Learning from Structured Data with High Dimensional Structured Input and Output Domain

By

Hongliang Fei

Submitted to the graduate degree program in Electrical Engineering & Computer Science
and the Graduate Faculty of the University of Kansas
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Jun Huan, Chairperson

Brian Potetz

Committee members

Bo Luo

Arvin Agah

Hongguo Xu

Date defended: _____

The Dissertation Committee for Hongliang Fei certifies
that this is the approved version of the following dissertation :

Learning from Structured Data with High Dimensional Structured Input and Output Domain

Jun Huan, Chairperson

Date approved: _____

Abstract

Structured data is accumulated rapidly in many applications, e.g. Bioinformatics, Cheminformatics, social network analysis, natural language processing and text mining. Designing and analyzing algorithms for handling these large collections of structured data has received significant interests in data mining and machine learning communities, both in the input and output domain.

However, it is nontrivial to adopt traditional machine learning algorithms, e.g. SVM, linear regression to structured data. For one thing, the structure information in the input domain and output domain is ignored if applying the normal algorithms to structured data. For another, the major challenge in learning from many high-dimensional structured data is that input/output domain can contain tens of thousands even larger number of features and labels. With the high dimensional structured input space and/or structured output space, learning a low dimensional and consistent structured predictive function is important for both robustness and interpretability of the model.

In this dissertation, we will present a few machine learning models that learn from the data with structured input features and structured output tasks. For learning from the data with structured input features, I have developed structured sparse boosting for graph classification, structured joint sparse PCA for anomaly detection and localization. Besides learning from structured input, I also investigated the interplay between structured input and output under the context of multi-task learning. In particular, I designed a multi-task learning algorithms that performs structured feature selection & task relationship Inference. We

will demonstrate the applications of these structured models on subgraph based graph classification, networked data stream anomaly detection/localization, multiple cancer type prediction, neuron activity prediction and social behavior prediction. Finally, through my intern work at IBM T.J. Watson Research, I will demonstrate how to leverage structural information from mobile data (e.g. call detail record and GPS data) to derive important places from people's daily life for transit optimization and urban planning.

Acknowledgements

I would like to express my gratitude to these faculties, colleagues and my families for their support and assistance with my dissertation.

First, I sincerely thank Dr. Jun Huan for his guidance during my graduate studies. His mentorship was great and consistent my long-term career goals. He trained me to become an independent researcher with critical thinking rather than a programmer. In reviewing my paper drafts, he offered valuable comments and insights on problem formalization and professional writing. Furthermore, he provided me with financial support for my education and conference travel. He has always taken time to introduce me to people within the discipline. I learned a lot from him, not only research skills but also his dedication towards research. I also gratefully thank my dissertation committee members: Dr. Brian Potetz, Dr. Bo Luo, Dr. Arvin Argh and Dr. Hongguo Xu. They offered professional and valuable suggestions on my proposal and dissertation.

Second, I would also like to thank all my colleagues in our research group, especially Aaron Smalter, Brian Quanz and Ruoyi Jiang. These three friends and co-workers helped a lot with my research. Aaron's code on chemical data processing and learning initialized my work in Chapter 3. Brian's broad knowledge on machine learning and optimization further helped me to dig deeper in my discipline through our discussion. Ruoyi Jiang's creative thinking and experimental skills contributed most in Chapter 4 of my dissertation.

Third, I want to thank Dr. Ming Li, Dr. Sambit Sahu and Dr. Millind Naphade from IBM T.J. Watson Research. I appreciate their offer with a platform where I

can utilize my domain knowledge in industrial problems. They gave their insights on mobility data mining and collected data sets from clients, which contributed the model and experimental design in Chapter 7. Without their help, I could not have a fulfilling and productive internship.

Finally but the most importantly, I want to thank my parents Haixue Fei, Qingfen Meng and two elder sisters Hongmei Fei, Hongying Fei. My parents led a very simple life to support my education from elementary school to college. Their unselfish love makes me strong and warm even I stayed tens of thousands of miles away from them. My two sisters, Hongmei Fei and Hongying Fei, inspired me to have ambition and provided me with endless encouragement and support in my research and study.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Structured data refers to the data that has both information contents and the organization of contents. Such data is accumulated rapidly in many applications, e.g. Bioinformatics [106, 109], Cheminformatics [154], social network analysis [30, 140], natural language processing [48, 167] and text mining [64]. Designing and analyzing algorithms for handling these large collections of structured data has received significant interests in data mining and machine learning communities over recent years, both in the input [44, 84, 106, 158] and output domain [95, 167, 138, 141].

Structured input refers to the situation that the samples or features are organized in a certain meaningful way, e.g. chain, tree or a graph. For instance, in microarray classification, we often use genes as features and genes form biological networks, captured in various biological network databases [106, 109]. In text mining where key words are features, we have additional information about synonyms or antonyms of the features. Such information is usually captured with a word net [48]. In sensor networks, at a given time point regarding the state of the full sensor network, the features are the readings of the sensors, and we usually know the topology of the network or the physical location of the sensors [84].

Besides structured input, data may have structured output. Unlike structured input, structured output may either refer to the case that output labels have structured relationship

or the scenario that predictive functions for generating the output are structured. For example, in Natural Language Parsing, given a sentence we aim to derive its grammar parse tree according to the general grammar rules [167]. In text categorization, we often have label/class taxonomy which is organized either in a hierarchical tree [141] or a DAG [11]. In multiple cancer prediction from microarray data, each task corresponds to a function that predicts whether a particular cancer exists and these functions (a.k.a models) can be organized as groups [79] or graphs [88]. In location based social network analysis, we aim to annotate the missing place information from a set of all possible locations, and the locations have spatial relationship or hierarchical relationship [183]. The rest examples can be found named entity recognition [18, 138] and label sequence learning [18, 167].

For normal machine learning problems, the modeling practise is to learn a function: $f : \mathcal{X} \to \mathcal{Y}$, in which $\mathcal{X}$ is typically a data matrix in a vector space and $\mathcal{Y}$ belongs to $\mathcal{R}$, $\mathcal{N}$ or $\{-1, 1\}$ for different learning purposes. For structured data learning problems, the target is still learn a function mapping from input to output domain, but both domains may have structural information and the structure can be captured by a certain data model, such as chain, tree or graph.

However, it is nontrivial to adopt normal machine learning algorithms, e.g. SVM, linear regression to structured data. For one thing, the structure information in the input domain and output domain is ignored if applying the normal algorithms to structured data. For another, the major challenge in learning from many high-dimensional structured data is that input/output domain can contain tens of thousands even larger number of features and labels. With the high dimensional structured input space and/or structured output space, learning a low dimensional and consistent structured predictive function is important for both robustness and interpretability of the model.

In this dissertation, we will present a few learning models that learn from the data with one or more of the following aspects.

- The data has known structural relationship among input features.

- The data has known/unkown structural relationship among output tasks.
- The data has limited training samples but high dimensional feature space.
- Features of the data are not atomic but have internal complexity.

The focus of this dissertation is on the first three cases. The fourth case happens when dealing with more complex data sets,such as graphs, and use subgraph as features to represent the graph. In that case, the features themselves contain complex structure and have spatial/partial overlapping relationship detailed in Chapter 3.

As we know, it is often the case that only a limited training samples can be collected, due to such factors as time and cost. The situation becomes worse if the feature dimensionality is high. When labeled data is limited, it becomes more important to make use of any additional sources of information available, which can be in the form of different but related sets of data (multiple tasks), different relationships of the data (task relationship) and information about the relationships between features of the data. Leveraging the additional knowledge and integrating into learning provides us some prior information so that we can maximize the utility of data given limited training samples.

In general, the characteristics of the data along with the specific form of auxiliary information as listed above determines the specific learning problem. For instance when analyzing Microarray data from one particular cancer type (e.g. breast cancer), the data is typically characterized by low sample size and high dimensionality (case 3). Moreover, the features of the data are genes and genes have known structural relationship (case 1) that can be captured by Biological pathways (a group of genes carry a certain Biological function). It may be desirable to make use of the additional information in learning a predictive model. One line of my previous research with structured data is on utilizing auxiliary information in the form of a known relationship between features of the data [42, 43, 44, 47, 84]. These works include subgraph based graph classification [42, 43, 44], networked feature selection [47], anomaly detection and localization [84], and they are discussed chronologically in Chapter 3

and Chapter 4, of this dissertation, comprising preliminary study on learning from structured input.

Another line of my work with structured data is to utilize the structural information from features or tasks under the context of multi-task learning [4, 111, 112, 136]. Multi-task Learning (MTL) aims to enhance the generalization performance of supervised classification or regression by learning multiple related tasks simultaneously, in which all the tasks share the same feature representation. Following up with the example of Microarray data analysis but extending one step further, we considering the problem of predicting cancer status based on several Microarray data sets, where there are different types of cancers. Each data set is composed of multiple Microarray data from patients who have or do not have the specific cancer. Some cancers are "similar" to each other (e.g. breast cancer vs ovary cancer) while some are quite different (e.g. breast cancer vs prostate cancer). We aim to transfer some knowledge from one task to a related one with the purpose of Leveraging commonality among tasks. Towards this end, we have developed a multi-task learning algorithm in which feature selection and task relationship learning are performed simultaneously. The algorithm has been applied to multiple cancer type prediction, neuron activity prediction [45] and social behavior prediction [46]. We provide details in 5 and Chapter 6.

Last but not least, my knowledge on "structured data" has been also applied spatial and temporal data mining for urban planning, e.g. transit optimization and dynamic population density estimation. In particular, through my intern work at IBM T.J. Watson Research, I will demonstrate how to leverage structural information from mobile data (e.g. call detail record and GPS data) to accurately derive important places from people's life as well as daily traveling profile, including origin and destination (OD) and time of day origin and destination (TOD). All the mined information is indispensable for urban planning, especially for transit optimization.

1.1 Contribution

The dissertation provides a theoretic framework and efficient and effective algorithms for structured data with structured input features and/or output tasks including unsupervised learning, single task/multi-task learning. Collectively, the theoretic framework and the algorithms will provide the research community much better tools to mine and learn even more complex data set with structured input and structured output. More specifically, our contributions are:

- We have investigated a broad range of learning problems from structured data covering different applications, including Cheminformatics, Bioinformatics, social network analysis, telecommunication and computational neuroscience.

- We have designed a novel way to incorporate the prior knowledge of structured input features into learning framework and achieved sparsity and smoothness in the feature space.

- To our best knowledge, we are the first to study the interplay between structure feature selection and structured output tasks relationship inference under the multi-task learning framework.

- We have formalized each learning problem into structured risk minimization under a certain regularization and proposed efficient optimization algorithms to solve them.

The remainder of the dissertation is as follows. First in chapter 2, a background section will cover more details about structured data, single task learning and multi-task learning. A few motivating examples is listed. Next, preliminary study on learning from structured input or output is given in the following three chapters. The first part, Chapter 3 is on the work of structured sparse boosting algorithm that incorporates the structured relationship between base learners for graph classification [42, 43, 44]; the second part, Chapter 4, is about structured sparse PCA by adapting network topology for anomaly detection and

localization [84], and the final part of the preliminary study, Chapter 5, is a multi-task learning algorithm with known task relationship on utilizing latent social network structure induced by common interests for social behavior prediction [46]. Afterwards, a more general framework that investigates the interplay of structured input and output under multi-task learning [45] is given in Chapter 6. In Chapter 7, my work in leveraging structure information of mobile data is given. Finally, conclusions and future work are discussed in Chapter 8.

Chapter 2

Background

This chapter outlines the background of unstructured data, structured data, supervised learning, single task vs multi-task learning, regularization as well as the motivated applications. Besides, the contribution of the thesis is provided in this chapter. Bellow we give the notations.

2.1 Notations

Throughout the proposal, all matrices are boldface uppercase letters, vectors are boldface lowercase letters, sets are uppercase calligraphic letters and Lagrange multipliers are Greek letters $\{\lambda, \lambda_1, \lambda_2 ...\}$. n is the number of samples in the training data set, d is the data dimensionality, and k is the number of tasks. The ith sample in the training data set is denoted as $\mathbf{x}_i \in \mathcal{R}^d$, and its corresponding label is denoted as $\mathbf{y}_i \in \{-1, 1\}^k$, where $\mathbf{y}_i(j) = 1$ if $\mathbf{x}_i$ belongs to class j and $\mathbf{y}_i(j) = -1$ otherwise. $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]^T \in \mathcal{R}^{nd}$ represents the input data matrix and $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n]^T \in \{0, 1\}^{n \times k}$ is the output label or output task matrix. In our discussion, we assume that each instance in the training data set is represented by a feature vector and an associated label set. For those applications with semi-structured data such as chemical protei interaction prediction, we assume a certain procedure has been applied on the data to derive its feature vector, e.g. generating frequent

subgraphs from graph data sets and representing each sample graph with a binary vector ($x_{ij} = 1$ denoting that the jth subgraph occurs in ith graph).

We use $\|\mathbf{A}\|_1 = \sum_{i,j}^{p} |a_{ij}|$ to denote the L_1 norm of $\mathbf{A}$, $\|\mathbf{A}\|_F$ to denote the Frobenius norm, $\|\mathbf{a}\|_2 = \sqrt{\sum_{i=1}^{d} a_i^2}$ to represent the L_2 norm of vector $\mathbf{a}$, $< \mathbf{A}, \mathbf{B} >= tr(\mathbf{A}^T\mathbf{B})$ to represent the inner product between two matrices where $tr(.)$ is the trace of matrix. Furthermore, given matrix $\mathbf{A} \in \mathcal{R}^{d\times k}$, $\mathbf{A}_{i,:}$ is the ith row, $\mathbf{A}_{:,j}$ is the jth column and $\|A\|_{1,q} = \sum_{i=1}^{d} \|\mathbf{A}_{i,:}\|_q$ is the L_1/L_q norm. Unless stated otherwise, all vectors are column vectors.

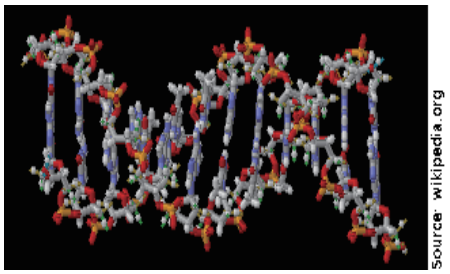
2.2 Background

In this section, we provide the details for structured data, single task/multi-task learning, and regularization respectively.

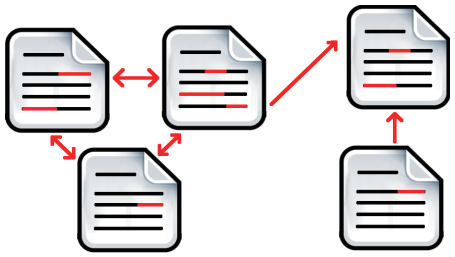
2.2.1 Unstructured data and Structured data

There is no formal definition for unstructured data vs structured data. Unstructured data (or unstructured information) refers to information that either does not have a pre-defined data model and/or does not fit well into relational tables (wikipedia). Typically unstructured data contains content only, e.g. body of an email, video and audio file.

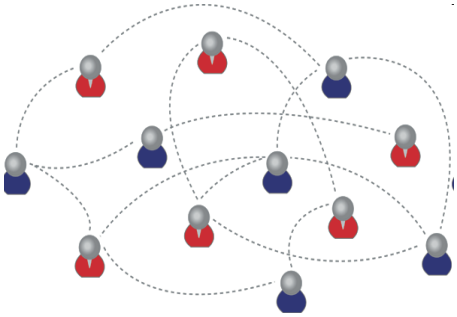
To the contrary, structured data refers to the data that has both information contents and the organization of contents. For example in Figure 2.1, we show four types of data. For protein structure data, it contains both animo acids and 3D structure. For web document data, each document has bag of words and there are hyper-links among web documents. Similarly, for social network data and genomic data, the network topology and biological pathways define the structure of data.



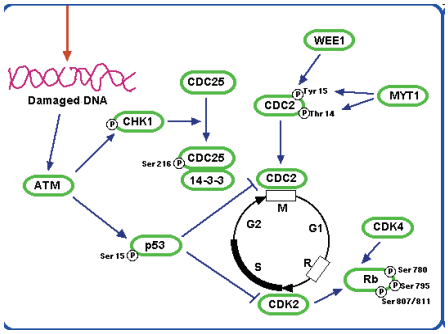
Protein Structure (source: wikipedia)



Document/Hyper Text (Source: Lampert'11)



Social Network (Source: web)



Gene Rb pathway (Source: http://dna.brc.riken.jp)

Figure 2.1: Examples of structured data. Top left: Protein structure data; Top right: web document data; Lower left: social network data; Lower right: gene Rb pathway data.

2.2.2 Supervised Learning

The general goal of machine learning is to learn a predictive function: $f : \mathcal{X} \to \mathcal{Y}$, in which $\mathcal{X}$ is typically a data matrix in a vector space and $\mathcal{Y}$ belongs to $\mathcal{R}$ for regression, $\mathcal{N}$ for ranking or $\{-1, 1\}$ for binary classification. Supervised learning seeks the predictive function f over a set of functions $\mathcal{F}$ by minimizing empirical loss on a training data set consisting of a set of data and label pairs, $\{\mathbf{x}_i, y_i\}_{i=1}^n \in \mathcal{X} \times \mathcal{Y}$.

$$\min_{f \in \mathcal{F}} \sum_{i=1}^{n} \ell(y_i, f(\mathbf{x}_i)) \tag{2.1}$$

where $\ell(.,.)$ is loss function measuring fitness and it could be 0-1 loss, hinge loss, exponential loss or negative binomial likelihood.

For normal supervised learning problem, there is no structural information on either input domain $\mathcal{X}$ or $\mathcal{Y}$. For structured data learning problems, both domains may have structural information and the structure can be captured by a certain data model, such as chain, tree or graph. For example in Figure 2.1, the protein structure data and web documents has structural information in the input domain. For the social network and the genomic data, the structural information could be found in either input or output domain based on different learning purposes. If one is interested in identifying social communities, the structure is on the input domain. But if one tries to do behavior targeting, then the structure is on the output domain.

2.2.3 Single Task Learning vs Multi-task Learning

Based on how many tasks are involved in the learning process, supervised learning can be divided into single task learning and multi-task learning. For single task learning, only one task is performed such as classifying breast cancer vs normal from Microarray data and classifying handwritten digit "6" vs "b". Traditional learning algorithms e.g. SVM, logistic regression and boosting, belong to this category.

For multi-task learning, there are several tasks that are learned jointly. As shown in Figure 2.2, there are three tasks and each task corresponds to a classification problem on a particular type of cancer.

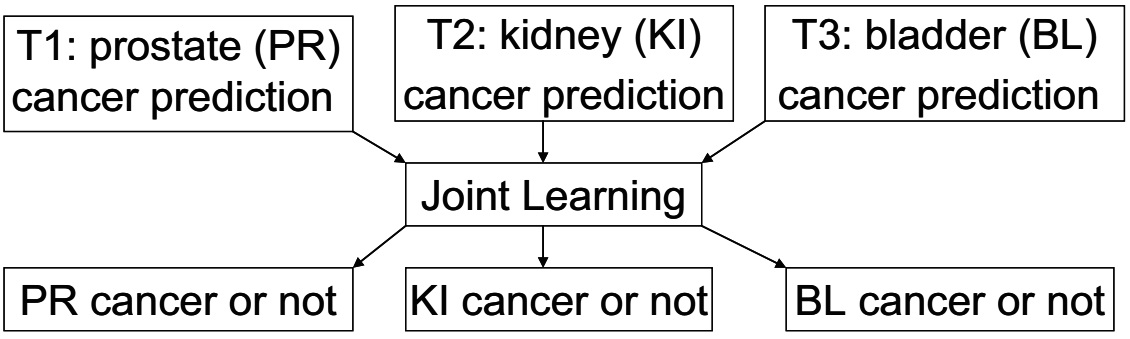


Figure 2.2: An MTL example from multiple cancer prediction.

In this thesis, we focus on multi-task linear model. W.L.O.G., suppose we are given k tasks $\{T_i\}_{i=1}^k$. For the ith task T_i, the training set $\mathcal{D}_i$ consists of n samples $(\mathbf{x}_j^i, y_j^i)$, $j = 1, \cdots, n_i$, where $\mathbf{x}_j^i \in \mathcal{R}^p$ and $y_j^i \in \{0, 1\}$. For simplicity, we assume all the tasks have the same number of training samples. The goal of the modeling practice is to learn a function $f_i(\mathbf{x})$ to map the sample to the output, where $f_i(\mathbf{x}) = \mathbf{w}_i^T\mathbf{x}$. The learning task is to seek $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_k]$ with $\mathbf{w}_i$ corresponding to the ith task, such that:

$$\min_{\mathbf{W}} \sum_{i=1}^{k} \sum_{j=1}^{n} \ell(y_j^i, f_i(\mathbf{x}_j^i)) \tag{2.2}$$

(5.1) is minimized.

We use linear regression with least square loss function $\ell(y_j^i, f_i(\mathbf{x}_j^i)) = 1/2(y_j^i - f_i(\mathbf{x}_j^i))^2$ to perform classification, which is equivalent to a linear discriminant analysis (LDA) for binary classification [66]. Such a procedure is also widely used in other MTL algorithms for classification problems [26, 111, 190].

2.2.4 Motivating Applications

Structured data has diverse applications, e.g. Bioinformatics [106, 109], Cheminformatics [154], social network analysis [30, 140], natural language processing [48, 167] and text mining [64]. Since single task learning is a special case of multi-task learning, we only list a few applications of structured data learning under multi-task framework.

Text Categorization Real-world documents often involves multiple categories, for example, a web page introducing the release of the newest android may be categorized as business and technology. The task of text Categorization is to classify text documents under one or more of a set of predefined categories or subjects. Typically, the problem can be cast as a multi-task learning problem, in which each task corresponds to classifying a document to one category. The predefined labels (or categories) in text categorization are usually not

assumed to be mutually exclusive that can be captured by a certain structure e.g. hierarchy [158], thus the text categorization can naturally be modeled as a multi-task learning problem with structured output tasks.

Neuron-activity prediction An important goal in computational neuroscience is to analyze the association between neuron activity and external stimulus such as viewing a pictures or hearing a word of certain semantic categories, including tools, buildings and animals [35, 111, 122]. The task of neuron-activity prediction is to predict the activity value given stimulus, e.g. a word. Computational linguists have analyzed the statistics of very large text corpora and have demonstrated that a word's meaning is captured to some extent by the distribution of words and phrases with which it commonly co-occurs [122], therefore a natural feature representation for the stimulus word is the intermediate semantic features extracted from trillion-word text corpus such as google-trillion words.

Since multiple related neurons tend to fire with similar stimulus [35], it is natural to model the activity of a set of neuron jointly rather than a single one. In [111], authors proposed a multi-task learning approach to predict the activity of several neurons. For example, we show the scheme with 3 tasks in Figure 2.3. In this example, the structured input information can be found from the input keyword features that can be captured by wordnet [48] or co-occurrence statistics [145]. The output tasks are also structured, since similar neurons are tend to be fired together given the same stimulus.

Social behavior prediction In social behavior prediction, we are interested in social activity prediction in a social network i.e., to predict a user's response (e.g., comment or like) to their friends' postings (e.g., blogs, tweets) or to click a particular advertisement link recommended from his followers. Similar to traditional supervised learning algorithm, the information content (sample) is rep-resented as a high dimensional feature vector and its labels indicate the responses of users towards the information. Social behavior prediction has diverse applications ranging from behavior targeting [178], personalized news delivery

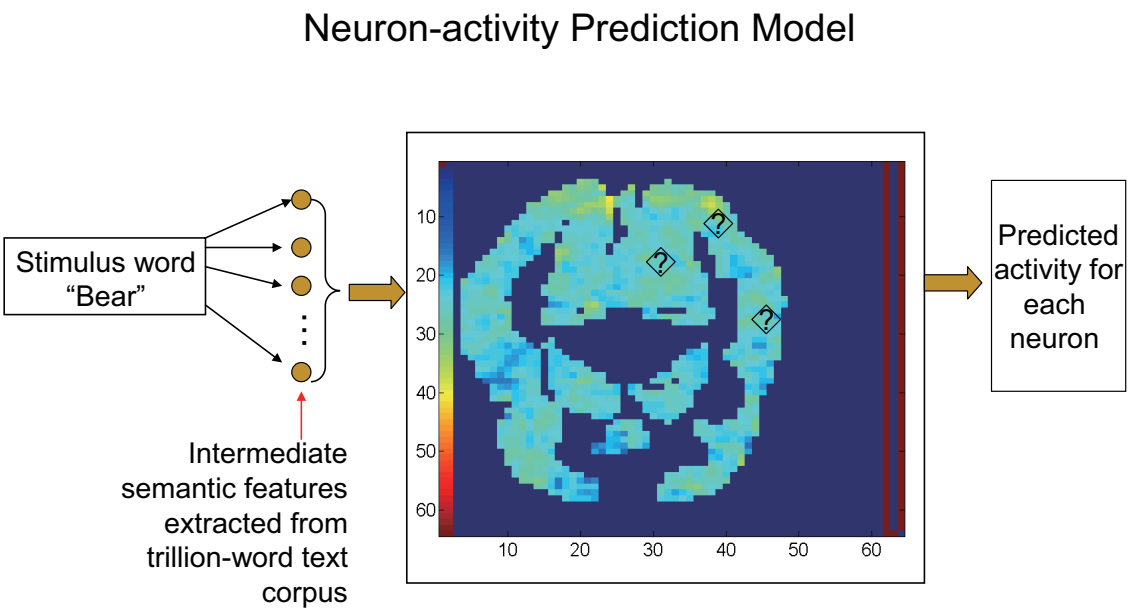


Figure 2.3: The scheme for MTL approach to neuron activity prediction: each neuron corresponds to a task and the features is the co-occurrence rate extracted from text corpus. “?” denotes the neuron activity value to be predicted.

[107] and enhanced search [176]. The major challenges of this problem is the sparsity and heterogeneity, where sparsity means only a small number of actions per-user distributed in a large number of samples and heterogeneity refers to the situation that the topic and social linkage are heterogenous. MTL is applicable to conquer the challenges since MTL increases effective sample size and hence boosts the generalization performance of learned models by learning several related tasks simultaneously. As discussed before, the task space is structured since each user corresponds to a task and there are links among them.

Chapter 3

Preliminary Study I: Boosting with Structural Sparsity

3.1 Introduction

Boosting is a very successful classification algorithm that produces a linear combination of "weak" classifiers (a.k.a. base learners) to obtain high quality classification models [52, 55, 146, 147]. Recently, the boosting algorithm has been successfully extended to tasks such as multi-class classification [108], multi-label classification [179], cost sensitive learning [118], semi-supervised learning [194], manifold learning [115], classification with missing-value [65], and transfer learning [33] among others.

In this paper we propose a new boosting algorithm where base learners have structure relationships in the functional space. Our work is particularly motivated by the emerging topic of pattern based classification for semi-structure data including graphs [85, 143, 161, 168, 180]. For example, Kudo *et al.* [100] recently applied boosting to graph classification using subgraphs as base learners and showed the connection of graph boosting to support vector machine with the R-convolution kernel. Nowozin *et al.* [129, 144] combined subgraph mining and graph boosting for classifying graphs representing images.

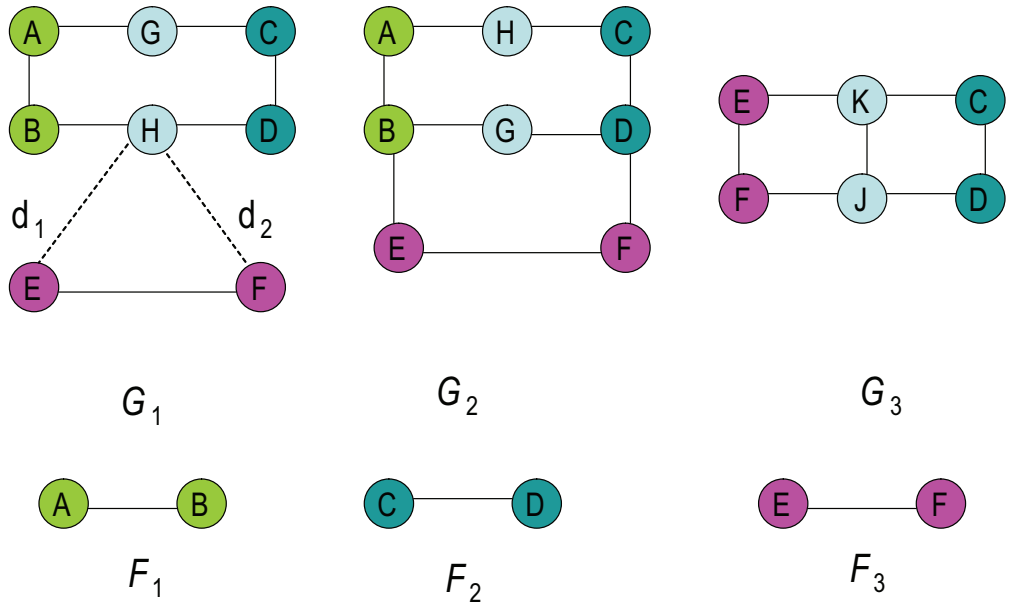


Figure 3.1: Three subgraph features in three graphs. Dashed edge means that the two nodes are connected by a path with varying length ¿1.

Though graph boosting has demonstrated promising results, the limitations of the current algorithms are that they totally ignore the structure relationships among subgraph base learners and hence may not provide the optimal results for graph classification. We illustrate the point with the following example:

Consider the three labeled graphs G_1, G_2, G_3 and three subgraph features F_1, F_2, F_3 shown in Figure 3.1. Suppose that the class labels for graphs G_1, G_2, G_3 are $Y = [1, 1, -1]^T$. We may construct three base learners $h_1(G)$, $h_2(G)$ and $h_3(G)$ in the format $h_i(G) = 1\ if\ F_i \subseteq G$ and $h_i(G) = -1$ otherwise ($i \in \{1, 2, 3\}$). These decision rules are derived based on a majority voting of subgraph coverage on positively and negatively labeled graphs.

Considering a boosting algorithm that iteratively selects base classifiers to build ensemble models, since h_1 is perfectly correlated with class labels as evaluated on the three training samples, h_1 will be selected first. h_2 and h_3 produce the same prediction for all the graphs in the training data set and hence may be perceived to have the same discriminative power. This is not true in this example. Subgraph F_1 and F_2 occur in every positive graph sample and are clustered with a consistent relative spatial position. F_3 occurs in every graph, but

in contrast to F_1 and F_2, it has quite different spatial distribution as compared to F_1 and F_2 and hence we consider F_3 as a spurious pattern. Once F_1 is selected, we argue that we should select F_2 rather F_3 to build more stable and interpretable classification models. However, current boosting methods are not designed to perform such model selection since the structure relationships of base learners are not considered in any case.

The spatial relationship is special cases of possible relationships of base learners. Another example is the partial overlapping relationship. We call the possible information regarding to the relationships of base learners as structure relationships. Here we hypothesize that the structure relationship of subgraph features carries important information regarding the importance of the base learners in boosting. Towards an efficient incorporation of such information, we design a general model where we use an undirected graph to capture the relationship of subgraph-based base learners. We combined L_1 norm and Laplacian based L_2 norm penalty with Logit loss function of Logit Boost [55]. In this approach, we enforce model sparsity and smoothness in the functional space spanned by the basis functions. We derive efficient optimization algorithms based on coordinate decent for the new boosting formulation and theoretically prove that it exhibits a natural grouping effect for nearby spatial or overlapping features. Using comprehensive experimental study and comparing with the state-of-the-art, we have demonstrated the effectiveness of the proposed learning method.

We believe the new formalization is applicable to a variety of boosting applications where (i) base learners have a known structure relationship and (ii) the optimal ensemble of base learner functions is sparse in the functional space. The proposed method can be naturally extended to other semi-structured data such as sequences and trees where patterns such as frequent subsequences and frequent subtrees are widely used for classification [104].

3.1.1 Related Work

Subgraph based supervised learning on graphs has recently attracted extensive research interest [85, 143, 161, 168, 180]. For example, Yan et. al [180] proposed Leap algorithm with two concepts: structural leap search and frequency descending to reduce search space and mine informative patterns faster than previous methods. However, LEAP only considers individual pattern rather than a set of patterns [85]. Moreover, the discriminate power of a pattern is evaluated entirely on the occurrence information of the pattern and misses interaction among patterns. gPLS [143] applies partial least square regression to graph mining and performs feature selection and classifier construction simultaneously, but the model interpretability is low due to the use of latent variables [85]. In addition, the structure relationship among features is neglected. COM [85] is a newly proposed method that mines co-occurrence rules. COM is prone to giving high number of false positives and fails to consider the structure information among features as well.

Recently, a significant amount of progress has been made on developing supervised learning algorithms for feature selection from data with structured features [36, 79, 94, 106, 145, 163, 187, 193]. In these models, features may be naturally partitioned into groups [36, 79, 187] or ordered in some meaningful way, such as a chain [94, 163], a tree [193] or a graph [106, 145]. These approaches demonstrate the importance of incorporating prior structure information among features to build highly accurate and interpretable models. However, all these algorithms handles vector data and hence are not applicable to graphs.

In the context of structured feature selection of boosting for other types of data, the most related work to ours is the *spatially informed boosting* for fMRI data analysis [174]. In their work, they apply L_2 norm regularized Gaussian kernel matrix to guiding the boosting algorithm to select spatially clustered image voxels or pixels. But their method did not provide a more general approach of encoding the spatial relationship. It is possible that Gaussian kernel matrix works for some data, but fails for others. Furthermore, they use exponential loss function which sensitive to outliers [55].

Though subgraph based feature selection on graph data has been studied for a long time, none of the existing method considers the structure relationships among subgraph features and hence may not provide the optimal results for graph classification. The objective of this paper is to incorporate the structural information on features into learning and build a more accurate and interpretable graph boosting model.

3.2 Background

Here we introduce notations and preliminaries for graph, graph kernel functions, and Boosting.

3.2.1 Graph Theory

A *labeled graph* G is described by a finite set of nodes V and a finite set of edges $E \subset V \times V$. In most applications, a graph is labeled, where labels are drawn from a label set σ. A labeling function $\lambda : V \cup E \to \Sigma$ assigns labels to nodes and edges. Without loss of generality, we handle fully-labeled graphs where both nodes and edges are labeled in this paper. We do not assume any structure of label set Σ now; it may be a field, a vector space, or simply a set.

Following convention, we denote a graph as a quadruple $G = (V, E, \Sigma, \lambda)$ where V, E, Σ, λ are explained before. A graph $G = (V, E, \Sigma, \lambda)$ is a *subgraph* of another graph $G' = (V', E', \Sigma', \lambda')$, denoted by $G \subseteq G'$, if there exists a 1-1 mapping $f : V \to V'$ such that

- for all $v \in V, \lambda(v) = \lambda'(f(v))$
- for all $(u, v) \in E, (f(u), f(v)) \in E'$
- for all $(u, v) \in E, \lambda(u, v) = \lambda'(f(u), f(v))$

In other words, a graph G is a subgraph G' of another graph if there exits a 1-1 node mapping f preserving the node labels, edge relations, and edge labels. The 1-1 mapping

f is a *subgraph isomorphism* from G to G' and the range of the mapping f, $f(V)$, is an *embedding* of G in G'.

3.2.2 Graph Kernel Function

Kernel functions are powerful computational tools to analyze large volumes of graph data [67]. The advantage of kernel functions is due to their capability to map a set of data to a high dimensional Hilbert space without explicitly computing the coordinates of the structure. This is done through a special function K. Specifically a binary function $K : X \times X \to \mathcal{R}$ is a *positive semi-definite* function if

$$\sum_{i,j=1}^{n} c_i c_j K(x_i, x_j) \geq 0 \tag{3.1}$$

for any $m \in \mathcal{N}$, any selection of samples $x_i \in X$ $(i = [1, n])$, and any set of coefficients $c_i \in \mathcal{R}$ $(i = [1, n])$. In addition, a binary function is symmetric if $K(x, y) = K(y, x)$ for all $x, y \in X$. A symmetric, positive semi-definite function ensures the existence of a Hilbert space $\mathcal{H}$ and a map $\Phi : X \to \mathcal{H}$ such that

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle \tag{3.2}$$

for all $x, x' \in X$. $\langle x, y \rangle$ denotes an inner product between two objects x and y. The result is known as the Mercer's theorem and a symmetric, positive semi-definite function is also known as a Mercer kernel function [149], or *kernel* function for simplicity. In this paper, we focus on graph random walk based kernels, where we use subgraph as features and kernels are defined on pairwise subgraph features.

3.3 Preliminaries

We use the following notations throughout the rest of the paper. We use lowercase letters to represent scalar values, lower-case letters with an arrow to represent vectors (e.g. $\vec{\beta}$), uppercase letters to represent matrices, $\{\lambda, \lambda_1, \lambda_2...\}$ to represent Lagrange multiplier, and uppercase calligraphic letters to represent sets. Unless state otherwise, all vectors in this paper are column vectors.

Given training instances $T = \{x_i, y_i\}_{i=1}^n$ where $y_i \in \{-1, +1\}$, $x_i \in \mathcal{X}$, we construct a set of base learners $\mathcal{H} = \{h_j : \mathcal{X} \mapsto \{-1, +1\}, j = 1 \cdots p\}$. In this paper, we do not assume any type of $\mathcal{X}$; it may be a vector space, or simply a set. The objective of boosting is to train a composite binary classifier with weight vector $\vec{\beta}$ taking the form of $h_{\vec{\beta}}(x_i) = sgn(\sum_{j=1}^p \beta_j h_j(x_i))$ such that the following empirical loss function $\ell(\mathcal{X}, \vec{y}; \vec{\beta})$ is minimized.

$$\mathcal{L}(\mathcal{X}, \mathcal{Y}, \vec{\beta}) = \sum_{i=1}^{n} l(y_i, h_{\vec{\beta}}(x_i)) \tag{3.3}$$

where l is a loss function.

AdaBoost [53] takes the exponential loss function:

$$l(y_i, h_{\vec{\beta}}(x_i) = exp(-y_i \sum_{j=1}^{p} \beta_j h_j(x_i))) \tag{3.4}$$

and LogitBoost [55] takes the logit loss function:

$$l(y_i, h_{\vec{\beta}}(x_i) = log(1 + exp(-y_i \sum_{j=1}^{p} \beta_j h_j(x_i))) \tag{3.5}$$

Duchi et. al [36] modified AdaBoost by imposing L_1/L_2 or L_1/L_∞ penalty on weight vectors in a multi-task learning framework. However, they neglect the structure information among base learners. We consider a simple yet effective modification to Logit Boost [55] that incorporates a composite penalty with L_1 and L_2 regularization encoding the structural information among base learners on the weight vector, which is detailed in the following

section.

3.4 Boosting with Structure Information in the Functional Space

We capture the structure relationships among base learners as an undirected graph G, whose nodes correspond to the set of p base learners. Edges in the graph G are weighted, with $w_{i,j}$ indicating the "closeness" between the two features and 0 indicating that the two features have no relationship. We call the graph G "feature graph" and explore approaches for building a feature graph in Section 3.5.2.

We incorporate the priori domain knowledge by adding a Tikhonov regularization factor $\frac{1}{2}\sum_{i,j} w_{i,j}(\beta_i - \beta_j)^2$ in a convex fitness function $\ell(\mathcal{X}, \vec{y}; \vec{\beta})$ to enforce that the feature coefficients vary smoothly for neighboring features. The Tikhonov regularization factor could be conveniently written in matrix format $\vec{\beta}^T L \vec{\beta}$ where L is the *Laplacian* of G given by: $L = D - W$. W is the p by p edge weight matrix $W = (w_{i,j})_{i,j=1}^{p}$, and D is the density matrix of W, defined as $D = (d_{i,j})_{i,j=1}^{p}$ where $d_{i,j} = \begin{cases} \sum_{k=1}^{p} W_{i,k} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$

To avoid having any feature "dominate" the penalization function, we use the *normalized Laplacian* $\mathcal{L}$ following [32] to normalize the weight of each feature, where the elements of $\mathcal{L}$ are defined by

$$\mathcal{L}_{i,j} = \begin{cases} 1 - w_{i,j}/d_{i,i} & \text{if } i = j \text{ and } d_{i,i} \neq 0 \\ -w_{i,j}/\sqrt{d_{i,i} d_{j,j}} & \text{if } i \text{ and } j \text{ are adjacent} \\ 0 & \text{otherwise} \end{cases} \tag{3.6}$$

Tikhonov regularization does not lead to the sparsity of the model. To obtain a sparse solution, we add the L_1 norm of $\vec{\beta}$ to the convex function $\ell(\mathcal{X}, \vec{y}; \vec{\beta})$. Specifically, we seek to identify a vector $\vec{\beta}$ that minimizes the following loss function:

$$g(\mathcal{X}, \vec{y}; \vec{\beta}) = \ell(\mathcal{X}, \vec{y}; \vec{\beta}) + \lambda_1 ||\vec{\beta}||_1 + \frac{1}{2}\lambda_2 \vec{\beta}^T \mathcal{L} \vec{\beta} \tag{3.7}$$

where $\lambda_1 > 0$, $\lambda_2 > 0$, $||.||_1$ is L_1 norm. In our implementation, we use the logitloss [55]:

$$\ell(\mathcal{X}, \vec{y}; \vec{\beta}) = \sum_{i=1}^{n} \log(1 + \exp(-y_i \sum_{j}^{p} \beta_j h_j(x_i))) \tag{3.8}$$

The major challenge in fitting the model described in Equation (3.7,3.8) to data is to estimate the parameter $\vec{\beta}$ efficiently and accurately. In the following subsection, we provide the optimization algorithm.

3.4.1 Optimization Algorithm

We discuss the optimization algorithm for Equation (3.7) bellow. We first show that the structurally regularized boosting with logit loss function can be interpreted as an additive logistic regression with the same regularization in the functional space spanned by base learners. We then provide the optimization algorithm based on coordinated decent to solve the equivalent regularized logistic regression towards the base learners. For simplicity, let $F(x) = \sum_j^p \beta_j h_j(x)$ be the decision function on the sample x. For a fixed training data set, we denote all the predicted labels for the training data using functions in $\mathcal{H}$ as an n by p matrix H, where n is the sample size, p is the number of base learners. $H_{i,j} = h_j(x_i)$ is the label given by base learner $h_j \in \mathcal{H}$ on the training sample x_i. We call H "object-prediction" matrix. We use H_i to denote ith row of object-prediction matrix H (the predictions of all the base learners on the sample x_i) and $H_{.j}$ to represent jth column of H (the predictions of h_j on the training data).

We use the following Lemma to show that the minimizer of the expected loss function $J(F) = E(\log(1 + \exp(-yF(x))))$ is the symmetric logistic transform of $P(y = 1|x)$.

Lemma 3.4.1. *$E(\log(1 + \exp(-yF(x))))$ is minimized at $F(x) = log(\frac{P(y=1|x)}{P(y=-1|x)})$. Hence $P(y = 1|x) = \frac{1}{1+e^{-F(x)}}$ and $P(y = -1|x) = \frac{1}{1+e^{F(x)}}$.*

Proof. Since E imposes expectation over the joint distribution of y and x, we have $E(\log(1+\exp -yF(x))) = P(y=1|x)\log(1+\exp(-yF(x))) + P(y=-1|x)\log(1+\exp(yF(x)))$. Then it is sufficient to minimize $J(F)$ by computing the first derivative with respect to $F(x)$: $\frac{\partial J(F)}{\partial F(x)} = -\frac{P(y=1|x)e^{-F(x)}}{1+e^{-F(x)}} + \frac{P(y=-1|x)e^{F(x)}}{1+e^{F(x)}}$. The result follows by setting the derivative to zero. □

With Lemma 3.4.1, the structurally regularized boosting can be interpreted as logistic regression with the same regularization function. Let $y^* = (y+1)/2$, taking values of 0, 1, and parameterize the binomial probabilities by $P(y=1|x) = p(x) = \frac{1}{1+e^{-F(x)}}$, it is sufficient to derive that the logit loss function is equivalent to negative binomial log-likelihood:

$$\begin{aligned} l_b(y^*, p(x)) &= -[y^* log(p(x)) + (1-y^*) log(1-p(x))] \\ &= \log(1+\exp(-yF(x))) \end{aligned} \tag{3.9}$$

By plug $p(x)$ into (3.9), we can reduce (3.9) to $l_b(y^*, p(x)) = \log(1+e^{F(x)}) - y^*F(x)$. Now we rewrite (3.7) in terms of negative binomial log-likelihood with y^*:

$$\begin{aligned} g(\mathcal{X}, \vec{y}; \vec{\beta}) &= \textstyle\sum_{i=1}^n [\log(1+\exp(\sum_j^p \beta_j h_j(x_i))) - y_i^* \sum_j^p \beta_j h_j(x_i)] + \lambda_1 ||\vec{\beta}||_1 + \frac{1}{2}\lambda_2 \vec{\beta}^T \mathcal{L} \vec{\beta} \\ &= \textstyle\sum_{i=1}^n [\log(1+\exp(H_i\vec{\beta})) - y_i^* H_i \vec{\beta}] + \lambda_1 ||\vec{\beta}||_1 + \frac{1}{2}\lambda_2 \vec{\beta}^T \mathcal{L} \vec{\beta} \end{aligned} \tag{3.10}$$

After transforming logit loss to negative binomial log-likelihood, we followed the general framework of coordinated decent algorithm proposed in [56] recently proposed by Friedman *et al.* for L_1 norm regularized logistic regression. Their approach relies on the connection between the Newton's method for optimizing logistic regression and the least square formulation. The Newton's method amounts to using Taylor expansion, up to a quadratic function, to approximate the logit function. In this way, applying Newton's method can be viewed as solving a series of least squares problem (also called *iterative reweighted least squares fitting* [56]). Applying Taylor's expansion at current estimate $\tilde{\vec{\beta}}$ to negative log-likelihood function

(3.9), we have the reweighted least square problem:

$$l_Q(\vec{\beta}) = -\sum_{i=1}^{n} w_i(z_i - H_i\vec{\beta})^2 + C(\tilde{\vec{\beta}}) \tag{3.11}$$

where $z_i = H_i\tilde{\vec{\beta}} + (y_i^* - \tilde{p}(x_i))/(\tilde{p}(x_i)(1-\tilde{p}(\vec{x}_i)))$, $w_i = \tilde{p}(x_i)(1-\tilde{p}(x_i))$ and $C(\tilde{\vec{\beta}})$ is a constant.

In the remaining discussion, we show an extension of Friedman's work to solve a reweighted least square fitting (3.11) with Laplacian weighted L_2 and L_1 norm regularization. To handle the new mixture penalty, we derive a modified coordinate descent scheme in Lemma 3.4.2 extending the work presented in [56].

Lemma 3.4.2. *Suppose that the data set contains n observations and p predictors, with the response vector* $Y = (y_1, \cdots$

$, y_n)^T$ *and the data matrix* $X = (\vec{x}_1, \cdots, \vec{x}_n)^T$. *We also assume that the predictors are standardized and the response is centered so that for all* j, $\sum_{i=1}^n x_{ij} = 0$, $\sum_{i=1}^n x_{ij}^2 = 1$ *and* $\sum_{i=1}^n y_i = 0$. *The Lagrange form of the network constrained objective function (with least squares fitness function) is:*

$$L(\lambda_1, \lambda_2, \vec{\beta}) \;=\; \tfrac{1}{2}(Y - X\vec{\beta})^T(Y - X\vec{\beta}) + \tfrac{1}{2}\lambda_2\vec{\beta}^T\mathcal{L}\vec{\beta} + \lambda_1||\vec{\beta}||_1 \tag{3.12}$$

The coordinate-wise update has the form (for each β_j*):* $\hat{\beta}_j = S(\sum_{i=1}^n x_{ij}(y_i - \tilde{y_i}^{(j)}) - \lambda_2\sum_{k\neq j}^p \mathcal{L}_{jk}\hat{\beta}_k, \lambda_1)/(1 + \lambda_2\mathcal{L}_{jj})$ *where* $\tilde{y_i}^{(j)} = \sum_{l\neq j} x_{il}\hat{\beta}_l$ *is the fitted response value excluding the contribution from* x_{ij} *and* $S(z, \gamma) = sign(z)(|z| - \gamma)_+$ *is the soft thresholding operator where:*

$$sign(z)(|z| - \gamma)_+ = \begin{cases} z - \gamma & \text{if } z > 0 \text{ and } \gamma < |z| \\ z + \gamma & \text{if } z < 0 \text{ and } \gamma < |z| \\ 0 & \text{if } \gamma \geq |z| \end{cases}$$

Suppose that we have estimates of $\hat{\beta}_l$ for $l \neq j$ and we wish to partially optimize the objective function with respect to β_j. We would like to compute the gradient at $\beta_j = \hat{\beta}_j$,

which only exists if $\hat{\beta}_j \neq 0$. If $\hat{\beta}_j > 0$, then the gradient for equation 3.12 is given by

$$\frac{\partial L(\lambda_1,\lambda_2,\vec{\beta})}{\partial \beta_j} = -\sum_{i=1}^{n} x_{ij}(y_i - \sum_{k\neq j} x_{ik}\hat{\beta}_k - x_{ij}\beta_j) + \lambda_2 \sum_{k\neq j}^{p} \mathcal{L}_{jk}\hat{\beta}_k + \lambda_2 \mathcal{L}_{jj}\beta_j + \lambda_1 \tag{3.13}$$

Since X is standardized, by setting 3.13 to 0, we obtain $\beta_j = \frac{\sum_{i=1}^{n} x_{ij}(y_i - \tilde{y_i}^{(j)}) - \lambda_2 \sum_{k\neq j}^{p} \mathcal{L}_{jk}\hat{\beta}_k - \lambda_1}{1+\lambda_2 \mathcal{L}_{jj}}$. A similar closed form exists for $\hat{\beta}_j < 0$. Combining two cases we will get Lemma 3.4.2.

We notice that our solution is not constrained in L_1 and L_2 penalty, but can be extended to L_∞, which recently attracted research interest [36], since L_∞ norm is differentiable everywhere except singular points ($\vec{\beta} = 0$) [198].

We summarize what is discussed previously in the algorithm called LPGB. Given the training data $T = \{\mathcal{X}, \vec{y}\}$, the n by p object-prediction matrix $H = \{h_{i,j}\} = \{h_j(x_i)\}$ constructed from base learners, regularization parameters λ_1, λ_2 and convergence parameter ϵ, our algorithm iteratively solves (3.10). Here we transform $\vec{y}$ to $\vec{y^*}$ using 0/1 to represent the outcome and $p(x) = P(y = 1|x) = P(y^* = 1|x) = 1/(1 + \exp(-\sum_{j=1}^{p} \beta_j h_j(x)))$.

Algorithm 1 LPGB($\lambda_1, \lambda_2, H, \vec{y}*, MaxIteration, \epsilon$)

1: Initialize $\hat{\vec{\beta}}^{(0)} = \vec{0}$;
2: **for** i=1 to MaxIteration **do**
3: Compute the quadratic approximation for (3.9);
4: Use the coordinate descent method in lemma 3.4.2 to solve the reweighted least squares problem with mixture penalty and obtain the updated $\hat{\vec{\beta}}^{(i)}$;
5: **if** $||\hat{\vec{\beta}}^{(i)} - \hat{\vec{\beta}}^{(i-1)}||_1 \leq \epsilon$ **then**
6: Break;
7: **end if**
8: **end for**
9: return $\hat{\vec{\beta}} = \hat{\vec{\beta}}^{(i)}$;

As evaluated in our experimental study in Section 3.6, the regularized LPGB algorithm usually has better classification performance and are insensitive to outliers and class label noises, comparing to the unregularized gBoosting [100]. We believe that these advantages are contributed to the capability of LPGB to select clustered base learners in the functional space. We call this phenomenon the "grouping effect" and we provide theorems to explain

the "group effect" below. Our proof is similar to that presented in [106] where we consider a simple case of two base learners that are linked. We show that the related L_2 regularization ensures that the difference of the estimated coefficients have an upper bound based on the sample size and the regularization coefficients.

3.4.2 Grouping Effect

We derive an upper bound of the difference of coefficients between two neighboring features. Motivated from a similar proof in [106] where a linear regression framework with L_1 and L_2 regularization, we study the special case in which only two features are connected to each other in the feature graph.

Theorem 3.4.3. *Give training data $T = \{x_i, y_i\}_{i=1}^n$ where $x_i \in \mathcal{X}$ and fixed scalars λ_1, λ_2 and let $\hat{\vec{\beta}}(\lambda_1, \lambda_2)$ be the optimal solution to (3.10), we suppose that $\hat{\beta}_i(\lambda_1, \lambda_2)\hat{\beta}_j(\lambda_1, \lambda_2) > 0$, and the two features F_i and F_j are only linked to each other on the feature graph. Define $D_{\lambda_1,\lambda_2}(i,j) = |\hat{\beta}_i(\lambda_1, \lambda_2) - \hat{\beta}_j(\lambda_1, \lambda_2)|$, then $D_{\lambda_1,\lambda_2}(i,j) \leq \sqrt{2(1-\rho)}/\lambda_2$, where ρ is the correlation between the normalized $H_{\cdot i}$ and $H_{\cdot j}$.*

Proof. Since $\hat{\vec{\beta}}(\lambda_1, \lambda_2)$ is the optimal solution to (3.10), $\hat{\vec{\beta}}(\lambda_1, \lambda_2)$ satisfies $\frac{\partial g(\lambda_1,\lambda_2,\vec{\beta})}{\partial \beta_k}|_{\vec{\beta}=\hat{\vec{\beta}}(\lambda_1,\lambda_2)} = 0$ if $\hat{\beta}_k(\lambda_1, \lambda_2) \neq 0$. More specifically, for $\hat{\beta}_i$ and $\hat{\beta}_j$, we have

$$-H_{\cdot i}^T(\vec{y^*} - \vec{p}(\mathcal{X})) + \lambda_1 sgn(\hat{\beta}_i) + \lambda_2\hat{\beta}_i - \lambda_2 \sum_{u \neq i} w_{u,i} \frac{\hat{\beta}_u}{\sqrt{d_{u,u}d_{i,i}}} = 0 \tag{3.14}$$

$$-H_{\cdot j}^T(\vec{y^*} - \vec{p}(\mathcal{X})) + \lambda_1 sgn(\hat{\beta}_j) + \lambda_2\hat{\beta}_j - \lambda_2 \sum_{v \neq j} w_{v,i} \frac{\hat{\beta}_v}{\sqrt{d_{v,v}d_{j,j}}} = 0 \tag{3.15}$$

where $\vec{p}(\mathcal{X}) = 1/(1 + \exp(-H\vec{\beta}))$, $\vec{y^*} = (\vec{y} + \vec{1})/2$ and H is the object-prediction matrix. Subtracting (3.14) from (3.15)and taking the absolute value with the assumption that $d_{i,i} = d_{j,j} = w_{i,j}$ and $sgn(\hat{\beta}_i) = sgn(\hat{\beta}_j)$ gives

$$|\hat{\beta}_i - \hat{\beta}_j| = \frac{|H_{\cdot i}^T - H_{\cdot j}^T||\vec{y^*} - \vec{p}(\mathcal{X})|}{\lambda_2} \tag{3.16}$$

and by the definition of $D_{\lambda_1,\lambda_2}(i,j)$,

$$D_{\lambda_1,\lambda_2}(i,j) = \frac{|\hat{\beta}_i - \hat{\beta}_j|}{\vec{y^*}} = \frac{|H_{.i}^T - H_{.j}^T||\vec{y^*} - \vec{p}(\mathcal{X})|}{\lambda_2 \vec{y^*}} \tag{3.17}$$

By Cauchy-Schwartz inequality,

$$|H_{.i}^T - H_{.j}^T||\vec{y^*} - \vec{p}(\mathcal{X})| \leq \|H_{.i}^T - H_{.j}^T\|_2 \|\vec{y^*} - \vec{p}(\mathcal{X})\|_2$$

Also, because $\hat{\vec{\beta}}$ is the optimal solution to problem (3.10), we have:

$$\|\vec{y^*} - \vec{p}(\mathcal{X})\|_2 \leq \|\vec{y^*}\|_2$$

By the normalization of H, $\|H_{.i}^T - H_{.j}^T\|_2^2 = 2 - 2\rho$, hence we have $D_{\lambda_1,\lambda_2}(i,j) \leq \sqrt{2(1-\rho)}/\lambda_2$

□

The upper bound of $D_{\lambda_1,\lambda_2}(i,j)$ provides two insights of our method: 1) smoothness: the coefficients of neighboring base learners are close to each other due to the L_2 norm regularized feature graph Laplacian penalty term. 2) Grouping effect: Once a base learner is selected, its spatially neighboring base learners will be more likely selected. Thus our boosting algorithm can select groups of spatially neighboring base learners.

3.5 Application to Graph Data

We show how to apply the LPGB algorithm to graph classification bellow.

3.5.1 Base Learner Construction

In our model, we use frequent subgraphs as features and construct base learners (decision stamps) from these features. Given training data $\{\mathcal{X}, \vec{y}\}$ and a set of frequent subgraphs,

the decision stamp classifier for subgraph F_i is given by:

$$h_i(x) = \begin{cases} \hat{y} & \text{if } F_i \subseteq x \\ -\hat{y} & \text{otherwise} \end{cases}$$

The prediction $\hat{y}$ for F_i given training data $\mathcal{X}$ is found by:

$$\hat{y} = arg \max_{y \in \{\pm 1\}} \sum_{j=1}^{n} y_j h_i(x_j)$$

This criteria is to perform a majority voting to obtain prediction of the decision stamp based on the percentage of positive (or negative) graphs where the feature occurs. gBoosting [100] uses a similar strategy to construct base learners.

3.5.2 Feature Graph Construction

One challenge of processing graph data is that there is no natural approach to define the structure relationship of base learners. We notice a few recent studies that are moving towards the direction of defining the relationship among features in graphs and sets. For example in the recently defined graph Graphlet Spectrum kernel [96], the spatial relationship of graph feature (called graphlets) are explored in an algebraic framework for measuring the structure similarity of graph adjacency matrices. In addition, recently developed association net uses a graph model to represent a set of association rules [133]. However, these work could not be directly applied in our current framework since the graphlet spectrum method models the spatial relationship of graphlet in an implicit approach and the association rule net only explore the overlapping relationship of features.

Here we adopted our previous work [42, 43] to construct feature graphs. In [42], we formalize a concept which we called "feature consistency map". A feature consistency map is a undirected graph in which each node represents a feature and each edge encodes the spatial consistency relationship between two features. We measure the minimum distance

between two features using the average shortest path connecting a node in one feature to a node in the other feature. We compute the variance of the minimal distance between the occurrences of the two subgraphs in the training data. If the variance is bellow a threshold, we consider the two features are in a consistent spatial relationship. In our experiment study, we adopt the feature consistency map as an approach to construct a feature graph.

In addition, we also explored the possibility of evaluating the structure-overlapping relationship of features as did in [43]. Towards that end, we compute a kernel function for the set of features. A graph kernel function is a positive semi-definite function that maps graphs to a Hilbert space in order to evaluate the similarity of graphs in the space. Many kernel functions have been designed for graphs and we use the random walk based Marginalized Graph kernel function [87] to compute the kernel function for the set of subgraph features. We convert such kernel matrix to a feature graph where nodes are features and edges are labeled with the inner product (as evaluated with a graph kernel function) of the two features. To avoid a complete connected graph, we use a threshold. If the inner product between two features is less than a threshold, we set the weight of the edge to zero (and hence canceling the edge). The aforementioned approach provides another way to construct a feature graph.

3.6 Experimental Study

We have performed a rigorous evaluation of our algorithm in terms of modeling accuracy and feature selection performance using 6 Protein structure data sets, obtained from [85]. We implemented a prototype of our method in Matlab. We have compared our method with state-of-the-art methods including Support Vector Machine Recursive Feature Elimination (SVM-RFE) [63], gBoosting [100], graph partial least square regression (gPLS) [143], graph classification based Pattern Co-occurrence (COM) [85]. We obtained the SVM-RFE executable along with the *spider machine learning toolbox* from `http://www.kyb.tuebingen.mpg.de/bs/people/spider/`. For gBoosting, we use the gboost toolbox [142]. We obtained

gPLS and COM directly from the original authors of the methods. All the experiments were conducted on a PC with a 2.8Ghz duo core CPU and 3GB memory.

3.6.1 Data sets

To evaluate our methods, we utilized 6 protein-structure graph data sets that were originally studied in [85]. Each data set is a set of geometric graphs representing a set of three-dimensional protein structures. Nodes in such graphs represent amino acids in a protein structure and are labeled with the amino acid type. Edges represent the pairwise Euclidian distance of amino acids (defined between C_α atoms) and are labeled with the discretized distances.

Graphs in the data sets are labeled. Positive samples are sampled from a selected protein family. Negative samples are randomly sampled from the Protein Data Bank. On average a graph contains 250 nodes and 1600 edges. Protein-structure graphs are much larger than chemical-structure graphs, which usually contain about hundreds of nodes and thousands of edges, and contain much large number of patterns. Working with protein structure graphs are hence more challenging for constructing sparse predictive models.

In Table 5.1, we summarize the characteristics of the 6 protein-structure graph data sets. For each data set, we list the data set index, the related protein family ID in the SCOP database [125], the description of the protein family, the number of positive samples and the number of negative samples. See [85] for a comprehensive description of the data collection process.

Table 3.1: Data set: the symbol of the data set. P: total number of positive samples, N: total number of negative samples

Data set	SCOP_ID	Family Name	P	N
P_1	48623	Vertebrate phospholipase A2	29	29
P_2	52592	G proteins	33	33
P_3	48942	C1 set domains	38	38
P_4	56437	C-type lectin domains	38	38
P_5	56251	Proteasome subunits	35	35
P_6	88854	Protein kinases, catalytic subunit	41	41

3.6.2 Experimental Protocol

We use standard cross validation to generate training and testing data sets. We apply FFSM [71] to generating frequent subgraphs from the training data set with $min\_sup = 0.30$ and with subgraph size between 2 and 6. Such subgraphs are used as feature for feature based classification(e.g. SVM, SVM_RFE) or as base learners for boosting based classification including gBoosting and our methods.

For SVM_RFE, we encode each graph sample as a binary feature vector, indexed by the mined subgraphs, with values indicate the presence (1) or absence (0) of the related features. We perform feature selection using SVM_RFE and use LibSVM [24] with linear kernel to construct the best model. We use 5-fold cross validation in the training data set to select important parameter C for SVM.

For COM, we set $t_p = 0.3$ and $t_n = 0$ as proposed in [85], where t_p is the minimal positive frequency for a classification rule and t_n represents the maximal negative frequency permitted. For gPLS, we use $min\_sup = 0.3$ and examine the combinations of $n = \{2, 4, 8, 16\}$ and $k = \{2, 4, 8, 16\}$ for optimal setting. For gBoosting, we also set $min\_sup = 0.30$ and search the optimal parameter μ (misclassification cost) in the range of $\{0.04, 0.06, \ldots, 0.18, 0.20\}$. All the parameter selection are based on another 5-fold cross validation on the training data only.

For our own methods, we utilize two approaches to model the spatial correlation of base learners (i.e. subgraphs). The first approach, LPGBK, is to construct a kernel function for the subgraphs, utilizing the the Marginalized kernel [87]. The second approach, LPGBCMP, is to construct the feature consistency map, as investigated in [42]. We fix $max\_var = 1$ for feature consistency map building threshold and $\delta = 0.25$ for overlapping threshold. Empirical study shows that there is no significant change if we change these two parameters within a wide range. Further details of the two spatial correlation computation methods can be found in [42, 43].

Below we summarize the model construction and model evaluation.

Model Construction. For each data set, we partition the data set into 5-folds to perform 5-fold cross-validation (CV) with 4 folds for training and 1 fold for testing. We use another 5-fold CV on the training data set to select the optimal parameters for each method. We then generate a single model from the entire training set with the selected parameters and apply the model to the testing data set for prediction.

Model Comparison. For model comparison, we collect the sensitivity (TP/(TP+FN)), specificity (TN/(TP+FP)) and accuracy ((TP+TN)/S) of the trained model, where TP stands for true positive, FP stands for false positive, TN stands for true negative, FN stands for false negative, and S stands for the total number of samples. All the values reported are collected from the testing data set only and are averaged across 10 replicates of the 5-fold cross validation in a total of 50 experiments.

3.6.3 Classification Performance

In this subsection, we show the performance of our methods compared with SVM-RFE, gPLS, gBoosting and COM. The accuracy is shown in Fig 3.2. Since the standard deviation is around 2%-5% for all these methods, we do not list it here.

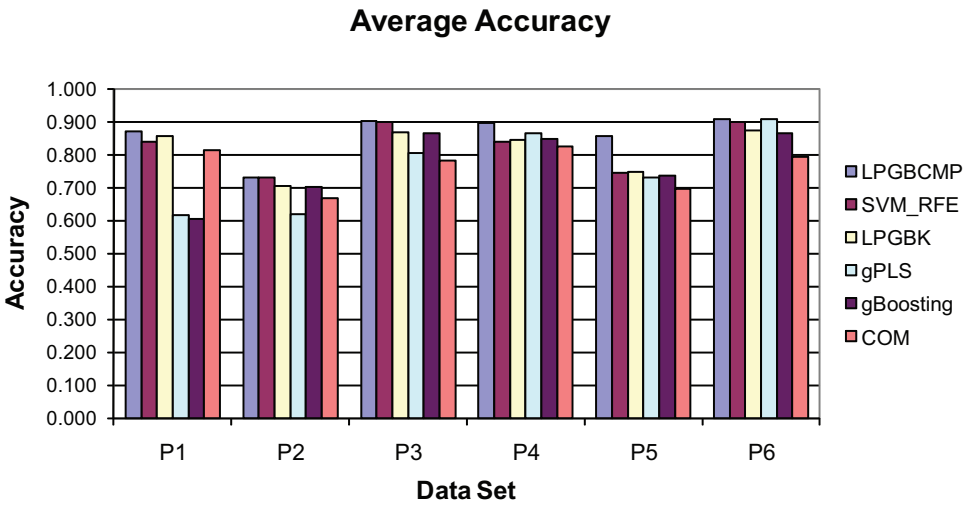


Figure 3.2: Accuracy comparison of on 6 data sets.

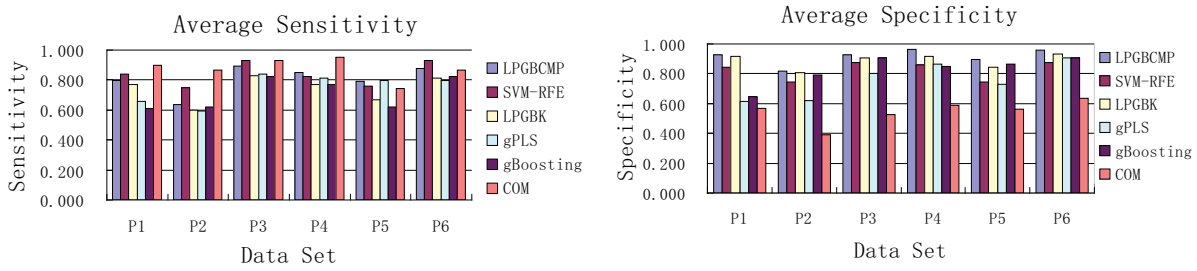


Figure 3.3: Left: Sensitivity comparison. Right: Specificity comparison

In Fig 3.2, we observe that the accuracy of all these methods has the same trend with different data sets. gBoosting and gPLS have comparable performance in the 6 data sets. SVM_RFE outputs gBoosting, gPLS, and COM in three out of six data sets and have comparable performance for the rest. Comparing two versions of our methods, LPGBCMP outperforms LPGBK on all data sets. In fact LPGBCMP performs best among all the evaluated data sets though the margin may be small for 3 data sets when compared with SVM_RFE.

To better understand the accuracy differences, we plot the average sensitivity and average specificity of all methods in Fig 3.3. It is clear that COM provides the best sensitivity among the majority of data sets. COM utilizes a rule-based classification algorithm where it classifies a graph sample as positive if a co-occurrence pattern-rule is satisfied. This algorithm is not specific enough, as compared to other methods (shown in the right panel of Fig 3.3). Interesting enough, all boosting based methods, including gBoosting, LPGBCMP, and LPGBK, have very high specificity comparing to the rest of the methods. Overall, the regularized boosting methods such as LPGBCMP and LPGBK seem to have a good compromise between specificity and sensitivity. This observation provides experimental evidence supporting our hypothesis that structure information among base classifier should be considered in order to build a highly accurate predictive model for semi-structure data such as graphs.

3.6.4 Grouping Selection Effect and Stable Spatial Distribution

To evaluate the capability of the LPGBCMP algorithm for selecting grouped base learners, we visualize the spatial distribution of selected base learners in original graphs. By ranking the base learners by the learned coefficients, we select the top three features for LPGBCMP and gBoosting. We plot the embedding of the three subgraphs in two proteins: protein 1EGI and protein 1H8U belonging to the same protein family in Fig 3.5. We rotate the protein structures and highlight the occurrence of the features with circles for a better demonstration.

In Fig 3.5, the upper row shows the spatial distribution of the top three features for LPGBCMP in two proteins and the lower row shows the distribution for the top three features from gBoosting. Each column uses the same protein for demonstration. From Fig 3.5, we observe that F_1, F_2 and F_3 from LPGBCMP have a consistent spatial distribution on the two proteins. F_1 and F_2 are clustered and both are close to F_3. In contrast, features from gBoosting do not have a stable spatial distribution in the two proteins. The observation supports our claim that our method can select grouped features with stable spatial distribution among the graph data.

3.6.5 Method Robustness

A common concern with boosting is that the method is usually sensitive to outliers and errors in the training data set due to the exponential loss function. We use logit loss function that is less sensitive to outliers. However, as claimed in [116], any convex loss function may degenerate to random guess with a certain level of random classification noise. L_2 regularization in linear regression has been shown to stabilize the learning function [66]. In our algorithm design, we used the Laplacian based L_2 regularization and this may reduce the boosting algorithms' sensitivity to outliers and random classification noise. To test the robustness of our method experimentally, we singled out the P_4 data set and performed 5 fold cross validation with class label errors. In particular, for each fold, we change certain percentage of the class labels in the training data, train a model with changed training data,

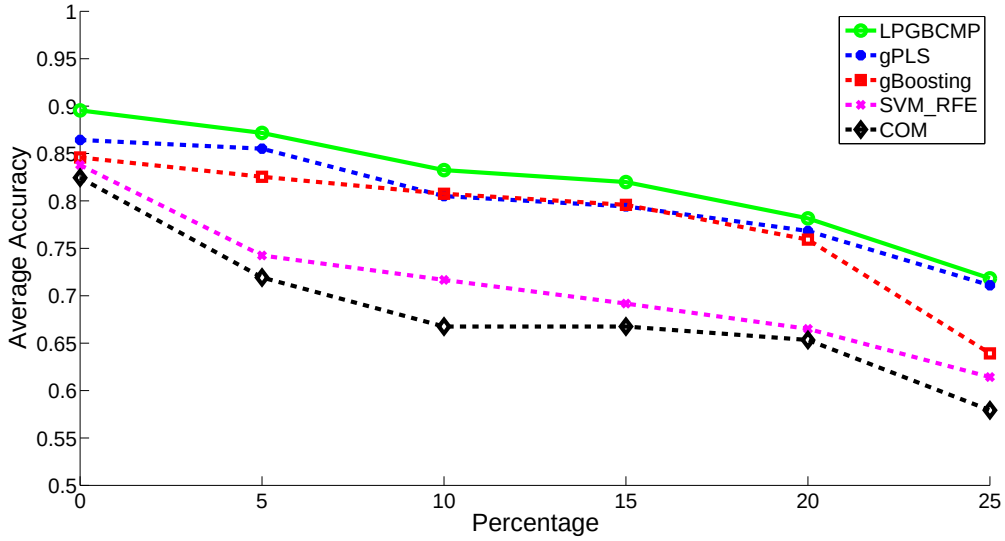


Figure 3.4: Average accuracy with different percentage of flipped training labels

and apply trained model to normal test data. In Fig 3.4, we report the average accuracy with error rate ranging from 0% to 25% for LPGBCMP, gPLS, gBoosting, SVM_RFE and COM.

From Fig 3.4, a clear trend is that the accuracy of all methods decreases as more errors are introduced in the training data set. There is a sharp deceasing from 0 to 5% for SVM_RFE and COM. The regularized boosting method remains the best over all the settings, even though the performance gain is not significant. From the test, we conclude that LPGBCMP is at least as sensitive (if not less) to noises as other classifiers including SVM and partial least square based methods.

In addition, we evaluate the robustness of the regularized boosting algorithm by changing different parameter values. Among the parameters that may affect the performance of the regularized boosting algorithm, we test the parameter max_var, which is used to derive the feature consistency map. With a large value of max_var, the edge number of feature consistency map increases and with smaller value of max_var, the edge number of feature consistency map decreases.

Fig 3.6 indicates average accuracy on 5 fold cross validation for each value of threshold max_var from 0.5 to 8. From the result, we observe that the accuracy remains stable within

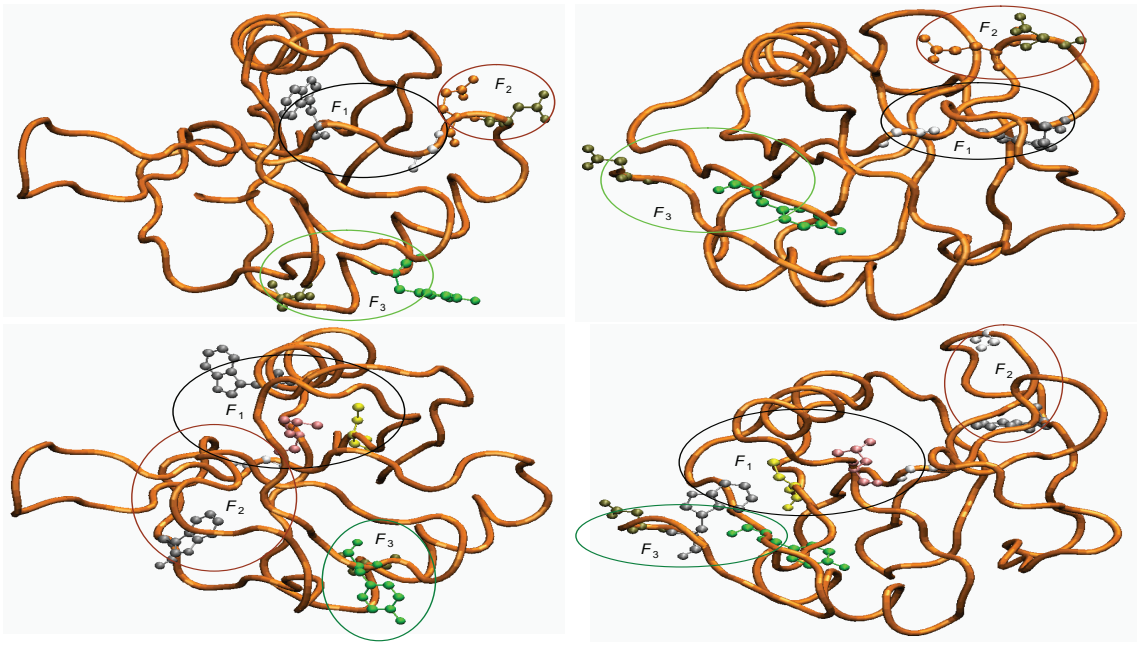


Figure 3.5: Top Left: Spatial distributions of the top 3 features from LPGBCMP in protein 1EGI. Top Right: Spatial distributions of the same 3 features from LPGBCMP in protein 1H8U. Lower Left: Spatial distributions of the top 3 features from gBoosting in protein 1EGI. Lower Right: Spatial distributions of the same 3 features from gBoosting in protein 1H8U.

a relatively wide range of threshold and the best accuracy can be obtained around 1 to 2. Furthermore, the relationship between the performance and parameter *mar_var* is revealed. When *max_var* is quite small, the structure information among features is ignored and our method will degenerate to regular logit boosting with elastic net regularization [197]; when *max_var* is large, the feature graph will be a complete graph and our method may possibly introduce less discriminative features hence undermine the performance.

Overall, the regularized boosting method is effective and achieves good accuracy within a wide range parameters and a certain number of outliers.

3.7 Conclusions

In this paper, we presented a novel boosting algorithm that considered the structure relationship of base learners in the functional space. We model the structure relationship as an undirected graph and incorporate such information by introducing a L_2 norm regularized

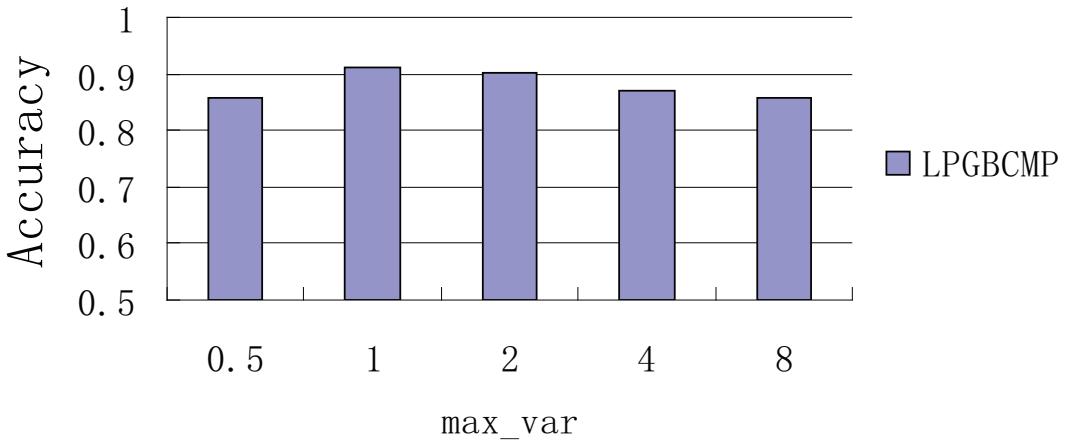


Figure 3.6: Average accuracy with different *max_var*

graph Laplacian to standard boosting formalization. Though the new algorithm may be applied to many applications, we specifically focus on constructing supervised graph learning models in this paper. Using a comprehensive experimental study with protein structure graphs and comparing with current state-of-the-art, we demonstrate that the new algorithm selects clustered features with stable spatial relationship, and achieves better predictive performance.

Chapter 4

Preliminary Study II: Structured Joint Sparse Principal Component Analysis

4.1 Introduction

Determining anomalies in data streams that are collected and transformed from various types of networks has recently attracted significant research interest in the data mining community [19, 77, 151, 188]. Applications of the work could be found in network traffic data [188], sensor network streams [19], social networks [151], cloud computing [128], and finance networks [77] among others.

The common limitation of aforementioned methods is that they are incapable of determining the sources that contribute most to the observed anomalies, or *anomaly localization*. With fast-accumulating stream data, an outstanding data analysis issue is *anomaly localization*, where we aim to discover the specific sources that contribute most to the observed anomalies. Anomaly localization in network data streams is apparently critical to many applications, including monitoring the state of buildings [175], or locating the sites for flooding

and forest fires [51]. In the stock market, pinpointing the change points in a set of stock price time series is critical for making intelligent trading decisions [114]. For network security, localizing the sources of the most serious threats in computer networks helps ensure security in networks [101].

Principal Component Analysis (PCA) is arguably the most widely applied unsupervised anomaly detection technique for network data streams [101, 72, 102]. However, a fundamental problem of PCA, as claimed in [139], is that the current PCA based anomaly detection methods can not be applied to anomaly localization. We believe that the major obstacle for extending PCA techniques to anomaly localization lies in the mixed nature of the abnormal space. In particular, the projection of the data streams in the abnormal subspace is a combination of data from all the sources, which makes any localization difficult. Our key observation is that if we manage to identify a low dimensional approximation of the abnormal subspace using a subset of sources, we "localize" the abnormal sources. The starting point of our investigation hence is the recently studied sparse PCA framework [196] where PCA is formalized in a sparse regression problem where each principle component (PC) is a sparse linear combination of the original sources. However, sparse PCA does not fit directly into our problems in that sparse PCA enforces sparsity randomly in the normal and abnormal subspaces. In this paper, we explore two directions in improving sparse PCA for anomaly detection and localization.

First, we develop a new regularization scheme to simultaneously calculate the normal subspace and the sparse abnormal subspace. In the normal subspace, we do not add any regularization but use the same normal subspace as ordinary PCA for anomaly detection. In the abnormal subspace, we enforce that different PCs share the same sparse structure hence it is able to do anomaly localization. We call this method *joint space PCA* (JSPCA).

Second, we observe that abnormal streams are usually correlated to each other. For example in stock market, index changes in different countries are often correlated. For incorporating stream correlation in anomaly localization we design a *graph guided sparse*

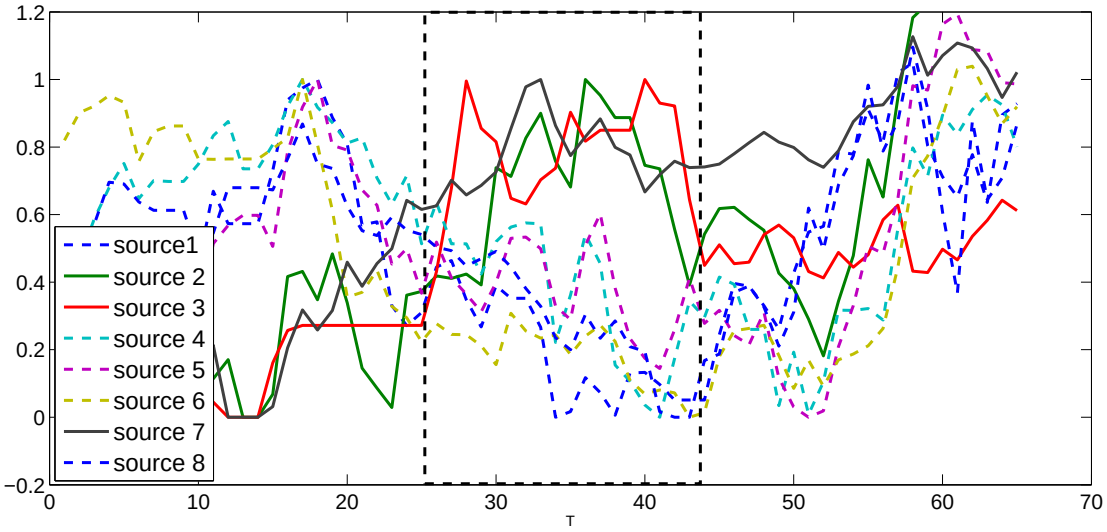


Figure 4.1: Illustration of time-evolving stock indices data. Index 2,3,7 in solid lines are abnormal.

PCA (GJSPCA) technique. Our experimental studies demonstrate the effectiveness of the proposed approaches on three real-world data sets from financial markets, wireless sensor networks, and machinery operating condition studies.

A major drawback of PCA based anomaly detection methods is that the performance of the methods is very sensitive to the number of PCs representing the normal subspace. In order to overcome this problem, we introduce a multi-dimensional Karhunen Loève Expansion (KLE) as an extension of PCA (one dimensional KLE) to consider the spatial correlation among different sources and the temporal correlation among different time stamps [17]. The corresponding methods are named *joint space KLE* (JSKLE) and *graph guided sparse KLE* (GJSKLE) respectively. The experimental results demonstrate that the JSKLE and GJSKLE effectively stabilize localization performance when changing the number of PCs representing the normal subspace.

As an example of anomaly detection and anomaly localization in network data streams, we show the normalized stock index streams of eight countries over a period of three months in Figure 4.1. We notice an anomaly in the marked window between time stamps 25 and 42. In that window sources 1, 4, 5, 6, 8 (denoted by dotted lines) are normal sources. Sources 2, 3, 7 (denoted by solid lines) are abnormal ones since they have a different trend from that

of the other sources. In the marked window, the three abnormal sources clearly share the same increasing trend while the rest share a decreasing trend.

4.2 Related Work

Existing work on anomaly localization from network data streams could be roughly divided into two categories: those at the source level and those at the network level. The source level anomaly localization approaches embed detection algorithm at each stream source, resulting in a fully distributed anomaly detection system [128, 62, 103]. The major problem of these approaches is that source level anomalies may not be indicative of network level anomalies due to the ignorance of the rest of the network [72].

To improve source level anomaly localization methods, several algorithms have been recently proposed to localize anomaly at the network level. Brauckhoff [16] applied association rule mining to network traffic data to extract abnormal flows from the large set of candidate flows. Their work is based on the assumption that anomalies often result in many flows with similar characteristics. Such an assumption holds in network traffic data streams but may not be true in other data streams such as finance data. Keogh *et al.*[92] proposed a nearest neighbor based approach to identify abnormal subsequences within univariate time series data by sliding windows. They extracted all possible subsequences and located the one with the largest Euclidean distance from its closest non-overlapping subsequences. However, the method only works for univariate time series generated from a single source. In addition, if the data is distributed on a non-Euclidean manifold, two subsequences may appear deceptively close as measured by their Euclidean distance [160]. L. Fong *et al.*developed a nonparametric change-point test based on U-statistics to detect and localize change-points in high-dimensional network traffic data [119]. The limitation is that the method is specifically designed for the Denial of Service (DOS) attack in communication networks and cannot be generalized to other types of network data streams easily.

Closely related to our work, Ide *et al.*[75, 76] measured the change of neighborhood graph for each source to perform anomaly localization and developed a method called Stochastic Nearest Neighbor (SNN). Hirose *et al.*[69] designed an algorithm named Eigen Equation Compression (EEC) to localize anomalies by measuring the deviation of covariance matrix of neighborhood sources. In these two studies, we have to build a neighborhood graph for each source for each time interval, which is unlikely to scale to a large number of sources. Another closely related work to ours is the Stream Projected Outlier Detector (SPOT) [189], in which a subspace is learned with genetic algorithm from a potential huge number of subsets of sources and the outliers in temporal domain are detected in the reduced subspace. The limitation of their work is that they used a genetic algorithm to select a subset of sources. The computational complexity to find the optimum set grows exponentially with the number of features and there is no guarantee that we will reach the optimal subset of sources. Our work formalizes anomaly localization via a sparse regularization framework and solved it efficiently with convex optimization technique. Furthermore, the anomalies may not be observable in the original space. Instead of coping with original space, we localize anomalies in abnormal subspace in which the anomalous behaviors of data are significant. Compared with [189], Yang *et al.*[182] learned the subspace with locally linear embedding and PCA and then detect outliers in the reduced space. However, there is no mapping between the newly learned space and original data space therefore it is not applicable for anomaly localization. Cao *et al.*[21] partitioned data streams within a window into clusters based on their similarity and outliers were detected on each individual cluster. A stream is a outlier if the number of streams lies within a predefined distance is smaller than k. However, if normal instances do not have enough close neighbors or if the abnormal instances that have enough close neighbors, the technique may have high level of false positive and false negative.

We have investigated the anomaly localization problem in our previous publications [83, 84]. In [83], we proposed a two step approach where we first compute normal subspace from

ordinary PCA and then derive a sparse abnormal subspace on the residual data subtracted from the original data. The critical limitation of the two stage method is that after removing the abnormal subspace, the resulting data is a linear combination of all the sources. It is very difficult to identify which sources contributes most to the observed anomaly. In [84], we designed a single step approach to jointly learn normal subspace for anomaly detection and sparse abnormal subspace for anomaly localization. In this paper, we substantially extended [84] by generalizing our proposed joint sparse PCA framework to Karhunen-Loeve Expansion (KLE). KLE considers both temporal and spatial correlation of data and it has been shown to reduce the sensitivity from the choice of number of PCs [17]. We also extended the experiment study by adding one more data set. Our experimental studies demonstrate the effectiveness of the proposed method over the state-of-the-art.

Principal Component Analysis (PCA) is extensively applied to network data streams anomaly detection [101, 72, 102]. For example, Lakhina et al. applied PCA to detect network traffic anomalies. Huang [72] developed a distributed PCA anomaly detector by equipping a local filter in each source. Brauckhof [17] considers both the temporal and spatial correlation of streamed data by extending PCA to Karhunen-Loeve Expansion (KLE) and solve the sensitivity problem of PCA proposed by Ringberg [139]. The major limitation of these works, as pointed out in in [139], is that PCA can not be applied to anomaly localization.

4.3 Preliminaries

We introduce the notations used in this paper and background information regarding PCA and sparse PCA.

Table 4.1: Notations in the paper.

Symbol	Notation
$\mathcal{S}$	a set
$\mathbf{X}$	a matrix
x_{ij}	the entry of the ith row and the jth column of matrix $\mathbf{X}$
$\mathbf{x}$	a column vector $\mathbf{x}$
x_i	the ith entry of the vector $\mathbf{x}$
$\mathbf{x}_i$	the ith column of the matrix $\mathbf{X}$

4.3.1 Notation

We use bold uppercase letters such as $\mathbf{X}$ to denote a matrix and bold lowercase letters such as $\mathbf{x}$ to denote a vector. Greek letters such as λ_1, λ_2 are Lagrangian multipliers. $\langle \mathbf{A}, \mathbf{B} \rangle$ represents the matrix inner product defined as $\langle \mathbf{A}, \mathbf{B} \rangle = tr(\mathbf{A}^T\mathbf{B})$ where tr represents the matrix trace. Given a matrix $\mathbf{X}$ we use x_{ij} to denote the entry of $\mathbf{X}$ at the ith row and jth column. We use x_i to represent the ith entry of a vector $\mathbf{x}$. $||\mathbf{x}||_p = (\sum_{i=1}^{n} |x_i|^p)^{\frac{1}{p}}$ denotes the l_p norm of the vector $\mathbf{x} \in \mathcal{R}^n$. Given a matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^T \in \mathcal{R}^{n \times p}$, $\|\mathbf{X}\|_{1,q} = \sum_{i=1}^{n} \|\mathbf{x}_i\|_q$ is the l_1/l_q norm of the matrix $\mathbf{X}$, where $\tilde{\mathbf{x}}_i$ is the ith row of $\mathbf{X}$ in column vector form. Unless stated otherwise, all vectors are column vectors. In Table 4.1, we summarize the notations in our paper.

4.3.2 Network Data Streams

Our work focuses on data streams that are collected from multiple sources. We call the set of data stream sources together as a network since we often have information regarding the structure of the sources.

Following [34], *Network Data Streams* are multi-variate time series $\mathcal{S}$ from p sources where $\mathcal{S} = \{S_i(t)\}$ and $i \in [1, p]$. p is the dimensionality of the network data streams. Each function $S_i : \mathcal{R} \to \mathcal{R}$ is a *source*. A source is also called a "node" in the communication network community and a "feature" in the data mining and machine learning community.

Typically we focus on time series sampled at (synchronized) discrete time stamps $\{t_1, t_2, \ldots, t_n\}$.

In such cases, the network data streams are represented as a matrix $\mathbf{X} = (x_{i,j})$ where $i \in [1, n]$, $j \in [1, p]$ and $x_{i,j}$ is the reading of the stream source j at the time sample t_i.

4.3.3 Applying PCA for Anomaly Localization

Our goal is to explore a Principal Component Analysis (PCA) based method for performing anomaly detection and localization simultaneously. PCA based anomaly detection technique has been widely investigated in [101, 72, 102]. In applying PCA to anomaly detection, one first constructs the normal subspace $\mathbf{V}^1$ by the top k PCs and the abnormal subspace $\mathbf{V}^2$ by the remaining PCs, then projects the original data on $\mathbf{V}^{(1)}$ and $\mathbf{V}^{(2)}$ as:

$$\mathbf{X} = \mathbf{X}\mathbf{V}^{(1)}\mathbf{V}^{(1)^T} + \mathbf{X}\mathbf{V}^{(2)}\mathbf{V}^{(2)^T} = \mathbf{X}_n + \mathbf{X}_a, \tag{4.1}$$

where $\mathbf{X} \in \mathcal{R}^{n \times p}$ is the data matrix with n time stamps from p data sources, $\mathbf{X}_n$ and $\mathbf{X}_a$ are the projections of $\mathbf{X}$ on normal subspace and abnormal subspace respectively. The underlying assumption of PCA based anomaly detection is that $\mathbf{X}_n$ corresponds to the regular trends and $\mathbf{X}_a$ captures the abnormal behaviors in the data streams. By performing statistical testing on the squared prediction error $SPE = tr(\mathbf{X}_a^T \mathbf{X}_a)$, one determines whether an anomaly happens [101, 72]. The larger SPE is, the more likely an anomaly exists.

Although PCA has been widely studied for anomaly detection, it is not applicable for anomaly localization. The fundamental problem, as claimed in [139], lies in the fact that there is no direct mapping between two matrices $\mathbf{V}_{(1)}$, $\mathbf{V}_{(2)}$ and the data sources. Specifically, let $\mathbf{V}^{(2)}$ be the last $p - k$ PCs that spans the abnormal subspace, $\mathbf{X}_a$ is essentially an aggregated operation that performs linear combination of all the data sources, as follows:

$$\mathbf{X}_a = \mathbf{X}\mathbf{V}^{(2)}\mathbf{V}^{(2)^T} = \left\{ \sum_{j=1}^{p} \mathbf{x}_j \tilde{\mathbf{v}}_j^T \tilde{\mathbf{v}}_i \right\}_{i=1,\cdots,p} \tag{4.2}$$

where $\mathbf{x}_j$ is the data from the jth source and $\tilde{\mathbf{v}}_j$ is the transpose of the jth row of $\mathbf{V}^{(2)}$.

Considering the ith column of $\mathbf{X}_a$: $\sum_{j=1}^{p} \mathbf{x}_j \tilde{\mathbf{v}}_j \tilde{\mathbf{v}}_i^T$, there is no correspondence between the original ith column of $\mathbf{X}$ and ith column of $\mathbf{X}_a$. Such an aggregation makes PCA difficult to identify the particular sources that are responsible for the observed anomalies.

Although all the previous works claim PCA based anomaly detection methods *cannot* do localization, we solve the problem of anomaly localization in a reverse way. Instead of locating the anomalies directly, we filter normal sources to identify anomalies by employing the fact that normal subspace captures the general trend of data and normal sources have little or no projection on abnormal subspace. The following provides a sufficient condition for data sources to have no projection on abnormal subspace.

Suppose $\mathcal{I} = \{i | \tilde{v}_i = \mathbf{0}\}$ is the set that contains all the indices for the zero rows of $\mathbf{V}^{(2)}$, then $\forall i \in \mathcal{S}$, $\mathbf{x}_i$ has no projection on the abnormal subspace. In other words, these sources have no contribution to the abnormal behavior. Let $\mathbf{V}^{(2)} = [\tilde{\mathbf{v}}_1, \tilde{\mathbf{v}}_2, \cdots, \tilde{\mathbf{v}}_p]^T$ and consider the squared prediction error $SPE = tr(\mathbf{X}_a^T \mathbf{X}_a)$ and plug equation (4.2) in:

$$\begin{aligned} tr(\mathbf{X}_a^T \mathbf{X}_a) &= tr(\mathbf{X}_a \mathbf{X}_a^T) \\ &= tr(\mathbf{V}_2^T X^T X \mathbf{V}_2) \\ &= tr((\textstyle\sum_{j=1}^{p} \mathbf{x}_j \tilde{\mathbf{v}}_j^T)^T (\sum_{j=1}^{p} \mathbf{x}_j \tilde{\mathbf{v}}_j^T)) \end{aligned} \tag{4.3}$$

From equation (4.3), it is clear that $\forall i \in \mathcal{I}$, the data $\mathbf{x}_i$ from the source i has no projection on the abnormal subspace and hence could be excluded from the statistics used for anomaly detection. We call such a pattern with an entire row of zeros *"joint sparsity"*.

Unfortunately ordinary PCA does not afford sparsity in PCs. Sparse PCA is a recently developed algorithms where each PC is a sparse linear combination of the original sources [196]. However existing sparse PCA method has no guarantee that different PCs share the same sparse representation and hence has no guarantee for the joint sparsity. To illustrate the point, we plotted the entries of each PC for ordinary PCA (left plot of Figure 4.2) and for sparse PCA (right plot of Figure 4.2) for the stock data set shown in figure 4.1. White blocks indicate zero entries and the darker color indicates a larger absolute loading. Sparse

PCA produces sparse entries but that alone does not indicate sources that contribute most to the observed anomaly.

Below we present our extensions of PCA that enable us to reduce dimensionality in the abnormal subspace.

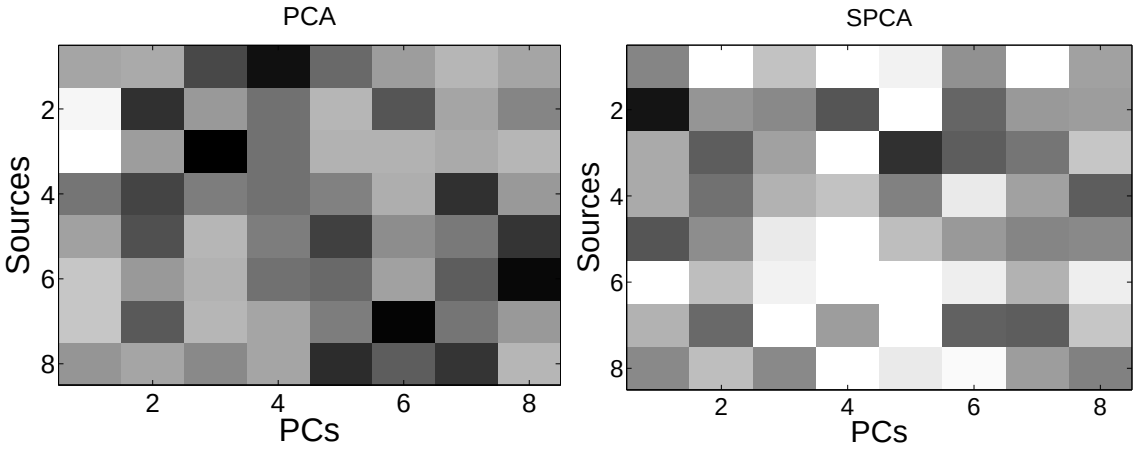


Figure 4.2: Comparing PCA and Sparse PCA. Left: PCA. Right: SPCA.

4.4 Methodology

In this section, we propose a novel regularization framework called joint sparse PCA (JSPCA) to enforce joint sparsity in PCs in the abnormal space while preserving the PCs in the normal subspace so that we can perform simultaneous anomaly detection and anomaly localization. Starting from JSPCA, we proposed two extensions. In the first extension, we consider the network topology in the original data and incorporate such topology into JSPCA and develop an approach called Graph JSPCA (GJSPCA). In the second, we extend JSPCA and GJSPCA to JSKLE and GJSKLE, which taking the temporal correlation into account as well as spatial correlation considered in JSPCA and GJSPCA.

Before formally providing the detailed methods, we give an overall work flow of our method as shown in Figure 4.3 for JSPCA. Note that the rest methods share the same flow and we only show JSCPA for simplicity.

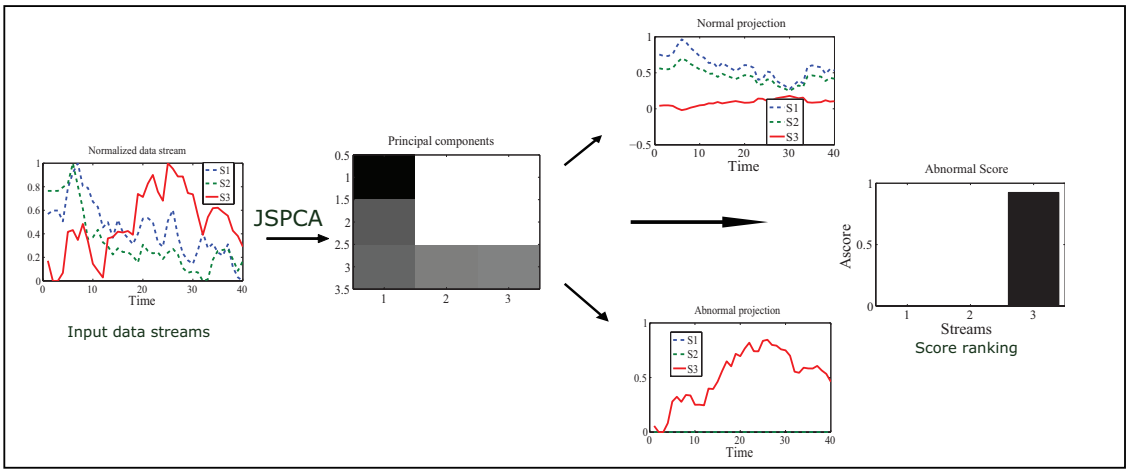


Figure 4.3: Demonstration of the the system architecture of JSPCA on three network data streams with one anomaly (solid line) and two normal streams (dot lines).

Given several data streams, the first step is to calculate a set of principal components with an ordinary normal subspace and abnormal subspace with joint sparsity. Our example is a network with 3 sources, then a 3×3 principal component matrix is calculated by JSPCA. The first principal component with non-zero entries represents the normal subspace. The subtraction between original data and the projection on normal subspace is used for anomaly detection. The remaining two principal components represent abnormal subspace, with the first two rows being zero but the last row being non-zero. Based on the abnormal subspace, the second step is to calculate the abnormal scores. A larger score indicates larger possibility of the corresponding source is abnormal. Therefore, we complete the task of anomaly detection and localization simultaneously.

4.4.1 Joint Sparse PCA

Our objective here is to derive a set of PCs $\mathbf{V} = [\mathbf{V}^{(1)}, \mathbf{V}^{(3)}]$ such that $\mathbf{V}^{(1)}$ is the normal subspace and $\mathbf{V}^{(3)}$ is a sparse approximation of the abnormal subspace with the joint sparsity.

The following regularization framework guarantees the two properties simultaneously:

$$\begin{aligned} \min_{\mathbf{V}^{(1)},\mathbf{V}^{(3)}} \quad & \frac{1}{2}||\mathbf{X} - \mathbf{X}\mathbf{V}^{(1)}\mathbf{V}^{(1)^T} - \mathbf{X}\mathbf{V}^{(3)}\mathbf{V}^{(3)^T}||_F^2 + \lambda||\mathbf{V}^{(3)}||_{1,2} \\ \text{s.t.} \quad & \mathbf{V}^T\mathbf{V} = I_{p\times p}. \end{aligned} \tag{4.4}$$

Using one variable $\mathbf{V}$, we simplify equation (4.4) as:

$$\begin{aligned} \min_{\mathbf{V}} \quad & \frac{1}{2}||\mathbf{X} - \mathbf{X}\mathbf{V}\mathbf{V}^T||_F^2 + \lambda||\mathbf{W}\circ\mathbf{V}||_{1,2} \\ \text{s.t.} \quad & \mathbf{V}^T\mathbf{V} = I_{p\times p}. \end{aligned} \tag{4.5}$$

Here $\circ$ is the *Hadamard product* operator (entry-wise product), λ is a scalar controlling the balance between sparse and fitness, $W = [\tilde{\mathbf{w}}_1, \cdots, \tilde{\mathbf{w}}_p]^T$ with $\tilde{\mathbf{w}}_j$ is defined below:

$$\tilde{\mathbf{w}}_j = [\underbrace{0, \cdots, 0}_{k}, \underbrace{1, \cdots, 1}_{p-k}]^T, \quad j = 1, \cdots, p. \tag{4.6}$$

The regularization term $\|\mathbf{W}\circ\mathbf{V}\|_{1,2}$ is called group lasso penalty [187], in which L_2 norm is used to aggregate the coefficients within a group and L_1 norm is applied to achieve sparsity among groups. In our framework, each group is corresponding to a row of the abnormal subspace matrix $\mathbf{V}^{(3)}$ and L_1/L_2 penalty enforces joint sparsity for each source across the abnormal subspace.

The major disadvantage of equation (4.5) is that it poses a difficult optimization problem since the first term (the trace norm) is concave and the second term (the L_1/L_2 norm) is convex. The similar situation was first investigated in sparse PCA [196] with elastic net penalty [197], in which two variables and an alternative optimization algorithm were introduced. Here we share the first least square loss term but adopt a different regularization term. Motivated by [196], we consider a relaxed version:

$$\begin{aligned} \min_{A,B} \quad & \frac{1}{2}||\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}^T||_F^2 + \lambda||\mathbf{W}\circ\mathbf{B}||_{1,2} \\ \text{s.t.} \quad & \mathbf{A}^T\mathbf{A} = I_{p\times p}, \end{aligned} \tag{4.7}$$

where $\mathbf{A}, \mathbf{B} \in \mathcal{R}^{p \times p}$. The advantage of the new formalization is two folds: first, equation (4.7) is convex to each subproblem when fixing one variable and optimizing the other. As asserted in [196] disregarding the Lasso penalty, the solution of equation (4.7) corresponds to exact PCA; second, we only impose penalty on the remaining $p - k$ PCs and preserve the top k PCs representing the normal subspace from ordinary PCA. Such a formalization will guarantee that we have the ordinary normal subspace for anomaly detection and the sparse abnormal subspace for anomaly localization. Note that Jenatton *et al.*recently proposed a structured sparse PCA [81], which is similar to our formalization. But their structure is defined on groups and cannot be directly applied for anomaly localization.

Figure 4.4 (left) demonstrates the principal components generated from JSPCA for the stock market data shown in figure 4.1. Joint sparsity across the PCs in abnormal subspace pinpoints the abnormal sources 2,3,7 by filtering out normal sources 1, 4, 5, 6, 8. Such result matches the truth in figure 4.1.

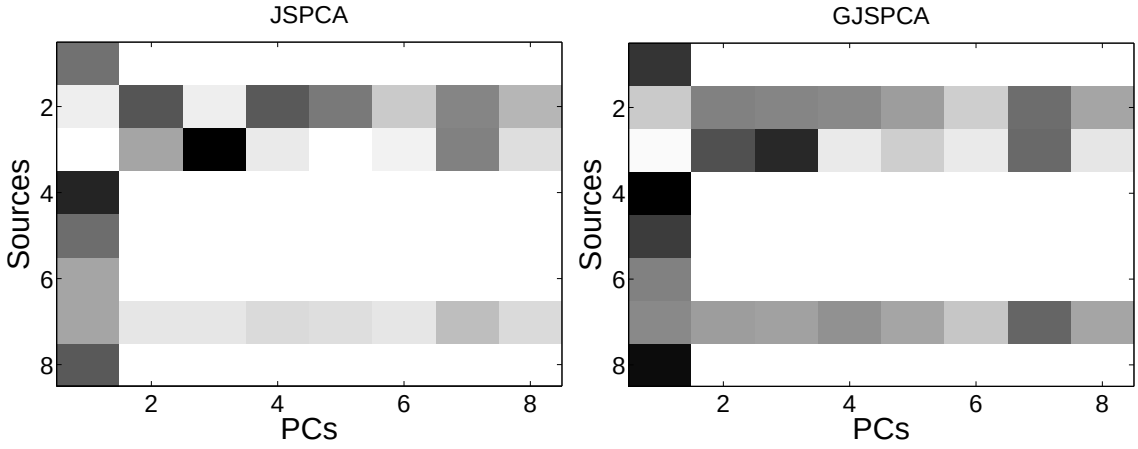


Figure 4.4: Comparing *joint sparse PCA* (JSPCA) and *graph joint sparse PCA* (GJSPCA). Left: JSPCA; Right: GJSPCA.

4.4.2 Anomaly Scoring

To quantitatively measure the degree of anomalies for each source, we define anomaly score and normalized anomaly score as following.

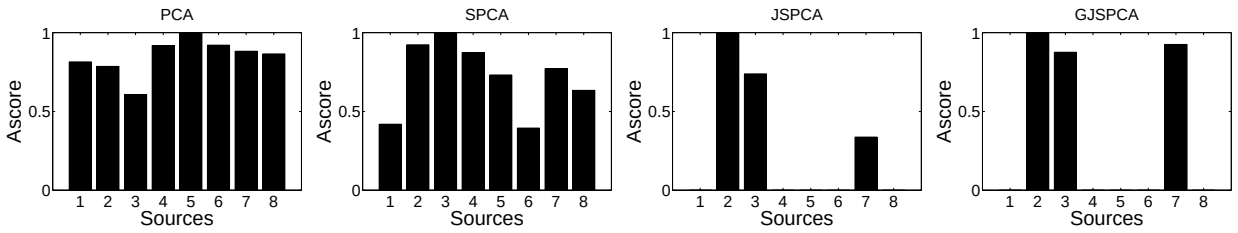


Figure 4.5: Comparing different anomaly localization methods. From left to right: PCA, sparse PCA, JSPCA, and GJSPCA.

Definition 4.4.1. *Given p sources and the abnormal subspace $\mathbf{V}^{(3)} = [\mathbf{v}_{k+1}, \cdots, \mathbf{v}_p]$ from JSPCA, the anomaly score for source i, $\quad i = 1 \cdots p$ is defined on the L_1 norm of the ith row of $\mathbf{V}^{(3)}$, divided by the size of the row:*

$$\zeta_i = \frac{\sum_{j=k+1}^{p} |\tilde{v_{ij}}|}{p-k}, \tag{4.8}$$

where $\tilde{v_{ij}}$ is the ith entry of $\mathbf{v}_j$.

For each input data matrix $\mathbf{X}$, (4.8) results in a vector $\zeta = [\zeta_1, \cdots, \zeta_p]^T$ of anomaly scores. The normalized score for source i is defined as $\tilde{\zeta_i} = \zeta_i / \max\{\zeta_i, i = 1, \cdots p\}$.

A higher score indicates a higher probability that a source is abnormal. We show the anomaly scores obtained from PCA, SPCA, JSPCA, for the stock data in Figure 4.5. JSPCA succeeds to localize three anomalies by assigning nonzero scores to anomalous sources and zero to normal ones, while PCA and SPCA both fail. With abnormal scores, we can rank abnormality or generate ROC curve to evaluate localization performance. Bellow, we give a skeleton of algorithm for computing abnormal score and the detailed optimization algorithm is introduced later.

Algorithm 2 Anomaly Localization with JSPCA

1: Input: $\mathbf{X}$, k and λ_1.
2: Output: anomaly scores.
3: Calculate a set of PCs $\mathbf{V} = [\mathbf{V}^{(1)}, \mathbf{V}^{(3)}]$ (matrix $\mathbf{B}$ in equation (4.7)), $\mathbf{V}^{(1)}$ is normal subspace, $\mathbf{V}^{(3)}$ is abnormal subspace with joint sparsity;
4: Compute abnormal score for each source by the definition (4.4.1);

4.4.3 Graph Guided Joint Sparse PCA

In many real-world applications, the sources generating the data streams may have structure, which may or may not change with time. As the example mentioned in figure 4.1, stock indices from source 2, 3 and 7 are closely correlated over a long time interval. If source 2 and 3 are anomalies as demonstrated in left Figure 4.4, it is very likely that source 7 is an anomaly as well. This observation motivates us to develop a regularization framework that enforce smoothness across features. In particular, we model the structure among sources with an undirected graph, where each node represents a source and each edge encodes a possible structure relationship. We hypothesize that incorporating structure information of sources we can build a more accurate and reliable anomaly localization model. Below, we introduce the graph guided *joint sparse* PCA, which effectively encodes the structure information in the anomaly localization framework.

To achieve the goal of smoothness of features, we add an extended l_2 (Tikhonov) regularization factor on the graph laplacian regularized matrix norm of the $p-k$ PCs. This is an extension of the l_2 norm regularized Laplacian on a single vector in [44]. With this addition, we obtain the following optimization problem:

$$\begin{aligned} \min_{\mathbf{A},\mathbf{B}} \quad & \frac{1}{2}||\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}^T||_F^2 + \lambda_1\|\mathbf{W} \circ \mathbf{B}\|_{1,2} + \\ & \tfrac{1}{2}\lambda_2 tr((\mathbf{W} \circ \mathbf{B})^T L(\mathbf{W} \circ \mathbf{B})) \\ \text{s.t.} \quad & \mathbf{A}^T\mathbf{A} = I_{p\times p}, \end{aligned} \tag{4.9}$$

where L is the *Laplacian* of a graph that captures the correlation structure of sources [44].

In Figure 4.4 we show the comparison of applying JSPCA and GJSPCA on the data shown in figure 4.1. Both JSPCA and GJSPCA correctly localize the abnormal sources 2,3,7. Comparing JSPCA and GJSPCA, we observe that in GJSPCA the entry values corresponding to the three abnormal sources 2,3,7 are closer (a.k.a. smoothness in the

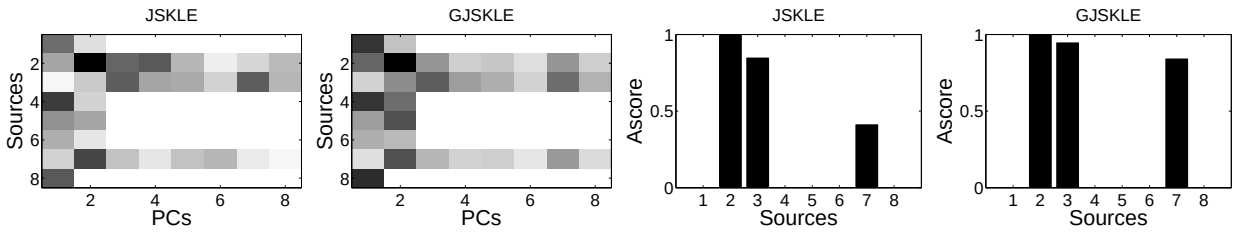


Figure 4.6: From left to right: PC space for JSKLE and GJSKLE, abnormal score for JSKLE, and GJSKLE.

feature space). In the raw data, we observe that sources 2,3,7 share an increasing trend. The smoothness is the reflection of the shared trend and helps highlight the abnormal source 7. As evaluated in our experimental study, GJSPCA outperforms JSPCA. We believe that the additional structure information utilized in GJSPCA helps.

The same observation is also shown in Figure 4.5. Comparing JSPCA and GJSPCA we find that JSPCA assigns higher anomaly scores to source 2 and 3 but a lower score to source 7, and GJSPCA has smooth effect on the abnormal scores. It assigns similar scores for the three sources. The similar scores demonstrate the effect of smooth regularization term induced by the graph Laplacian. The smoothness also sheds light on the reason why GJSPCA outperforms JSPCA a little in anomaly localization in our detailed experimental evaluation.

4.4.4 Extension with Karhunen Loève Expansion

A limitation of PCA is it only considers the spatial correlation but ignores the temporal correlation. As an extension of PCA, Karhunen Loève Expansion Karhunen Loève Expansion (KLE) was introduced in to solved this problem in [17] by taking both spatial and temporal correlation into consideration. In [17], Brauckhoff et al. claimed that by extending PCA to KLE, they stabilized the anomaly detection performance and reduced the sensitivity of PCA when changing the number of principal components representing the normal subspace [139]. Since JSPCA and GJSPCA are based on PCA, they both involve the same problem proposed in [139].

In this section, we extend our regularization framework to KLE, called JSKLE and GJSKLE respectively. Our contributions is to formalize a regularized joint sparse PCA with KLE for *localization* and design efficient optimization algorithms to solve the objective with KLE. Our goal is to stabilize localization performance and reduce the localization performance sensitivity. Such advantage will be illustrated in our experimental studies.

KLE was first considered as a representation of a stochastic process on an infinite linear combination of orthogonal functions [57], and usually named as continuous KLE. Later on, discrete KLE was then given [98] and the its one dimensional version (PCA) has been successfully applied to a broad domain of applications [101, 38]. Generalize PCA to KLE amounts for expanding the original data matrix $X \in \mathcal{R}^{n\times p}$ to $X' \in \mathcal{R}^{(n-N+1)\times pN}$ in both spatial and temporal domain as follows:

$$X'^{T} = \begin{bmatrix} x_1(1) & \cdots & x_1(t) & \cdots & x_1(n-N+1) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_1(N) & \cdots & x_1(t+N-1) & \cdots & x_1(n) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_p(1) & \cdots & x_p(t) & \cdots & x_p(n-N+1) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_p(N) & \cdots & x_p(t+N-1) & \cdots & x_p(n) \end{bmatrix} \tag{4.10}$$

where N is the offset moving forward in temporal domain.

Our staring point is a one dimensional stochastic process $x(t)$ with zero mean over time interval $t \in [a, b]$. By the definition of KLE, $x(t)$ admits a decomposition [148]:

$$x(t) = \sum_{i=1}^{\infty} \alpha_i \psi_i(t) \tag{4.11}$$

where α_i are pairwise uncorrelated random variables and the function $\psi_i(t)$ are continuous

orthogonal deterministic functions such that

$$\int_D \psi_i(t)\psi_j(t)dt = \delta_{ij}$$
$$\delta_{ij} = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases} \tag{4.12}$$

Suppose $K_x(t,s)$ is the continuous covariance function of $x(t)$, s.t.: $K_x(t,s) = \mathbf{E}[X(t)X(s))$, ψ_i are eigenfunctions of $K_x(.,.)$ and derived by solving the Fredholm integral equation:

$$\int_a^b K_x(t,s)\psi_j(s)ds = \lambda_i\psi_i(t) \tag{4.13}$$

The uncorrelated random coefficients α_i are calculated as $\alpha_i = \int_a^b x(t)\psi_i(t)dt$.

In real world applications, we can only access to discrete and finite processes. When applying to a discrete and finite process, KLE discretizes the parameter t to obtain the discrete version on temporal domain. Suppose a continuous stochastic process $x(t)$ is sampled at an equal interval $\triangle t$ and a n dimension vector $\mathbf{x}$ is

$$\mathbf{x} = [x(1), x(2) \dots x(n)]^T \tag{4.14}$$

where $n = \frac{b-a}{\triangle t}$. In discrete version, covariance function $K_x(t,s)$ turns into covariance matrix:

$$\Gamma_{xx} = E(\mathbf{x}\mathbf{x}^T) \tag{4.15}$$

To estimate the covariance matrix Γ_{xx}, we use sliding window averaging algorithm as the covariance estimator [120]. In this algorithm, computation of the estimated covariance matrix essentially involves the averaging of outer products of a sliding window over $\mathbf{x}$. More specifically, a window of fixed size N moves forward in $\mathbf{x}$. Each time it forms a N-dimensional vector and the outer product is calculated. Averaging those outer products over all the

vectors yields the estimated covariance matrix.

Definition 4.4.2. *Given a scaler time series* $\mathbf{x}$*, the estimate of covariance matrix* Γ_{xx} *using a sliding window approach is defined as:*

$$\Gamma_{xx} = \sum_{i=1}^{n-N+1} \mathbf{x_i} \mathbf{x_i}^T \tag{4.16}$$

where $\mathbf{x_i} = [x_i, x_{i+1}, \ldots, x_{i+N-1}]^T$ *is the subvector of vector* $\mathbf{x}$ *with length* N*. A normalization factor is ignored, since it is irrelevant for the eigenvectors of* Γ_{xx}*.*

The summation function in (4.16) can be given in matrix format $\Gamma_{xx} = \mathbf{X}^T\mathbf{X}$, with the following expanded data matrix X from a single vector $\mathbf{x}$ in (4.14):

$$\mathbf{X^T} = \begin{bmatrix} x(1) & x(2) & \ldots & x(n-N+1) \\ x(2) & x(3) & \ldots & x(n-N+2) \\ \vdots & \vdots & \ddots & \vdots \\ x(N) & x(N+1) & \ldots & x(n) \end{bmatrix} \tag{4.17}$$

The integral equation (4.13) becomes a matrix eigenvector problem to solve the KLE vector (or principal component) associated with $\mathbf{X}$: $\Gamma_{xx}\psi_i = \lambda_i\psi_i$

The eigenvectors ψ_i capture the temporal correlation of one discrete stochastic process (one stream) while the ordinary PCA we refereed previously, considers the spatial correlation among different streams. In order to take both temporal and spatial correlation into account, we extended KLE from one dimension to multi-dimensions to deal with multiple stochastic processes.

From [148], a p-dimensional stochastic process from p sources is defined: $\mathbf{X} = [\mathbf{x}_1^T, \mathbf{x}_2^T, \cdots \mathbf{x}_p^T]^T$. The ith component $\mathbf{x}_i$ from the ith source takes the form in (4.14). Followed the equation (4.15), covariance matrix is defined as:

$$\Gamma_{XX} = E(\mathbf{X}\mathbf{X}^T) \tag{4.18}$$

with the following covariance structure:

$$\Gamma_{XX} = \begin{bmatrix} \Gamma_{\mathbf{x}_1\mathbf{x}_1} & \cdots & \Gamma_{\mathbf{x}_1\mathbf{x}_p} \\ \vdots & \ddots & \vdots \\ \Gamma_{\mathbf{x}_p\mathbf{x}_1} & \cdots & \Gamma_{\mathbf{x}_p\mathbf{x}_p} \end{bmatrix}$$

Consider the covariance matrix estimator for one dimension KLE in equation (4.17) and its corresponding data matrix format in (4.17), we have the data matrix X' for multi-dimensional KLE defined in (4.10). The corresponding eigen vectors, which can be found by solving $\Gamma_{XX}\psi_i = \lambda_i\psi_i$ considering both the temporal and spatial correlation.

However, it is nontrivial to adopt the regularization framework proposed in (4.7) and (4.9) to expanded data matrix $\mathbf{X}'$ because the data stream from each source has been extended from a vector to a matrix. The model parameters corresponding to each source also become a matrix, namely $B = [\mathbf{B}_1^T, \mathbf{B}_2^T, \cdots, \mathbf{B}_p^T]^T$ where $\mathbf{B}_i$ is a N by pN matrix. The top k PCs of $\mathbf{B}$ representing the normal subspace in regular PCA will become kN PCs after KLE extension. Similarly, abnormal subspace is the rest $(p-k)N$ PCs of $\mathbf{B}$. More specifically, we consider the following optimization problem similar to the objective of JSKLE:

$$\begin{aligned} \min_{\mathbf{A},\mathbf{B}} \quad & \frac{1}{2}||\mathbf{X}' - \mathbf{X}'\mathbf{B}\mathbf{A}^T||_F^2 + \lambda_1 \sum_{j=1}^{p} ||\mathbf{W}_j \circ \mathbf{B}_j||_F \\ \text{s.t.} \quad & \mathbf{A}^T\mathbf{A} = I_{pN\times pN}, \end{aligned} \tag{4.19}$$

where $\mathbf{W}_j \in \{0,1\}^{N\times pN}$ is the jth matrix block of $\mathbf{W}^T = [\mathbf{W}_1, \mathbf{W}_2, \cdots, \mathbf{W}_p]$ similar to (4.6) with first kN columns being 0s and the rest being 1s:

$$\mathbf{W}_j = \begin{bmatrix} 0 & \cdots & 0 & 1 & \cdots & 1 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & \cdots & 1 \end{bmatrix}$$

For GJSKLE, we have to adjust the structured trace regularization component for ex-

tended data. Since each source has been extended to multiple streams, we take average values across the N extended streams and make the average values smooth according to the network topology. More formally, considering the following objective:

$$\begin{aligned} \min_{\mathbf{A},\mathbf{B}} \quad & \frac{1}{2}||\mathbf{X}' - \mathbf{X}'\mathbf{B}\mathbf{A}^T||_F^2 + \lambda_1 \sum_{j=1}^{p} ||\mathbf{W}_j \circ \mathbf{B}_j||_F \\ & \tfrac{1}{2N}\lambda_2 tr((\mathbf{W} \circ \mathbf{B})^T P^T L P (\mathbf{W} \circ \mathbf{B})) \\ \text{s.t.} \quad & \mathbf{A}^T\mathbf{A} = I_{pN \times pN}, \end{aligned} \tag{4.20}$$

where $\mathbf{P} \in \{0,1\}^{p \times pN}$ is used to summing each block of $\mathbf{B}$ and defined as:

$$\mathbf{P} = \begin{bmatrix} 1 & \cdots & 1 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 1 & \cdots & 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & \cdots & 1 & \cdots & 1 \end{bmatrix}$$

In Figure 4.6, we show the PC space computed from JSKLE and GJSKLE. There are two principal components representing the normal subspace and the rests presenting the abnormal subspace. Both JSKLE and GJSKLE highlight the abnormal sources while GJSKLE shows a smooth effect on 3 abnormal sources 2, 3, 7.

For JSKLE and GJSKLE, the definition of abnormal score is a little different from that of JSPCA and GJSPCA. Suppose the abnormal subspace is given by $\mathbf{V}^{(3)^T} = [\mathbf{V}^{(3)}{}_1, \mathbf{V}^{(3)}{}_2, \cdots, \mathbf{V}^{(3)}{}_p]$ (the rest $(p-k)N$ columns of $\mathbf{B}$ from (4.19) or (4.20)), the anomaly score for source $i, \quad i = 1 \cdots p$ is

$$\zeta_i = \frac{||\mathbf{V}_i^{(3)}||_1}{(p-k)N} \tag{4.21}$$

where $\mathbf{V}_i^{(3)}$ is the ith matrix block of $\mathbf{V}^{(3)}$.

Abnormal scores computed by JSKLE and GJSKLE are shown in Figure 4.6. JSKLE and GJSKLE performs similarly to JSPCA and GJSPCA but they are insensitive to the number of PCs representing the normal subspace, which will be studied in our experimental

studies.

4.4.5 Optimization Algorithms

We present our optimization technique to solve equations (4.7), (4.9), (4.19) and (4.20) based on accelerated gradient descent [126] and projected gradient scheme [15]. Since (4.19) and (4.20) are similar to (4.7) and (4.9), our following discussion will focus on (4.7) and (4.9). The solutions for (4.19) and (4.20) can be obtained by the same procedure with only minor changes on calculating gradient and gradient projection.

Although equations (4.7) and (4.9) are not joint convex for $\mathbf{A}$ and $\mathbf{B}$, they are convex for $\mathbf{A}$ and $\mathbf{B}$ individually. The algorithm solves $\mathbf{A}$, $\mathbf{B}$ iteratively and achieves a local optimum. Due to the space constrain, we provide our optimization algorithm in appendix.

A given B: If $\mathbf{B}$ is fixed, we obtain the optimal $\mathbf{A}$ analytically. Ignoring the regularization part, equation (4.7) and equation (4.9) degenerate to

$$\begin{aligned} \min_{\mathbf{A}} \quad & \tfrac{1}{2}||\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}^T||_F^2 \\ \text{s.t.} \quad & \mathbf{A}^T\mathbf{A} = I_{p\times p}. \end{aligned} \tag{4.22}$$

The solution is obtained by a reduced rank form of the Procrustes Rotation. We compute the SVD of $\mathbf{G}\mathbf{B}$ to obtain the solution where $\mathbf{G} = \mathbf{X}^T\mathbf{X}$ is the gram matrix:

$$\begin{aligned} \mathbf{G}\mathbf{B} &= \mathbf{U}\mathbf{D}\mathbf{V}^T \\ \hat{\mathbf{A}} &= \mathbf{U}\mathbf{V}^T. \end{aligned} \tag{4.23}$$

Solution in the form of Procrustes Rotation is widely discussed, see [196] for example for a detailed discussion.

B given A: If $\mathbf{A}$ is fixed, we consider equation (4.9) only since equation (4.7) is a special

case of equation (4.9) when $\lambda_2 = 0$. Now the optimization problem becomes:

$$\min_{\mathbf{A},\mathbf{B}} \quad \frac{1}{2}||\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}^T||_F^2 + \lambda_1\|\mathbf{W} \circ \mathbf{B}\|_{1,2} + \frac{1}{2}\lambda_2 tr((\mathbf{W} \circ \mathbf{B})^T L(\mathbf{W} \circ \mathbf{B})). \quad z \tag{4.24}$$

Equation (4.24) can be rewritten as $\min_{\mathbf{B}} \quad F(\mathbf{B}) \stackrel{\text{def}}{=} f(\mathbf{B}) + R(\mathbf{B})$, where $f(\mathbf{B})$ takes the smooth part of equation(4.24)

$$f(\mathbf{B}) = \frac{1}{2}||\mathbf{X}' - \mathbf{X}'\mathbf{B}\mathbf{A}^T||_F^2 + \frac{1}{2}\lambda_2 tr((\mathbf{W} \circ \mathbf{B})^T L(\mathbf{W} \circ \mathbf{B})) \tag{4.25}$$

and $R(\mathbf{B})$ takes the nonsmooth part, $R(\mathbf{B}) = \lambda_1||\mathbf{W} \circ \mathbf{B}||_{1,2}$. It is easy to verify that (4.25) is a convex and smooth function over $\mathbf{B}$ and the gradient of f is: $\nabla f(\mathbf{B}) = \mathbf{G}(\mathbf{B} - \mathbf{A}) + \lambda_2 L(\mathbf{W} \circ \mathbf{B})$.

Considering the minimization problem of the smooth function $f(\mathbf{B})$ using the first order gradient descent method, it is well known that the gradient step has the following update at step $i + 1$ with step size $1/L_i$:

$$\mathbf{B}_{i+1} = \mathbf{B}_i - \frac{1}{L_i}\nabla f(\mathbf{B}_i). \tag{4.26}$$

In [10, 126], it has shown that the gradient step equation (4.26) can be reformulated as a linear approximation of the function f at point $\mathbf{B}_i$ regularized by a quadratic proximal term as $\mathbf{B}_i = \operatorname{argmin}_{\mathbf{B}} f_{L_i}(\mathbf{B}, \mathbf{B}_i)$, where

$$f_{L_i}(\mathbf{B}, \mathbf{B}_i) = f(\mathbf{B}_i) + \langle \mathbf{B} - \mathbf{B}_i, \nabla f(\mathbf{B}_i)\rangle + \frac{L_i}{2}\|\mathbf{B} - \mathbf{B}_i\|_F^2 \tag{4.27}$$

Based on the relationship, we combine equations (6.12) and $R(B)$ together to formalize the

generalized gradient update step:

$$\begin{aligned} Q_{L_i}(\mathbf{B}, \mathbf{B}_i) &= f_{L_i}(\mathbf{B}, \mathbf{B}_i) + \lambda_1 ||\mathbf{W} \circ \mathbf{B}||_{1,2} \\ q_{L_i}(\mathbf{B}_i) &= \underset{\mathbf{B}}{\operatorname{argmin}}\ Q_{L_i}(\mathbf{B}, \mathbf{B}_i). \end{aligned} \tag{4.28}$$

The insight of such a formalization is that by exploring the structure of regularization $R(.)$ we can easily solve the optimization in equation (6.13), then the convergence rate is the same as that of gradient decent method. In this paper, we use accelerated gradient descent [126] to handle the smooth part and projected gradient scheme [15] to tackle nonsmooth part.

Our goal is to find $\mathbf{B}$ at current $\mathbf{B}_i$ to minimize $Q_{L_i}(\mathbf{B}, \mathbf{B}_i)$ composed of smooth and nonsmooth components. Rewriting the optimization problem in equation(6.13) and ignoring terms that do not depend on B, the objective can be expressed as:

$$\begin{aligned} q_{L_i}(\mathbf{B}_i) = \underset{\mathbf{B} \in \mathcal{M}}{\operatorname{argmin}} \Big(&\frac{1}{2}\|\mathbf{B} - (\mathbf{B}_i - \frac{1}{L_i}\nabla f(\mathbf{B}_i))\|_F^2 + \\ &\frac{\lambda_1}{L_i}||\mathbf{W} \circ \mathbf{B}||_{1,2}\Big). \end{aligned} \tag{4.29}$$

With ordinary first order gradient method for smooth problems, the convergence rate is $O(1/\sqrt{\epsilon})$ [126] where ϵ is the desired accuracy. In order to have a better convergence rate, we apply the Nestrerov accelerated gradient descent method [126] with $O(1/\sqrt{\epsilon})$ convergence rate, and solve the *generalized gradient update step* in equation (6.13) for each gradient update step. Such a procedure has demonstrated scalability and fast convergence in solving various sparse learning formulations [29, 82, 112]. Below we present the accelerated projected gradient algorithm. The stopping criterion is that the change of the objective values in two successive steps is less than a predefined threshold (e.g. 10^{-4}).

Now we focus on how to solve the generalized gradient update in equation (6.14). Let

Algorithm 3 Accelerated Projected Gradient Descent

Input: $\mathbf{B}_0, \mathbf{W} \in \mathcal{R}^{p \times p}$, $L_1 > 0$, $F(.), Q_L(.,.)$ and max-iter.
Output: $\mathbf{B}$.
Initialize $\mathbf{B}_1 := \mathbf{B}_0, t_{-1} := 0, t_0 := 1$;
for $i = 1$ to max-iter **do**
  $\alpha_i := (t_{i-2} - 1)/t_{i-1}$;
  $\mathbf{S} := \mathbf{B}_i + \alpha_i(\mathbf{B}_i - \mathbf{B}_{i-1})$;
  while (true) **do**
    Compute $q_{L_i}(S)$ in Eq. (6.14);
    if $F(q_{L_i}(S)) > Q_{L_i}(q_{L_i}(S), S)$ **then**
      $L_i := 2 \times L_i$;
    else
      break;
    end if
  end while
  $\mathbf{B}_{i+1} := q_{L_i}(S), L_{i+1} := L_i$;
  $t_i := \frac{1}{2}(1 + \sqrt{1 + 4t_{i-1}^2})$;
  if (Convergence) **then**
    $\mathbf{B} := \mathbf{B}_{i+1}$, break;
  end if
end for
return $\mathbf{B}$;

$\mathbf{C} = \mathbf{B}_i - \frac{1}{L_i}\nabla f(\mathbf{B}_i)$ and $\bar{\lambda} = \lambda_1/L_i$, equation (6.14) can be represented as:

$$\begin{aligned} q_{L_i}(\mathbf{B}_i) &= \underset{\mathbf{B}}{\operatorname{argmin}}(\tfrac{1}{2}||\mathbf{B}-\mathbf{C}||_F^2 + \bar{\lambda}||\mathbf{W}\circ\mathbf{B}||_{1,2}) \\ &= \underset{\tilde{\mathbf{b}}_1,\cdots,\tilde{\mathbf{b}}_p}{\operatorname{argmin}} \textstyle\sum_{j=1}^{p}(\tfrac{1}{2}||\tilde{\mathbf{b}}_j - \tilde{\mathbf{c}}_j||_2^2 + \bar{\lambda}||\tilde{\mathbf{w}}_j \circ \tilde{\mathbf{b}}_j||_2) \end{aligned} \tag{4.30}$$

where $\tilde{\mathbf{b}}_j^T, \tilde{\mathbf{c}}_j^T$ and $\tilde{\mathbf{w}}_j^T \in \mathcal{R}^p$ are row vectors denoting the jth row of matrices $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{W}$. By the additivity of equation (6.15), we decompose equation (6.15) into p subproblems. For each subproblem, we ignore the row index j:

$$\min_{\mathbf{b}} \quad \frac{1}{2}||\mathbf{b}-\mathbf{c}||_2^2 + \bar{\lambda}||\mathbf{w}\circ\mathbf{b}||_2. \tag{4.31}$$

The following theorem provides the analytical solution of equation (6.16).

Theorem 4.4.1. *Given* $\bar{\lambda}, \mathbf{w} = [\mathbf{0}_{1\times k}, \mathbf{1}_{1\times(p-k)}]^T$ *and* $\mathbf{c} = [\mathbf{c}_1^T, \mathbf{c}_2^T]^T$ *where* $\mathbf{c}_1 = [c_1, \cdots, c_k]^T$, $\mathbf{c}_2 = [c_{k+1}, \cdots, c_p]^T$ *and* k *is the number of PCs representing the normal subspace, the optimal solution for (6.16)* $\mathbf{b}^* = [\mathbf{b}_1^{*T}, \mathbf{b}_2^{*T}]^T$ *is given by:*

$$\mathbf{b}_1^* = \mathbf{c}_1$$

and

$$\mathbf{b}_2^* = \begin{cases} (1 - \frac{\bar{\lambda}}{||\mathbf{c}_2||_2})\mathbf{c}_2 & ||\mathbf{c}_2||_2 > \bar{\lambda} \\ 0 & \textit{otherwise.} \end{cases} \tag{4.32}$$

Proof. By the definition of the l_2 norm, the equation (6.16) can be rewritten as:

$$\min_{\mathbf{b}_1,\mathbf{b}_2} \frac{1}{2}||\mathbf{b}_1 - \mathbf{c}_1||_2^2 + \frac{1}{2}||\mathbf{b}_2 - \mathbf{c}_2||_2^2 + \bar{\lambda}||\mathbf{b}_2||_2 \tag{4.33}$$

where $\mathbf{b} = [\mathbf{b}_1^T, \mathbf{b}_2^T]^T$. The solution can be found by decomposing (4.33) into two subproblems and solving one ordinary least square problem and one least square problem with l_2 norm regularization. Since there is no regularization on $\mathbf{b}_1$ and the two subproblems are

independent, the optimal solution of the ordinary least square problem is $\mathbf{b}_1^* = \mathbf{c}_1$. With optimal $\mathbf{b}_1^*$, (4.33) degenerates to

$$\min_{\mathbf{b}_2} \frac{1}{2}||\mathbf{b}_2 - \mathbf{c}_2||_2^2 + \bar{\lambda}||\mathbf{b}_2||_2. \tag{4.34}$$

The analytical solution of equation (4.34) is given in equation (6.17) and can be found by forming Lagrangian dual. A detailed proof can be found in [112]. □

For JSKLE and GJSKLE, we perform the similar procedure but on a set of matrices $\mathbf{B}_i \in \mathcal{R}^{N\times(p-k)N}$ due to the KL expansion. Then the solution $\mathbf{B}^* = [\mathbf{B}_1^*, \cdots, \mathbf{B}_p^*]^T$ given $\mathbf{A}$ is obtained:

$$\mathbf{B}_i^* = \begin{cases} (1 - \frac{\bar{\lambda}}{\sqrt{tr(\mathbf{C}_i\mathbf{C}_i^T)}})\mathbf{C}_i & \sqrt{tr(\mathbf{C}_i\mathbf{C}_i^T)} > \bar{\lambda} \\ 0 & \text{otherwise} \end{cases} \tag{4.35}$$

where $\mathbf{C}_i$ is the ith matrix block of $\mathbf{C} = [\mathbf{C}_1, \mathbf{C}_2, \cdots, \mathbf{C}_p]^T = \mathbf{B} - \frac{1}{L}\nabla f(\mathbf{B})$, and $\mathbf{B}$ is computed from (6.13), (6.15) in an extended data matrix and principal components.

We summarize what is briefly discussed previously for GJSPCA in Algorithm XX. Note that JSPCA is a special case of GJSPCA, we obtain the algorithm for JSPCA by setting $\lambda_2 = 0$. For JSKLE and GJSKLE, the only changes are the gradient of smooth parts in the objective (4.19), (4.20) and projected gradient. Given data matrix $\mathbf{X} \in \mathcal{R}^{n\times p}$ and the number of PCs representing normal subspace k and regularization parameters λ_1, λ_2, GJSPCA optimizes two matrix variables alternatively and returns the matrix $\mathbf{B}$ composed of ordinary PCs representing normal subspace and joint sparse PCs representing the abnormal subspace.

4.5 Experimental Studies

We have conducted extensive experiments with three real-world data sets to evaluate the performance of JSPCA and GJSPCA on anomaly localization. We implemented our version

Algorithm 4 Graph Joint Sparse PCA (GJSPCA)

Input: $\mathbf{X}$, k, λ_1, λ_2 and max_iter.
Output: $\mathbf{B}$.
$\mathbf{A} := I_{p\times p}, \mathbf{G} := \mathbf{X}^T\mathbf{X}$;
for $iter = 1$ to max_iter **do**
    Compute $\mathbf{B}$ given $\mathbf{A}$ using the accelerated gradient descent and gradient projection as shown in the appendix;
    Compute $\mathbf{A}$ given $\mathbf{B}$ via (4.23);
    if (Converge) **then**
        break;
    end if
end for
return $\mathbf{B}$;

of two state-of-the-art anomaly localization methods at the network level: stochastic nearest neighbor (SNN) [76] and eigen equation compression (EEC) [69] since no executables were provided by the original authors. We implemented all four methods with Matlab and performed all experiments on a desktop machine with 6 GB memory and a Intel core i7 2.66 GHz CPU.

4.5.1 Data Sets

We used four real-world data sets from different application domains. For each data set, we singled out several intervals with anomalies. The anomalies are either labeled by the original data provided or manually labeled by ourselves when no labeling is provided. Note that we are only interested in the intervals where anomalies really exist since we focus on localizing anomalies. We used a sliding window with fixed size L and offset $L/2$ to create multiple data windows from the given intervals. The sliding window moves forward with the offset $L/2$ until it reaches the end of the intervals. We run all four methods on each data window to evaluate and compare their performances.

To run GJSPCA we calculated the pair-wise correlation between any two sources within the window. We produced a correlation graph for the data streams with a correlation threshold δ in that if the correlation between two sources is greater than δ, we connect the

two sources with an edge. This construction is meaningful because for highly correlated data, streams influence each other and such influence has been shown critical for better anomaly localization, as evaluated in our experimental studies.

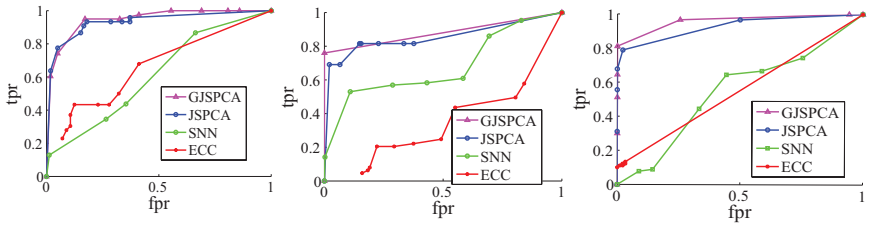
Below we briefly discuss the data collection and data preprocessing procedures for the three data sets. In Table 4.2, we list the intervals that we selected, the dimensionality of the network data streams, the sliding window size L, and the total number of data windows W for each data set. For KDD99 intrusion data set, T is the number of connections and p is the number of features.

Table 4.2: Characteristics of Data Sets. D: Data sets. D1: Stock Indices, D2: Sensor, D3: MotorCurrent, D4: Network Traffic. T: total number of time stamps, p: dimensionality of the network data streams, I: total number of intervals, $Indices$: starting point and ending point of the abnormal intervals, W: total number of data windows, L: sliding window size, -: not applicable.

D	T	p	I	$Indices$	W	L
D1	2396	8	4	[261-300], [361- 400] [761-800], [1631-1670]	12	20
D2	11000	7	4	[2371-2530],[3346-3550] [7191-7215], [8841-8870]	37	20
D3	1500	20	1	[1-1500]	29	50
D4						
(DOS)	391458	41	1	[1-391458]	-	-
(Probe)	4107	41	1	[1-4107]	-	-
(U2R)	52	41	1	[1-52]	-	-
(R21)	1126	41	1	[1-1126]	-	-

The Stock Indices Data Set: The stock indices data set includes 8 stock market index streams from 8 countries: Brazil (Brazil Bovespa), Mexico (Bolsa IPC), Argentina (MERVAL), USA (S&P 500 Composite), Canada (S&P TSX Composite), HK (Heng Seng), China (SSE Composite), and Japan (NIKKEI 225). Each stock market index stream contains 2396 stamps recording the daily stock price indices from January 1st 2001 to March 5th 2010.

Since this data set has no ground truth, we manually labeled all the daily indices for the selected intervals. In our labeling we followed the criteria list in [23] where small turbulence and co-movements of most markets are considered as normal, dramatic price changes or



Dataset \ Method / AUC	SNN	ECC	JSPCA	GJSPCA
Stock Indices	0.6119	0.6510	0.9316	0.9457
Sensor	0.6783	0.3491	0.8527	0.8798
MotorCurrent	0.5444	0.5515	0.9273	0.9601

Figure 4.7: ROC curves and AUC for different methods on three data sets. From left to right: ROC for the stock indices data, ROC for the sensor data, ROC curve for MotorCurrent data, AUC for the three ROC plots

significance deviation from the co-movement trend (e.g. one index goes up while the others in the market drop down) are considered as abnormal.

The Sun Spot Sensor Data Set: We collected a sensor data set in a car trial for transport chain security validation using seven wireless Sun Small Programmable Object Technologies (SPOTs). Each SPOT contains a 3-axis accelerometer sensor. In our data collection, seven Sun SPOTs were fixed in separated boxes and were loaded on the back seat of a car. Each Sun SPOTs recorded the magnitude of accelerations along x, y, z axis with a sample rate of 390ms. We simulated a few abnormal events including box removal and replacement, rotation and flipping. The overall acceleration $\sqrt{(x^2+y^2+z^2)}$ was used to detect the designed anomalous events.

The Motor Current Data Set: The Motor Current Data is the current observation generated by the state space simulations available at UCR Time Series Archive [91]. The anomalies are the simulated machinery failure in different components of a machine. The current value was observed from 21 different motor operating conditions, including one healthy operating mode and 20 faulty modes. For each motor operating condition, 20 time series were recorded with a length of 1,500 samples. Therefore, there are 20 normal time series and 400 abnormal time series altogether.

In our evaluation, we randomly extracted 20 time series out of 420 with the length 1500. 10 time series are from normal series and the rest are from abnormal series.

KDDCup 99 Intrusion Detection Data Set: The KDDCup99 intrusion detection data set is obtained from UCI Repository [50]. The 10% training data set consisting of

Table 4.3: Features Indexes in KDD 99 Intrusion Detection Data set

List of Features	
Basic Features	1. duration, 2. protocol type, 3. service, 4. flag, 5. source bytes, 6. destination bytes
Content Features	7. land, 8. wrong fragment, 9. urgent, 10. hot, 11. failed logins, 12. logged in, 13. # compromised, 14. root shell, 15. su attempted, 16. # root 17. # file creations, 18. # shells, 19. # access files, 20. # outbound cmds, 21. is host login, 22. is guest login
Traffic Features	23. count, 24. srv count 25. serror rate 26. srv serror rate, 27. rerror rate, 28. srv rerror rate, 29. same srv rate, 30. diff srv rate, 31. srv diff host rate
Host-based Traffic Features	32. dst host count, 33. dst host srv count, 34. dst host same srv rate, 35. same srv rate, 36. dst host same src port rate, 37. dst host srv diff host rate, 38. dst host serror rate, 39. dst host srv serror rate, 40. dst host rerror rate, 41. dst host srv rerror rate

494,021 connection records is used. Each connection can be classified as normal traffic or one of 22 different classes of attacks. All attacks fall into four main categories: Denial-of-service (DOS), Remote-to-local (R2L), User-to-root (U2R), and Probing (Probe). For each connection, 41 features are recorded, including 7 discrete features and 34 continuous features. Since our algorithm is calculated for continuous features, the discrete features such as protocol (TCP/UDP/ICMP), service type (http/ftp/telnet/...) and TCP status flag (SF/REJ/...) are mapped into distinct positive integers from 0 to $W-1$ (W is the number of states for a specific discrete feature). For three features spanning over a very large range, namely "duration", "src bytes" and "dst bytes", logarithmic scale is applied to reduce the ranges. Finally all the 41 features are linearly scaled to the range [0,1]. The task of anomaly localization on the intrusion detection data set is to identify the set of features most relevant to a specific anomaly, which is similar to feature selection.

4.5.2 Model Evaluation

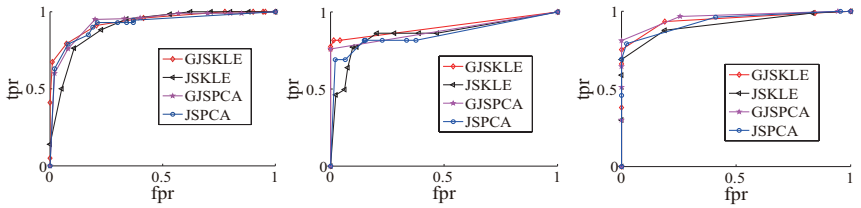
For evaluation, since our focus is anomaly localization, we did not evaluate anomaly detection although our method is able to do both. We used the standard ROC curves and area under ROC curve (AUC) to evaluate the anomaly localization performance. There is no training phase because our framework is unsupervised. PCA is a Below we introduce the details regarding the construction of ROC curves.

As defined in equation 4.8, a higher abnormal score indicates a higher probability the source is abnormal, which is the same as that of the baseline methods [69, 76] for comparison. To have a fair comparison, we compared the normalized abnormal score among each method. The reason for normalization is that the anomaly scores generated by the baseline methods have different orders of magnitude. We used the term "anomaly score" to refer to the normalized abnormal score in the following analysis.

For each data window, the abnormal score vector $\tilde{\zeta} = [\tilde{\zeta}_1, \cdots, \tilde{\zeta}_p]^T$ was generated and compared with a a cut-off threshold between $[0, 1]$ to separate abnormal sources and innocent sources. We performed the same procedure for all the data windows and finally we obtained a prediction matrix with size w by p, such that w is the number of data window and p is the number of sources. Each entry in the prediction matrix is 0 or 1 to indicate whether the source is normal or abnormal. Comparing the prediction matrix with the ground truth resulted in a pair of true positive rate (TPR) and false positive rate (FPR), where TPR is the total number of true detected abnormal sources over the total number of abnormal sources, and FPR is the total number of incorrect detected abnormal sources over the total number of normal sources in W windows. By changing the threshold, we obtained the ROC curve and the AUC value.

For network traffic data set, we evaluated our method in a qualitative because there is no ground truth about which features contribute to the observed anomaly, also there is no way to do manually label. For each kind of anomaly, we show the abnormal score of each feature and analyze with some prior knowledge such as what is the cause of a specific attack, and

how this attack effects the 41 features. To better demonstrate the effectiveness of JSPCA and GJSPCA, we also compare our results with those obtained from other feature selection methods such as information gain [89] and SVM [123] on KDDCUP 99 data set.



Dataset \ Method AUC	JSPCA	GJSPCA	JSKLE	GJSKLE
Stock Indices	0.9316	0.9457	0.9277	0.9405
Sensor	0.8527	0.8798	0.8342	0.8703
MotorCurrent	0.9273	0.9601	0.9421	0.9674

Figure 4.8: ROC curve for KLE extension methods on three data sets. From left to right: ROC for the stock indices data, ROC for the sensor data, ROC curve for MotorCurrent data

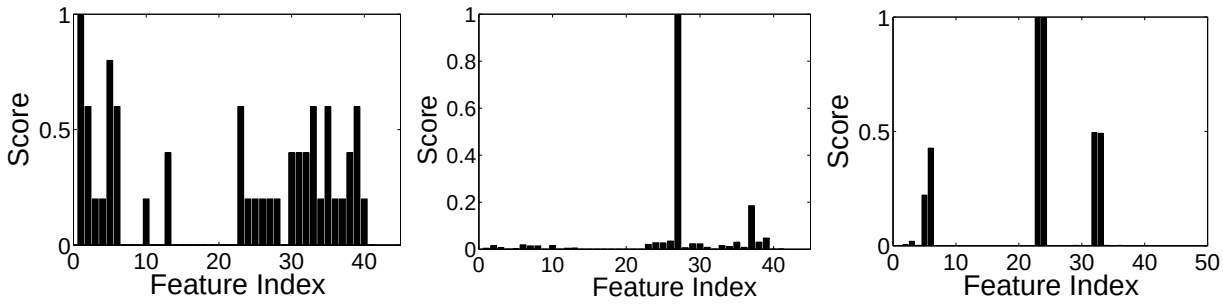


Figure 4.9: Anomaly Localization Comparison of Stochastic Nearest Neighborhood, Eigen-Equation Compression, GJSPCA on Network Intrusion Data Set(DoS Attack)

4.5.3 Anomaly Localization Performance

We have two parameters to tune in JSPCA: λ_1: controlling the sparsity, and k: the dimension of normal subspace. GJSPCA has two more parameters: λ_2: controlling the smoothness, and δ, the correlation threshold to construct the correlation graph. For the other two methods, we need to select the number of neighbors k for SSN and the number of clusters c for EEC. We first performed a grid search for each method to identify the optimal parameters and then compared the performance. The performances of different methods depend on the parameter selection. We evaluated the sensitivity of our results in the next selection.

For each data set, we tuned λ_1, λ_2 within $\{2^{-8}, 2^{-7}, \cdots, 2^8\}$ and δ from 0.1 to 0.9. k was tuned from 1 to 4 for the stock market and sensor data, and from 2 to 7 for the motor

current data. All the ranges were set by empirical knowledge. Our empirical study showed that the performance did not change significantly as the parameters vary in a wide range, which reduced the parameter search space significantly.

Table 4.6 lists the best parameter combination for JSPCA and GJSPCA. For SNN, we tuned the number of neighbors k in the range $2 \sim 6$ (for stock index data set and sensor data) and in the range $2 \sim 10$ (for motorcurrent data) respectively. For EEC method, the number of clusters c was tuned between $2 \sim 4$.

In Figure 4.7, we show the performances for the four methods on three different data sets. JSPCA and GJSPCA clearly outperform the other two methods. The AUC value of JSPCA and GJSPCA are both above 0.85 on three data sets, while that of EEC and SNN are around $[0.5 \sim 0.6]$. Compared with JSPCA, GJSPCA is slightly better, which supports our hypothesis on the importance of incorporating the structure information of network data streams into anomaly localization. SNN clearly outperforms EEC on Sensor data, and is comparable with EEC for the other two data sets.

We did a case study in which PCA, SPCA, JSPCA and GJSPCA are compared on a selected time interval. As shown in Figure 4.5, PCA is not able to identify the abnormal sources. SPCA fails to localize source 7 and introduces many false positives when threshold is wrongly selected. To further support the argument that PCA and SPCA are inadequate for anomaly localization, we did experiment on stock indices data and calculated AUC. We found the AUC value of PCA and SPCA are 0.667 and 0.6703 respectively, which are much lower compared to that of JSPCA. We do not extend our experiments to the other data sets since the two methods are clearly not competitive.

We also test the KLE extension of localization methods. In Figure 4.8, we show the performance of JSKLE and GJSKLE in comparison with JSPCA and GJSPCA with $N = 2$. From the Figure, we observe that KLE extension does not outperform JSPCA and GJSPCA on anomaly localization with the expense of introducing more computational complexity due to the data matrix expansion. However, KLE extension stabilizes localization performance

Table 4.4: Most relevant features for different attacks (JSPCA)

Attack	Feature Index
DOS	3,5,6,23,24,32,33
Probe	5
U2R	1,5,6,32,33
R21	1,3,5,6,32,33

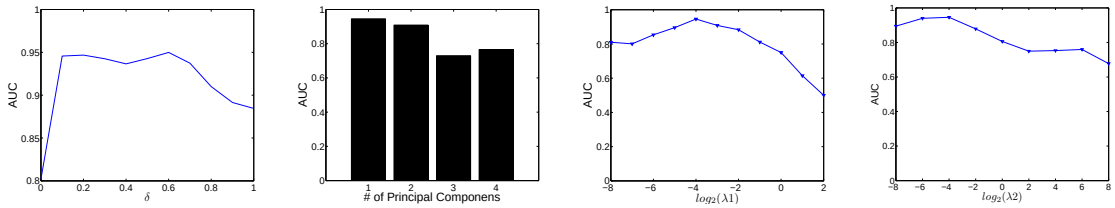


Figure 4.10: Sensitivity analysis of GJSPCA on stock indices data set. From left to right: δ, the dimension of the normal subspace, λ_1 and λ_2.

as shown in the section of parameter selection.

4.5.4 Feature Selection Performance

As mentioned earlier, anomaly localization on the KDDCUP 99 intrusion detection data set performs as a feature relevant analysis. Localizing abnormal data streams amounts to identify features most related to a specific anomaly. More specifically, our algorithm aims to identify a set of relevant features among all the 41 features for each type of attacks. The features are indexed and given in Table 4.3.

In Figure 4.9, we show the abnormal scores for the 41 features under the attack of Denial of Service (DOS) computed by SNN, EEC and GJSPCA respectively. Since four joint sparse methods provide similar abnormal scores, we just show the result of GJSPCA in the rightmost of Figure 4.9. Feature 5, 6, 23, 24, 32, 33 are the most relevant for DOS attack, which is reasonable since the nature of DOS attacks involves many connections to some host(s) in a very short period of time. In Table 4.4, we summarize the most relevant features for each attack from our method GJSPCA. Our result is consistent with the relevant features found in Mukkamala et al. using SVM [124].

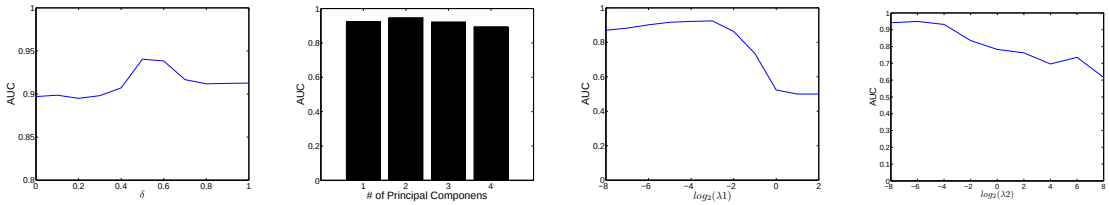


Figure 4.11: Sensitivity analysis of GJSKLE on stock indices data set. From left to right:δ, the dimension of the normal subspace, λ_1 and λ_2.

Table 4.5: Optimal parameters combinations on three data sets. J:JSPCA, GJ: GJSPCA.

	λ_1		k		λ_2	δ
Data set	J	GJ	J	GJ	GJ	GJ
Stock	2^{-3}	2^{-4}	1	1	2^{-4}	0.6
Sensor	2^{-7}	2^{-5}	1	1	2^{-6}	0.6
Motor	2^{-2}	2^{-2}	5	5	2^{-8}	0.5

4.5.5 Parameter Selection

In this section, we evaluated the sensitivity of our methods to different modeling parameters. In order to do so, we selected one parameter at a time, systematically changed its value while fixing the others at their optimal values. Although our approaches have more parameters than the other two methods, the sensitivity analysis shows that performances of our methods are remarkably stable over a wide range of parameters. Next we show the sensitivity study on the stock indices data set for the parameters λ_1 and λ_2, δ, k. Similar results are observed on the other two data sets.

In Figure 4.10, we show the stability by changing λ_1 in GJSPCA. We observe that AUC is quite stable over a wide range of λ_1. A similar phenomenon is also observed when changing λ_2. On the middle part of Figure 4.10, we performed sensitivity analysis on parameter δ. We observe that AUC remains stable for $\delta \in [0.15, 0.6]$. When $\delta = 0$, the graph is a complete graph and the smoothness regularization will penalize the loadings of each source across the PCs to be similar each other. Hence very low δ leads to a worse performance. On the other hand, when $\delta = 1$, the graph is just a set of isolated sources. The structure information is missing, therefore the performance is not optimal.

An important parameter in PCA based anomaly detection methods is k, the number

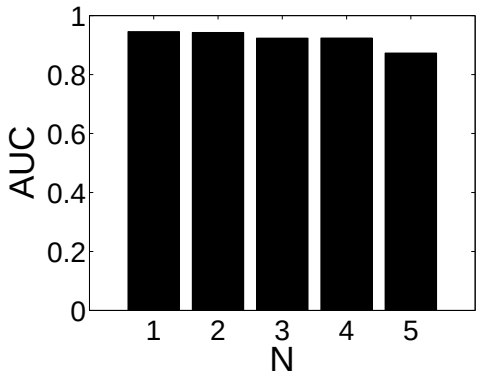


Figure 4.12: Sensitivity analysis of GJSKLE on stock indices data set on N.

Table 4.6: Optimal parameters combinations on three data sets. JK:JSKLE, GJK: GJSKLE.

	λ_1		k		λ_2	δ	N
Data set	JK	GJK	JK	GJK	GJK	GJK	(G)JK
Stock	2^{-3}	2^{-4}	2	2	2^{-4}	0.5	2
Sensor	2^{-6}	2^{-5}	2	2	2^{-6}	0.7	2
Motor	2^{-2}	2^{-2}	7	8	2^{-8}	0.6	2

of PCs spanning the normal subspace. In [139], Ringberg *et al.*claimed that the anomaly detection performance was sensitive to k. Such a claim is confirmed in the 4th subfigure of Figure 4.10, where the overall AUC gradually decreases from 0.96 to 0.72 as k changes from 1 to 3 and then increases to 0.77 at $k = 4$. Compared with GJSPCA, GJSKLE significantly stabilize the localization performance when δ (the threshold for deriving network topology) and k (dimension of normal space) change. As shown in Figure 4.11, when δ changes from 0 to 1 with step size 0.1, AUC increases to its optimum 0.94 at $\delta = 0.5$, and then decreases 3% to its minimum 0.91 at $\delta = 1$. Furthermore, AUC remains above 0.9 for $k \in [1, 4]$.

JSKLE and GJSKLE involves one more parameter: the temporal offset N. To test the sensitivity of N, we repeated the experiments of KLE with different Ns from 1 to 5 on the finance data set. Note that (G)JSPCA is a special case of (G)JSKLE when $N = 1$. The result is shown in Figure 4.12. With the change of N, AUC performance is very stable. The difference between the optimal case ($N = 1$) and the worse case ($N = 5$) is just 0.07. It may be apparent that $N = 1$ (degenerated to (G)JSPCA) is better than other cases. However by selecting $N = 2$, AUC of GJSKLE is stabilized when changing δ and k as shown in Figure

4.11 compared with GSPCA in Figure 4.10.

4.6 Conclusions and Future Work

Previous work on PCA based anomaly detection claimed that PCA cannot be used for anomaly localization. We proposed two novel approaches, *joint sparse PCA* (JSPCA) and *graph joint sparse PCA* (GJSPCA), for anomaly detection and localization in network data streams. By enforcing joint sparseness on PCs and incorporating the structure information of network via regularization, we significantly extended the applicability of PCA based technique for localization. Moreover, we developed JSKLE and GJSKLE based on multi-dimensional Karhunen Loève Expansion (KLE) that considers both spatial and temporal domains of data streams to stabilize localization performance. Our experimental studies on three real world data sets demonstrates the effectiveness of our approach. Our future works will focus on two directions: (a) how to efficiently and effectively select model parameters; and (b) how to extend our approach to kernel PCA.

Chapter 5

Preliminary Study III: Multi-task Learning with Structured Output Tasks for Social Behavior Prediction

5.1 Introduction

Online social networking sites are becoming extremely popular among Internet users, especially in the younger generation. The massive adoption of online social networks has introduced significant impacts to users' information sharing and socialization behaviors. Numerous efforts have been devoted to social networking research. In particular, the study of *social information flow* is to analyze the principles and mechanisms of social information distribution, which is one of the fundamental problems in social networking research [73, 164]. Full understanding and control of information flow in social networks is essential for a number of tasks. For instance, to effectively deliver personalized advertising or recommendation, we need to identify messages that are most interesting to the user. Meanwhile, to efficiently distribute emergency notifications in online social networks (e.g. [173]), it is important to discover the most influential nodes to inject the message. On the contrary, to

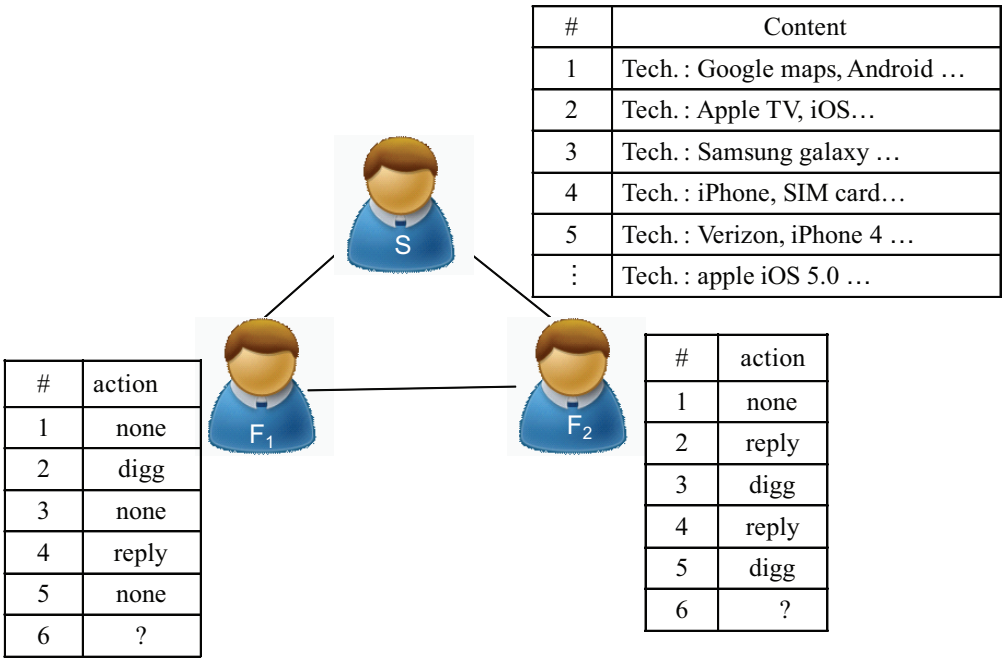
stop rumor dissemination, we need to identify key hubs to enforce countermeasures.

Most of the existing approaches study network information flow based on the social network graph topology, e.g. [90, 155]. For instance, maximum flow and betweenness centrality are the basic measurements employed to assess overall information flow and nodes' specific contributions to it [171]. However, topology itself can not accurately reflect the user interests or activities. It has been widely observed that it is more likely for a message to propagate between users that are mutually interested in the message.

Example 5.1.1. *In Figure 5.1, we present a subgraph of three users collected from a social networking site* digg.com*. In the network, S is an active user and F_1, F_2 are the followers of S and are also very active. In the subgraph S are connected to F_1 and F_2 with the following relationship. In addition, F_1 and F_2 are connected since they follow each other as well (mutual following is allowed in digg). From the perspectives of graph topology and social activity, F_1 and F_2 are highly symmetric. However, they have demonstrated different behaviors in response to S's posts of technology articles. As we have observed, F_2 responses to most of such posts, while F_1 only shows moderate interests in technology-related topics.*

As we observe from the example, in modeling and predicting socialization behaviors, it is important that we take both information content and user interests into consideration. Recently [140] performs a large scale trace on information diffusion in Twitter, and discovers that there are fundamental differences of diffusion behaviors across different topics. The phenomena discovered in [140] further confirms that information content should play a major role in modeling social information flow. However, [140] did not provide a solution of how to quantitatively model or predict social information diffusion process with regard to information content.

In this paper, we adopt a "microeconomics" approach to study social information diffusion and aim to answer the question that how social information flow and socialization behaviors are related to content similarity and user interests. In particular, we study content-based activity prediction, i.e., to predict a user's response (e.g., comment or like) to their friends'



#	Content
1	Tech. : Google maps, Android …
2	Tech. : Apple TV, iOS…
3	Tech. : Samsung galaxy …
4	Tech. : iPhone, SIM card…
5	Tech. : Verizon, iPhone 4 …
⋮	Tech. : apple iOS 5.0 …

#	action
1	none
2	digg
3	none
4	reply
5	none
6	?

#	action
1	none
2	reply
3	digg
4	reply
5	digg
6	?

Figure 5.1: A tiny snapshot of online social network digg.com with three users. The table besides S is his/her recent posts and the rest two tables record his follower's action.

postings (e.g., blogs, tweets) w.r.t. message content. Accurate social behavior prediction is a critical and indispensable step of social network information diffusion modeling and analysis, with a wide range of applications. For instance, social highlights (selected recent activities from friends) are provided to users when they login to social network sites, such as Facebook. With accurate predictions on user responses, we can provide highlights that better fit the interests of the users. In addition, in targeted advertisement (e.g. [156]), companies can recommend new products to users who are most likely to purchase the product, based on the previous actions of users.

In our solution, we cast the social behavior prediction problem as a multi-task learning problem, in which each task corresponds to a user. Similar to traditional supervised learning algorithm, the information content (sample) is represented as a high dimensional feature vector and its labels indicate the responses of users towards the information. We choose multi-task learning as the starting point of our investigation since MTL increases effective sample size and hence boosts the generalization performance of learned models by learning

several related tasks simultaneously [6, 40, 88, 150].

Our contributions in this paper are multifaced. At the conceptual level, we take the first step to model social network information flow w.r.t. information content in social networks. For modeling algorithms, we formalize the related modeling problem as a multi-task learning problem and provide a novel algorithm specifically designed for learning with information flow in social networks. To our best knowledge, we present the first case of developing and applying multi-task learning to the social behavior prediction. Finally we have derived a practical solution based on an advanced optimization technique. Experiment results show that our approach significantly improves prediction accuracy using real world data sets.

5.2 Related Work

Recently, social network has become a popular research area in WWW, information retrieval and knowledge discovery communities, e.g., information extraction and knowledge discovery from social networks [58], community evolution [59, 49], socialization behaviors [153], improving user experience [170], security and privacy protection [61, 99], etc. Among these topics, information diffusion (also referred to as information propagation or adoption) studies how information is distributed in social networks through user socialization activities [1, 2, 90, 110, 155, 181]. In particular, [90] provides an algorithm to select a subset of nodes whose information adoption activities can trigger a large cascade of information flow. [110] studies Internet chain-letter data, and finds that the letters spread in a "narrow but very deep tree-like pattern", instead of spreading widely. [155] models social information flow with a diffusion-rate based model and continuous-time Markov chain. While most existing approaches model information diffusion using graph topology (i.e. information follows the links to propagate from nodes to neighbors), [181] takes a different route that models information flow based on observed infection history. Last but not least, from the social science perspective, different types of social relationships (e.g. strong ties vs. weak ties) [54, 68]

have been used to study information flow.

On the other hand, *multi-task learning* (MTL) has been widely investigated from different researchers and domains, hence it is impossible to cover all the related works in depth. We roughly summarize current MTL methods into two categories based on how they utilize the task relationship: MTL feature learning and MTL with known task relationships. Multi-task feature learning [4, 111, 112, 136] assume all the tasks are uniformly related, and aim to learn a common low dimensional representation without actually learning the task relationships. The common features are learned by block regularization such as l_1/l_2 [4, 112, 36], l_1/l_∞ [29, 111, 136]. MTL with known task relationships [6, 40, 88, 150] utilizes the prior knowledge on task relationships via trace norm regularization to learn model parameters so that similar tasks share similar parameters. They use all the features to build MTL model, hence they are not suitable for high dimensional data. Besides, the task relationship is homogenous.

Though information flow analysis and MTL have been studied for a long time, none of the existing method considers formalizing the content based information flow analysis as MTL problem while considering the heterogenous social relationships. The objective of this paper is to incorporate the heterogenous relationships on tasks into MTL and build a more accurate and interpretable prediction model.

5.3 Methodology

We formalize the user behavior prediction problem with the following approach. Considering a social network with millions of users, we focus on one user, the "seed" user. We represent each article published by the seed user with a bag-of-words model, where features are terms extracted from all the articles the seed published and the value of a feature is the TF-IDF (term frequency times inverse document frequency of the term), as widely used in IR and text mining. There are a group of users actively receiving articles published by the seed (the "followers"). We treat each follower as a learning task. If the follower performed an

action on an article (e.g. writing comments about the article), we record the user response as positive (1). Otherwise, the user response is negative (-1). Figure 5.2 illustrate this data representation approach.

5.3.1 Learning Challenges

Developing and applying machine learning techniques to perform social behavior prediction is challenging. First the data set size is large. Typically an active seed may publish hundreds of articles and with hundreds of followers. Second the data set is often imbalanced. It is quite often that a follower only responds to a small fraction of the articles published by the seed. Third the data set could be quite heterogenous. A follower could be very active in technology while quite inactive for articles published in other categories.

The starting point of our investigation for designing better machine learning techniques for social behavior prediction is multi-task learning. In multi-task learning, we group tasks with similar characteristics in order to increase the effective sample sizes and hence achieve better prediction results for imbalanced and heterogenous data sets. Adopting existing MTL to social behavior prediction is not straightforward. For example current multi-task feature learning methods [4, 111, 112, 136] assume all tasks are uniformly related, which may not be true in social networks. In addition feature selection in MTL with given task relationship [6, 40, 88, 150] has been barely touched. Moreover, although current MTL methods can incorporate the topology information of social networks, they failed to consider the heterogeneity of tasks in social networks. We illustrate these points with the previous example in Figure 5.1.

The core problem in adopting MTL for social behavior prediction, as briefly discussed in the previous paragraph, is how we may group tasks with similar characteristics. Below we lay out three possible strategies:

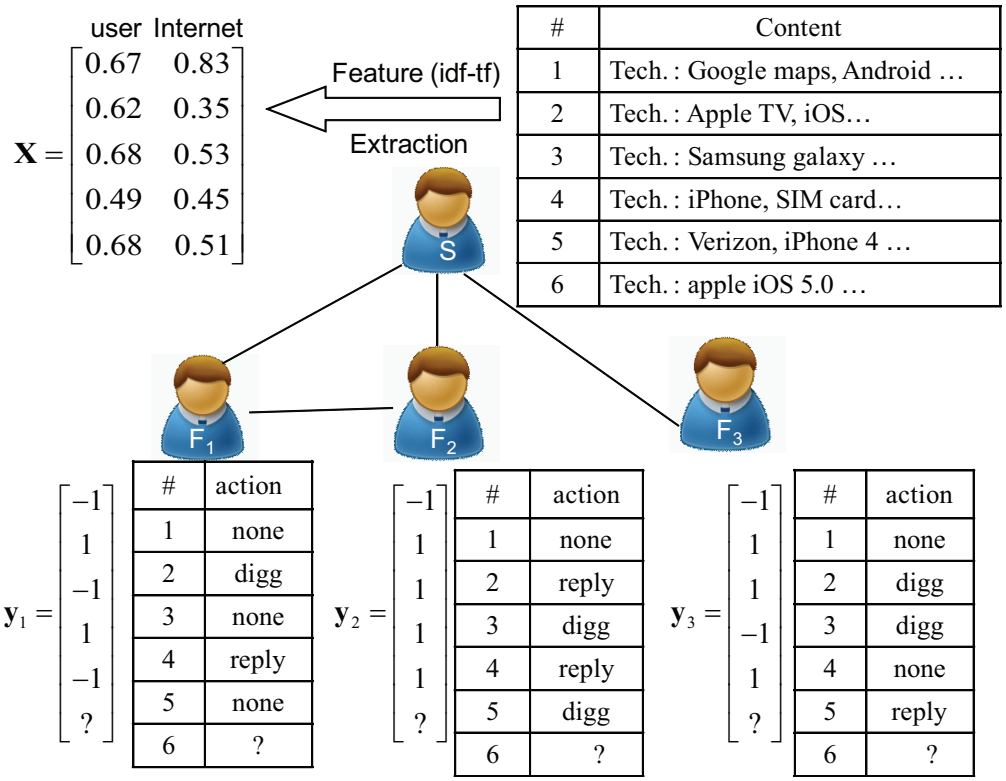
- Group all tasks in a single group and totally ignore the possible difference of tasks [29, 36, 111, 136].

- Use social network topology to model tasks relationships.
- Use previous history of tasks to estimate the possible structure of tasks.

In our experimental study, we have implemented all three strategies and done a case study as shown in Table 5.2. Our result shows that the third strategy is the best over all. The result is not surprising. For example, in Figure 5.2, we show three users. Clearly F_1 and F_2 are somehow related since they are follower of each other. After careful investigation, we conclude that F_1 and F_2 are following each other is due to the fact that both of the users are active in readying and posting entertainment related articles. Following this observation in our multitask learning practice, if our objective is to model the information flow for entertainment related articles, F_1 and F_2 should be group together due to the common interest. However, if our objective is to model the information flow for technology related articles, F_1 and F_2 have quite different interest. Learning F_1 and F_2 together will confuse any learning algorithm.

In summary, we observe that the relationship between tasks in social networks is multilayered in the sense that the relationship may change based on the content of the information. Based on the observation, we have designed a multigraph representation of task relationship. With the multigraph representation, we have modified an existing MTL algorithm by incorporating additional constraint based on the multigraph representation of tasks relationships and investigated the related optimization techniques. In the following subsections we elaborate the description by focusing on four important problems: (i) content based task similarity definition, (ii) multigraph representation of task similarity, (iii) MTL with multigraph constraints, and (iv) efficient optimization. Using comprehensive experimental study, we have demonstrated the effectiveness of the proposed learning method compared with single task learning algorithm SVM [169] and MTL feature learning algorithm without considering the heterogenous relationships [112].

Before we present our mathematical model, we list notations in this paper. We use lowercase letters to represent scalar values, lower-case bold letters to represent vectors (e.g.



#	Content
1	Tech. : Google maps, Android …
2	Tech. : Apple TV, iOS…
3	Tech. : Samsung galaxy …
4	Tech. : iPhone, SIM card…
5	Tech. : Verizon, iPhone 4 …
6	Tech. : apple iOS 5.0 …

#	action
1	none
2	digg
3	none
4	reply
5	none
6	?

#	action
1	none
2	reply
3	digg
4	reply
5	digg
6	?

#	action
1	none
2	digg
3	digg
4	none
5	reply
6	?

Figure 5.2: Data Representation of Content Based Social Behavior Prediction. Five articles with actions of three followers and two words with tf-idf are shown for demonstration only. $\mathbf{X}$ is the object-feature matrix with each row representing an article and each column representing a feature.

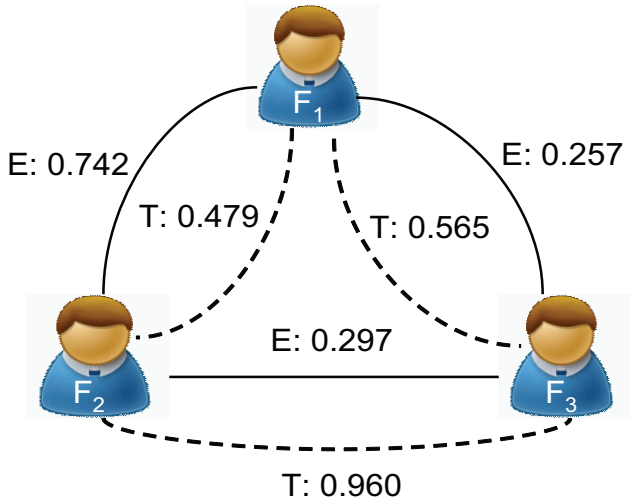


Figure 5.3: Heterogenous social relationships between F_1, F_2 and F_3 for category T: technology and E: entertainment. Dashed line represents the technology connection and solid line represents the entertainment connection. The number for each connection represents the similarity of two users detailed in Equation 5.2.

a), uppercase bold letters to represent matrices (e.g. $\mathbf{A}$), Greek letters $\{\lambda, \lambda_1, \lambda_2...\}$ to represent Lagrange multipliers, and uppercase calligraphic letters to represent sets. Unless state otherwise, all vectors in this paper are column vectors. We use $\|\mathbf{A}\|_1 = \sum_{i,j}^{p} |a_{ij}|$ to denote the l_1 norm of $\mathbf{A}$, $\|\mathbf{A}\|_F$ to denote the Frobenius norm, $\|\mathbf{a}\|_2 = \sqrt{\sum_{i=1}^{p} a_i^2}$ to represent the l_2 norm of vector $\mathbf{a}$, $< \mathbf{A}, \mathbf{B} >= tr(\mathbf{A}^T\mathbf{B})$ to represent the inner product between two matrices where $tr(.)$ is the trace of matrix. Furthermore, given matrix $A \in \mathcal{R}^{p\times k}$, $\|A\|_{1,q} = \sum_{i=1}^{p} \|\mathbf{A}_{i,:}\|_q$ is the l_1/l_q norm, $\mathbf{A}_{i,:}$ is the ith row and $\mathbf{A}_{:,j}$ is the jth column. Unless state otherwise, all vectors in this paper are column vectors.

5.3.2 Problem Statement

Formally, suppose we are given k users (tasks) $\{T_i\}_{i=1}^{k}$. For the ith user T_i, the training set $\mathcal{D}_i$ consists of n articles (samples) $(\mathbf{x}_j^i, y_j^i)$, $j = 1, \cdots, n_i$, where $\mathbf{x}_j^i \in \mathcal{R}^p$. We collect $y_j^i \in \{-1, 1\}$ for the response of the user T_i on article $\mathbf{x}_j^i$. For simplicity, we assume all the tasks have the same number of training samples. The goal of the modeling practice is to learn a function $f_i(\mathbf{x})$ to map the content of the article to the user response, where $f_i(\mathbf{x}) = \mathbf{w}_i^T\mathbf{x}$. The learning task is to seek $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_k]$ with $\mathbf{w}_i$ corresponding to the ith user, such that:

$$\min_{\mathbf{W}} \sum_{i=1}^{k} \sum_{j=1}^{n} \ell(y_j^i, f_i(\mathbf{x}_j^i)) \tag{5.1}$$

(5.1) is minimized.

In this paper, we use linear regression with least square loss function $\ell(y_j^i, f_i(\mathbf{x}_j^i)) = 1/2(y_j^i - f_i(\mathbf{x}_j^i))^2$ to perform classification, which is equivalent to a linear discriminant analysis (LDA) for binary classification [66]. Such a procedure is also widely used in other MTL algorithms for classification problems [26, 111, 190].

Equation (5.1) is ill-posed for high dimension low sample size problems. To remedy the problem, we add l_1 regularization [162] on $\mathbf{W}$ to stabilize (5.1) and to obtain a sparse solution. However, the resulting model neglects the heterogenous structural relationship

among tasks. We addressed task heterogeneities in the following sections.

5.3.3 Content Based User Similarity

With the heterogeneity of social networks, we cannot rely on network topology as we discussed in the introduction section, but have to consider the information content. Additionally, in order to build a more interpretable model, we expect the users that share the same interest will have similar prediction models when seeing the articles from their favorite information categories.

We collect the follower response to quantify the similarity between two users. More specifically, for each follower, we build an activity profile for each content category and calculate pair-wise user similarity as:

Definition 5.3.1. *Suppose that the data set contains k users and covers t categories with q possible actions in the social network (excluding no action), the* user profile $\mathbf{P}^{(l)}$ *for the lth user is a $q \times t$ matrix with entries $p_{ij}^{(l)}$ representing the number of action i on category j, where $1 \leq k \leq n$, $1 \leq i \leq q$ and $1 \leq j \leq t$. Furthermore, the similarity $a_{ij}^{(l)}$ between user i and j for the lth category is the cosine value between vector $\mathbf{P}_{:,l}^{(i)}$ and $\mathbf{P}_{:,l}^{(j)}$:*

$$a_{ij}^{(l)} = \frac{< \mathbf{P}_{:,l}^{(i)}, \mathbf{P}_{:,l}^{(j)} >}{\|\mathbf{P}_{:,l}^{(i)}\| \|\mathbf{P}_{:,l}^{(j)}\|} \tag{5.2}$$

where $< ., . >$ denotes the inner product, $\|.\|$ represents the vector norm and $\mathbf{P}_{:,l}^{(i)}$ is the lth column of profile matrix of user i.

Let $\mathbf{A}^{(l)} = \{a_{ij}^{(l)}\}_{i,j=1}^{n}$, we can view $\mathbf{A}^{(l)}$ as an adjacency matrix for a weighted graph $G^{(l)}$ capturing the structure of users for category l. Since the categories are often diverse in social network, the relationship among users is heterogenous. In the following section, we detail how to incorporate the heterogenous relationship into learning framework.

5.3.4 Heterogenous Task Relationship Incorporation

We capture the structure relationships among tasks for t categories as an undirected multi-graph $G = \{G^{(l)}\}_{l=1}^{t}$, whose nodes correspond to the set of k tasks. Edges in the graph G are multi-edges and weighted, with $\mathbf{a}_{ij} \in \mathcal{R}^t$ defined in Equation (5.2) representing the similarity vector between user i and j. In Figure 5.3, we have shown a multi-graph with two categories on three users.

Given n posts (training samples) and the multi-graph collected from the whole social network for k users, we further assume all the tasks share the same training data since our goal is to predict the user activities towards these n samples. We incorporate the heterogenous structure information by adding a Tikhonov regularization factor $\sum_{l=1}^{n}\sum_{i,j=1}^{k}\mathbf{I}_{:,l}^{T}$ $\mathbf{a}_{ij}\|\mathbf{w}_i-\mathbf{w}_j\|_2^2$ to enforce that the task parameters vary smoothly for neighboring users, where $\mathbf{I}_{t\times n}$ is the indicator matrix with $I_{jk} = 1$ if the kth article belongs to the jth category and 0 otherwise.

The Tikhonov regularization factor can be conveniently written in matrix format in terms of graph Laplacian matrix for individual graph $G^{(l)},\ (1 \le l \le t)$ defined on each category of information content $\sum_{i=1}^{t} r_i tr(\mathbf{W}\mathbf{L}_i\mathbf{W}^T)$, where $r_i = \sum_j I_{ij},\ (1 \le i \le t)$ is the sum of the ith row in matrix $\mathbf{I}$ to summarize the number of posts belonging to the ith category. Note that we also allow category overlapping, which means that each column of $\mathbf{I}$ may have multiple 1s.

Combining with l_1 penalty, the composite regularization function is:

$$R(\mathbf{W}) = \lambda_1\|\mathbf{W}\|_1 + \frac{\lambda_2}{2}\sum_{i=1}^{t} r_i tr(\mathbf{W}\mathbf{L}^{(i)}\mathbf{W}^T) \tag{5.3}$$

where $\lambda_1 > 0,\ \lambda_2 > 0$ are the regularization parameters, $\mathbf{L}^{(i)}$ is the Laplacian matrix of $G^{(i)}$ given by $\mathbf{L}^{(i)} = \mathbf{D}^{(i)} - \mathbf{A}^{(i)}$. $\mathbf{A}^{(i)}$ is the k by k adjacency matrix for category i and $\mathbf{D}^{(i)}$ is the density matrix of $\mathbf{A}^{(i)}$, defined as $\mathbf{D}^{(i)} = (d_{j,l}^{(i)})_{j,l=1}^{k}$ where $d_{j,l}^{(i)} = \begin{cases} \sum_{p=1}^{k} A_{j,p}^{(i)} & \text{if } j = l \\ 0 & \text{otherwise} \end{cases}$. To

avoid any user (task) or category dominate (5.3), we use normalized graph laplacian defined in [32] and normalized category vector $\mathbf{r} = [r_1, \cdots, r_t]^T$ by dividing $max\{r_i | 1 \leq i \leq t\}$. Without state otherwise, $\mathbf{r}$ and graph Laplacian $\{\mathbf{L}^{(i)}\}_{i=1}^t$ are all normalized.

The interpretation of regularization function (5.3) is two folds: (1) we penalize each task individually via l_1 norm rather than block regularization such as l_1/l_2 to select features due to the nonuniformity diverse interests of users; (2) we encourage the followers that show interests on the same categories occurred in training data to have similar solutions.

5.3.5 MTL with Heterogenous Task Relationships

By plugging (5.3) into (5.1), we have the following objective function:

$$\min_{\mathbf{W}} \sum_{i=1}^{k} \sum_{j=1}^{n} \ell(y_j^i, f_i(\mathbf{x}_j)) + R(\mathbf{W}) \tag{5.4}$$

Since all tasks share the same design matrix, (5.4) can be further simplified as:

$$\min_{\mathbf{W}} \frac{1}{2} \|\mathbf{Y} - \mathbf{X}\mathbf{W}\|_F^2 + R(\mathbf{W}) \tag{5.5}$$

where $Y_{n \times k} = [\mathbf{y}_1, \cdots, \mathbf{y}_k]$ is the response matrix with ith column $\mathbf{y}_i \in \mathcal{R}^n$, $X_{n \times p} = [\tilde{\mathbf{x}}_1, \cdots, \tilde{\mathbf{x}}_n]^T$ is the data matrix with the ith row $\tilde{\mathbf{x}}_i \in \mathcal{R}^p$.

Equation (5.5) treats all samples equally, which is only suitable for balanced data sets. Due to the unbalanced sample ratio (a user only responses to a limited number of messages), we introduce a weighting scheme based on positive and negative sample ratio for each task to guarantee that the misclassification cost is more on rare samples. Consider the following optimization problem:

$$\min_{\mathbf{W}} \frac{1}{2} \|\mathbf{B} \otimes (\mathbf{Y} - \mathbf{X}\mathbf{W})\|_F^2 + \lambda_1 \|\mathbf{W}\|_1 + \frac{\lambda_2}{2} \sum_{i=1}^{t} r_i tr(\mathbf{W}\mathbf{L}^{(i)}\mathbf{W}^T) \tag{5.6}$$

where $\otimes$ is the element-wise product and $\mathbf{B}_{n \times k}$ is the weight matrix. For the ith task, let

$\mathcal{P} = \{j|y_j^i = 1\}$ denotes the indices of positive samples, then the weights for positive samples in ith task are given by $B_{\mathcal{P},i} = 1 - |\mathcal{P}|/n$ and negative sample weights are $B_{[1,k]\setminus\mathcal{P},i} = |\mathcal{P}|/n$, where $|\mathcal{P}|$ is the cardinality of set $\mathcal{P}$.

The major challenge in fitting the model described in Equation (5.6) to data is to estimate the parameters $\mathbf{W}$ efficiently and accurately. In the following subsection, we provide the optimization algorithm.

5.3.6 Optimization Algorithms

We propose an efficient algorithm to solve (5.6) based on accelerated gradient decent [126] and projected gradient [15]. The convergence rate of ordinary first order gradient method is $O(1/\epsilon)$ [126] for smooth problems, where ϵ is the desired accuracy. To have a better convergence rate, we use Nestrerov accelerated gradient descent method [127] with $O(1/\sqrt{\epsilon})$ convergence rate, and solve the *generalized gradient update step* for each gradient update step. Such a procedure has demonstrated good scalability and fast convergence in solving various MTL formulations [26, 112].

First, it is straightforward to verify that Equation (5.6) is convex w.r.t. $\mathbf{W}$, hence we can guarantee a global optimal solution. This is because the first two terms are convex, and the sum of trace norm in the third term is also convex due to the positive semi-definite property of graph laplacian and nonnegativity of the normalized category summarization.

Second, Equation (5.6) can be decomposed into two parts: smooth parts and nonsmooth parts. Let $F(\mathbf{W}) = f(\mathbf{W}) + \lambda_1||\mathbf{W}||_1$ with $f(\mathbf{W})$ taking the smooth part:

$$f(\mathbf{W}) = \frac{1}{2}\|\mathbf{B} \otimes (\mathbf{Y} - \mathbf{X}\mathbf{W})\|_F^2 + \frac{\lambda_2}{2}\sum_{i=1}^{t} r_i tr(\mathbf{W}\mathbf{L}^{(i)}\mathbf{W}^T) \tag{5.7}$$

For simplicity, let $\mathcal{M} = \mathcal{R}^{p\times k}$. It is easy to verify that (6.9) is a convex and smooth function

over $\mathbf{W}$ with Lipschitz continuous gradient satisfying:

$$\|\nabla f(\mathbf{W}_x) - \nabla f(\mathbf{W}_y)\|_F \leq L_f \|\mathbf{W}_x - \mathbf{W}_y\|_F, \ \forall \mathbf{W}_x, \mathbf{W}_y \in \mathcal{M} \tag{5.8}$$

where L_f is the Lipschitz constant.

Considering the minimization problem of the smooth function $f(\mathbf{W})$ without l_1 regularization using first order gradient descent method, it is well known that the gradient step has the following update at step $i+1$ with step size $1/L_i$:

$$\mathbf{W}_{i+1} = \mathbf{W}_i - \frac{1}{L_i} \nabla f(\mathbf{W}_i) \tag{5.9}$$

In [126], it has shown that the gradient step (6.11) can be reformulated as a linear approximation of the function f at point W_i regularized by a quadratic proximal term as $\mathbf{W}_i = \underset{\mathbf{W}}{\operatorname{argmin}} \ f_{L_i}(\mathbf{W}, \mathbf{W}_i)$, where

$$f_{L_i}(\mathbf{W}, \mathbf{W}_i) = f(\mathbf{W}_i) + \langle \mathbf{W} - \mathbf{W}_i, \nabla f(\mathbf{W}_i) \rangle + \frac{L_i}{2} \|\mathbf{W} - \mathbf{W}_i\|_F^2 \tag{5.10}$$

Based on the relationship, we combine (6.12) and nonsmooth part together to formalize the *generalized gradient update step*:

$$\begin{aligned} Q_{L_i}(\mathbf{W}, \mathbf{W}_i) &= f_{L_i}(\mathbf{W}, \mathbf{W}_i) + \lambda_1 ||\mathbf{W}||_1 \\ q_{L_i}(\mathbf{W}_i) &= \underset{\mathbf{W}}{\operatorname{argmin}} \ Q_{L_i}(\mathbf{W}, \mathbf{W}_i) \end{aligned} \tag{5.11}$$

The insight of such a formalization is that by exploring the structure of l_1 regularization, we can easily solve the optimization in (6.13), then the convergence rate is the same as that of gradient decent method. Rewriting the optimization problem in (6.13) and ignoring terms that do not depend on W, the objective can be expressed as:

$$q_{L_i}(\mathbf{W}_i) = \underset{\mathbf{W} \in \mathcal{M}}{\operatorname{argmin}} \ (\frac{1}{2} \|\mathbf{W} - (\mathbf{W}_i - \frac{1}{L_i} \nabla f(\mathbf{W}_i))\|_F^2 + \frac{\lambda_1}{L_i} ||\mathbf{W}||_1) \tag{5.12}$$

(6.14) can also be interpreted as gradient projection [15] on a convex set specified by the mixture norm $R(W)$. In this paper, we only consider the equivalent Lagrange form.

As mentioned previously, we employ Nesterov's method to obtain a better convergence rate. Nesterov's method amounts for using two sequences $\{\mathbf{W}_i\}$ and $\{\mathbf{S}_i\}$ in which $\{\mathbf{W}_i\}$ is the sequence of feasible solutions and $\{\mathbf{S}_i\}$ is the sequence of search points. At each step, $\mathbf{S}_i = \mathbf{W}_i + \alpha_i(\mathbf{W}_i - \mathbf{W}_{i-1})$, where α_i is the combination coefficient specified in algorithm 1. Bellow we present the accelerated projected gradient algorithm and the stopping criteria is the change of objective values in two successive steps is less than some predefined threshold (eg. in this paper 10^{-6}).

Algorithm 5 Accelerated Projected Gradient Descent Algorithm

Input: $\mathbf{W}_0 \in \mathcal{R}^{p\times k}$, $L_1 > 0$, $F(.), Q_L(.,.)$ and max-iter.
Output: $\mathbf{W}$.
Initialize $\mathbf{W}_1 := \mathbf{W}_0, t_{-1} := 0, t_0 := 1$;
for $i = 1$ to max-iter **do**
  $\alpha_i := (t_{i-2} - 1)/t_{i-1}$;
  $\mathbf{S} := \mathbf{W}_i + \alpha_i(\mathbf{W}_i - \mathbf{W}_{i-1})$;
  while (true) **do**
    Compute $q_{L_i}(\mathbf{S})$ in generalized gradient update in (6.14);
    if $F(q_{L_i}(\mathbf{S})) > Q_{L_i}(q_{L_i}(\mathbf{S}), \mathbf{S})$ **then**
      $L_i := 2 \times L_i$;
    else
      break;
    end if
  end while
  $\mathbf{S}_{i+1} := q_{L_i}(\mathbf{S}), L_{i+1} := L_i$;
  $t_i := \frac{1}{2}(1 + \sqrt{1 + 4t_{i-1}^2})$;
  if (Convergence) **then**
    $\mathbf{W} := \mathbf{W}_{i+1}$, break;
  end if
end for
return $\mathbf{W}$;

Now we focus on how to solve the generalized gradient update in (6.14). Let $\mathbf{C} =$

$\mathbf{W}_i - \frac{1}{L_i}\nabla f(\mathbf{W}_i)$ and $\tilde{\lambda} = \lambda_1/L_i$, (6.14) can be represented as:

$$\begin{aligned} q_{L_i}(\mathbf{W}_i) &= \underset{\mathbf{W}}{\operatorname{argmin}}(\tfrac{1}{2}||\mathbf{W}-\mathbf{C}||_F^2 + \tilde{\lambda}||\mathbf{W}||_1) \\ &= \underset{w_{ij}}{\operatorname{argmin}} \sum_{i=1}^{p}\sum_{j=1}^{k}(\tfrac{1}{2}(w_{ij}-c_{ij})^2 + \tilde{\lambda}|w_{ij}|) \end{aligned} \tag{5.13}$$

where w_{ij} is the ijth element of $\mathbf{W}$. By the additivity of (6.15), we decompose (6.15) into $p \times k$ subproblems. For each subproblem, we ignore the index i, j:

$$\min_{w} \quad \frac{1}{2}(w-c)^2 + \tilde{\lambda}|w| \tag{5.14}$$

For simplicity, c and w are scalars. Problem (6.16) is a one dimensional optimization and the analytical solution can be easily found. The optimal solution for (6.16) is given by:

$$w^* = \begin{cases} (1 - \frac{\tilde{\lambda}}{|w|})w & |c| > \tilde{\lambda} \\ 0 & \text{otherwise} \end{cases} \tag{5.15}$$

With Eq. (6.15) and (6.17), the problem of generalized gradient update (6.14) can be solved efficiently in a linear time complexity.

5.4 Experiment

We have conducted experiments with four real world data sets crawled from digg.com. To evaluate the performance of our MTL algorithm (MTLTLap), we compared our method with: (1) single task learning algorithm linear kernel SVM [169] with feature selection method SVMRFE [63]; (2) Multi-task feature learning with l_1/l_2 regularization (MTLF) [112] without considering task relationship; and (3) MTL with homogenous networked task relationship (MTLALap) [88]. We have carefully implemented our method MTLTLap, MTLALap and used LIBSVM [24] integrated with spider toolbox [172] for SVM and SLEP package [113] for MTLF.

5.4.1 Data sets

We have crawled four data sets from digg.com, in which there are 10 categories of articles. Due to the tremendous number of users in digg.com, it is impossible to perform analysis for every user. Instead we randomly select four users as "seeds" and collect their article contents as well as their followers' activities towards these articles. The activities are comment, digg, comment & digg and no action. For a seed user, we treat each article as a sample with bag-of-words representation, where the words (features) are extracted from all the articles the seed user submitted. We remove stop words, normalize words and calculate their TF-IDF values with porter stemmer available at `http://sourceforge.net/projects/wvtool/`. Each follower corresponds to a task, where the action of comment or digg are treated as 1 and no action as -1. We eliminate these followers whose total number of comments and diggs on the seed is less than 5. To build the multigraph for capturing the heterogenous relationships among the followers, we also collect the number of submissions, diggs and comments of these followers on each category. Each follower has a 3 by 10 user profile matrix. In Table 5.1, we summarize the characteristics of the five data sets.

Table 5.1: Data set: the symbol of the data set. $\#T$: total number of tasks (followers), $\#S$: total number of samples (stories), $\#F$: total number of features, $\#C$: total number of categories

Seed	username	$\#T$	$\#S$	$\#F$	$\#C$
S_1	nichewp	11	71	942	3
S_2	buhlerchelsey	15	61	3217	5
S_3	GIVINGAWAY	12	57	1167	2
S_4	arjunchauhan24	10	41	1416	5

5.4.2 Evaluation Criteria

Model Construction: We partition each data set into 5 folds to perform 5-fold cross-validation (CV). For MTL methods, we use another 5-fold CV on the training data set to select the regularization parameters λ_1 and/or λ_2 with a simple grid search in the range of $[2^{10}, 2^9, \cdots, 2^{-10}]$. For SVM, we first use 5-fold CV to select the number features in the

rage of $\{25, 50, 75, \cdots, 300\}$, then another 5-fold CV to select parameter C in the range of $[2^{10}, 2^9, \cdots, 2^{-10}]$.

Model Evaluation: We collect the following metrics:

$$\begin{aligned} precision &= TP/(TP + FP) \\ recall &= TP/(TP + FN) \\ F_1 &= 2 * precision * recall/(precision + recall) \end{aligned}$$

where TP stands for true positive, FP stands for false positive, TN stands for true negative, FN stands for false negative. All the values reported are collected from the testing data set only and are averaged across 5-fold CV with 6 replicates. Note that since thee is even no positive samples available in testing data during cross validation due to the imbalanced class ratio, we skip such folds when averaging the final F_1 score.

5.4.3 Experiment Performance

We compare our method with SVM, MTLF and MTLALap in terms of the average F_1 score for four different data sets in Figure 5.4. The standard deviation for each task is around 8%-15% for all the methods and we do not report them for simplicity. From Figure 5.4, we observe that the performance of single task SVM is very unstable compared with MTL approaches. For example, the average F_1 score is 0 for the 5th task of nichewp (S_1) because SVM predicts all the samples as negative (no comment or digg action) when the class ratio is unbalanced. However for the 2nd task, the performance is comparable to MTLF because the class ratio is balanced. Such an observation demonstrates the advantages of MTL vs STL for improving the generalization performance especially when training samples are limited and imbalanced.

Among the MTL methods, MTLALap and our method MTLTLap outperform MTLF for most tasks for four data sets, which confirms that considering the relationship among tasks will boost the learning performance. Finally, compared with MTLALap without dif-

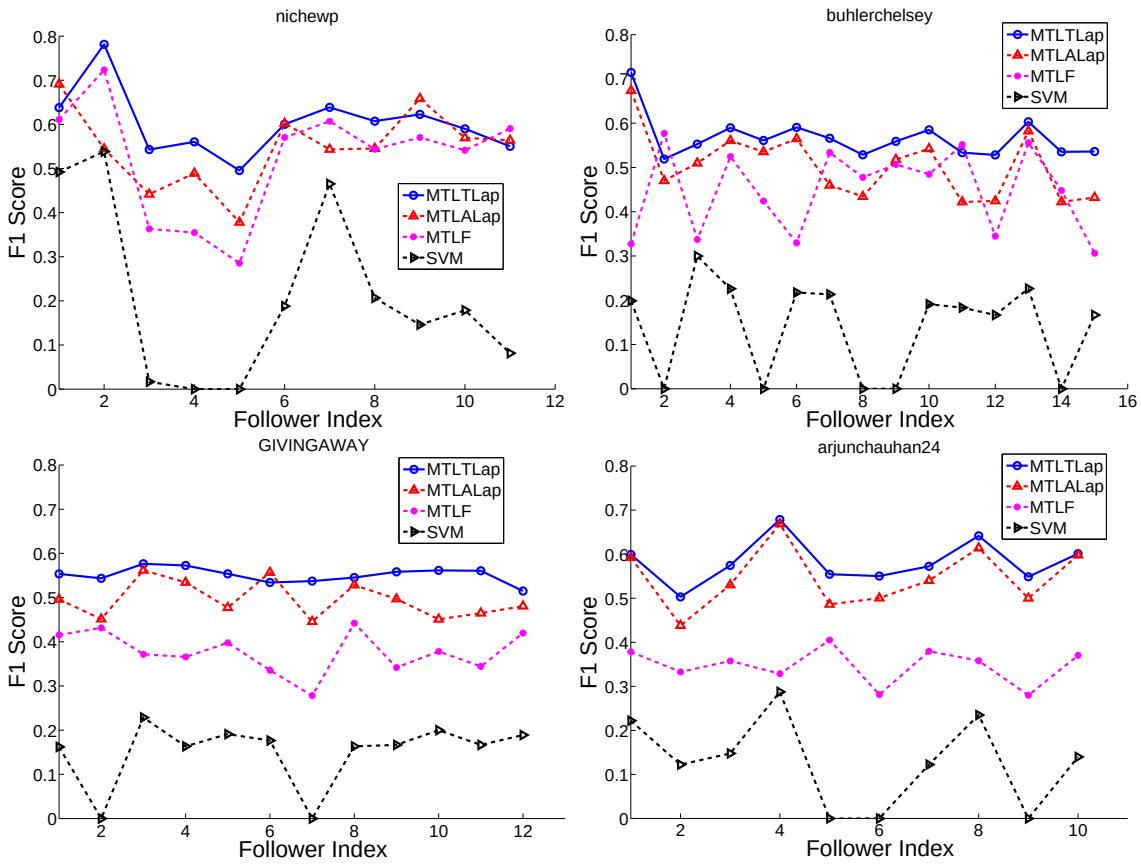


Figure 5.4: Average F_1 score for 4 seed users. Each figure's title corresponds to a username.

Table 5.2: Average F_1 score, Precision and Recall of three MTL methods on 3 tasks of seed S_2. black fonts denote the highest values among all competing methods for a task.

	T_2			T_4			T_{10}		
Method	F_1	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall
MTLTLap	0.519	0.402	**0.733**	**0.590**	0.485	**0.751**	**0.585**	**0.446**	**0.850**
MTLALap	0.470	0.363	0.667	0.561	0.461	0.714	0.542	0.413	0.789
MTLF	**0.577**	**0.573**	0.581	0.525	**0.517**	0.533	0.485	0.419	0.576

ferentiating the heterogenous social connections, our approach MTLTLap performs better, although the difference is subtle for some tasks. The reason is that when the training samples are dominated by only one or two categories (i.e. S_2), our method may not perform very well compared with MTLALap especially when the true topology actually reflects the user interests.

To further explore why our approach provides a reasonable performance compared with the other methods, we select three followers 2, 4 and 10 from S_2, which corresponds to

the example shown in Figure 5.2. The data set of S_2 covers 5 categories dominated by technology and game and the result is shown in Table 5.2. We see that MTLTLap and MTLALap outperforms MTLF for two out of three tasks, which confirms the importance of task relationship incorporation. Meanwhile, the F_1 score and recall of the third strategy MTLTLap is consistently better than MTLALap, which demonstrates that our approach can more effectively identify positive samples in an unbalanced classification task.

To explain this phenomenon, we recall the similarities of the three followers as shown in Figure 5.3: F_1, F_2 and F_3 all like technology but with different levels; Meanwhile, F_1 and F_2 share interests in entertainment. Disregarding the information content category in the training data, the heterogenous relationship among followers will be either mixed (i.e. MTLALap) or ignored (i.e. MTLF). However in our approach, by introducing heterogenous task relationship induced by different categories and the weight of each category in the training data, we enforce similar tasks to have similar parameters based on the categories specified in the training data. Hence our prediction result is more stable and the resulting model is more interpretable.

5.5 Conclusions

In this paper, we tackle the problem of predicting user behavior to friends' postings in social networks. We argue that social information flow and socialization behaviors are not only related to social relationships (i.e. graph topology), but also information contents and user interests. Therefore, we should integrate all these factors to construct a better model. Towards that end, we presented a multi-task learning algorithm with heterogenous task relationships, in which we capture the heterogenous relationships induced by different information categories among users as an undirected multigraph and incorporate such information by introducing a trace norm regularized graph Laplacian to standard MTL formalization. Using a comprehensive experimental study with social network data collected

from digg.com and comparing with current state-of-the-art, we demonstrate that the new algorithm achieves better prediction performance.

Chapter 6

Multi-task Learning with Structured Input and Output

In Chapter 3 and 4, we have discussed one line of my research on learning from the data with structured input. In particular, the input features have structural information and the structural relationship is given as a prior. In Chapter 5, we have demonstrated the utility of multi-task learning with structured output tasks in the application of social network analysis. It is worthwhile to investigate the problem of multi-task learning with both structured input and output.

Multi-task Learning (MTL) aims to enhance the generalization performance of supervised regression or classification by learning multiple related tasks simultaneously. In this chapter, we aim to extend the current MTL techniques to high dimensional data sets with structured input and structured output (SISO), where the SI means the input features are structured and the SO means the tasks are structured. We investigate a completely ignored problem in MTL with SISO data: the interplay of structured feature selection and task relationship modeling. We hypothesize that combining the structure information of features and task relationship inference enables us to build more accurate MTL models. Based on the hypothesis, we have designed an efficient learning algorithm, in which we utilize a task covariance

matrix related to the model parameters to capture the task relationship. In addition, we design a regularization formulation for incorporating the structured input features in MTL. We have developed an efficient iterative optimization algorithm to solve the corresponding optimization problem. Our algorithm is based on the accelerated first order gradient method in conjunction with the projected gradient scheme. Using two real-world data sets, we demonstrate the utility of the proposed learning methods.

6.1 Introduction

Multi-task learning (MTL) has recently attracted extensive research interest in the data mining and machine learning community [25, 26, 29, 41, 111, 112]. It has been observed that learning multiple related tasks simultaneously often improves modeling accuracy and leads to better feature selection, especially in cases where each task has very limited number of training samples. MTL has been applied to a wide range of application areas including information retrieval [25], computational neuroscience [111], genetic analysis [191], disease progression prediction [195], image classification [26] and collaborative filtering [186].

We aim to extend the current MTL techniques to high dimensional data sets with structured input and structured output (SISO) [111, 88, 189, 191]. SISO data types could be found in a diverse set of application domains including health care [130, 152, 157], information flow analysis in social networks [140], computational neuroscience [111, 122] and information retrieval [25]. For example in information flow analysis of social networks [140], we model a user's behavior regarding the social information (e.g. whether they recommend a video to a user group in YouTube or whether they participate in a discussion) to quantify the user interest in the information content [28, 181]. Such modeling usually involves multiple users that can be divided into different communities with different interests and background (SO). The social information content is usually represented as a bag of words with a high dimensionality where words have relationships (SI) such as being synonymous

or antonymous [48].

Recognizing the fact that not all tasks are uniformly related, there is substantial research interest in modeling task relationships in the state-of-the-art MTL methods [5, 13, 79, 177, 189]. For example, several recent MTL algorithms modeled task relationship as a covariance matrix [13, 79, 189] or positive definite matrices linked to the task parameters [5] and learned such matrices from data. The common concern of these methods is that there is no feature selection in the modeling process and that makes those models less attractive for high dimensional data.

In this paper, we investigate a completely ignored problem in MTL with SISO data: the interaction of structured feature selection and task relationship modeling. Although feature selection has been widely utilized in MTL to improve modeling accuracy [4, 111, 112, 136, 191], the core limitation of these feature selection methods in MTL, when applied to complex real-world data sets, is the ignorance of the potential interplay between the structured input and the task relationship. We illustrate the limitation with the following example.

Considering a problem of predicting cancer status based on Microarray data sets, where there are multiple different data sets for different types of cancers. Each data set is composed of multiple Microarray data from patients who have or do not have the specific cancer. Some cancers are "similar" to each other (e.g. breast cancer vs ovary cancer) while some are quite different (e.g. breast cancer vs prostate cancer). For similar cancer types (tasks), learning models built for those similar cancer types (tasks) are expected to share similar features; for dissimilar ones, learning models are expected to select different features. However, current feature selection methods for MTL [4, 111, 112, 136, 191] select a common subset of features across all the tasks. Moreover, features in the case study are genes and they have structured input since genes are typically organized as pathways. Such pathway information is known important for predictive model construction in multiple studies [47, 106]. In this example, the gene pathway information provides the possible structure information of features (genes) and the cancer type similarity provides possible structure information regarding learning

tasks (predicting whether a patient has a specific cancer type). The challenges of designing learning algorithms is how to (i) incorporating such information for efficient MTL and (ii) inferring such structure information, if necessary, to gain insights of the data. Further details of the case study could be found in the section of our experimental studies.

We hypothesize that combining structured feature selection and task relationship inference enables us to build more accurate MTL models for SISO data. Our hypothesis is based on the following insights: (1) discriminative and informative features will guide more accurate task relationship inference; and (2) accurate task relationship will benefit feature selection.

Towards an efficient incorporation of structured input and task relationship inference, we have designed a regularized MTL model where we use an undirected graph defined on features (feature graph) to capture the structured input and learn a task covariance matrix related to the task parameters to measure the task relationship. To enable that dissimilar tasks select different subsets of features, we use l_1 regularization to penalize each task individually, and use trace regularization on the task covariance matrix to encourage similar tasks to share a similar subset of features. We have derived efficient optimization algorithms based on the Nestrerov's accelerated gradient descent algorithm [127] and the projected gradient scheme [15] to solve the corresponding optimization problem.

Though our methodology is generic, our paper is particularly motivated by two real-world problems from the health care domain, that of micoraray based cancer prediction and that of neuron response prediction. Our experimental studies demonstrate the effectiveness of the proposed MTL method as compared to the state-of-the-art MTL algorithms on the two real-world applications.

6.2 Related Work

In this section, we summarize the related work. Since we propose a feature selection method for multi-task learning, we first briefly introduce feature selection and then discuss Multi-task Learning.

Current methods for feature selection can be roughly divided into two categories: feature extraction and feature selection. Feature extraction methods [60, 184], such as Principle Component Analysis (PCA) and Linear Discriminative Analysis LDA [184], project data to a lower dimensional space and hence obtain a small number of latent features. Feature selection methods (filtering and wrapper methods) select individual/subset of informative features that are relevant to class labels. For example, Kong et al. [97] proposed a branch-and-bound feature selection algorithm for multi-label graph classification; Nizar Bouguila et al. [14] adopted feature selection in mixture model for text and image categorization; Zhang et al. [192] developed a feature selection method via supervised dimensionality reduction while preserving the locality of data points. In [12], a comprehensive study of feature selection methods is evaluated on synthetic data.

In parallel, MTL has been widely investigated in data mining and machine learning communities. The state-of-the-art multi-task learning algorithms may be roughly divided into three categories based on how they utilize the task relationship. Multi-task feature learning [4, 111, 112, 136] assumed all the tasks were homogenous and learned a common low dimensional representation without considering the task relationship. The common features were learned by block regularization such as l_1/l_2 [4, 112] and l_1/l_∞ [29, 111, 136]. MTL with known task relationship [6, 40, 88, 150] utilized the prior knowledge on task relationship via trace norm regularization to learn model parameters so that similar tasks share similar parameters. However, such methods used all the features to build MTL model and were not suitable for high dimensional data. Besides, the task relationship is not always available beforehand. MTL with task relationship inference [5, 13, 79, 189] learned a task covariance matrix [13, 189], a spectral function linked to the task covariance matrix [5] or a general

positive semi-definite matrix [79] describing clustered tasks from data. Nevertheless, such methods used all the features and hence afford no sparsity in their solutions. As pointed in [117], taking advantage of sparsity in multi-task learning is very important for improving the generalization performance, especially for tasks with high dimensional feature space and low sample size.

To alleviate the problem of existing methods, Zhang et al. [191] recently extended their previous work in [189] to perform feature selection and task relationship inference simultaneously by employing a block-regularization with the l_1/l_q norm regularization where $1 < q < \infty$. The limitations of the work are that (i) the method ignores the possible structured input information among features; and (ii) the method selects a subset of features for all tasks regardless the relatedness the tasks.

Recently, there is growing interest to detect irrelevant (outlier) tasks in the development of the multi-task learning algorithms [27, 185]. For example, Chen et al. captured the relationship of multiple related tasks using a low-rank structure and meanwhile identified the outlier tasks using a group-sparse structure. However, these works only detect task outliers without indicating how relevant the remaining tasks are. Furthermore, the potential structured input information is not utilized.

We summarize the most recent related work in Table 6.1. x means the method has the corresponding property. For each category, we select one representative method. The symbol of the table is as following: SF, Sparse Feature; TRF, Fixed Task Relationship; TRI, task relationship inference; SI, Structured Input; MTLasso, Multi-task feature learning [111]; MTLFTR, MTL with fixed task relationship [88]; MTLTR, MTL with task relationship inference [189]; MTLPTR: MTL with feature learning and task relationship inference [191].

Though MTL has been studied for a long time, none of the existing methods considers the interaction of structured input information among features and heterogeneous task relationship inference simultaneously. The objective of this paper is to incorporate structured feature selection and heterogenous task relationship inference into MTL to build a more

Table 6.1: Summarization of related work.

	SF	TRF	TRI	SI
MTLasso	x			
MTLKTR		x		
MTLTR			x	
MTLPTR	x	x		

accurate and interpretable model.

6.3 Methodology

In this section, we describe the proposed MTL framework. Our framework is an extension to [4, 111, 112], i.e., task parameters lie in a linear manifold and share a common linear subspace. As mentioned earlier, due to the existence of heterogenous tasks, it is impractical that all the tasks share a common linear subspace (subset of features).

In our approach, we adopt the technique in [189] that utilized a task covariance matrix to model the task relationship. Moreover, rather than jointly selecting features across all the tasks via block regularization as the previous work [4, 111, 112], we allow each task to select its own subset of features and similar tasks to share similar model parameters as well as common features. In addition, we incorporate the structured input information of features into MTL so that the selected features are clustered or tended to be connected on the graph for better model interpretation.

6.3.1 MTL with Sparse Features and Task Relationship Inference

We first consider the problem of MTL with heterogenous tasks and our aim is to derive model parameters and task relationship simultaneously. The same problem was investigated in [191], in which the problem was cast in a probabilistic framework and the task relationship was measured by a task covariance matrix specific to the columns of model parameters. To achieve a sparse solution, the authors assumed that task parameters share a common a subset

of features and employed l_1/l_q $(1 < q < \infty)$ block regularization to model parameters.

As what we mentioned above, each task in our framework selects its own subset of features and the feature sets of two closely related tasks have common features. Adopted the task relationship regularization component in [189], we penalize each task individually and solve the following optimization problem:

$$\begin{aligned} \min_{W,\Omega} \quad & \sum_{i=1}^{k}\sum_{j=1}^{n_i} \ell(y_j^i, f_i(\mathbf{x}_j^i)) + \lambda_1||W||_1 + \frac{\lambda_2}{2} tr(W\Omega^{-1}W^T) \\ \text{s.t.} \quad & \Omega \succeq 0 \quad tr(\Omega) = k \end{aligned} \tag{6.1}$$

where W is the model parameters with each column corresponding to a task, Ω is the task covariance matrix and $\lambda_1, \lambda_2 > 0$ are regularization parameters that controls the model sparsity and smoothness across tasks. The l_1 regularization penalizes each task individually and the trace regularization term enforces task parameters to be similar according to the similarity encoded in the covariance matrix inverse, also known as precision matrix, whose elements have an interpretation in terms of partial correlations.

6.3.2 Structured Input Incorporation

We capture the structured input information among features as an undirected graph G whose nodes represent the features. Edges represent a particular relationship between pairwise features and are weighted with a_{ij} denoting the weight between feature i and feature j. We call such a graph defined on features *feature graph*. For example, if features are genes or bag-of-words, the feature graph can be constructed either from domain knowledge (e.g. gene pathways, wordnet [48]) or derived from data [145]. We will detail how to build the feature graph for the two real world data sets in our experimental studies.

We incorporate the structure information of features by adding a Tikhonov regularization factor $\sum_{i,j=1}^{p} a_{ij}\|\tilde{\mathbf{w}}_i - \tilde{\mathbf{w}}_j\|_2^2$ to enforce that the parameters vary smoothly for neighboring features, where $\tilde{\mathbf{w}}_i$ is the ith row of W. The Tikhonov regularization factor can be conveniently

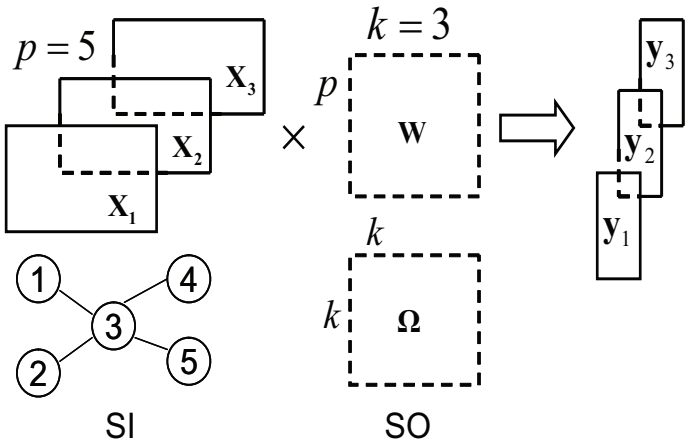


Figure 6.1: A demo for the Multi-task linear model with structured input (SI) and task relationship inference (SO) with 5 features and 3 tasks. Solid line square represents input and dashed line square represents output.

written in matrix format in terms of graph Laplacian matrix L as $tr(W^T LW)$, where L is the *Laplacian* of G given by: $L = D - A$. A is the p by p adjacency matrix $A = (a_{i,j})_{i,j=1}^{p}$. D is the density matrix of A, defined as $D = (d_{i,j})_{i,j=1}^{p}$ where $d_{i,j} = \begin{cases} \sum_{k=1}^{p} a_{i,k} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$.

To avoid having any feature "dominate" the penalization function, we use the *normalized Laplacian* $D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$ to normalize the weight of each feature.

A similar formalization was proposed in [44] for single task learning in the boosting, but it is a special case of our MTL framework when there is only one task. Combining the sparse regularization and task relationship modeling, we consider the following objective function:

$$\begin{aligned} \min_{W,\Omega} \quad & \sum_{i=1}^{k} \sum_{j=1}^{n_i} \ell(y_j^i, f_i(\mathbf{x}_j^i)) + \lambda_1 ||W||_1 + \\ & \tfrac{\lambda_2}{2} tr(W^T LW) + \tfrac{\lambda_3}{2} tr(W \Omega^{-1} W^T) \\ \text{s.t.} \quad & \Omega \succeq 0 \quad tr(\Omega) = k \end{aligned} \tag{6.2}$$

The trace regularization term $tr(W^T LW)$ in (6.2) imposes smoothness across features, in other words, the selected features tend to be connected in the feature graph.

In Figure 6.1, we show our scheme for a data set with $p = 5$ features and $k = 3$ tasks.

The structured input in this example is the graph defined on 5 features and the output are the model parameters W and task covariance matrix Ω.

6.3.3 Relationship with existing MTL algorithms

As we discussed before, some existing MTL algorithms [4, 111, 112] assume a uniform task relationship to jointly select features, while others [88, 79, 189, 191] treat tasks non-uniformly and either assume the task relationship is given as a prior or learn the relationship from data. In this section, we discuss the relationship our method with these existing regularized MTL algorithms.

The objective function for multi-task feature learning [4, 111, 112] can be summarized in the following form:

$$\min_{W} \sum_{i=1}^{k} \sum_{j=1}^{n_i} \ell(y_j^i, f_i(\mathbf{x}_j^i)) + \lambda R(W) \tag{6.3}$$

where $R(W)$ is the regularization function on model parameters and it could be l_1/l_2 or l_1/l_∞. In our framework, we use l_1/l_1 to enforce sparsity and feature laplacian L_2 regularization to add smoothness on feature selection.

The methods in [40, 88] assume the task relationship is given and they incorporate it into learning via $R(W) = tr(WLW^T)$, where L is the task relationship graph laplacian. In our framework, $\Omega^{-1} = L$. Obviously, a limitation of these methods is that only feature selection is ignored.

The most related work to ours are the MTL task relationship learning (MTLTR) [189] and MTL feature selection and task relationship learning (MTLPTR) [191]. In MTLTR, the objective function takes the following form:

$$\min_{W,\Omega} \sum_{i=1}^{k} \sum_{j=1}^{n_i} \ell(y_j^i, f_i(\mathbf{x}_j^i)) + \lambda_1 tr(WW^T) + \lambda_2 tr(W\Omega^{-1}W^T) \tag{6.4}$$

where Ω is the task covariance matrix and the second term is used to stabilize the solution.

Compared with MTLTR. our formulation (6.2) considers feature selection and it shows a superior performance for high dimensional data. For MTLPTR [191], they extended their work in [189] to perform feature selection via l_1/l_q where $1 < q < \infty$. Though considering feature selection, the method neglects the structured input information. Furthermore, the block regularization selects features regardless of task relevance since once one feature is selected by a few tasks, the same feature has to be chosen by other tasks as well. We will detail the issue in our experimental study.

In summary, compared with existing methods, our method is very appealing in that it can not only learn task relationships, but use the task relationship and structured input information on features to guide feature selection. This makes it easy to identify the tasks and relevant features that are useful for multi-task learning.

6.3.4 Optimization

We propose an efficient algorithm to solve (6.2) based on the accelerated gradient decent method [126] and the projected gradient scheme [15]. The convergence rate of ordinary first order gradient method is $O(1/\epsilon)$ [126] for smooth problems, where ϵ is the desired accuracy. To have a better convergence rate, we use the Nestrerov accelerated gradient descent method [127] with $O(1/\sqrt{\epsilon})$ convergence rate, and solve the *generalized gradient update step* for each gradient update step. Such a procedure has demonstrated good scalability and fast convergence in solving various sparse learning formulations [26, 82, 112].

First, we characterize the convexity of (6.2) in the following theorem.

Theorem 6.3.1. *The problem (6.2) is jointly convex for W and Ω.*

Proof. It is obvious that the first two terms are convex w.r.t. W. For the third term, the trace regularization term regularization $tr(W^T L W) = \sum_{i=1}^{k} \mathbf{w}_i^T L \mathbf{w}_i$ is also convex since L is the graph Laplacian and hence positive semi-definite. The last term $tr(W\Omega^{-1}W^T) = \sum_{i=1}^{p} \tilde{\mathbf{w}}_i \Omega^{-1} \tilde{\mathbf{w}}_i^T$ is also convex w.r.t. W and Ω under the constrain $\Omega \succeq 0$ [15, 189], where

$\tilde{\mathbf{w}}_i$ is the ith row of W. Hence the objective function (6.2) and the constraints are convex with respect to all variables and hence problem (6.2) is jointly convex. □

Since (6.2) is jointly convex for W and Ω, a global optimal solution is guaranteed. However, it is difficult to optimize two variables simultaneously. Below, we present an algorithm to solve (6.2), which optimizes W, Ω iteratively and alternatively.

Ω **given** W: If W is fixed, we can ignore the regularization part independent of Ω. Now (6.2) degenerate to

$$\begin{aligned} &\min_{\Omega} tr(W\Omega^{-1}W^T) \\ &\text{s.t. } \Omega \succeq 0 \quad tr(\Omega) = k \end{aligned} \tag{6.5}$$

The solution is given by:

$$\Omega = \frac{kA^{\frac{1}{2}}}{tr(A^{\frac{1}{2}})} \tag{6.6}$$

where $A = W^TW$ is the gram matrix. We can obtain the solution (6.6) using Cauchy-Schwarz inequality on equation (6.5). Refer to [189] therein for the proof.

W **given** Ω: On the other hand, if Ω is fixed, the optimization becomes:

$$\begin{aligned} \min_{W} \quad & \sum_{i=1}^{K}\sum_{j=1}^{n_i} \ell(y_j^i, f_i(\mathbf{x}_j^i)) + \lambda_1||W||_1 + \\ & \tfrac{\lambda_2}{2}tr(W^TLW) + \tfrac{\lambda_3}{2}tr(W\Omega^{-1}W^T) \end{aligned} \tag{6.7}$$

Now (6.7) can be rewritten as:

$$\min_{W} \quad F(W) \stackrel{\text{def}}{=} f(W) + R(W) \tag{6.8}$$

where $f(W)$ takes the smooth parts of (6.7)

$$\begin{aligned} f(W) = \quad & \sum_{i=1}^{k}\sum_{j=1}^{n_i} \ell(y_j^i, f_i(\mathbf{x}_j^i)) + \frac{1}{2}\lambda_2 tr(W^TLW)) \\ & + \tfrac{\lambda_3}{2}tr(W\Omega^{-1}W^T) \end{aligned} \tag{6.9}$$

and $R(W)$ takes the nonsmooth part,

$$R(W) = \lambda_1 \|W\|_1 \tag{6.10}$$

Considering the minimization problem of the smooth function $f(W)$ without regularization $R(W)$ using first order gradient descent method, it is well known that the gradient step has the following update at step $i+1$ with step size $1/L_i$:

$$W_{i+1} = W_i - \frac{1}{L_i} \nabla f(W_i) \tag{6.11}$$

In [126], it has shown that the gradient step (6.11) can be reformulated as a linear approximation of the function f at point W_i regularized by a quadratic proximal term as $W_i = \underset{W}{\operatorname{argmin}}\ f_{L_i}(W, W_i)$, where

$$f_{L_i}(W, W_i) = f(W_i) + \langle W - W_i, \nabla f(W_i) \rangle + \frac{L_i}{2} \|W - W_i\|_F^2 \tag{6.12}$$

Based on the relationship, we combine (6.12) and (6.10) together to formalize the *generalized gradient update step*:

$$\begin{aligned} Q_{L_i}(W, W_i) &= f_{L_i}(W, W_i) + \lambda_1 ||W||_1 \\ q_{L_i}(W_i) &= \underset{W}{\operatorname{argmin}}\ Q_{L_i}(W, W_i) \end{aligned} \tag{6.13}$$

The insight of such a formalization is that by exploring the structure of regularization $R(.)$, we can easily solve the optimization in (6.13), then the convergence rate is the same as that of gradient decent method. Rewriting the optimization problem in (6.13) and ignoring terms that do not depend on W, the objective can be expressed as:

$$q_{L_i}(W_i) = \underset{W \in \mathcal{M}}{\operatorname{argmin}}\ (\frac{1}{2} \|W - (W_i - \frac{1}{L_i} \nabla f(W_i))\|_F^2 + \frac{\lambda_1}{L_i} ||W||_1) \tag{6.14}$$

(6.14) can also be interpreted as gradient projection [15] on a convex set specified by $R(W)$. In this paper, we only consider the equivalent Lagrange form.

As mentioned previously, we employ Nesterov's method to obtain a better convergence rate. Nesterov's method amounts for using two sequences $\{W_i\}$ and $\{S_i\}$ in which $\{W_i\}$ is the sequence of feasible solutions and $\{S_i\}$ is the sequence of search points. At each step, $S_i = W_i + \alpha_i(W_i - W_{i-1})$, where α_i is the combination coefficient specified in algorithm 1. Bellow we present the accelerated projected gradient algorithm. The stopping criteria is that the change of objective values in two successive steps is less than a predefined threshold (e.g. 10^{-4}).

Algorithm 6 Accelerated Projected Gradient Descent Algorithm

Input: $W_0 \in \mathcal{R}^{p\times k}$, $\Omega \in \mathcal{R}^{k\times k}$, $L_1 > 0$, $F(.), Q_L(.,.)$ and max-iter.
Output: W.
Initialize $W_1 := W_0, t_{-1} := 0, t_0 := 1$;
for $i = 1$ to max-iter **do**
  $\alpha_i := (t_{i-2} - 1)/t_{i-1}$;
  $S := W_i + \alpha_i(W_i - W_{i-1})$;
  while (true) **do**
    Compute $q_{L_i}(S)$ in generalized gradient update;
    if $F(q_{L_i}(S)) > Q_{L_i}(q_{L_i}(S), S)$ **then**
      $L_i := 2 \times L_i$;
    else
      break;
    end if
  end while
  $W_{i+1} := q_{L_i}(S), L_{i+1} := L_i$;
  $t_i := \frac{1}{2}(1 + \sqrt{1 + 4t_{i-1}^2})$;
  if (Convergence) **then**
    $W := W_{i+1}$, break;
  end if
end for
return W;

Now we focus on how to solve the generalized gradient update in (6.14). Let $C =$

$W_i - \frac{1}{L_i}\nabla f(W_i)$ and $\tilde{\lambda} = \lambda_1/L_i$, (6.14) can be represented as:

$$\begin{aligned} q_{L_i}(W_i) &= \underset{W}{\operatorname{argmin}}(\tfrac{1}{2}||W - C||_F^2 + \tilde{\lambda}||W||_1) \\ &= \underset{w_{ij}}{\operatorname{argmin}} \sum_{i=1}^{p}\sum_{j=1}^{k}(\tfrac{1}{2}(w_{ij} - c_{ij})^2 + \tilde{\lambda}|w_{ij}|) \end{aligned} \tag{6.15}$$

where w_{ij} is the ijth element of W. By the additivity of (6.15), we decompose (6.15) into $p \times k$ subproblems. For each subproblem, we ignore the index i, j:

$$\min_{w} \quad \frac{1}{2}(w - c)^2 + \tilde{\lambda}|w| \tag{6.16}$$

For simplicity, c and w are scalars here. Problem (6.16) is a one dimensional optimization problem and the analytical solution can be easily found. The optimal solution for (6.16) is given by:

$$w^* = \begin{cases} (1 - \frac{\tilde{\lambda}}{|w|})w & |c| > \tilde{\lambda} \\ 0 & \text{otherwise} \end{cases} \tag{6.17}$$

With Eq. (6.15) and (6.17), the problem of generalized gradient update (6.14) can be solved efficiently with the time complexity of $O(pk)$.

We summarize what is briefly discussed previously in the algorithm (MTLapTR) bellow. Given training data $\{X_i \in \mathcal{R}^{n_i \times p}\}_{i=1}^k$, $\{\mathbf{y}_i \in \mathcal{R}^{n_i}\}_{i=1}^k$ and regularization parameters $\lambda_1, \lambda_2, \lambda_3$, we optimize two matrix variables alternatively and return the coefficient matrix W and covariance matrix Ω.

6.4 Experimental Studies

We have performed a comprehensive evaluation of our algorithm (MTLapTR) on modeling accuracy, task relationship inference and feature selection performance using two real-world data sets. We have compared with the state-of-the-art methods including Multi-task Lasso (MTLasso) [111], MTL with task relationship inference (MTLTR) [189] and MTL with fea-

Algorithm 7 Main Algorithm (MTLapTR)

Input: $\{X_i \in \mathcal{R}^{n_i \times p}\}_{i=1}^k$, $\{\mathbf{y}_i \in \mathcal{R}^{n_i}\}_{i=1}^k$, $\lambda_1, \lambda_2, \lambda_3$ and max_iter.
Output: W, Ω.
$\Omega := I_{k \times k}$;
for $iter = 1$ to max_iter **do**
  Compute W given Ω using Algorithm 1;
  Compute Ω given W via 6.6;
  if (Converge) **then**
    break;
  end if
end for
return W, Ω;

ture learning and task relationship inference (MTLPTR) [191]. We obtained the source code for MTLasso from the authors and implemented the other two methods since their executables are not available. To validate our hypothesis on the importance of incorporation of structured input and task relationship inference, we have implemented two special cases of our method: MTLapTR without Laplacian regularization (MTL1TR) and MTLapTR without task relationship inference (MTLap). To demonstrate the utility of multi-task learning, we also compared multi-task algorithms with single task learning algorithms: support vector machine (SVM) and support vector regression (SVR) [169].

6.4.1 Data Sets

We utilized two real world data sets: fMRI data [111, 122] from computational Neuroscience and Microarray data sets [130, 152, 157] for cancer diagnostics. The following details the collection and preprocessing of the two data sets.

Microarray: The data set was composed of multi-category cancer tumors for human collected from [130, 152, 157]. All the studied data sets were collected from Affymetrix arrays HG-U95 or Hu6800, and expression values (average difference units) were computed using the Affymetrix GENECHIP analysis software. In our experiment, we studied 8 binary classification tasks, where 5 were from [157], 2 were from [130] and 1 was from [152]. From

[157], we singled out 5 types of tumors of 11 in total: breast (BR), ovary (OV), kidney (KI), liver (LI) and bladder (BL) as positive samples and perform random sampling from the collection of the rest 6 categories as negative samples, resulting in 5 tasks. In [130], four types brain tumors were investigated and we selected two challenging pairs: Brain Classific GBM (BCG) VS Brain Non-classic GBM (BNG) and Brain Classic AO (BCO) VS Brain Nonclassic AO (BNO) [1]. In [152], we used the data of Prostate tumor (PR) VS normal tumor for the 8th task. From the original probe sets, we first removed those genes with the variance of < 0.3, then filtered out those without a valid mapping to a KEGG gene name. In our final data set, we had 8925 common genes shared for the three Microarray data sets. From

Table 6.2: Microarray data sets for 8 tasks. 8925 features are shared for these tasks. #S: total number of samples; #P: number of positive samples; #N: number of negative samples.

	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8
Data	BR	KI	OV	LI	BL	BCG	BCO	PR
#S	54	22	54	14	16	21	29	102
#P	27	11	27	7	8	14	14	50
#N	27	11	27	7	8	7	15	52

the KEGG database [86], we collected 212 human pathways, then merged those pathways to generate a giant network and extracted the subnetwork incident on the 8925 genes as the feature graph. In Table 6.2, we summarize the 8 tasks and their corresponding data sets.

fMRI: The fMRI data set was collected using the functional magnetic resonance imaging technique (fMRI) at CMU [122]. Nine subjects were presented with 60 different words and were asked to think about each word for several seconds while their neuron activities were recorded. For each subject, there were 360 (60×6) fMRI images taken for the 60 words with 6 replications. Each voxel (volume-element) in an image recorded a neuron's activity in the brain. There were over 20,000 different voxels in a typical fMRI image. Following the same procedure in [111], we averaged the fMRI images for each word and hence had 60 images for the 60 words.

[1]GBM, glioblastoma; AO, Anaplastic Oligodendroglioma

Each word in the data set is encoded as a feature vector with co-occurrence statistics from the Google Trillion words. We used 5,000 features ($p = 5000$) for each word and constructed the feature graph from the co-occurrence statistics [145] with the threshold 0.25.

6.4.2 Experiment Protocol

Below we present our approaches for model construction and model evaluation.

Model Construction. For the Microarray data, we created training and testing data using the standard 5-fold cross-validation (CV). We performed another 5-fold CV on the training data set to select the regularization parameters. Once those parameters were selected, we generated a model from the entire training set with the selected parameters and applied the model to the testing data set for prediction. All the regularization parameters are tuned using grid search in the range of 2^8 to 2^{-8} with the power decreased by -1.

For the fMRI data set, we used the exact experimental protocols described in [111] for generating training and testing samples. The only difference was that we performed feature selection and model construction simultaneously rather than selecting features first then building a regression model with the selected features via the ridge regression. Specifically, the leave-two-out-cross-validation was performed. For the training set with 58 words, we randomly divided them into two subsets with 80% and 20% and tuned regularization parameters on the two subsets. We also selected top 500 stable voxels or top 250 stable plus 250 unstable voxels based on the stability score[2] defined in [122] to verify the importance of incorporation of structured input and task relationship. Similarly, the regularization parameters are tuned in the range of 2^8 to 2^{-8} for regularized MTL algorithms.

In applying single task learning algorithm to the two multi-task data sets, we separately apply SVM to each Microarray cancer classification task (8 SVM models in total) and ϵ-SVR to each voxel activity prediction task of fMRI data (500 models in total). We use libsvm [24] with linear kernel for both SVM and SVR and tune C from 2^8 to 2^{-8} and ϵ from 2^1 to 2^{-6}.

[2]Stability score measure the variation of voxel activity across the 58 training stimuli.

Table 6.3: Average accuracy for 8 tasks. Bold text denotes the best performance and $*$ means the method statistically better than the rest.

	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8
SVM	0.91 ±0.05	0.90±0.11	0.98±0.02	0.93±0.12	0.67±0.20	0.76±0.21	0.72±0.04	0.90±0.05
MTLasso	0.94±0.08	0.93±0.09	0.99±0.03	0.98±0.07	0.83±0.09	0.57±0.18	0.79±0.04	0.83±0.07
MTLap	0.96±0.05	0.94±0.10	0.99±0.02	0.99±0.07	0.85±0.11	0.61±0.19	0.80±0.07	0.84±0.06
MTLapTR	**0.98±0.05**	**0.95±0.11**	**1.00±0.00**	0.99±0.05	**0.90±0.10**	**0.78±0.19**	**0.88±0.06**	**0.91±0.09**
MTL1TR	0.95±0.08	0.94±0.11	**1.00±0.00**	0.99±0.05	0.87±0.11	0.66±0.19	0.80±0.06	0.87±0.09
MTLPTR	0.93±0.09	0.93±0.12	0.98±0.04	**1.00±0.00**	0.85±0.08	0.64±0.17	0.81±0.07	0.86±0.10
MTLTR	0.92±0.06	0.92±0.10	0.99±0.03	0.97±0.09	0.82±0.16	0.62±0.18	0.76±0.17	0.82±0.10

Model Evaluation. For the Microarray data, we obtained binary prediction based on the sign of the outcome from the test data. We collected accuracy ((TP+TN)/S) of the trained model, where TP stands for true positive, TN stands for true negative, and S stands for the total number of samples. All the accuracy reported was collected from the testing data set only and were averaged across 5-fold CV with 10 replicates.

For the fMRI data set, we followed the same procedure in [111, 122] to derive accuracy for each test set: (1) Predict the neuron response of the 500 selected voxels for the two testing words; (2) Compute the cosine similarity of each prediction with each of the held out images; (3) Based on the combined similarity scores, choose which prediction goes with each held out image; (4) Test if the joint labeling was correct, which leads to an output of 0 or 1; (5) repeat the process several times and compute the ratio of correlated labeling over the number of all trials. We repeated these above steps 400 times.

6.4.3 Experiment Results

In this section, we report the experiment results in term of accuracy. Bellow, we first report the average classification accuracy with standard deviation for Microarray data set.

6.4.3.1 Microarray Results

In Table 6.3, we provide the average classification accuracy of the multi-task learning algorithms and single task SVM for the 8 cancer types. We compared MTL algorithms with

single task learning algorithm SVM. As shown in Table 6.3, MTL algorithms outperform SVM in 6 out of 8 tasks, which proves the power of Multi-task learning v.s. single task learning, especially when each task has limited training samples.

Among these MTL algorithms, we also observe that the method MTLTR without feature selection performs worse than the rest 5 MTL methods with feature selection, which demonstrates the importance of feature selection in MTL for high dimensional data sets. Second, we find that the three methods (MTLapTR, MTL1TR and MTLPTR) with task relationship modeling performs slightly better than those without (MTLasso and MTLap). Finally, among the three methods with task relationship modeling, our approach MTLapTR with structured input incorporation performs better than the other two methods in 7 out of the 8 tasks though the difference of performance may not be large. Among these 7 tasks, there are 5 tasks with statistical significance ($\alpha = 5\%$).

Task Relationship Study In this section, we evaluate the task relationship modeling performance compared with MTLTR [189] and MTLPTR [191]. We averaged the learned task relationship covariance matrix Ω from 50 experiments for each method and created a 3D embedding of Ω via the Ndaona package [105]. In Figure 6.2, we show the 3D plot for each method. From the figure, we observe that there are approximate 4 groups for our method MTLapTR in total: (I) T_1 and T_3; (II) T_6 and T_7; (III) T_2, T_5 and T_8; (IV) T_4. The embedding is consistent with our knowledge, for example: T_1 is Breast cancer prediction and T_3 is ovary cancer prediction. T_1 and T_3 are close to each other. Similarly, T_2, T_5 and T_8 are close to each other since they are related with uropoietic system.

To the contrary, MTLPTR (joint feature selection and task relationship learning) groups T_1, T_2, T_3 and T_5 together but T_8 is far away from T_2 and T_5, which does not make sense since prostate cancer is closely related with kidney and bladder. MTLTR (task relationship learning) only groups T_1, T_3 and T_2, T_5 together, but leaving T_6, T_7 separated. The low dimensional embedding of task relationship demonstrates the effectiveness of our approach

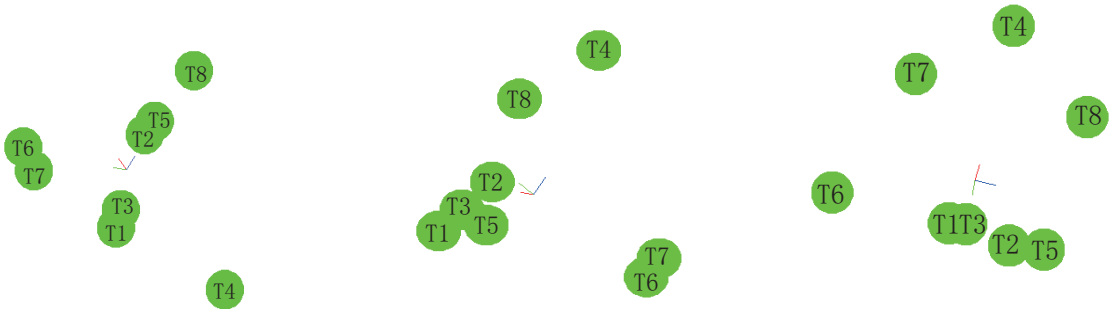


Figure 6.2: Task Relationship embedding for 3 methods in 3D space from Ndaona. Left: Our method MTLapTR; Middle: MTLPTR; Right: MTLTR

in terms of task relationship modeling.

We want to mention that the task (cancer) relationship is empirically learnt from data hence the relationship may not be all correct. For example according to the "factsheet" of NIH [131], breast cancer can easily be spread to liver site resulting in Metastatic Cancer. However, none of the three methods capture the relationship. The possible reason is that we do not have any patients with Metastatic cancer that spread from breast to liver.

Feature Selection Performance One important aspect for measuring the quality of MTL models is the sparsity of the learned models. To evaluate the feature selection performance of our MTL approach, we singled out the selected features for each task in 50 experiments (5 fold CV with 10 replications) with frequency at least 20 times. We also collected the number of pathways for each task, in which these selected features occur. In Table 6.4, we summarize the number of features and pathways for each task.

Table 6.4: Number of selected features and pathways per task. #F: number of features; #P: number of pathways.

	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8
#F	2305	1207	2134	1056	887	959	1084	1624
#P	153	148	152	146	118	132	140	138

For comparison, we also collect all the features that appear at least 20 times in MTLPTR

models in the 50 trials. We obtain 4301 features that belong to 196 pathways [3]. Clearly our approach builds much sparser models. We believe that the the block regularization utilized in MTLPTR is the root cause for leading MTLPTR to select more features and pathways. Using a block regularization, once a feature is selected by one task, the block regularization enforces all the other tasks to select the same feature. With heterogeneous tasks, different tasks favor different features and hence there are a large number of selected features.

As discussed before, our approach enables each task to select features that are specific to the task. Different tasks are encouraged to select the same set of features if they are similar to each other with the trace regularization term on the feature graph. To demonstrate the power of the trace regularization term, we calculate the common features among selected features for each task. We find that similar tasks share a large portion of features, but dissimilar tasks may have different choices. For example, T_1 and T_3 are closely related. They share 563 features. T_1 and T_4 are quite different and they only share 152 features.

Structured Input Inference One possible extension of our method is to infer the structured input given the task relationship. For that purpose, we calculated the average task covariance matrix over 50 trials from the Microarray data and use that matrix to learn the graph Laplacian L for the input data. We calculated the average L over 5 fold CV, dropped positive off-diagonal entries, and enforced the negative entries with absolute value bellow 10^{-6} to 0. From the learned Laplacian, we extract the recovered graph on features. Since it is difficult to visualize a giant graph, we show our result on a subgraph with 13 nodes [4] from pathway hsa04070 (Phosphatidylinositol signaling system). We chose this pathway since it is highly cancer related [74] and is relative small comparing with common pathways such as p53. The learned pathway topology as well as that from KEGG is shown in figure 6.3.

We observe that our approach recovered 21 out of total 25 edges, and produced 8 additional edges. Our pathway topology is more balanced while the KEGG pathway has more

[3] MTLPTR selects the same feature set for all tasks due to the block regularization, hence a single number is reported.

[4] The 13 genes' KEGG IDs: 534, 535, 536, 1067, 1068, 2211, 2224, 2229, 2284, 3262, 4978, 5068, 7190.

edges among the top 6 nodes (node 1-3, 11-13) than the bottom ones. This observation may be an artifact due to the trace regularization that we used (which encourage a balanced topology) or the additional 8 edges have biological meaning. Investigating the domain relevance of the input structure inference is one of our future research directions.

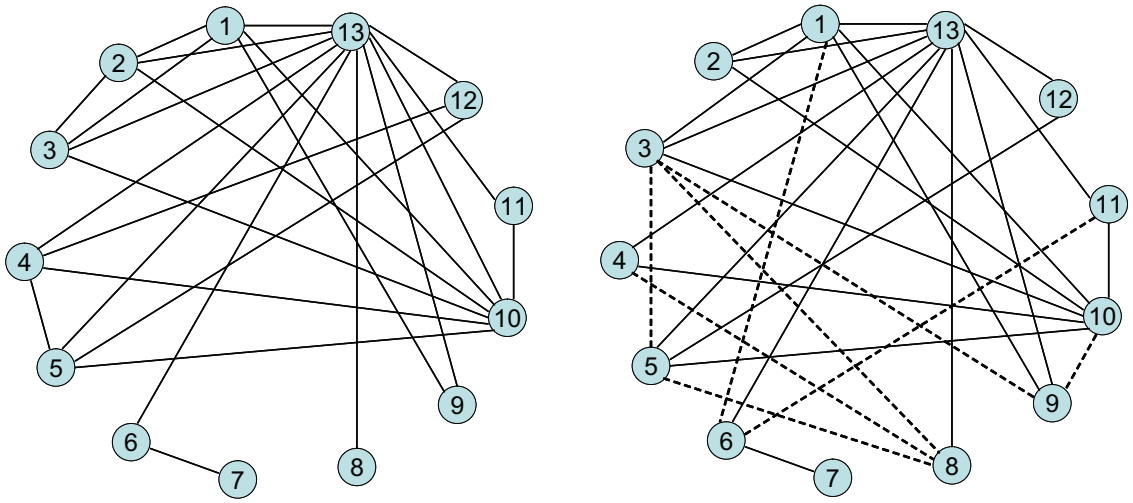


Figure 6.3: Comparing KEGG pathway (left) and learned pathway (right) for the Phosphatidylinositol signaling pathway. Solid lines represent edges from KEGG and dashed lines represents additional edges learned from our algorithm.

6.4.3.2 fMRI Results

In the experiment study on Microarray data, we have demonstrated the importance of incorporating structured input and task relationship inference in MTL. Here we report the experimental study of the same methodology to a totally different application domain: that of fMRI data analysis.

In Table 6.5, we report the prediction accuracy on 9 subjects with 500 homogenous tasks, in which the top 500 stable voxels extracted from training data for each trial were used. In the table, we first observe that the performance of MTLTR is worse than the other 5 methods even comparable with single task SVR, which confirms that feature selection is important to boost the MTL performance for high dimension data. Among the methods with feature selection, MTLap and MTLapTR work slightly better than the other 3 methods, which

Table 6.5: Prediction accuracy for 9 FMRI Participants with 500 homogenous tasks.

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
SVR	0.760	0.720	0.728	0.723	0.627	0.497	0.576	0.510	0.700
MTLasso	0.790	0.733	0.717	0.797	0.657	0.493	0.603	**0.527**	0.723
MTLap	0.818	**0.757**	0.760	0.817	**0.720**	0.502	0.590	0.447	0.743
MTLapTR	**0.830**	0.753	**0.767**	**0.820**	0.680	**0.577**	**0.606**	0.460	**0.763**
MTLL1TR	0.820	0.719	0.695	0.807	0.677	0.447	0.560	0.353	0.743
MTLPTR	0.815	0.740	0.719	0.813	0.594	0.238	0.538	0.297	0.667
MTLTR	0.745	0.667	0.677	0.750	0.659	0.518	0.529	0.494	0.700

Table 6.6: Prediction accuracy for 9 FMRI Participants with 250 homogenous tasks and 250 heterogenous tasks.

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
SVR	0.740	0.720	0.682	0.703	0.610	0.477	0.591	**0.526**	0.707
MTLasso	0.613	0.660	0.627	0.703	0.600	0.443	0.517	0.447	0.510
MTLap	0.790	0.670	0.677	0.803	0.660	0.473	0.656	0.287	0.660
MTLapTR	**0.803**	**0.762**	**0.733**	**0.843**	**0.703**	**0.527**	**0.672**	0.417	**0.757**
MTL1TR	**0.803**	0.723	0.719	0.813	0.687	**0.527**	0.640	0.420	0.723
MTLPTR	0.780	0.700	0.643	0.797	0.620	0.223	0.609	0.248	0.610
MTLTR	0.723	0.700	0.646	0.781	0.688	0.506	0.567	0.498	0.653

demonstrates the power of incorporation of structured input. Finally, we do not observe an obvious advantage of task relationship inference in this study since the tasks are homogenous.

We then designed a new experiments where we introduced some heterogeneous tasks. In the new experiment, we mixed 250 stable voxels with 250 unstable voxels. The importance of task relationship modeling in this study becomes clearer. From Table 6.6, we first observe that the accuracy of MTLasso and MTLap is decreased by around 5%-10% , but for the other methods that employ task relationship modeling, the accuracy remains pretty stable. In addition, MTLTR performs better than MTLasso and MTLap, which have no task relationship modeling. These observations confirm our hypothsis regarding the importance of task relationship modeling for heterogenous MTL. Among the three methods with task relationship modeling and feature selection, our approach MTLapTR with structured input incorporation performs better than the other two methods in 8 out of the 9 tasks though

the difference may not be large (as observed in the Microarray data set).

Another information we obtain from Table 6.6 is that when tasks are not uniformly related or there are outlier tasks, uniform task relationship assumption can lead to even worse results than single task learning. We can compare SVR with MTLasso, MTLap and MTLTR to draw the conclusion.

6.4.4 Discussion

An interesting question in MTL is how to choose appropriate regularization (sparse l_1, block sparse l_1/l_q and et al.) to fit the underlying structure of data without prior knowledge. Recent research has focused on the use of l_1/l_q norm block-regularization with $q > 1$ for block-sparse structured problems. However, due to the existence of non-uniformly related tasks, it is impractical that all the MTL problems share the same block sparsity.

Recently, Jalali et al. proposed a method so called "dirty MTL model", which employed l_1 regularization and l_1/l_∞ block regularization on task parameters [80] to fit the "dirty data", which may not fall into a single structural bracket (all block-sparse, or all low-rank and so on). The block regularization selects a subset of features across all the tasks, and then l_1 penalty enforces a few entries to be 0. Although appealing, this approach actually did not reveal how tasks related and how similar the solutions are for similar tasks. In order to build a more accurate and interpretable MTL model, it is necessary to model task relationship and use task relationship to guide feature selection.

In our model, we use l_1 regularization to penalize each task individually but employ structured input and task relationship regularization to guide feature selection. On the task level, each task selects features specific to itself and ones common to similar tasks; on the feature level, by incorporating the structured input information into MTL, the selected features are tended to be connected on the feature graph and exhibit a grouping effect, which has been demonstrated in our previous work [44] and the experimental study on Microarray data.

6.5 Conclusions

Multi-task Learning (MTL) aims to enhance the generalization performance of supervised classification or regression models by learning multiple related tasks simultaneously. A key factor to ensure the success of MTL in the presence of high dimensional data with heterogeneous tasks is an efficient feature selection procedure.

In this paper, we present a linear multi-task learning formalization for learning sparse features and task relationship from multiple heterogenous tasks. In our algorithm, we utilize a task covariance matrix related to task parameters to model the task relationship and learn the matrix from data. Meanwhile, motivated from the data with structured input such as Microarray where genes are features and genes form biological pathways, we propose a regularization formulation for incorporating the structured input on features into MTL. We have designed an efficient iterative optimization algorithm to solve task parameters and task relationship matrix based on accelerated first order gradient method in conjunction with projected gradient scheme. We have evaluated our approach from two real-world data sets and the experimental results demonstrate the effectiveness of the proposed learning methods.

Chapter 7

Leveraging Structural Information from Mobile Device Data for Meaningful Location Detection

We have discussed a few models in learning from the data with structured input and output from Chapter 3 to Chapter 6. In particular, we have demonstrated how to utilize the structure information from data to build more accurate and interpretable models. The data sets we have focused on are typically "wide" data, a.k.a. high dimensionality and low sample size from single source and multiple sources. In this Chapter, we will taste the flavor of "tall" data with high sample size but low feature dimensionality. More specifically, the data analyzed in this chapter is from telecommunication, e.g. call detail record data, GPS data from smart phone. There are potentially millions of mobile subscribers and the only available features are geographical location coordinates (latitude, longitude). We will demonstrate that utilizing the "structural" information (spatial relationship) of the data sets enables us to find more accurate meaningful locations and origin-destination information from people's daily life.

7.1 Introduction

With the highly advanced telecommunication technology, mobile devices such as GPS and mobile phones have been adopted faster than any other technology [7]. The number mobile phone subscribers has climbed from a few million worldwide in 1999 to more than 6 billion today [1]. Recently, there is growing interest of utilizing Call Detailed Records (CDR) to sense the locations of large populations and model the city dynamics among networking and urban computing communities [7, 8, 78, 134, 166]. In particular, the study of identifying "meaningful locations", a.k.a. a few key places where people spend a significant amount of time, is one of the most fundamental problems. The knowledge of meaningful locations from people's lives is essential for a number of tasks. For instance, to understand the traffic flow, it is important to discover the meaningful locations of individual people first and then to summarize how many people stay at each zone and how many people travel from one zone to the other [20]. Other applications of meaningful location discovery can be found in mobile advertising [137], social security [165] and social event detection [166].

Traditional ways to collect the meaningful location information are usually through census, GPS summarization, or the combination of these two. But census and GPS summarization have very limited coverage. Moreover, urbanization is fast in Modern cities, e.g. Beijing or New York, in which people move in and out frequently. It is impossible to perform census every Quarter of each year due to the high cost of time and endeavor.

To automatically identify meaningful locations, several computational methods on CDR and GPS data have been proposed. Since the technology for GPS data is more mature, we list a few important works for CDR data and cover the algorithms for GPS data in related work. For CDR data, there are both unsupervised and supervised algorithms developed., though there are less supervised methods compared with unsupervised ones. In supervised case, Isaacman et al. [78] recently proposed a method that combines clustering and logistic regression. Through volunteer's data, a clustering procedure is performed on spatial locations

[1] `http://www.cnn.com/2012/05/09/opinion/sachs-global-childrens-health/index.html`

of cell towers and a set of features, such as number of days, call frequency and night time call v.s. day time call frequency ratio are extracted from each cluster. Then a logistic regression model is trained on the clusters and applied to the clusters from testing data to predict whether a cluster is important or not.

In unsupervised case, Phithakkitnukoon et al. [135] divided the local region into 500 meter by 500 meter square grid cells (zones), then counted the frequency the subject occupy at each zone. Those zones with a certain days above a certain threshold are determined as meaningful locations, i.e. home and work locations are estimated as the zones in which the subjects occupy most frequently during the night and day hours. Calabrese et al. [20] first aggregated trajectory points into several small areas with a certain radius within which the subject stays, then segmented CDR trajectories into several trips based on the temporal domain information, i.e. the time interval between two successive points is more than a certain threshold e.g. 10 minutes. The resulting trips contain a set of origin and destination points, which are the meaningful locations. Furthermore, a home/work detection is also proposed in [20] by tower clustering and number of call days counting.

However, there are a few limitations of existing works in detecting meaningful locations from CDR data: 1) current unsupervised studies [20, 135] are typically based on frequency counting and largely focus on long term data, i.e. people have several months' record. When only short term data is available (i.e. one week), it is unreliable to claim a place as a meaningful location by frequency. 2) CDR data is normally unlabeled, namely no ground truth about where people are and whether the places are important. The supervised method proposed in [78] used 17 volunteers'data to train a model and apply to millions of people. Since the traveling pattern is diverse from people and regions, it is hard to generalize a model from those people in a small town to the people in metropolitan area, e.g. Lawrence, KS vs NYC. 3) For CDR data, a major problem existing in cellular network connectivity is that a cell phone may hop between multiple towers even when the subscriber is not moving. When applying the methods in [20, 135] to short term data, tower hopping may incur large

detection errors. 4) possible data record type information is ignored. For example, when mobile subscribers commute from one LAC zone to another zone, a location update (LU) event type is triggered at the boundary. Such additional information reflects whether people are moving and should be incorporated into learning, especially for short term CDR data.

In this paper, we propose a unsupervised learning algorithm for meaningful location identification for CDR and GPS data within a short period. In particular, we the contributions of this paper are as follows:

- We design and implement a framework for discovering meaningful locations from CDR data and GPS data with low sample rate. Our method addresses the tower hopping problem by a spatial clustering procedure on geo-locations of cell towers. We found tower hopping always happen among spatially adjacent towers. Hence we can eliminate tower hopping by leveraging structural information of CDR data in spatial domain.

- Instead of using call frequency to measure the importance of a cluster of towers or a zone, we utilize the criteria of "Duration of Stay" (DoS), which denotes how long a user dwells in a cluster/zone in temporal domain. We propose an algorithm to calculate DoS for each cluster under two scenarios: 1) record data type is unavailable; 2) record data type is available. When the data type is unavailable, we provide an approximate estimation. When the type is available, we leverage different data types, including RTT, SMS, heartbeat (HB), location update (LU) and handover (HO) to obtain more closed estimation.

- We derive origin and destination matrices among meaningful locations, which accounts for the meaningful trips. From the public agencies' point of view, reporting origin-destination matrix (OD) among zones is critical for planning public transit and urban development. More specifically categorized OD based on purposes of trips can give rich information for designing/providing transportation services to different groups of residents.

- We have validated our framework by three real-world data sets, including two CDR data and one GPS data with low sample rate. We identified home and work locations, all other meaningful locations for each data set and home/work OD commute distance. By comparing with [20, 135], we have demonstrated the utility of our framework.

7.2 Related Work

There are two families of methods closely related to ours. One line is a body of work that determines meaningful locations based on GPS or wifi beacon trace. The other line is based on CDR data.

For GPS/wifi data, Kim et al. [93] developed a method called "PlaceSense" to detect semantically meaningful places from Pervasive RF-Beacons by detecting place entrance and departure. Cao et al. [22] designed a location ranking scheme via random walk over the graph that captures the relationship among locations and user-locations. Zheng [194] et al. combined clustering and user's travel experience to derive meaningful locations. Although accurate, these words require much finer granularity and a longer time period (e.g. several months) than CDR data. Moreover, the GPS data investigated in the works [22, 93, 194] has limited coverage within 100 users. In contrast, we focus on CDR data with vast coverage and coarser granularity on millions of users collected within a short time period.

For CDR data, Phithakkitnukoon et al. [135] identified meaningful locations by dividing map into regions and counting record frequencies. Calabrese et al. [20] discovered meaningful locations by trip segmentation and "number of call days" counting. But these methods also depend on long term data since Issacman et al. [78] developed a supervised method by combing cell tower clustering and logistic regression to predict whether a cluster is important or not. But in real world CDR data, the data is typically unlabeled and it is hard to generalize the model trained from a small town to big cities.

Besides meaningful location detection, there are a few works attempting to discover

human mobility pattern. For example, Becker et al. [9] studied individual mobility patterns and aggregate them into a summary of city dynamics for a city in NJ. Bayir et al. [7] extracted most popular trips from MIT Reality Mining data set [37]. Since our focus is on meaningful location detection and origin-destination analysis, we do not cover all related work on mobility pattern mining.

Though there are a few works proposed for discovering meaningful location from CDR data, ours is the first to focus on short term CDR data and utilize different event data types if they are possible. Besides, we proposed more elegant duration of stay estimation scheme to measure the dwelling time at a certain region.

7.3 Background

In this section, we cover background knowledge in the telecommunication domain.

7.3.1 Call Detail Record

The Call detail record (CDR) data analyzed in this paper consists of anonymous location measurements generated each time a device connects to the cellular network, including call placing/receiving, message placing receiving and internet 2G/3G connection. Unlike previous studies [7, 78, 134, 166], the data was collected from cell tower rather than end user. All cell phone numbers were anonymized and hashed to a unique ID, which was kept for at most one week or 10 days. In other words, the CDR record for each user can have at most 10 days' data. We collect data from all towers's data in a certain area, extract each user's records and sort them by timestamp.

More formally, we provide definition of Call Detail Record (CDR) and CDR trajectory bellow.

Definition 7.3.1. *Call Detail Record (CDR): A CDR $\mathcal{CD}$ is a four-tuple $< uid, time, lat, lon >$, where uid is the user id, time is the time stamp, lat, lon are the latitude and longitude of the*

cell tower that serves this call.

Definition 7.3.2. *CDR trajectory: A CDR trajectory $\mathcal{TR}$ of a user is a series of CDR $\mathcal{CD}$ for the user that are ordered by the time stamp of the records as $\mathcal{TR} = \mathcal{CD}_1 \rightarrow \cdots \mathcal{CD}_i \rightarrow \cdots \mathcal{CD}_n$.*

7.3.2 Record Data Type

On a few occasions, we might have more information about the CDR data, such as the data types, which are typically obtainted from carriers. In this paper, we have the following data types available from one data set:

- RTT: placing or receiving a call
- SMS: sending or receiving a message
- Handover/handoff (HO): transferring an ongoing call or data session from one cell tower to another tower. HO typically happens when the user is moving from one tower's coverage to that of another one and holding an ongoing call.
- Location Update (LU): moving from one location area to another area, where a "location area" is group of cell towers that serve for a particular area. LU typically happens when the user is moving from one location area to another one.
- SGSN: 2G/3G internet connection

Among these event types, HO and LU are indicating the user is moving and we call them "moving data type". For the rest types, it is difficult to judge.

7.3.3 Tower Hopping

Tower hopping is referred to the phenomenon that a user may be assigned to a number of spatially nearby cell towers even when the user is not moving. It is commonly seen in cellular

network due to the load balancing factor. To illustrate what is tower hopping, we show an example: In Figure 7.1, the user has two events on two towers simultaneously at 11:04, then

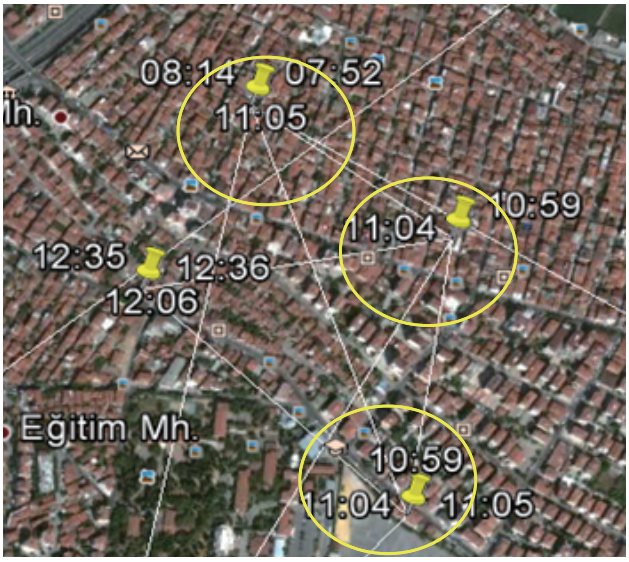


Figure 7.1: Demonstration of tower hopping from a user's CDR trace in Singapore. Each pinpoint is a cell tower with a set of events that happened. Yellow circles highlight tower hopping among three towers.

has another event on a third tower at 11:05. It is impossible for a human to make such transitions in one minute, hence the user is assigned to other towers when he/she is static.

7.3.4 Duration of Stay

Duration of stay (DoS) is the length of stay at a particular area, which could be represented as a rectangular zone or the location that a set of cell towers cover. DoS could be aggregated at daily/weekly basis or a short time within which the user continuously stayed in the same region.

7.4 Methodology

Given call detail record of mobile subscribers within a short period (i.e. one week), our goal is to identify their meaningful locations, such as home and work. Our assumption is that the

locations where people spend a significant of time are meaningful locations. The proposed method is a unsupervised approach based on spatial clustering on locations of cell towers and duration of stay (DoS) estimation on CDR events in temporal domain. If record types are available, we further improve the reliability of DoS estimation by utilizing the properties of different data types. Our current analysis is per-user based and we have not considered the relationship among different users. Without statement otherwise, the terms e.g. "cell towers", "CDR Trajectory" are collected from a particular user.

Our method (clusterDos) derives meaningful locations from CDR data based on 4 major steps: 1) spatial clustering, 2) duration of stay calculation, 3) cluster-zone map generation and 4) meaningful location generation.

7.4.1 Spatial Clustering on Cell Tower Locations

As discussed previously, tower hopping always exists in cellular networks. We observed from real-world data that tower hopping always happens among spatially nearby cell towers, hence we can leverage the spatially structural information to eliminate tower hopping. In particular, we utilize a spatial clustering procedure to group spatially nearby towers together. Typical spatial clustering algorithms that can be used here are Density Spatial Clustering (DBSCAN) [39] or Ordering Points To Identify the Clustering Structure (OPTICS) [3].

In this paper, we use DBSCAN to cluster cell towers at a daily basis, which consists of the unique cell towers associated with one day's CDR trajectory. DBSCAN finds a number of clusters starting from the density of corresponding points hence does not require one to specify the number of clusters, as opposed to k-means. DBSCAN has two parameters: neighborhood radius ϵ and minimum number of points required to form a cluster $minPts$.

The basic procedures of DBSCAN are cluster generation and expansion. It starts with an arbitrary starting point that has not been visited. The point's ϵ-neighborhood is queried. If the number of points it contains is larger or equal than $minPts$, a cluster is created and the point is labeled as the core point. Otherwise, the point is labeled as noise. For

each point within the cluster, its ϵ-neighborhood is retrieved and added to the cluster. If the neighborhood point contains sufficient points ($>= minPts$), the neighborhood point's neighbors are added to cluster as well. This process is repeated until the density-connected cluster is completely found. Then, a new unvisited point is retrieved and processed, leading to the discovery of a further cluster or noise.

However, it is nontrivial to adopt DBSCAN to CDR data. On one hand, DBSCAN singles out noise points, which are isolated points that do not belong to any dense part of existing cluster points. In cellular network, it is possible to have isolated cell towers in remote area, hence they shouldn't be excluded. On the other hand, it is tricky to choose the parameter $minPts$ for CDR data. If $minPts >= 2$, then the point with only one neighbor within ϵ is denoted as noise. If $minPts = 1$, it is possible that all the points along a line with adjacent distance less than ϵ are clustered in one cluster.

In our implementation, we modified the basic DBSCAN algorithm to resolve the two problems. First, each "noise" point is output as a cluster with only one point. Second, we set $minPts = 1$ and allow DBSCAN to expand the cluster up to two hops away from the core point. Finally, we sort the cell towers in descending order based on number of days that the tower is used and the core points are selected from the list rather than randomly picking up any points. Such a procedure is proved to be helpful in [78].

Note that we consider two scenarios: data types are available or unavailable. If record data types are available, we cluster the cell towers having other types excluding LU, HO since it is unlikely for the locations of "moving data type" to be home, work or shopping centers. If no data type information available, we simply cluster on all cell towers the user has connected.

7.4.2 Duration of Stay Calculation

After spatially clustering, the user's trace can be summarized on a few clusters, in which each cluster contains a subset of towers associated several pieces of CDR trajectories. In

duration of stay (DoS) calculation, we calculate how long people stay at each cluster. Note that a user may stay at a location several times a day, e.g. early morning and late night at home, we calculate DoS piece by piece and aggregate them into daily DoS. Bellow we first consider the case when no data type information is available. Under this circumstance, all cell towers the user connects are clustered.

Suppose $< t_f, C_i >, < t_{f+1}, C_i >, \cdots, < t_l, C_i >, < t_{l+1}, C_{i+1} >$ is a subset of user's trace on cluster level, where $< t_f, C_i >, < t_l, C_i >$ represents the 1st and last record with time stamp t_f, t_l in the cluster C_i, and $< t_{l+1}, C_{i+1} >$ is the 1st record in cluster C_{i+1}. We aim to know how long the user spent in cluster C_i from t_f to t_{l+1}. A natural estimation of DoS [7] is the difference between the last event's time stamp and the first event's time stamp observed in the cluster C_i: $DoS_i = t_l - t_f$. But such calculation may underestimate the real duration of stay. There might be a long time between t_{l+1} and t_l resulting in an underestimate.

To overcome the drawback, we make compensation for the temporally adjacent clusters if there is a large gap in temporal domain. More specifically, we assign the DoS to cluster C_i and meanwhile update the start time for cluster C_{i+1} for future calculation DoS_{i+1}:

$$\begin{aligned} DoS_i &= t_l - t_f + \left[\frac{t_{l+1} - t_l - \Delta_{l,l+1}}{2}\right]_+ \\ t_{l+1} &= t_{l+1} - \left[\frac{t_{l+1} - t_l - \Delta_{l,l+1}}{2}\right]_+ \end{aligned} \tag{7.1}$$

where $[x]_+ = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$ and $\Delta_{l,l+1}$ is the approximated commute time from the centroid cluster C_i to that of C_{i+1}. We assume the user travels in a straight line in a constant speed e.g. $v = 30m/h$. The demonstration of this case is shown in the right of Figure 7.2.

Although we make an effort to compensate possible gaps between cluster C_i and C_{i+1}, we are still unclear when the user leaves from cluster C_i to C_{i+1}. It is unreliable to simply exclude commute time and divide the remains equally. Next, we show that if data type information is available, we can estimate DoS more accurately.

Recall that LU and HO data types can indicate movement of users, such information can be utilized. In this case, we only cluster the cell towers associated with the data types excluding LU/HO. If the user triggers a LU/HO event at t_{l+1} after the last event in cluster C_i and the LU/HO's cell tower does not belong to C_i, we can safely assign $t_{l+1} - t_f$ to C_i since the user is just leaving C_i. Bellow we give a similar DoS calculation rule for the left scenario in Figure 7.2:

$$\begin{aligned} DoS_i &= t_{l+1} - t_f \\ t_{l+2} &= t_{l+2} - \left[\frac{t_{2+1} - t_{l+1} - \Delta_{l+1,l+2}}{2}\right]_+ \end{aligned} \tag{7.2}$$

where $\Delta_{l+1,l+2}$ is the approximated commute time from LU/HO event tower to the centroid of C_{i+1}.

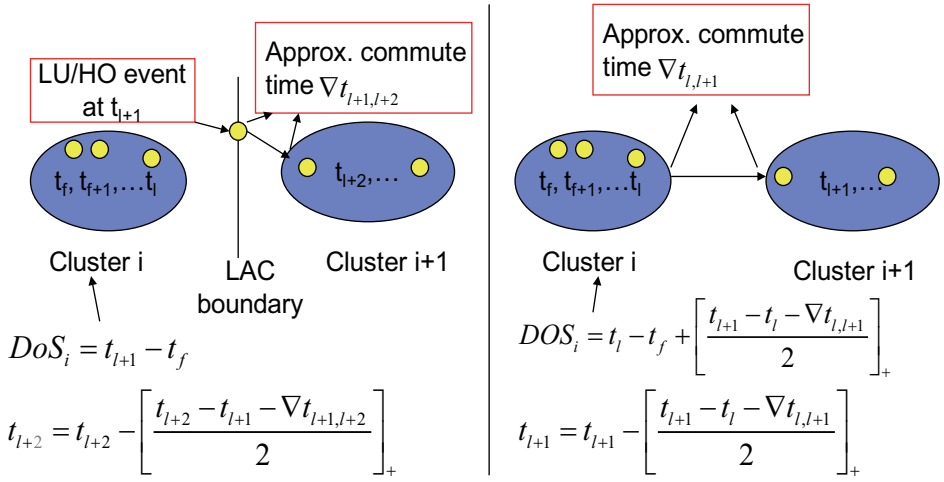


Figure 7.2: DoS calculation. Left: LU/HO record happens between two clusters. Right: No movement record.

After calculating several pieces of $DoS_i, i = 1, ..., k$ for one cluster in a day where k is the number of times the user shows up in the cluster, we can aggregate them into daily DoS.

7.4.3 Cluster-Zone map generation

The purpose of cluster-zone map generation is to make connections among different clusters that represent the same place. For example in Figure 7.3, we obtain two clusters around

"home" for two different days. The location of two centroids are different, but they represent the same home location, hence there is a demand to summarize spatially nearby clusters as one area. Towards that end, we divide the overall area into a set of zones, either in

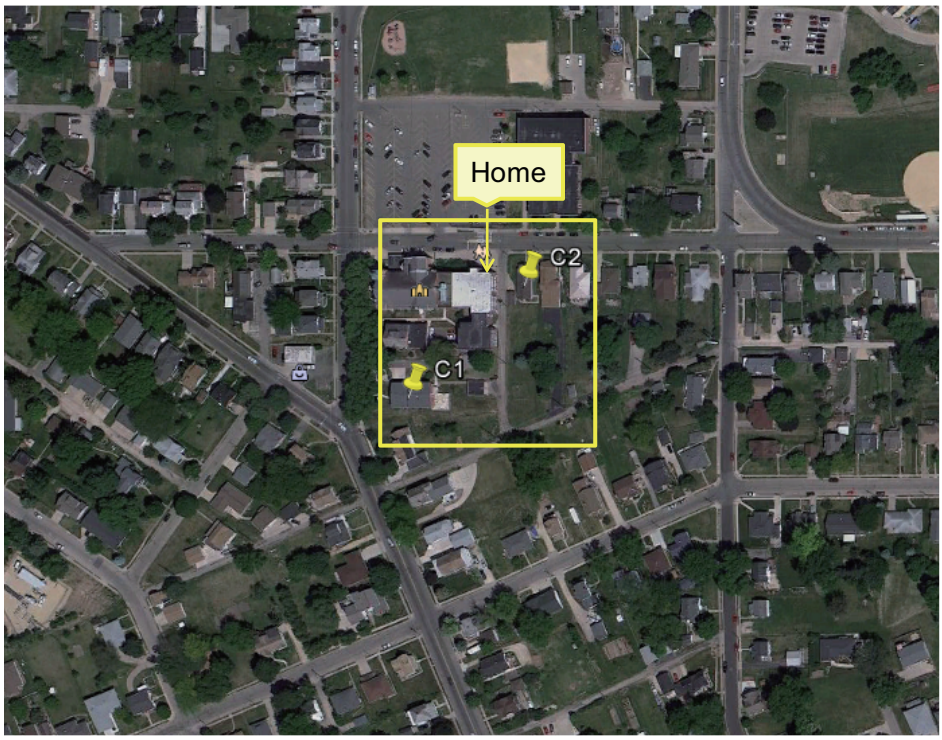


Figure 7.3: Rectangular zone example. Meaningful location is home as labeled. C_1, C_2 are cluster centroids from two days.

rectangular zones (e.g. 500m X 500m) or predefined irregular polygons. A zone is annotated with a unique ID and its coordinates are recorded as boundaries. For each cluster, we map its centroid to a zone. After this step, each zone is associated with duration of stay for one day.

7.4.4 Meaningful Location Generation

The we output these zones with duration of stay more than a certain threshold (e.g. 1 hour) at any day as meaningful locations. To further annotate these locations with semantic meaning, such as home, work and shopping center, we can combine common knowledge about home and work (maximum DoS in night time and day time in week days and repeating several

days) and points of interests within the area.

7.4.5 Home/work Detection

Among meaningful locations, home/work are the most important places of daily life. We proposed a home and work detection methods that identifies where people live and work respectively. The algorithms are similar to meaningful location detection but are independent and may select the same zone as both home and work.

We define home time from 9:00pm to 8:00am and work time as 10:00am to 6:00pm. Unlike the general meaningful location algorithm, we separately cluster the cell towers of home time and work time events. The purpose is to reduce the effect of day time events when analyzing home time data since it is normal to have more day time events than night time events. Once clustering is done, we derive the home/work zone having the following properties:

- the largest duration of stay at home time and work time on week days
- at least 2 days' call record with more than 1 hour DoS

7.5 Experiment

In this section, we evaluate our proposed method using four real-world data sets, including two call detail record data and two GPS data collected from smart phones. The two CDR data sets are collected from Istanbul, Turkey and Singapore respectively. For the Turkey data, we have data type information available. To demonstrate the utility of our method, we implemented and compared with home/work algorithm "zoneCount" [135] and "clusterCount" [20]. All the algorithms are implemented in Java.

7.5.1 Data sets

Unlabeled Smart phone/GPS data: The data is collected from Dubuque, Iowa in the period of October 2011-June 2012. There are 555 users carrying a smart phone equipped with GPS. Since people do not stay on the program for various reasons, we assemble one week's data by grouping daily events and selecting 7 days with maximum number of data points. The total number of events is 256,185 with 461 per person on average, therefore the average number of events is 65 for one day.

Labeled Smart phone/GPS data: The data is also collected from Dubuque, IA. There are additional 7 volunteers who are willing to share their daily trace between 05/10/12-05/19/12 and disclose their home and work locations. They also kept a diary of their stayed locations with departure and arrival time. These ground data enables us to quantify our algorithm's performance.

Turkey Cellular network data: The data is collected in Istanbul, Turkey from 02/12/2012 to 02/19/2012. There are 3353 cell towers and 46353 users with 24,732,562 events in total. For this data set, we have data type information available, including RTT, SMS, SGSN, LU and HO. The data is unlabeled since no users provide their real home/work location.

Singapore Cellular network data: This data is collected from 376 users in Singapore within 03/19/12-03/27/12, but not all users have complete record during the period. There are 75 people with only 1 day's record available. There are 2534 cell towers and 376 users with 152,844 events in total.

In Table 7.1, we summarize the characteristics of the four data sets.

7.5.2 Evaluation Criteria

Since three data sets have no ground truth, we focus on the evaluation on GPSLabel data from volunteers. For the rest three data sets, we show our detection results in plots and match them with domain knowledge, e.g. census map.

Table 7.1: Characteristics of the data set. $\#U$: total number of users, $\#T$: total number of cell, $\#E$: total number of events (records), Avg $\#E$: average number of events per user, Avg $\#T$: average number of towers per user used, Label: labeled or unlabeled, Data type: having data type information (Yes) or not (No)

Data	# U	# T	# E	Avg # E	Avg # T	Label	Data type
GPSNoLabel	555	NA	256,185	461	NA	Unlabeled	No
GPSLabel	7	NA	5365	766	NA	Labeled	No
TurkeyCDR	46353	3335	24,732,562	533	27	Unlabeled	Yes
SIGCDR	376	2534	152,844	406	31	Unlabeled	No

Model Construction Recall that we utilize the DBSCAN algorithm to cluster spatially nearby cell towers. At this step, we set the $minPts = 1$ and $\epsilon = 1$ mile for CDR data and $\epsilon = 0.5$ for GPS data. Emperical studies show that this configuration works well in practise. In our zone map generation step, we set the rectangular range as 1 mile $\times$ 1 mile for CDR data and 300 meter $\times$ 300 meter for GPS data. In meaningful location generation, we set the time threshold that determines the meaningful location to 30 minutes.

For the comparison method, we use the same parameter to determine zone size (1 mile for CDR, 300m for GPS), home time (9:00pm to 7am) and work time (10:00pm to 6:00pm) for home/work detection.

Model Evaluation We evaluated our home/work detection and meaningful location discovery algorithm by detection error for GPSLabel data. Given the ground meaningful location (e.g. home or work) coordinates $< x, y >$ and predicted location coordinates $< \hat{x}, \hat{y} >$, We define the detection error for one user's one location as:

$$err = dist(< x, y >, < \hat{x}, \hat{y} >) \tag{7.3}$$

where $dist(.,.)$ is the distance function that calculates Euclidean distance between two geographic coordinates based on [2].

In evaluation of home work algorithm, we only focus the location prediction error 7.4 in

[2] `http://www.movable-type.co.uk/scripts/latlong.html`

spatial domain. However, for meaningful location evaluation, we also consider duplication of meaningful locations in temporal domain, e.g. home and work, that appeared multiple times in a day. Our rule to generate meaningful location is based on duration of stay in a cluster. If the user has two series of stay record in a cluster with duration more than a certain threshold, we claim we find two meaningful locations in spatial and temporal domain, though they have the same geographical locations. For each user, we measure the following two metrics: detection rate (DR) and average detection error (avgErr) defined as:

$$\begin{aligned} DR &= \#\{detected\ meaningful\ locations\}/n \\ aveErr &= \textstyle\sum_i err_i/n \end{aligned} \tag{7.4}$$

where n is the total number of meaningful locations and err_i is the location error for ith meaningful location detection.

7.5.3 Home Work Detection Results

GPSLabel data results Since this data set is relatively small, we focus on quantifying how our home work detection works. For other type of analysis, such as home to work commute distance, we study them on the rest 3 data sets.

In Figure 7.4, we show the home/work prediction result for one volunteer. The blue pinpoint represents the predicted home location and the red one represents the work location. The yellow points represent the ground truth reported by the volunteer. There is one line connecting home and work locations representing the OD line. From Figure 7.4, we observe the predicted location is very close to the ground truth. To have a clearer idea how close they are, we show a zoom-in plot on the right panel. As shown in the Figure, the predictions are almost exactly in the same area. By measuring the geodesic distance between the prediction and truth as the prediction error, we find the prediction error is 0.08 miles for home and 0.06 for work.

For all the 7 volunteers, we show the comparison in Table 7.2. Our method achieved better

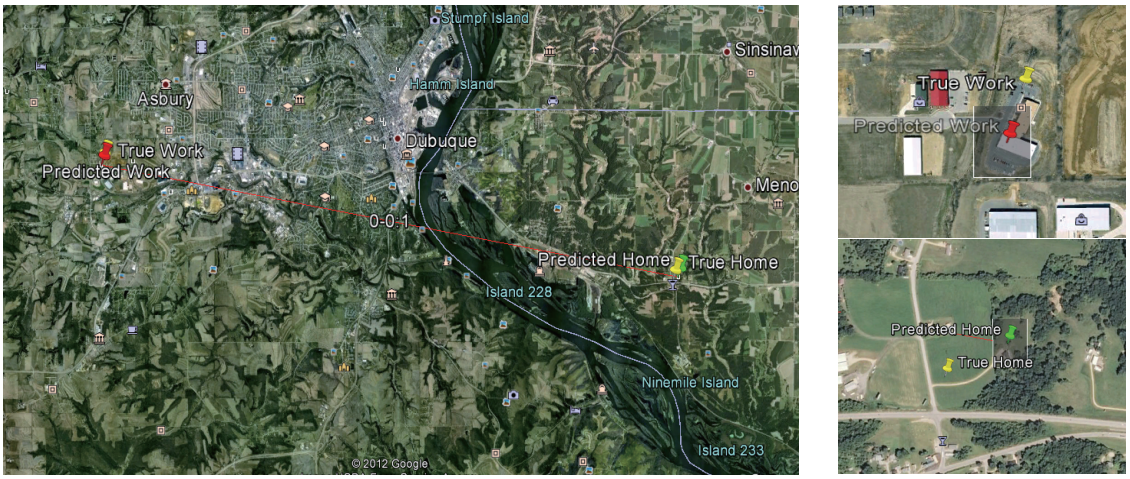


Figure 7.4: Home/work detection for volunteer 1. Left: overall plot for true home/work location and predicted location. The yellow pinpoints represent the ground truth. The blue and red represents the prediction. Right: zoomed in prediction vs ground truth.

performance for 5 out of 7 volunteers for home/work detection. The average home prediction error is 0.08 miles and work prediction error is 0.26 miles for our method clusterDos. For the competing baselines, zoneCount [135] has 3.2 miles error for home, 1.66 miles for work while clusterCount [20] has 1.16 for home, 6.27 for work.

Table 7.2: Home/work detection comparison. Error is in miles and the least error is highlighted in bold font for home and work separately. VID: volunteer ID, HError: Home Prediction Error, WError: Work Prediction Error.

	clusterDos		zoneCount		*clusterCount*	
VID	HError	WError	HError	WError	HError	WError
1	**0.08**	**0.06**	0.11	4.83	0.11	11.18
2	**0.03**	**0.54**	0.06	1.23	**0.03**	23.72
3	**0.09**	**0.07**	7.32	0.09	7.32	7.21
4	**0.03**	0.53	0.24	2.81	0.46	**0.06**
5	**0.01**	0.48	**0.01**	**0.35**	0.01	1.45
6	0.1	**0.07**	14.56	2.23	**0.04**	**0.07**
7	0.2	**0.07**	**0.06**	0.08	0.16	0.21

Home to work commute distance study Since the rest three data sets are unlabeled, it is impossible to compare our method with other two baselines. Instead, we run our home/work algorithm to detect home/work locations and study the home to work commute

distance since it is interest to urban planner and transportation research.

From the GPSNoLabel data set, our method clusterDos identified both home and work locations from 500 out of 550 people with the average commute distance 7.8 miles. The result is pretty close to the ground value of 7.1 miles for the GPSLabel data set from the same city, although 7 users cannot represent the whole population.

Beyond the 7 volunteers's ground data, we try to match our result with census/survey in US. According to the census of US Department of Transportation in 2003 [132], the average one way commute distance from home to work 15 miles all over the USA. Compared with census, our estimate is smaller but we argue that the result is reasonable. For one thing, our distance is given in Euclidean distance between origin and destination. In real life, people cannot travel in straight line hence the real travel distance should be greater than 7.8 miles. Actually there are 51% population in the survey with commute distance less than 10 miles, which is consistent with our findings. For another, the census was taken in both metropolitan and rural area hence the outcome is biased towards big cities due to the large population. It is hard to imagine the people of Dubuque, IA (a small town) travel 15 miles to work every day.

We performed similar analysis on the SIGCDR data. From the Singapore data, we detected both home and work locations from 274 out of 365 users. There are 75 users that are both "home less" and "workless" since they only have less than one day's record and fail to pass our 2 days' filter. The average commute distance for the 274 users is 4.03 miles, which is consistent with the claim (4.3 miles - 6.8 miles) in one Singapore resident's blog [3].

For the TurkeyCDR data, we identified both home and work locations from 33196 out of 46353 users with average home to work distance 4.12 miles (6.63km). Such an estimate is close to the finding in [121] that the distance of 5-6 km between home and work ranks highest in the Istanbul metropolitan.

[3] `http://www.mrbrown.com/blog/2008/07/mrbrowns-quick.html`

7.5.4 Meaningful Location Detection Result

In this section, we evaluate our algorithm for all meaningful locations include home and work. Likewise, we focus on the labeled data set GPSLabel and then summarize other characteristics about the meaningful locations on the other 3 data sets.

GPSLabel data results For GPSLabel data, we first check the user's diary to single out all locations with more than 30 minutes duration, then run our method to detect meaningful locations. From Table 7.3, we first observe that there is no big difference between the average

Table 7.3: Meaningful Location Detection Result. The best result of each method is highlighted by bold font. Notations: DR, detection rate; aveErr: average error among detected meaningful locations

	clusterDos		zoneCount		*clusterCount*	
VID	DR	aveErr	DR	aveErr	DR	aveErr
1	**0.88**	0.25	0.44	**0.20**	0.65	0.50
2	**0.69**	**0.22**	0.53	0.25	0.81	0.49
3	**0.85**	0.43	0.31	**0.26**	0.81	0.47
4	**1.00**	**0.35**	0.92	0.42	0.69	0.40
5	**1.00**	**0.10**	0.62	0.26	0.62	0.14
6	**0.92**	**0.26**	0.70	0.28	0.25	0.48
7	**1.00**	0.20	0.93	0.19	0.93	**0.10**

detection error. The reason is that if distance between the closed point from prediction and a ground truth location is greater than 1 mile, we claim a mismatch for the meaningful location. Among these detected locations, our method outperforms two baselines in 4 out of 7 users, though the difference is subtle. But for detection rate, our method is always better than zoneCount and clusterCount. The reason is that call frequency cannot tell the importance of a certain zone/cluster, especially for short term data.

Number of Meaningful Locations Study As mentioned before, without user-provided ground data for the rest three data sets, we do not know how many and where their meaningful locations are. Instead, we apply our algorithm to study the number of meaningful

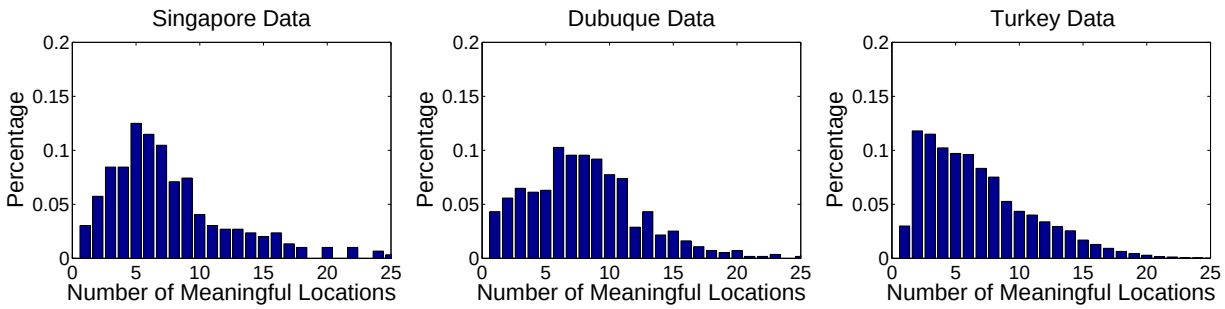


Figure 7.5: Bar chart of the number of meaningful locations vs population for three cities. Left: Singapore; Middle: Dubuque, IA; Right: Istanbul, Turkey.

locations for the residents of Istanbul, Turkey, Singapore and Dubuque, Iowa. Such an analysis allows us to study the mobility pattern across different countries.

In Figure 7.5, we plot the bar chart of the number of meaningful locations v.s. percentage of population. We have a few observations. First, around 50% people in Dubuque, IA and Singapore have between 3 to 8 meaningful locations and more than 50% people in Istanbul have between 2 to 6 meaningful locations. Our result from Dubuque data is consistent with the work in [78], in which they found majority of people in NYC and LA had 3 to 7 meaningful locations. Second, we have a heavy tail in the bar chart compared with the study in [78]. For example, around 1% users have more than 20 meaningful locations. we believe the reason is that our definition of "meaningful" is empirically determined by duration of stay without considering the semantical meaning. It is highly likely people get stuck in a traffic jam or whatever reasons stay at a certain place, which is not meaningful for them but identified by our algorithm. Finally, the mobility pattern of Turkish seems different from Americans since the peak points are at 2 to 4 for Turkish while 5-7 for Americans. A future study will be conducting survey about the users, e.g. percentage of housewife.

7.6 Conclusion

In this paper, we proposed meaningful location detection framework in which important places, e.g. home and work are identified from call detail record/GPS from mobile phones.

In our model, motivated by the fact that tower hopping happens among spatially nearby towers, we leveraged the spatial structure information of cell towers and designed a clustering method based on DBSCAN algorithm. We measured cluster's importance based on DoS (how long a user dwell in a cluster) and devised a method to calculate DoS. Based on experimental studies, we have demonstrated its utility on four real-world data sets.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

In this dissertation, we tackle the problem of learning from structured data with high dimensional structured input features and output tasks. With the high dimensional structured input space and/or structured output space, learning a low dimensional and consistent structured predictive function is important for both robustness and interpretability of the model.

We first presented a few machine learning models that learn from the data with structured input features and structured output tasks. For learning from the data with structured input features, we have developed structured sparse boosting for graph classification, structured joint sparse PCA for anomaly detection and localization. For learning from structured input, we investigated the interplay between structured input and output under the context of multi-task learning. In particular, we designed a multi-task learning algorithms that performs structured feature selection & task relationship Inference. We demonstrated the applications of these structured models on subgraph based graph classification, networked data stream anomaly detection/localization, multiple cancer type prediction, neuron activity prediction and social behavior prediction. Through extensive experimental studies, we demonstrated utility of our models across several application domains.

Besides the work on benchmarks, we also demonstrated how the "structure" information help solving industrial problems through my intern work at IBM Research. In particular, we proposed meaningful location detection framework in which important places, e.g. home and work are identified from call detail record/GPS from mobile phones. In our model, motivated by the fact that tower hopping happens among spatially nearby towers, we leveraged the spatial structure information of cell towers and designed a clustering method based on DBSCAN algorithm. We measured cluster's importance based on DoS (how long a user dwell in a cluster) and devised a method to calculate DoS. Based on experimental studies, we have demonstrated its utility on four real-world data sets.

8.2 Future Work

Overall this dissertation has only touched a small portion of structured data learning with structured input and output, and more generally learning with structured input features and output tasks. Below are some key directions of future work.

One key future direction is to investigate both high dimensional structured feature space and high dimensional structured output space. Our current work only handles a few hundred tasks with known/unknown task structures. However, in real-world applications, e.g. text categorization [64], gene function annotation [11] and location annotation in social network [183], the number of learning tasks could be very huge.

Recently, motivated by the potential high dimensional label/task space, researchers in started to investigate dimensionality reduction on label space [11, 70, 159] for multi-label learning. The major advantage of label space transformation is to reduce the problem size, i.e. transform k classification problems into m regression problems in the reduced label subspace [70, 159] where $k \gg m$. The common limitation of [70, 159] is that the label structure information is missing during encoding (label space reduction) and decoding (label prediction). Additionally, the regression analysis in the reduced label space still suffers from

the curse of dimensionality of potentially high dimensional feature space. The state-of-the-art algorithm for multi-label classification that utilizes label structure information and performs label space reduction is CSSA [11], which is again suffering from high dimensional feature space. Hence exploring a approach to cope with high dimensional structured feature and task/label space is beneficial.

Another key area of future work that is to accelerate our current optimization algorithms [44, 47, 45, 84], although most of them have achieved optimal convergence rate either globally or partially under single thread and single core platform. The bottleneck of our algorithm is objective function evaluation and gradient calculation at each step. However, both function evaluation and gradient calculation can be written in a certain "summation form", which allows them to be easily parallelized on multicore computers. As studied in [31], PCA and logistic regression could be significantly accelerated with the help of parallel computing. Therefore, how to adopt their insights into our structured PCA or multi-task logistic models is worthwhile to investigate.

Last but not the least, there is much future work possible for the case of mobile data (CDR) mining. Our current meaningful location detection scheme is purely based on duration of stay. It is highly likely that people get stuck in a traffic jam or whatever reasons stay at a certain place, which is not "meaningful" for users but identified by our algorithm. Hence one direction is to combine our algorithm with GIS information, e.g. the bus stop and shopping center distribution, to reduce false positives. Another direction is to infer trip purpose, e.g. home based work, home based shopping et al. More specifically categorized purposes of trips can give rich information for designing/providing transportation services to different groups of residents.

References

[1] E. Adar and L. A. Adamic. Tracking information epidemics in blogspace. In *Web Intelligence*, pages 207–214, 2005.

[2] M. A. Ahmad and A. Teredesai. Modeling spread of ideas in online social networks. In *Proceedings of the fifth Australasian conference on Data mining and analystics - Volume 61*, AusDM '06, pages 185–190, Darlinghurst, Australia, Australia, 2006. Australian Computer Society, Inc.

[3] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. Optics: Ordering points to identify the clustering structure. In A. Delis, C. Faloutsos, and S. Ghandeharizadeh, editors, *SIGMOD 1999, Proceedings ACM SIGMOD International Conference on Management of Data, June 1-3, 1999, Philadelphia, Pennsylvania, USA*, pages 49–60. ACM Press, 1999.

[4] A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. In *NIPS*, 2006.

[5] A. Argyriou, A. Micchelli, M. Pontil, and Y. Ying. A spectral regularization framework for multi-task structure learning. In *NIPS*, 2007.

[6] B. Bakker and T. Heskes. Task clustering and gating for bayesian multitask learning. *Journal of Machine Learning Research*, 4:2003, 2003.

[7] M. A. Bayir, M. Demirbas, and N. Eagle. Discovering spatiotemporal mobility profiles of cellphone users. In *WOWMOM*, pages 1–9, 2009.

[8] R. A. Becker, R. Cáceres, K. Hanson, J. M. Loh, S. Urbanek, A. Varshavsky, and C. Volinsky. A tale of one city: Using cellular network data for urban planning. *IEEE Pervasive Computing*, 10(4):18–26, 2011.

[9] R. A. Becker, R. Cáceres, K. Hanson, J. M. Loh, S. Urbanek, A. Varshavsky, and C. Volinsky. A tale of one city: Using cellular network data for urban planning. *IEEE Pervasive Computing*, 10(4):18–26, 2011.

[10] D. P. Bertsekas. *Nonlinear programming.* Athena Scientific. 2nd edition., 1999.

[11] W. Bi and J. T. Kwok. Multi-label classification on tree- and dag-structured hierarchies. In *ICML*, 2011.

[12] V. Bolón-Canedo, N. Snchez-Maroón, and A. Alonso-Betanzos. A review of feature selection methods on synthetic data. *Knowledge and Information Systems*, pages 1–37, 2012.

[13] E. V. Bonilla, K. M. Chai, and C. K. I. Williams. Multi-task gaussian process prediction. In *NIPS*, 2008.

[14] N. Bouguila and D. Ziou. A countably infinite mixture model for clustering and feature selection. *Knowledge and Information Systems*, pages 1–20, 2011. 10.1007/s10115-011-0467-4.

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

[16] D. Brauckhoff, X. Dimitropoulos, A. Wagner, and K. Salamatian. Anomaly extraction in backbone networks using association rules. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, IMC '09, pages 28–34, 2009.

[17] D. Brauckhoff, K. Salamatian, and M. May. Applying pca for traffic anomaly detection: Problems and solutions. In *INFOCOM*, pages 2866–2870. IEEE, 2009.

[18] U. Brefeld and T. Scheffer. Semi-supervised learning for structured output variables. In *ICML*, pages 145–152, 2006.

[19] S. Budhaditya, D.-S. Pham, M. Lazarescu, and S. Venkatesh. Effective anomaly detection in sensor networks data streams. *IEEE International Conference on Data Mining, ICDM2009*, 0:722–727, 2009.

[20] F. Calabrese, G. D. Lorenzo, L. Liu, and C. Ratti. Estimating origin-destination flows using mobile phone location data. *IEEE Pervasive Computing*, 10(4):36–44, 2011.

[21] H. Cao, Y. Zhou, L. Shou, and G. Chen. Attribute outlier detection over data streams. In *DASFAA (2)*, pages 216–230, 2010.

[22] X. Cao, G. Cong, and C. S. Jensen. Mining significant semantic locations from gps data. *PVLDB*, 3(1):1009–1020, 2010.

[23] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), 2009.

[24] C. Chang and C. Lin. Libsvm: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[25] O. Chapelle, P. K. Shivaswamy, S. Vadrevu, K. Q. Weinberger, Y. Zhang, and B. L. Tseng. Multi-task learning for boosting with application to web search ranking. In *KDD*, pages 1189–1198, 2010.

[26] J. Chen, J. Liu, and J. Ye. Learning incoherent sparse and low-rank patterns from multiple tasks. In *The Sixteenth ACM SIGKDD International Conference On Knowledge Discovery and Data Mining (SIGKDD 2010)*, 2010.

[27] J. Chen, J. Zhou, and J. Ye. Integrating low-rank and group-sparse structures for robust multi-task learning. In *KDD*, pages 42–50, 2011.

[28] W. Chen, Y. Yuan, and L. Zhang. Scalable influence maximization in social networks under the linear threshold model. In *ICDM*, pages 88–97, 2010.

[29] X. Chen, W. Pan, J. T. Kwok, and J. G. Carbonell. Accelerated gradient method for multi-task sparse learning problem. *IEEE International Conference on Data Mining (ICDM09)*, 0:746–751, 2009.

[30] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: user movement in location-based social networks. In *KDD*, pages 1082–1090, 2011.

[31] C.-T. Chu, S. K. Kim, Y.-A. Lin, Y. Yu, G. R. Bradski, A. Y. Ng, and K. Olukotun. Map-reduce for machine learning on multicore. In *NIPS*, pages 281–288, 2006.

[32] F. Chung. Spectral graph theory. *CBMS Reginal Conferences Series*, 92, 1997.

[33] W. Dai, Q. Yang, G. rong Xue, and Y. Yu. Boosting for transfer learning. In *International Conference on Machine Learning*, 2007.

[34] P. H. dos Santos Teixeira and R. L. Milidiú. Data stream anomaly detection through principal subspace tracking. In *SAC '10: Proceedings of the 2010 ACM Symposium on Applied Computing*, pages 1609–1616, New York, NY, USA, 2010. ACM.

[35] N. U. F. Dosenbach, B. Nardos, A. L. Cohen, D. A. Fair, J. D. Power, J. A. Church, S. M. Nelson, G. S. Wig, A. C. Vogel, C. N. Lessov-Schlaggar, K. A. Barnes, J. W. Dubis, E. Feczko, R. S. Coalson, J. R. Pruett, D. M. Barch, S. E. Petersen, and B. L. Schlaggar. Prediction of individual brain maturity using fmri. *Science*, 329(5997):1358–1361, 2010.

[36] J. Duchi and Y. Singer. Boosting with structural sparsity. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, pages 297–304, 2009.

[37] N. Eagle, A. S. Pentland, and D. Lazer. Inferring social network structure using mobile phone data. *PNAS*, 109(21), 2009.

[38] M. Eiermann, O. G. Ernst, and E. Ullmann. Computational aspects of the stochastic finite element method. *Comput. Vis. Sci.*, 10:3–15, February 2007.

[39] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, pages 226–231, 1996.

[40] T. Evgeniou, C. A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 2005.

[41] T. Evgeniou and M. Pontil. Regularized multi–task learning. In *KDD*, pages 109–117, 2004.

[42] H. Fei and J. Huan. Structure feature selection for graph classification. In *Proc. ACM 17th Conference on Information and Knowledge Management*, 2008.

[43] H. Fei and J. Huan. L2 norm regularized feature kernel regression for graph data. In *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*, pages 593–600, 2009.

[44] H. Fei and J. Huan. Boosting with structure information in the functional space: an application to graph classification. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 2010.

[45] H. Fei and J. Huan. Structured feature selection and task relationship inference for multi-task learning. In *Proceedings of the IEEE International Conference on Data Mining (ICDM'11)*, 2011.

[46] H. Fei, R. Jiang, Y. Yang, B. Luo, and J. Huan. Content based social behavior prediction: A multi-task learning approach. In *Proceedings of the 20th ACM Conference on Information and Knowledge Management (CIKM'11)*, 2011.

[47] H. Fei, B. Quanz, and J. Huan. Regularization and feature selection for networked features. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management (CIKM'10)*, 2010.

[48] C. Fellbaum. *WordNet: an electronic lexical database*. the MIT Press, 1998.

[49] G. W. Flake, S. Lawrence, and C. L. Giles. Efficient identification of web communities. In *KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 150–160, 2000.

[50] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[51] C. Franke and M. Gertz. Orden: outlier region detection and exploration in sensor networks. In *SIGMOD Conference*, pages 1075–1078, 2009.

[52] Y. Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121:256–285, 1995.

[53] Y. Freund and R. Shapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of the Second European Conference on Computational Learning Theory*, 1995.

[54] N. Friedkin. Information flow through strong and weak ties in intraorganizational social networks. *Social Networks*, 3:273–285, 1982.

[55] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28(2):337–407, 2000.

[56] J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *The Annals of Applied Statistics*, page to be appeared, 2009.

[57] K. Fukunaga. *Introduction to statistical pattern recognition (2nd ed.)*. Academic Press Professional, Inc., San Diego, CA, USA, 1990.

[58] D. Gale and S. Kariv. Bayesian learning in social networks. *Games and Economic Behavior*, 45(2):329–346, 2003.

[59] D. Gibson, J. Kleinberg, and P. Raghavan. Inferring web communities from link topology. In *HYPERTEXT '98: Proceedings of the ninth ACM conference on Hypertext and hypermedia : links, objects, time and space—structure in hypermedia systems*, pages 225–234, New York, NY, USA, 1998. ACM.

[60] J.-C. Gomez, E. Boiy, and M.-F. Moens. Highly discriminative statistical features for email classification. *Knowl. Inf. Syst.*, 31(1):23–53, 2012.

[61] R. Gross and A. Acquisti. Information revelation and privacy in online social networks. In *Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, WPES '05, pages 71–80, New York, NY, USA, 2005. ACM.

[62] X. Gu and H. Wang. Online anomaly prediction for robust cluster systems. In *ICDE*, pages 1000–1011, 2009.

[63] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422, 2002 January.

[64] V. Ha-Thuc and J.-M. Renders. Large-scale hierarchical text classification without labelled data. In *WSDM*, pages 685–694, 2011.

[65] G. Haffari, Y. Wang, S. Wang, G. Mori, and F. Jiao. Boosting with incomplete information. In *International Conference on Machine Learning*, 2008.

[66] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, 2009.

[67] D. Haussler. Convolution kernels on discrete structures. *Technical Report UCSC-CRL099-10, Computer Science Department, UC Santa Cruz*, 1999.

[68] C. Haythornthwaite. Social network analysis: An approach and technique for the study of information exchange. *Library and Information Science Research*, 18:323–342, 1996.

[69] S. Hirose, K. Yamanishi, T. Nakata, and R. Fujimaki. Network anomaly detection based on eigen equation compression. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1185–1194, New York, NY, USA, 2009. ACM.

[70] D. Hsu, S. M. Kakade, J. Langford, and T. Zhang. Multi-label prediction via compressed sensing. In *NIPS*, 2009.

[71] J. Huan, W. Wang, and J. Prins. Efficient mining of frequent subgraph in the presence of isomorphism. In *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM)*, pages 549–552, 2003.

[72] L. Huang, M. I. Jordan, A. Joseph, M. Garofalakis, and N. Taft. In-network pca and anomaly detection. In *In NIPS*, pages 617–624, 2006.

[73] R. Huckfeldt and J. Sprangue. Networks in context: The social flow of political information. *The Academy of Management Review*, 81(4):1179–1216, 1979.

[74] V. I and S. CL. The phosphatidylinositol 3-kinase akt pathway in human cancer. *Nat Rev Cancer.*, 2(7):489–501, 2002.

[75] T. Idé, A. C. Lozano, N. Abe, and Y. Liu. Proximity-based anomaly detection using sparse structure learning. In *SDM*, pages 97–108, 2009.

[76] T. Idé, S. Papadimitriou, and M. Vlachos. Computing correlation anomaly scores using stochastic nearest neighbors. In *ICDM '07: Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, pages 523–528, Washington, DC, USA, 2007. IEEE Computer Society.

[77] M. Isaac, B. Raul, E. Gerard, and G. Moisès. On-line fault diagnosis based on the identification of transient stages. In *in Proc. of 20th European Symposium on Computer Aided Process Engineering C ESCAPE20*. Elsevier B.V., 2010.

[78] S. Isaacman, R. A. Becker, R. Cáceres, S. G. Kobourov, M. Martonosi, J. Rowland, and A. Varshavsky. Identifying important places in people's lives from cellular network data. In *Pervasive*, pages 133–151, 2011.

[79] L. Jacob, F. Bach, and J.-P. Vert. Clustered multi-task learning: a convex formulation. In *Advances in Neural Information Processing Systems (NIPS09), 2009.*, 2009.

[80] A. Jalali, P. Ravikumar, S. Sanghavi, and C. Ruan. A dirty model for multi-task learning. In *In Proceedings of the Advances in Neural Information Processing Systems (NIPS 2010)*, pages 964–972, 2010.

[81] R. Jenatton, G. Obozinski, and F. Bach. Structured sparse principal component analysis. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS) 2010*, 2010.

[82] S. Ji and J. Ye. An accelerated gradient method for trace norm minimization. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, pages 457–464, New York, NY, USA, 2009. ACM.

[83] R. Jiang, H. Fei, and J. Huan. Anomaly localization by joint sparse pca in wireless sensor networks. In *Proceedings of the The 4th International Workshop on Knowledge Discovery from Sensor Data (SensorKDD-2010)*, 2010.

[84] R. Jiang, H. Fei, and J. Huan. Anomaly localization for network data streams with graph joint sparse pca. In *KDD*, pages 886–894, 2011.

[85] N. Jin, C. Young, and W. Wang. Graph classification based on pattern co-occurrence.

In *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*, pages 573–582, 2009.

[86] M. Kanehisa, S. Goto, M. Hattori, K. F. Aoki-Kinoshita, M. Itoh, S. Kawashima, T. Katayama, M. Araki, , and M. Hirakawa. From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Research*, 34:D354–357, 2006.

[87] H. Kashima, K. Tsuda, and A. Inokuchi. Marginalized kernels between labeled graphs. In *Proc. of the Twentieth Int. Conf. on Machine Learning (ICML)*, 2003.

[88] T. Kato, H. Kashima, and K. A. Masashi Sugiyama. Multi-task learning via conic programming. In *NIPS*, 2007.

[89] H. G. Kayacik, A. N. Zincir-Heywood, and M. I. Heywood. Selecting features for intrusion detection: A feature relevance analysis on kdd 99. In *PST*, 2005.

[90] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03, pages 137–146, New York, NY, USA, 2003. ACM.

[91] E. Keogh and T. Folias. The ucr time series data mining archive. Website, 2002. `http://www.cs.ucr.edu/eamonn/TSDMA/index.html`.

[92] E. Keogh, J. Lin, and A. Fu. Hot sax: Efficiently finding the most unusual time series subsequence. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, ICDM '05, pages 226–233, Washington, DC, USA, 2005.

[93] D. H. Kim, J. Hightower, R. Govindan, and D. Estrin. Discovering semantically meaningful places from pervasive rf-beacons. In *UbiComp*, pages 21–30, 2009.

[94] S. Kim and E. P. Xing. Structured feature selection in highdimensional space via

block regularized regression. In *In Proceedings of the 24th International Conference on Conference on Uncertainty in Artificial Intelligence*, 2008.

[95] S. Kim and E. P. Xing. Tree-guided group lasso for multi-task regression with structured sparsity. In *In proceedings of The 27th International Conference on Machine Learning (ICML 2010)*, 2010.

[96] R. I. Kondor, N. Shervashidze, and K. M. Borgwardt. The graphlet spectrum. In *ICML09*, volume 382, page 67. ACM, 2009.

[97] X. Kong and P. S. Yu. gmlc: a multi-label feature selection framework for graph classification. *Knowl. Inf. Syst.*, 31(2):281–305, 2012.

[98] D. Kosambi. Statistics in function space. 7:76–88, 1943.

[99] B. Krishnamurthy and C. E. Wills. On the leakage of personally identifiable information via online social networks. *SIGCOMM Comput. Commun. Rev.*, 40:112–117, January 2010.

[100] T. Kudo, E. Maeda, and Y. Matsumoto. An application of boosting to graph classification. In *NIPS*, 2004.

[101] A. Lakhina, M. Crovella, and C. Diot. Diagnosing network-wide traffic anomalies. In *In ACM SIGCOMM*, pages 219–230, 2004.

[102] A. Lakhina, M. Crovella, and C. Diot. Mining anomalies using traffic feature distributions. In *In ACM SIGCOMM*, pages 217–228, 2005.

[103] J.-G. Lee, J. Han, and X. Li. Trajectory outlier detection: A partition-and-detect framework. In *ICDE*, pages 140–149, 2008.

[104] C. Leslie, E. Eskin, and W. Noble. The spectrum kernel: a string kernel for svm protein classification. In *Pac Symp Biocomput*, pages 564–75, 2002.

[105] S. Levy. Interactive 3-d visualization of particle systems with partiview. *Proceedings of the International Astronomical Union Symposium on "Astrophysical Supercomputing Using Particles*, 208, 2001.

[106] C. Li and H. Li. Newwork-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics*, 24(9):1175–1182, 2008.

[107] L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 661–670, New York, NY, USA, 2010. ACM.

[108] P. Li. Adaptive base class boost for multi-class classification. In *International Conference on Machine Learning*, 2008.

[109] L. Liang, V. Mandal, Y. Lu, and D. Kumar. Mcm-test: a fuzzy-set-theory-based approach to differential analysis of gene pathways. *BMC Bioinformatics*, 9(Suppl 6):S16, 2008.

[110] D. Liben-Nowell and J. Kleinberg. Tracing information flow on a global scale using internet chain-letter data. *PNAS*, 105(12):4633–4638, 2008.

[111] H. Liu, M. Palatucci, and J. Zhang. Blockwise coordinate descent procedures for the multi-task lasso, with applications to neural semantic basis discovery. In *In proceedings of The 26th International Conference on Machine Learning (ICML 2009)*, 2009.

[112] J. Liu, S. Ji, and J. Ye. Multi-task feature learning via efficient $\ell_{2,1}$-norm minimization. In *Uncertainty in Artificial Intelligence*, 2009.

[113] J. Liu, S. Ji, and J. Ye. *SLEP: Sparse Learning with Efficient Projections*. Arizona State University, 2009.

[114] X. Liu, X. Wu, H. Wang, R. Z. 0003, J. Bailey, and K. Ramamohanarao. Mining distribution change in stock order streams. In *ICDE*, pages 105–108, 2010.

[115] N. Loeff, D. Forsyth, and D. Ramachandran. Manifoldboost: Stagewise function approximation for fully-, semiand un-supervised learning. In *International Conference on Machine Learning*, 2008.

[116] P. M. Long and R. A. Servedio. Random classification noise defeats all convex potential boosters. In *International Conference on Machine Learning*, pages 608–615, 2008.

[117] K. Lounici, M. Pontil, A. Tsybakov, and S. Van De Geer. Taking advantage of sparsity in multi-task learning. *Knowledge and Information Systems*, 20(1):109–348, 2009.

[118] A. C. Lozano and N. Abe. Multi-class cost-sensitive boosting with p-norm loss functions. In *ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, 2008.

[119] A. Lung-Yut-Fong, C. Lévy-Leduc, and O. Cappé. Distributed detection/localization of change-points in high-dimensional network traffic data. *CoRR*, abs/0909.5524, 2009.

[120] S. L. Marple. *Digital Spectral Analysis With Applications*. Prentice Hall, Australia, Sydney, 1987.

[121] Mehmet and Ocakci. Commuting to the istanbul historical core: The case of industrial employees. *European Planning Studies*, 9(1):117–127, 2001.

[122] T. M. Mitchell, S. V. Shinkareva, A. Carlson, and et al. Predicting human brain activity associated with the meanings of nouns. *Science*, 320:1191–1195, 2008.

[123] S. Mukkamala and A. H. Sung. Feature ranking and selection for intrusion detection systems using support vector machines. In *Proceedings of the Second Digital Forensic Research Workshop*, 2002.

[124] S. Mukkamala and A. H. Sung. Feature selection for intrusion detection using neural networks and support vector machines. *Journal of the Transportation Research Board of the National Academies*, pages 33–39, 2003.

[125] A. Murzin, S. Brenner, T. Hubbard, and C. Chothia. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology*, 247:536–40, 1995.

[126] Y. Nesterov. Introductory lectures on convex optimization: A basic course. 2003.

[127] Y. Nesterov. Gradient methods for minimizing composite objective function. *CORE Discussion Paper*, 76:265–286, 2007.

[128] H. Nguyen, Y. Tan, and X. Gu. Pal: Propagation-aware anomaly localization for cloud hosted distributed applications. In *PST*, Cascais, Portugal, 2011.

[129] S. Nowozin, K. Tsuda, T. Uno, T. Kudo, and G. Bakir. Weighted substructure mining for image analysis. In *Computer Vision and Pattern Recognition 2007. CVPR '07*, pages 1–8, 2007.

[130] C. L. Nutt, D. R. Mani, R. A. Betensky, P. Tamayo, J. G. Cairncross, C. Ladd, U. Pohl, C. Hartmann, M. E. McLaughlin, T. T. Batchelor, P. M. Black, A. von Deimling, S. L. Pomeroy, T. R. Golub, and D. N. Louis.

[131] N. I. of Health. Metastatic cancer, 2010. `http://www.cancer.gov/cancertopics/factsheet/Sites-Types/metastatic`.

[132] U. D. of Transportation and B. of Transportation Statistics. From home to work, the average commute is 26.4 minutes. *OmniStats*, 3(4), 2003.

[133] G. Pandey, S. Chawla, S. Poon, B. Arunasalam, and J. G. Davis. Association rules network: Definition and applications. *Stat. Anal. Data Min.*, 1(4):260–279, 2009.

[134] S. Phithakkitnukoon, F. Calabrese, Z. Smoreda, and C. Ratti. Out of sight out of mind-how our mobile social network changes during migration. In *SocialCom/PASSAT*, pages 515–520, 2011.

[135] S. Phithakkitnukoon and C. Ratti. Inferring asymmetry of inhabitant flow using call detail records. *Journal of Advances in Information Technology*, 2(4), 2011.

[136] A. Quattoni, X. Carreras, M. Collins, and T. Darrell. An efficient projection for l1,infinity regularization. In *In proceedings of The 26th International Conference on Machine Learning (ICML 2009)*, 2009.

[137] D. Quercia, G. D. Lorenzo, F. Calabrese, and C. Ratti. Mobile phones and outdoor advertising: Measurable advertising. *IEEE Pervasive Computing*, 10(2):28–36, 2011.

[138] E. Ricci, T. D. Bie, and N. Cristianini. Magic moments for structured output prediction. *Journal of Machine Learning Research*, 9:2803–2846, 2008.

[139] H. Ringberg, A. Soule, J. Rexford, and C. Diot. Sensitivity of pca for traffic anomaly detection. In *SIGMETRICS '07: Proceedings of the 2007 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 109–120, 2007.

[140] D. M. Romero, B. Meeder, and J. M. Kleinberg. Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on twitter. In *WWW*, pages 695–704, 2011.

[141] J. Rousu, C. Saunders, S. Szedmák, and J. Shawe-Taylor. Kernel-based learning of hierarchical multilabel classification models. *Journal of Machine Learning Research*, 7:1601–1626, 2006.

[142] H. Saigo and et. al. gboost: Graph learning toolbox for matlab, 2007. `http://www.kyb.tuebingen.mpg.de/bs/people/nowozin/gboost/`.

[143] H. Saigo, N. Krämer, and K. Tsuda. Partial least squares regression for graph mining. In *Proc. SIGKDD08*, 2008.

[144] H. Saigo, S. Nowozin, T. Kadowaki, T. Kudo, and K. Tsuda. gboost: a mathematical programming approach to graph classification and regression. *Journal of Machine Learning*, 75(1):69–89, 2009.

[145] T. Sandler, P. P. Talukdar, and L. H. Ungar. Regularized learning with networks of features. In *NIPS08*, 2008.

[146] R. Schapire. The strength of weak learnability. *Machine Learning*, 5:197–227, 1990.

[147] R. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37:297–336, 1999.

[148] C. A. Schenk and G. I. Schuëller. *Uncertainty Assessment of Large Finite Element Systems*, volume 24. Springer-Verlag, Berlin/Heidelberg/New York, 2005.

[149] B. Schölkopf and A. J. Smola. *Learning with Kernels*. the MIT Press, 2002.

[150] A. Schwaighofer, V. Tresp, and K. Yu. Learning gaussian process kernels via hierarchical bayes. In *In Advances in Neural Information Processing Systems (NIPS*, pages 1209–1216. MIT Press, 2004.

[151] J. Silva and R. Willett. Detection of anomalous meetings in a social network. In *42nd Annual Conference on Information Sciences and Systems, 2008. CISS 2008.*, pages 636 –641, 2008.

[152] D. Singh, P. G. Febbo, K. Ross, D. G. Jackson, J. Manola, C. Ladd, P. Tamayo, A. A. Renshaw, A. V. D'Amico, J. P. Richie, E. S. Lander, M. Loda, P. W. Kantoff, T. R. Golub, and W. R. Sellers. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 1(2):203 – 209, 2002.

[153] P. Singla and M. Richardson. Yes, there is a correlation: - from social networks to personal behavior on the web. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 655–664, New York, NY, USA, 2008. ACM.

[154] A. M. Smalter, J. Huan, and G. H. Lushington. Graph wavelet alignment kernels for drug virtual screening. *J. Bioinformatics and Computational Biology*, 7(3):473–497, 2009.

[155] X. Song, Y. Chi, K. Hino, and B. L. Tseng. Information flow modeling based on diffusion rate for prediction and ranking. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 191–200, New York, NY, USA, 2007. ACM.

[156] X. Song, B. L. Tseng, C.-Y. Lin, and M.-T. Sun. Personalized recommendation driven by information flow. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 509–516, New York, NY, USA, 2006. ACM.

[157] A. I. Su, J. B. Welsh, L. M. Sapinoso, and *et al.*. Molecular classification of human carcinomas by use of gene expression signatures. *Cancer Research*, 15(20):7388–93, 2001.

[158] L. Sun, B. Ceran, and J. Ye. A scalable two-stage approach for a class of dimensionality reduction techniques. In *KDD*, pages 313–322, 2010.

[159] F. Tahi and H.-T. Lin. Multi-Label Classification with Principle Label Space Transformation. In *2nd International Workshop on Learning from Multi-Label Data (MLD'10)*, pages 45–52, June 2010.

[160] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319 – 2323, 2000.

[161] M. Thoma, H. Cheng, A. Gretton, J. Han, H.-P. Kriegel, A. J. Smola, L. Song, P. S. Yu, X. Yan, and K. M. Borgwardt. Near-optimal supervised feature selection among frequent subgraphs. In *Proccedings of the 2009 SIAM Conference on Data Mining (SDM 2009)*, pages 1076–1087. Philadelphia, PA, USA, 2009.

[162] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society B*, 58:267–288, 1996.

[163] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *J. R. Statist. Soc.*, 67(1)):91–108, 2005.

[164] N. Tichy, M. Tushman, and C. Fombrun. Social network analysis for organizations. *The American Political Science Review*, 4(4):507–519, 1979.

[165] J. L. Toole, N. Eagle, and J. B. Plotkin. Spatiotemporal correlations in criminal offense records. *ACM TIST*, 2(4):38, 2011.

[166] V. A. Traag, A. Browet, F. Calabrese, and F. Morlot. Social event detection in massive mobile phone data using probabilistic location inference. In *SocialCom/PASSAT*, pages 625–628, 2011.

[167] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *ICML*, 2004.

[168] K. Tsuda. Entire regularization paths for graph data. In *ICML07*, 2007.

[169] V. N. Vapnik. The nature of statistical learning theory. *Springer-Verlag*, 1995.

[170] A. Vasalou, A. N. Joinson, and D. Courvoisier. Cultural differences, experience with social networks and the nature of "true commitment" in facebook. *Int. J. Hum.-Comput. Stud.*, 68:719–728, October 2010.

[171] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications.* New York: Cambridge University Press, 1994.

[172] J. Weston, A. Elisseeff, G. BakIr, and F. Sinz. Spider: General purpose machine learning toolbox in matlab, 2007. Software available at `www.kyb.mpg.de/bs/people/spider/`.

[173] C. White, L. Plotnick, J. Kushma, S. R. Hiltz, and M. Turoff. An online social network for emergency management. *International Journal of Emergency Management*, 6(3-4):369–382, 2009.

[174] Z. Xiang, Y. Xi, U. Hasson, and P. Ramadge. Boosting with spatial regularization. In *NIPS '09: Proceedings of the Neural Information Processing Systems (NIPS2009)*, 2009.

[175] N. Xu, S. Rangwala, and et al. A wireless sensor network for structural monitoring. In *IN SENSYS*, pages 13–24, 2004.

[176] S. Xu, S. Bao, B. Fei, Z. Su, and Y. Yu. Exploring folksonomy for personalized search. In S.-H. Myaeng, D. W. Oard, F. Sebastiani, T.-S. Chua, and M.-K. Leong, editors, *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008, Singapore, July 20-24, 2008*, pages 155–162. ACM, 2008.

[177] Z. Xu and K. Kersting. Multi-task learning with task relations. In *ICDM*, pages 884–893, 2011.

[178] J. Yan, N. Liu, G. Wang, W. Zhang, Y. Jiang, and Z. Chen. How much can behavioral targeting help online advertising? In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 261–270, New York, NY, USA, 2009. ACM.

[179] R. Yan, J. Tesic, and J. R. Smith. Model-shared subspace boosting for multi-label classification. In *ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pages 834–843, 2007.

[180] X. Yan, H. Cheng, J. Han, and P. Yu. Mining significant graph patterns by leap search. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 433–444. ACM, 2008.

[181] J. Yang and J. Leskovec. Modeling information diffusion in implicit networks. *Data Mining, IEEE International Conference on*, 0:599–608, 2010.

[182] S. Yang and W. Zhou. Anomaly detection on collective moving patterns: Manifold learning based analysis of traffic streams. In *SocialCom/PASSAT*, pages 704–707, 2011.

[183] M. Ye, D. Shou, W.-C. Lee, P. Yin, and K. Janowicz. On the semantic annotation of places in location-based social networks. In *KDD*, pages 520–528, 2011.

[184] H. Yu and J. Yang. A direct LDA algorithm for high-dimensional data with application to face recognition. *Pattern Recognition*, 34(10):2067–2070, 2001.

[185] S. Yu, V. Tresp, and K. Yu. Robust multi-task learning with t-processes. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, pages 1103–1110, New York, NY, USA, 2007. ACM.

[186] C. Yuan. Multi-task learning for bayesian matrix factorization. In *ICDM*, pages 924–931, 2011.

[187] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68:49–67, 2006.

[188] J. Zhang, Q. Gao, and H. Wang. Anomaly detection in high-dimensional network data streams: A case study. In *IEEE International Conference on Intelligence and Security Informatics, 2008. ISI 2008.*, pages 251 –253, June 2008.

[189] J. Zhang, Q. Gao, H. H. Wang, Q. Liu, and K. Xu. Detecting projected outliers in high-dimensional data streams. In *DEXA*, pages 629–644, 2009.

[190] Y. Zhang and D.-Y. Yeung. A convex formulation for learning task relationships in multi-task learning. In *In Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence (UAI 2010)*, pages 733–742, 2010.

[191] Y. Zhang, D.-Y. Yeung, and Q. Xu. Probabilistic multi-task feature selection. In *In Proceedings of the Advances in Neural Information Processing Systems (NIPS 2010)*, pages 2559–2567, 2010.

[192] Z. Zhang and N. Ye. Locality preserving multimodal discriminative learning for supervised feature selection. *Knowledge and Information Systems*, 27:473–490, 2011. 10.1007/s10115-010-0306-z.

[193] P. Zhao and B. Yu. Grouped and hierarchical model selection through composite absolute penalties. *Annals of Statistics*, 2006.

[194] L. Zheng, S. Wang, C. hoon Lee, and Y. Liu. Information theoretic regularization for semi-supervised boosting. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009.

[195] J. Zhou, L. Yuan, J. Liu, and J. Ye. A multi-task learning formulation for predicting disease progression. In *KDD*, pages 814–822, 2011.

[196] H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1430, 2001.

[197] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society B*, 67:301–320, 2005.

[198] H. Zou and M. Yuan. F∞ norm support vector machine. *Statistica Sinica*, 18:379–398, 2008.