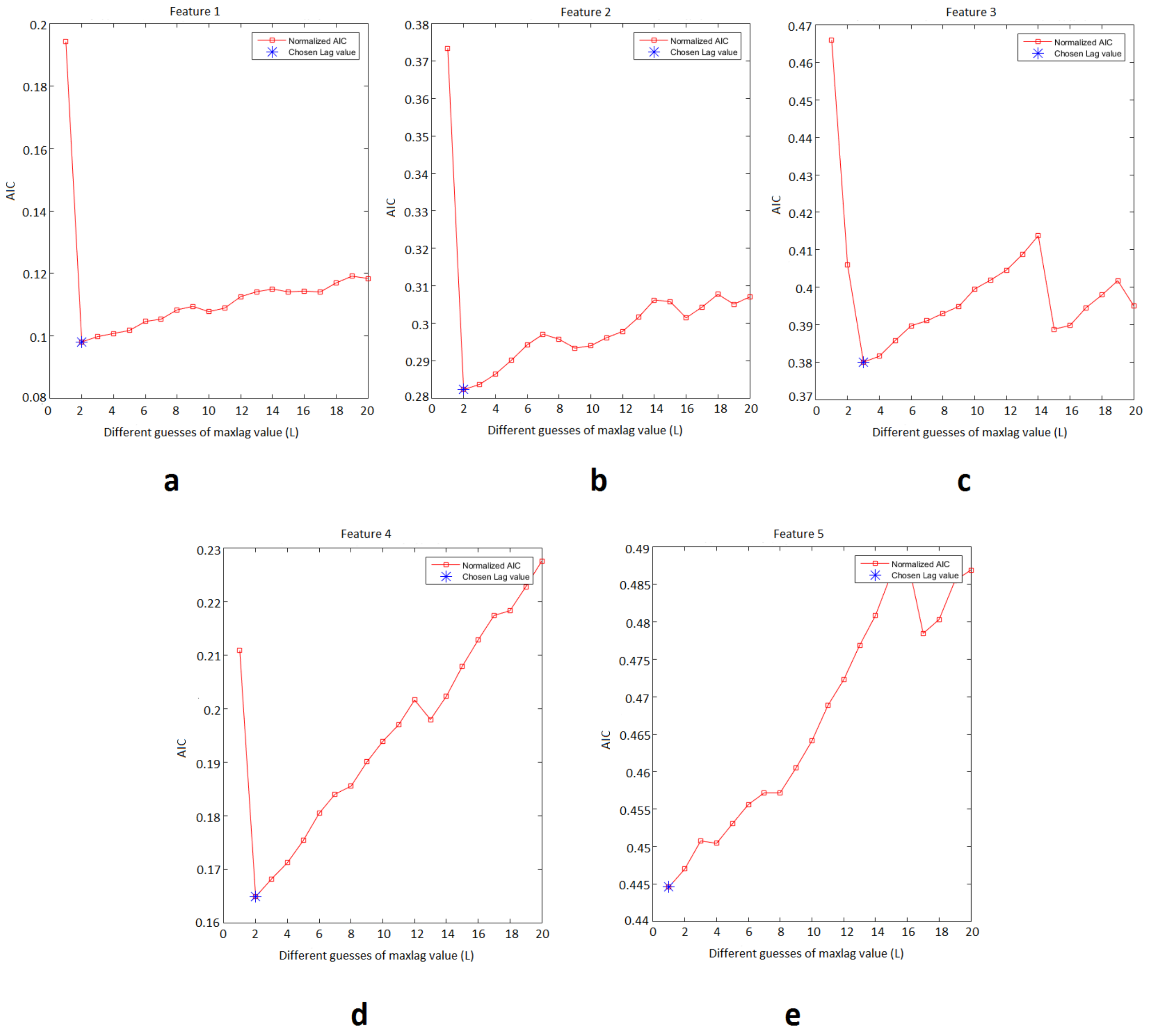
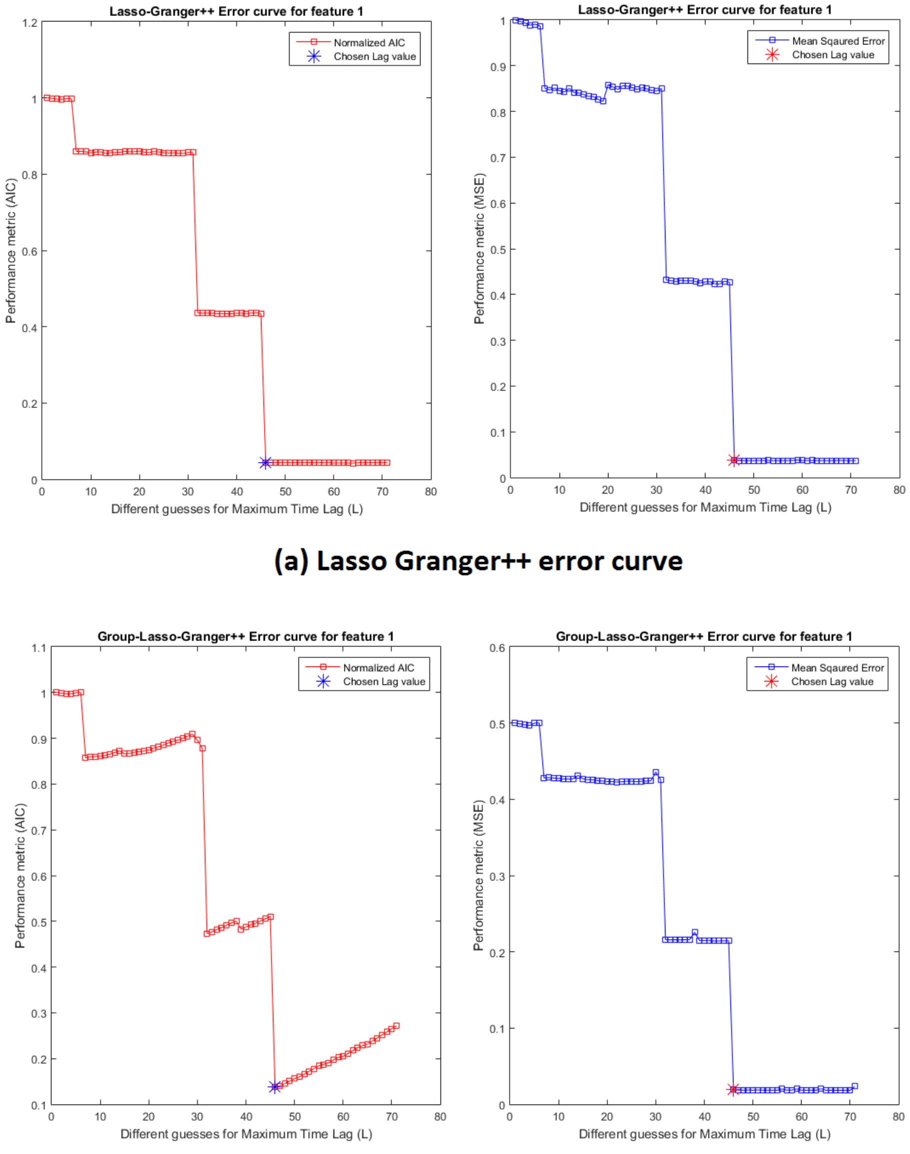


Figure 9.5: AIC error curve and maxlag selection for Group Lasso Granger++ for the 5 time series variables $\{x_1, x_2, x_3, x_4, x_5\}$

Synthetic Experiment 4 : The purpose of this experiment is to demonstrate the discovery of maxlag when the lag values of individual features are significantly different from each other and also the maxlag is somewhat large. Consider a star graph with $P = 5$ time series variables

$\{x_1, \dots, x_5\}$. Features x_2 through x_5 are noise variables simulated from the same distribution $\mathcal{N}(0, 1)$ independently. The only target variable is x_1 which is causally affected by all the other $P-1$ variables. The lag values (path delays) for $\{x_2, x_3, x_4, x_5\}$ are integers drawn uniformly at random from the interval $[1, 50]$, and their coefficients are also drawn from $\mathcal{U}(0, 1)$. The error curve for x_1 for both the algorithms is shown in figure 9.6.



(a) Lasso Granger++ error curve

(b) Group Lasso Granger++ error curve

Figure 9.6: Step wise error curve showing maxlag and the lag values of other causal features

In both parts (a) and (b) of this figure, the curve in red is the AIC curve against different guesses of maxlag and the blue curve is the MSE against different guesses of L. It is clear from this figure that the maxlag selection due to MSE-ϵ convergence criterion and the AIC criterion are asymptotically similar as the number of samples $n = T - L$ grows. In this figure, the lag values for $\{x_2, x_3, x_4, x_5\}$ were chosen randomly to be $\{46, 7, 46, 32\}$ and the maxlag value chosen by both the algorithms is 46 (marked with blue asterisk in the diagram). The stepping nature of the error curve reflects the significant difference in lag values of the individual features causally affecting x_1. The first step is observed at $L = 7$, the second one at $L = 32$ and finally the MSE converges (AIC is minimized) at $L = 46$ which is what we predict as $\hat{L}$. Given this information, the set of features which have causal influences at significantly different time lags, can also be extracted by simply looking up at the set of active features computed by our method at these different lag points ($L = 7$ and $L = 32$ in this case).

Synthetic Experiment 5 : The objective of this experiment is to demonstrate the efficiency, in terms of both space usage and time complexity, of our method (in particular Lasso Granger++) when compared against the brute force approach (without enforcing sparse selection at each step) of Lag prediction and the normal Lasso Granger with L fixed a priori. Figure 9.7 demonstrates the running time comparison of these algorithms. Of course, standard Lasso Granger (the curve in black) performs better than the other two algorithms since L is provided to it. But we claim that the running time behavior of our algorithm Lasso Granger++ (the red curve) compares favorably to it and is much superior to the brute force Lasso Granger (the blue curve) approach.

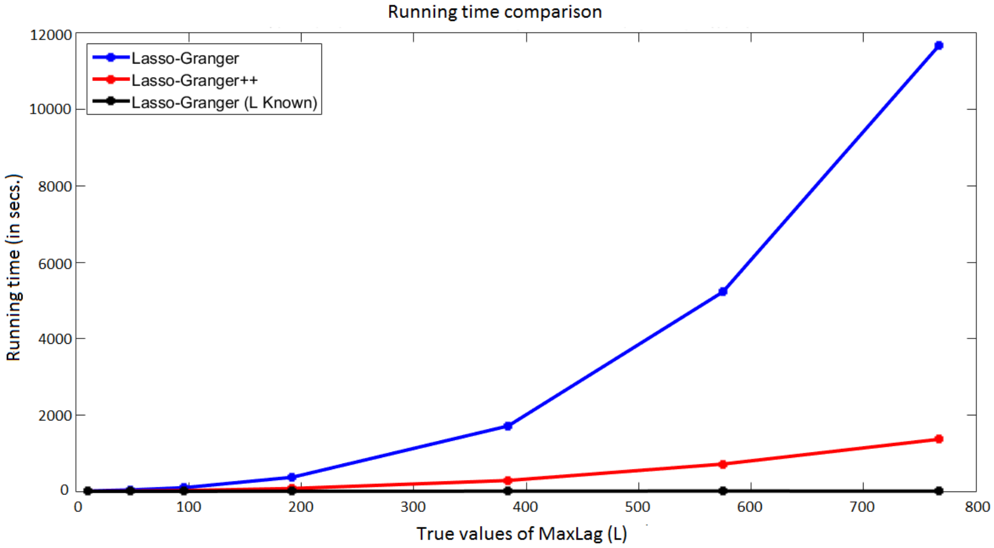


Figure 9.7: Running time as a function of maxlag (L)

Figure 9.8 demonstrates the efficiency of Lasso Granger++ with respect to the amount of memory space required as a function of maximum lag. Brute force Lasso Granger (the blue curve) shows a linear growth rate as a function of L since the effective number of features for a particular choice of L is PL whereas Lasso Granger++ (the red curve) shows a sub-linear growth rate since the effective number of features is much smaller than PL. We emphasize that if the number of features is in the thousands and the maximum lag is reasonably large, our algorithm still can be run on a standard laptop whereas this is not possible for the brute force Lasso Granger algorithm because of time and space constraints.

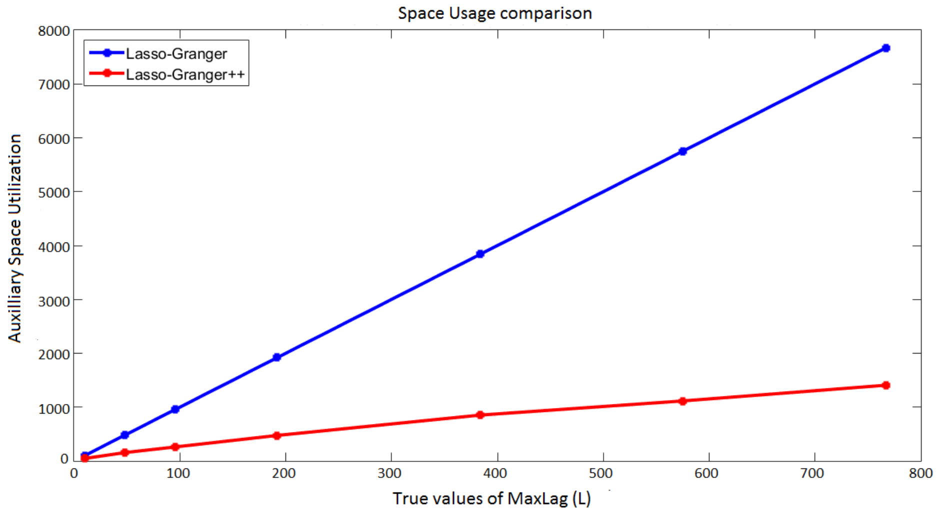


Figure 9.8: Space usage as a function of maxlag (L)

9.2 Real World Data

Experiment 1 : We applied our concurrent estimation method to the gene expression data of the human cancer cell cycle (HeLa S3) [WSS+02]. The data is public and available at http://genome-www.stanford.edu/Human-CellCycle/Hela/. We focus on the first four experiments having 12, 27, 48 and 19 data points resp. recording the expression levels of a handful of pre-identified genes well-studied by Whitfield et. al. [WSS+02] and Sambo et. al. [SDCT08]. For the first three experiments, cell synchronization is achieved by a double thymidine block, whereas for experiment 4 the same is done using a Thymidine-Nocodazole block. For all the experiments, we follow the same data pre-processing steps as detailed in [LALR09] (sec. 4). We cannot evaluate the performance of our method by comparing the "discovered" feature causal graph to the "true" graph since the latter is simply not known. Hence we focus on the particular subset of 9 genes as selected in [SDCT08] and compare the discovered gene-to-gene interactions to those reported in the de-facto BioGRID[1] database. However, it is important to note that the list of interactions reported in BioGRID is far from being exhaustive. Also some of the interactions may not be direct causal influences. So caution must be taken when interpreting the results of such comparison : *false positives* in the output causal network with respect to BioGRID may not necessarily be *false*, and may contain actual links that are unknown till date, or known but have not yet been incorporated into the database. These 9 genes are : (A) CCNA2, (B) E2F1, (C) CDC6, (D) CDC2, (E) CCNB1, (F) PCNA, (G) CCNE1, (H) RFC4 and (I) CDKN3. They are labeled using alphabets A to I for convenience. Figure 9.9 is the known network of interactions among these 9 genes as reported in the BioGRID database. We applied our method on data from both experiment 3 and 4. In experiment 3, the measurements are taken 1 hour apart, whereas for experiment 4 each observation is taken 2 hours apart. The networks discovered by our method, on data from experiment 3, in comparison with the known results, is shown in figure 9.10. The evaluation scores are listed in table 9.4.

Methods	P	R	F_1 score
Lasso Granger++	0.462	0.571	0.511
Group Lasso Granger++	0.435	0.476	0.455
Sambo et. al. [SDCT08]	0.360	0.440	0.400
Group Lasso Granger ($L = 4$ fixed) [LALR09]	0.500	0.720	0.590

Table 9.4: Results on gene data from experiment 3 [WSS+02]

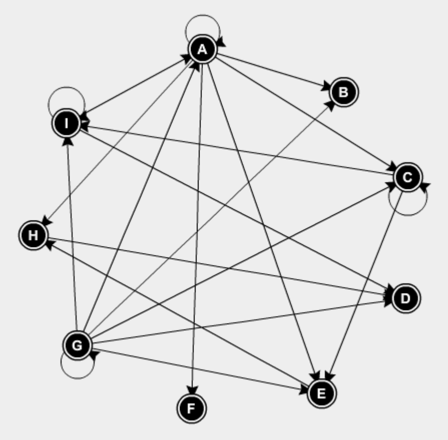


Figure 9.9: Network from the BioGRID database

[1] https://thebiogrid.org/

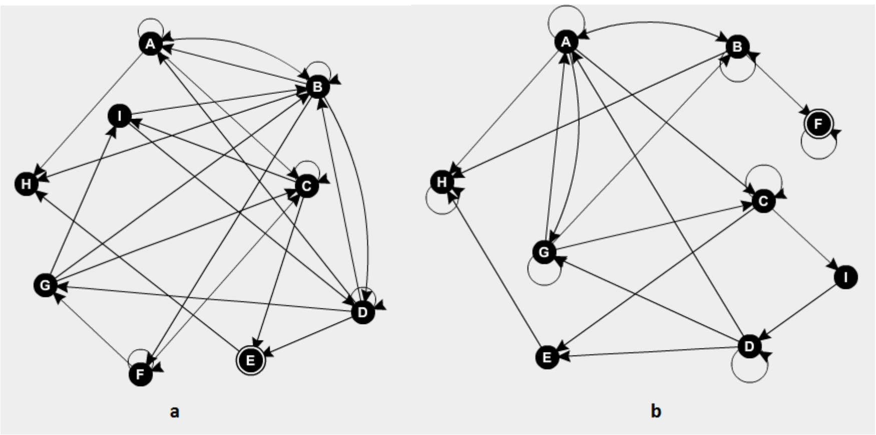


Figure 9.10: Networks discovered by our method - (a) Lasso Granger++ and (b) Group Lasso Granger++

For the extra edges reported by our method, we indeed found in literature [WSS+02, LALR09] that most of these genes are "regulated" genes (i.e. in-degree is high) except CCNE1 which is a "regulating" gene (i.e. out-degree is high) and E2F1 which is a "traffic/hub" gene. Also the maxlag values reported for this set of 9 genes are significantly large. For e.g. the maxlag values estimated (based on minimum AIC score) by our method using data from experiment 4 is shown in table 9.5.

Gene Numbers	A	B	C	D	E	F	G	H	I
$\hat{L}$ value	8	9	9	9	8	9	10	9	9

Table 9.5: Maxlag predicted for each gene (experiment 4 [WSS+02])

Experiment 2 : Climate change is one of the most critical issues today. An important challenge in understanding climate change is to uncover the causal relationships among the different climate enforcing agents which can be either natural (e.g. Solar radiation agents (SOL)) or human-impacted (e.g. green house gases such as CO_2, CH_4) etc. We use measurements of around 15 different climate enforcing variables, recorded *once* every month from the year 1990 till 2002, across 125 different regions in North America. The regions were chosen along a 2.5 × 2.5 spatial grid from latitude range [30.475, 47.975] and longitude range [-119.75, -82.25] as shown in figure 9.11. The data was collected from different sources such as NASA, CRU, NOAA, NCDC and processed as detailed in [LLNM+09] (sec 3.1). The aggregated and processed data is public and available at http://www-bcf.usc.edu/~liu32/data.html. The variables are listed in table 9.6. Some of them are grouped based on the category they belong to. For e.g - the first group (B, C, D, E) fall under the category of green house gases and the second group (A, F, G, H, I, J) represents the climate type.

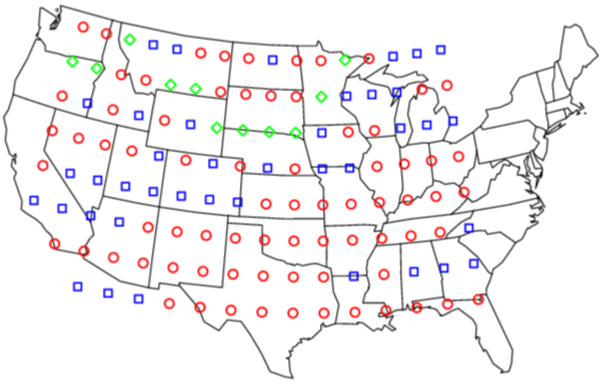


Figure 9.11: Climate data recorded across 125 regions of North America [LNMLL10]

Variable index	Variable Name
(B) CO_2	Carbon-Dioxide
(C) CH_4	Methane
(D) CO	Carbon-Monoxide
(E) H_2	Hydrogen
(A) TMP	Temperature
(F) WET	Wet Days
(G) CLD	Cloud Cover
(H) VAP	Vapor
(I) PRE	Precipitation
(J) FRS	Frost Days
(K) $UV(AER)$	Aerosol Index
(L) SOL	Solar Radiation agents

Table 9.6: Climate agents recorded across 125 regions of North America

We considered temperature (TMP) as the target variable for granger causal modeling since it is perhaps the most important factor related to global warming. Unlike Lozano et. al.'s "Group Elastic Net" formulation [LLNM+09], we ignored spatial lag and spatial penalizations. Instead we assume that a measurement at any particular location is only affected by past values of variables at that same location. We further assume that the climate temporal graph is time invariant but varies across locations. The time-invariant assumption may not be true over a long time period (e.g millions of years), but is a reasonable assumption for data in this case which spans across 13 years. The causal graph reported by our algorithm (*Lasso Granger++*)

is shown in figure 9.12. The vertices correspond to the climate enforcing variables and the directed edges capture causality. Each edge is labeled with a real number which represents the weight (causal strength) of the corresponding variable in influencing the target variable TMP. This weight is averaged over results from all the 125 regions.

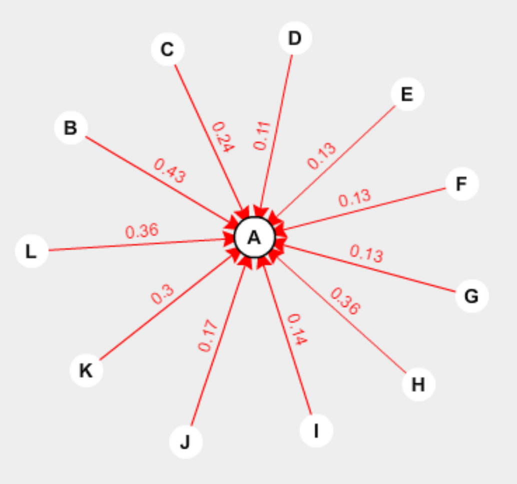


Figure 9.12: Output causal graph attributing changes in temperature

The causal graphs vary from location to location (confirms our assumption), probably due to the insufficient number of observations at each location. However, this seems reasonable by looking at the map. The green states (figure 9.11) correspond to the northern part of the country where the region is cold and hence temperature is more likely to be affected by the number of frost days, or cloud cover. The red circled states (figure 9.11) represent the most developed parts of the US where the green house gases (especially CO_2, CH_4) concentration is high due to more population and industrialization. But variables whose impact is common among all these regions are the solar radiation agents (SOL) and UV (AER). This is in accordance with the results reported in [LLNM$^+$09, LNMLL10]. The estimated maximum time lag reported by us was roughly 12 years. Getting a good estimate of the time lag associated with climatological effects is an important question in its own right, and we intend to target this problem using larger data sets in future work.

Chapter 10

Conclusion

The thesis contains description of the two problems we have studied, namely : (a) Feature Selection under multicollinearity, and (b) Temporal Lag estimation for Granger Causality on time series data.

In the first part, we considered the problem of feature selection in a high-dimensional linear model with strongly correlated predictors. In such situations, we have seen that determining the right subset of features is indeed difficult for any feature selection algorithm. We proposed to address this problem by : (i) Clustering the variables (features) first based on empirical correlation, then, (ii) Perform bootstrap enhanced projected gradient descent with hard-thresholding for supervised selection of appropriate clusters, and, (iii) finally another iterative hard-thresholding to eliminate noisy variables from the selected clusters thereof. Regarding the clustering step, we used agglomerative hierarchical clustering with complete linkage measure and sample correlation as the distance metric. But if preferred, it can also be replaced by another suitable clustering technique as long as it maintains the objective : intra-cluster correlation should be much higher than inter-cluster correlation. We presented empirical results which showcase that BoPGD is indeed an attractive choice for feature selection especially when it comes to the multi-collinear setting. We have also worked with proprietary datasets from Shell Labs, Bangalore. BoPGD has been applied successfully in the context of virtual high throughput screening for materials on a test case of descriptor selection in catalysts for CO2 electro-reduction, from a published database of 298 catalyst alloys. Our method has been found to demonstrate significant advantages over commonly used machine learning tools such as ANN and LASSO based methods as it provides higher stability in sparse estimation. Our results indicate that in addition to d-band characteristics, there is at least one other descriptor such as the Work Function and the Atomic Radius that can describe the CO binding energy on catalyst surfaces. The trends and fingerprint descriptors predicted by our method are found to have a strong chemical validation based on d-band theory and its extension. The article has been re-submitted to *The Journal of Chemistry of Materials, 2017*[1] after a minor review. Future efforts will be directed towards giving theoretical (provable) guarantees of BoPGD under different settings.

In the second part, we focused on the problem of “Granger Causal Inference” on *multivariate, high-dimensional* time series data. We followed the standard approach of modeling the time series as a vector autoregressive (VAR) process. But unlike existing methods, which fix a value of the maximum time lag (also called the model order of VAR), denoted by L, a priori, we didn’t. Instead we proposed and evaluated a pure data-driven and computationally efficient method which concurrently estimates the value of max lag while modeling Granger Causality on time series data by balancing the trade-off between “goodness-of-fit” and “model complexity”. Our empirical evaluations on both synthetic as well as real-world data have demonstrated that our proposed method has high predictive accuracy and low space requirements. We have also worked with another proprietary dataset from Shell, Bangalore in modeling the causal relationships of different process parameters involved in a chemical reactor such as input compositions, input and output flows, inlet and outlet pressure, temperature at various positions of the reactor etc. The Shell dataset had nearly 60000 data points with around 300 monitored process parameters each. Also the time interval and the number of monitored parameters were sufficiently diverse and large to capture any cause-effect relationship. Our results there too, is very promising and insightful, as validated with the subject matter experts. However, we could not put them down in the thesis because of some privacy and Non-disclosure agreement issues with Shell. In future, efforts will be made to further investigate and explore the range of real-world problems where our proposed method could add value.

[1] http://pubs.acs.org/journal/cmatex

Bibliography

[Aka69] Hirotugu Akaike. Fitting autoregressive models for prediction. *Annals of the institute of Statistical Mathematics*, 21(1):243–247, 1969. 55

[Aka98] Hirotogu Akaike. Information theory and an extension of the maximum likelihood principle. In *Selected Papers of Hirotugu Akaike*, pages 199–213. Springer, 1998. 14, 55, 58

[ALA07] Andrew Arnold, Yan Liu, and Naoki Abe. Temporal causal modeling with graphical granger methods. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 66–75. ACM, 2007. 5, 50, 52, 53

[Arn99] Frank Arntzenius. Reichenbach's common cause principle. 1999. 4

[BA03] Kenneth P Burnham and David Anderson. Model selection and multi-model inference. *A Pratical informatio-theoric approch. Sringer*, 2003. 5

[Bac08] Francis R Bach. Bolasso: model consistent lasso estimation through the bootstrap. In *Proceedings of the 25th international conference on Machine learning*, pages 33–40. ACM, 2008. 3, 18

[Bal06] David J Balding. A tutorial on statistical methods for population association studies. *Nature Reviews Genetics*, 7(10):781–791, 2006. 1

[BD09] Thomas Blumensath and Mike E Davies. Iterative hard thresholding for compressed sensing. *Applied and computational harmonic analysis*, 27(3):265–274, 2009. 23

[BJRL15] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015. 4

[BL13] Mohammad Taha Bahadori and Yan Liu. An examination of practical granger causality inference. In *Proceedings of the 2013 SIAM International Conference on data Mining*, pages 467–475. SIAM, 2013. 69, 70

[BR08] Howard D Bondell and Brian J Reich. Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with oscar. *Biometrics*, 64(1):115–123, 2008. 3, 19

[BRvdGZ13] Peter Bühlmann, Philipp Rütimann, Sara van de Geer, and Cun-Hui Zhang. Correlated variables in regression: clustering and sparse estimation. *Journal of Statistical Planning and Inference*, 143(11):1835–1858, 2013. 3, 20, 33

[BT09] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009. 2

[BVDG11] Peter Bühlmann and Sara Van De Geer. *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media, 2011. 27

[CG06] T Chu and C Glymour. Semi-parametric causal inference for nonlinear time series data. *J. of Machine Learning Res., submitted*, 2006. 46

[DCB06] Mingzhou Ding, Yonghong Chen, and Steven L Bressler. 17 granger causality: basic theory and application to neuroscience. *Handbook of time series analysis: recent theoretical developments and applications*, 437, 2006. 71

[Din14] Shutong Ding. Bayesian var models with asymmetric lags. 2014. 55

[DM09] Wei Dai and Olgica Milenkovic. Subspace pursuit for compressive sensing signal reconstruction. *IEEE Transactions on Information Theory*, 55(5):2230–2249, 2009. 23

[DS06] David Donoho and Victoria Stodden. Breakdown point of model selection when the number of variables exceeds the number of observations. In *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, pages 1916–1921. IEEE, 2006. 17

[EHJ$^+$04] Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004. 16, 50, 62

[ET94] Bradley Efron and Robert J Tibshirani. *An introduction to the bootstrap*. CRC press, 1994. 3, 19, 29

[Fou11] Simon Foucart. Hard thresholding pursuit: an algorithm for compressive sensing. *SIAM Journal on Numerical Analysis*, 49(6):2543–2563, 2011. 23

[FSGM$^+$07] André Fujita, João R Sato, Humberto M Garay-Malpartida, Rui Yamaguchi, Satoru Miyano, Mari C Sogayar, and Carlos E Ferreira. Modeling gene expression regulatory networks with the sparse vector autoregressive model. *BMC Systems Biology*, 1(1):1, 2007. 50

[FW74] George M Furnival and Robert W Wilson. Regressions by leaps and bounds. *Technometrics*, 16(4):499–511, 1974. 14

[GHW79] Gene H Golub, Michael Heath, and Grace Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223, 1979. 5

[GJY11] Dongdong Ge, Xiaoye Jiang, and Yinyu Ye. A note on the complexity of l p minimization. *Mathematical programming*, 129(2):285–299, 2011. 10, 23

[GK99] Mikael Gredenhoff and Sune Karlsson. Lag-length selection in var-models using equal and unequal lag-length procedures. *Computational Statistics*, 14(2):171–187, 1999. 55

[GK09] Rahul Garg and Rohit Khandekar. Gradient descent with sparsification: an iterative algorithm for sparse recovery with restricted isometry property. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 337–344. ACM, 2009. 23

[GMS08] Philip E Gill, Walter Murray, and Michael A Saunders. Users guide for sqopt version 7: Software for large-scale linear and quadratic programming. 2008. 20

[Gra69] Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society*, pages 424–438, 1969. 4, 43

[Gra80] Clive WJ Granger. Testing for causality: a personal viewpoint. *Journal of Economic Dynamics and control*, 2:329–352, 1980. 4, 43

[Hec98] David Heckerman. A tutorial on learning with bayesian networks. In *Learning in graphical models*, pages 301–354. Springer, 1998. 4, 42

[HK70] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970. 16

[Hsi81] Cheng Hsiao. Autoregressive modelling and money-income causality detection. *Journal of Monetary economics*, 7(1):85–106, 1981. 55

[HSK06] Patrik O Hoyer, Shohei Shimizu, and Antti J Kerminen. Estimation of linear, non-gaussian causal models in the presence of confounding latent variables. *arXiv preprint cs/0603038*, 2006. 46

[HT89] Clifford M Hurvich and Chih-Ling Tsai. Regression and time series model selection in small samples. *Biometrika*, 76(2):297–307, 1989. 58, 67

[JJR11] Ali Jalali, Christopher C Johnson, and Pradeep K Ravikumar. On learning discrete graphical models using greedy methods. In *Advances in Neural Information Processing Systems*, pages 1935–1943, 2011. 23

[JTK14] Prateek Jain, Ambuj Tewari, and Purushottam Kar. On iterative hard thresholding methods for high-dimensional m-estimation. In *Advances in Neural Information Processing Systems*, pages 685–693, 2014. 3, 10, 24, 25

[K$^+$95] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145, 1995. 5

[Kea01] John W Keating. Macroeconomic modeling with asymmetric vector autoregressions. *Journal of Macroeconomics*, 22(1):1–28, 2001. 55

[LALR09] Aurélie C Lozano, Naoki Abe, Yan Liu, and Saharon Rosset. Grouped graphical granger modeling for gene expression regulatory networks discovery. *Bioinformatics*, 25(12):i110–i118, 2009. 5, 51, 59, 77, 78, 79

[Lau96] Steffen L Lauritzen. *Graphical models*, volume 17. Clarendon Press, 1996. 53

[Lic13] M. Lichman. UCI machine learning repository, 2013. 37

[LLNM$^+$09] Aurelie C Lozano, Hongfei Li, Alexandru Niculescu-Mizil, Yan Liu, Claudia Perlich, Jonathan Hosking, and Naoki Abe. Spatial-temporal causal modeling for climate change attribution. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 587–596. ACM, 2009. 5, 79, 80, 81

[LNMLL10] Yan Liu, Alexandru Niculescu-Mizil, Aurelie C Lozano, and Yong Lu. Learning temporal causal graphs for relational time-series analysis. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 687–694, 2010. ix, 80, 81

[Lüt05] Helmut Lütkepohl. *New introduction to multiple time series analysis*. Springer Science & Business Media, 2005. 5, 54, 55

[Lüt11] Helmut Lütkepohl. *Vector autoregressive models*. Springer, 2011. 5

[LYF14] Ji Liu, Jieping Ye, and Ryohei Fujimaki. Forward-backward greedy algorithms for general convex smooth functions over a cardinality constraint. In *ICML*, pages 503–511, 2014. 23

[Mat00] Robert Matthews. Storks deliver babies (p= 0.008). *Teaching Statistics*, 22(2):36–38, 2000. 4

[MB06] Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the lasso. *The annals of statistics*, pages 1436–1462, 2006. 2, 5, 17, 52, 53

[Mee97] Christopher Meek. *Graphical Models: Selecting causal and statistical models*. PhD thesis, PhD thesis, Carnegie Mellon University, 1997. 4, 42

[Mei07] Nicolai Meinshausen. Relaxed lasso. *Computational Statistics & Data Analysis*, 52(1):374–393, 2007. 2, 17

[MPS08] Daniele Marinazzo, Mario Pellicoro, and Sebastiano Stramaglia. Kernel-granger causality and the analysis of dynamical networks. *Physical review E*, 77(5):056215, 2008. 45

[MS06] Alessio Moneta and Peter Spirtes. Graphical models for the identification of causal structures in multivariate time series models. In *JCIS*, 2006. 4, 42

[MY09] Nicolai Meinshausen and Bin Yu. Lasso-type recovery of sparse representations for high-dimensional data. *The Annals of Statistics*, pages 246–270, 2009. 17

[NT09] Deanna Needell and Joel A Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3):301–321, 2009. 23

[NYWR09] Sahand Negahban, Bin Yu, Martin J Wainwright, and Pradeep K Ravikumar. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. In *Advances in Neural Information Processing Systems*, pages 1348–1356, 2009. 23

[OE94] Patrick Onghena and Eugene S Edgington. Randomization tests for restricted alternating treatments designs. *Behaviour research and therapy*, 32(7):783–786, 1994. 4

[Pea09] Judea Pearl. *Causality*. Cambridge university press, 2009. 4

[RG99] Sam Roweis and Zoubin Ghahramani. A unifying review of linear gaussian models. *Neural computation*, 11(2):305–345, 1999. 46

[S$^+$78] Gideon Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978. 14, 55

[SDC03] Mark R Segal, Kam D Dahlquist, and Bruce R Conklin. Regression approaches for microarray data analysis. *Journal of Computational Biology*, 10(6):961–980, 2003. 1

[SDCT08] Francesco Sambo, Barbara Di Camillo, and Gianna Toffolo. Cnet: an algorithm for reverse engineering of causal gene networks. *NETTAB2008, Varenna, Italy*, 2008. 77, 78

[Set10] Anil K Seth. A matlab toolbox for granger causal connectivity analysis. *Journal of neuroscience methods*, 186(2):262–273, 2010. 73

[SGS00] Peter Spirtes, Clark N Glymour, and Richard Scheines. *Causation, prediction, and search*. MIT press, 2000. 4

[She08] Yiyuan She. *Sparse regression with exact clustering*. ProQuest, 2008. 3

[SSGS06] Ricardo Silva, Richard Scheine, Clark Glymour, and Peter Spirtes. Learning the structure of linear latent variable models. *Journal of Machine Learning Research*, 7(Feb):191–246, 2006. 46

[SSSZ10] Shai Shalev-Shwartz, Nathan Srebro, and Tong Zhang. Trading accuracy for sparsity in optimization problems with sparsity constraints. *SIAM Journal on Optimization*, 20(6):2807–2832, 2010. 23

[Tib96] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996. 2, 5, 16, 20, 50, 60

[VSSBLC$^+$05] Pedro A Valdés-Sosa, Jose M Sánchez-Bornot, Agustín Lage-Castellanos, Mayrim Vega-Hernández, Jorge Bosch-Bayard, Lester Melie-García, and Erick Canales-Rodríguez. Estimating brain functional connectivity with sparse multivariate autoregression. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 360(1457):969–981, 2005. 50

[Wei94] William Wu-Shyong Wei. *Time series analysis*. Addison-Wesley publ Reading, 1994. 5

[WSS$^+$02] Michael L Whitfield, Gavin Sherlock, Alok J Saldanha, John I Murray, Catherine A Ball, Karen E Alexander, John C Matese, Charles M Perou, Myra M Hurt, Patrick O Brown, et al. Identification of genes periodically expressed in the human cell cycle and their expression in tumors. *Molecular biology of the cell*, 13(6):1977–2000, 2002. x, 77, 78, 79

[YL06] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006. 20, 51, 61, 63

[ZH05] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005. 3, 18

[Zou06] Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429, 2006. 2, 17

[ZY06] Peng Zhao and Bin Yu. On model selection consistency of lasso. *Journal of Machine learning research*, 7(Nov):2541–2563, 2006. 2, 17