# A Family of Joint Sparse PCA Algorithms for Anomaly Localization in Network Data Streams

By

Ruoyi Jiang

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

Jun Huan, Chairperson

Committee members

Victor Frost

Bo Luo

Date defended: _____

The Master Thesis Committee for Ruoyi Jiang certifies
that this is the approved version of the following master thesis :

A Family of Joint Sparse PCA Algorithms for Anomaly Localization in Network Data Streams

Jun Huan, Chairperson

Date approved: _____

# Abstract

Determining anomalies in data streams that are collected and transformed from various types of networks has recently attracted significant research interest. Principal Component Analysis (PCA) is arguably the most widely applied unsupervised anomaly detection technique for networked data streams due to its simplicity and efficiency. However, none of existing PCA based approaches addresses the problem of identifying the sources that contribute most to the observed anomaly, or anomaly localization. In this paper, we first proposed a novel joint sparse PCA method to perform anomaly detection and localization for network data streams. Our key observation is that we can detect anomalies and localize anomalous sources by identifying a low dimensional abnormal subspace that captures the abnormal behavior of data. To better capture the sources of anomalies, we incorporated the structure of the network stream data in our anomaly localization framework. Also, an extended version of PCA, multidimensional KLE, was introduced to stabilize the localization performance. We performed comprehensive experimental studies on four real-world data sets from different application domains and compared our proposed techniques with several state-of-the-arts. Our experimental studies demonstrate the utility of the proposed methods.

# Contents

**1 Introduction** **1**

**2 Related Work** **4**

2.1 Current Anomaly Detection Techniques . . . . . 4

2.2 Current Anomaly Localization Techniques . . . . . 7

2.3 Applications of Anomaly Detection and Anomaly Localization . . . . . 8

**3 Preliminaries** **11**

3.1 Notation . . . . . 11

3.2 Network Data Streams . . . . . 12

3.3 Applying PCA for Anomaly Localization . . . . . 12

**4 Sparse PCA for Anomaly Localization** **16**

4.1 Joint Sparse PCA . . . . . 16

4.2 Anomaly Scoring . . . . . 18

4.3 Graph Guided Joint Sparse PCA . . . . . 20

4.4 Extension with Karhunen Loève Expansion . . . . . 22

4.5 Optimization Algorithms . . . . . 28

**5 Evaluation** **34**

5.1 Data Sets . . . . . 34

5.2 Experimental Protocol . . . . . 37

5.2.1 Localization Model Construction . . . . . 37
5.2.2 Detection Model Construction . . . . . 38
5.2.3 Model Evaluation . . . . . 38
5.2.4 Parameter Selection . . . . . 39
5.3 Anomaly Detection Performance . . . . . 40
5.4 Anomaly Localization Performance . . . . . 41
5.5 Trend Analysis on Abnormal Score . . . . . 45
5.6 Sensitivity of Parameter . . . . . 48

# List of Figures


3.1 Illustration of time-evolving stock indices data . . . . . 15

3.2 Comparing PCA and Sparse PCA. . . . . 15

4.1 Demonstration of JSPCA on three network data streams with one anomaly (solid line) and two normal streams (dot lines). . . . . 17

4.2 Comparing different anomaly localization methods. From left to right: PCA, sparse PCA, JSPCA, and GJSPCA. . . . . 19

4.3 Comparing *joint sparse PCA* (JSPCA) and *graph joint sparse PCA* (GJSPCA). 19

4.4 From left to right: PC space for JSKLE and GJSKLE, abnormal score for JSKLE, and GJSKLE. . . . . 22

5.1 ROC curve for anomaly detection on sensor dataset and motorCurrent dataset. AUC for sensor dataset is 0.7832, for motorCurrent dataset is 0.9688 . . . . 41

5.2 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 . . . . . 42

5.3 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 . . . . . 42

5.4 AUC for different methods on three data sets . . . . . 43

5.5 pairwise ANOVA testing . . . . . 43

5.6 Anomaly Localization Performance of GJSPCA, Stochastic Nearest Neighborhood, Eigen-Equation Compression on Network Intrusion Data Set(DoS Attack) . . . . . 44
5.7 Most relevant features selected for different attacks . . . . . 44
5.8 Left: original data in time interval $[2001, 3300]$ in sensor dataset. Right: time series of abnormal score calculated from left figure (with window size 20 and offset 10) . . . . . 45
5.9 Left: original data in time interval $[7391, 8000]$ in sensor dataset. Right: time series of abnormal score calculated from left figure (with window size 20 and offset 10) . . . . . 46
5.10 Left: original data in motorcurrent dataset. Right: time series of abnormal score calculated from left figure (with window size 50 and offset 25) . . . . . 46
5.11 Left: original data in time interval $[341, 420]$ on stock market dataset. Right: time series of abnormal score calculated from left figure (with window size 20 and offset 5) . . . . . 47
5.12 From left to right, sensitivity analysis of GSPCA on $\lambda_1$, $\lambda_2$, $\delta$, and the dimension of the normal subspace. . . . . 49
5.13 From left to right, sensitivity analysis of GJSKLE on $\lambda_1$, $\lambda_2$, $\delta$, and the dimension of the normal subspace. . . . . 50
5.14 Sensitivity analysis of GJSKLE on N. . . . . 50

# List of Tables

3.1 Notations in the paper. . . . . 12

5.1 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 for anomaly localization, $Indices$: starting point and ending point of the intervals for anomaly localization, $W$: total number of data windows for anomaly localization, $W2$: total number of data windows for anomaly detection $L$: sliding window size, -: not applicable. . . . . 35

5.2 Optimal parameters combinations on three data sets. J:JSPCA, GJ: GJSPCA, JK:JSKLE, GJK: GJSKLE. The best temporal offset is 2 for all data sets . . . . 40

5.3 Features Indexes in KDD 99 Intrusion Detection Data set . . . . . 45

# Chapter 1

# Introduction

*Anomaly detection* is an important problem that has been researched within diverse research areas and application domains. Anomaly detection refers to detecting the abnormal patterns in data that do not conform to established normal behavior. Those non-conforming patterns are referred to as outliers, change, deviation, surprise, aberrant, peculiarity, intrusion, etc. Over time, anomaly detection in data streams that are collected and transformed from various types of networks has recently attracted significant research interest in the data mining community [5, 24, 51, 59]. Applications of the work could be found in network traffic data [59], sensor network streams [5], social networks [51], cloud computing [44], and finance networks [24] among others. The importance of anomaly detection in network data stream is due to the fact that anomalies in network data stream is significant and critical information in a wide variety of application domains. For example, anomalous network traffic usually results from malicious activity such as break-ins and computer abuse which are interesting form a computer security perspective. An anomalous event in optical sensor network could mean that something is on fire. Anomalies in video surveillance may indicate insertion of foreign objects.

Besides anomaly detection, another 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 [58] to find the anomalous components, or locating the sites for flooding and forest fires [14]. In the stock market, pinpointing the change points in a set of stock price time series is also critical for making intelligent trading decisions [37]. For network security, localizing the sources of the most serious threats in computer networks helps quickly and accurately repair and ensure security in networks [32].

Principal Component Analysis (PCA) is arguably the most widely applied unsupervised anomaly detection technique for network data streams [21, 32, 33]. However, a fundamental problem of PCA, as claimed in [48], is that the current PCA based anomaly detection methods can not be applied to anomaly localization. Our key observation is that the major obstacle for extending the PCA technique to anomaly localization lies in the high dimensionality of the abnormal space. If we manage to identify a low dimensional approximation of the high dimensional abnormal subspace using a few sources, we "localize" the abnormal sources. The starting point of our investigation hence is the recently studied sparse PCA framework [62] 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 my thesis, we explore several 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*

*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.

Another drawback of PCA is it only considers the spatial correlation between different streams but ignores the temporal correlation between different time stamps [4]. In order to overcome this problem, we introduce a multi-dimensional Karhunen Loève Expansion (KLE) as an extension of PCA to take care of both temporal and spacial correlations. PCA is a special case of multi-dimensional KLE with only spacial dimension. The corresponding methods are called *joint space KLE* (JSKLE) and *graph guided sparse KLE* (GJSKLE) respectively. The experiments proves that the JSKLE and GJSKLE stabilizes localization performance effectively when considering both spatial and temporal correlations.

The remainder of the thesis is organized as follows. In chapter 2, we present related work of anomaly localization. In chapter 3, we discuss the challenge of applying PCA to anomaly localization. In chapter 4 we introduce the formulation of JSPCA and GJSPCA, and their extended version JSKLE, GJSKLE, and the related optimization algorithm. We present our experimental study in chapter 5 and conclude in chapter 6.

# Chapter 2

# Related Work

Anomaly detection and localization has been the topic of a number of articles and books. Next, I will first introduced several techniques used in anomaly detection and anomaly localization. A variety of anomaly detection and localization techniques have been developed in several research communities, some are specifically for certain application domain, some are generic and applicable for many domain. Then we will cover the applications of these techniques. Anomaly detection and localization have extensive use in a wide variety of applications such as intrusion detection for computer security, fraud detection for credit cards, insurance or health care, fault detection in condition monitoring systems and so on.

## 2.1 Current Anomaly Detection Techniques

There are a variety of methodologies used to do anomaly detection. Here we focus on data mining-based anomaly detection techniques. Data mining techniques are well suited to anomaly detection problem because it is a process of extracting "patterns" from large volume of data. Specifically, when applied to network anomaly detection, data mining techniques construct models that could automatically discover the consistent and useful patterns of normal behaviors from the network data, and use these patterns to recognize anomalies and intrusions.

Based on whether data samples are labeled or not, the approaches fall into two categories: supervised anomaly detection and unsupervised anomaly detection. Supervised outlier detection techniques require the availability of a labeled training data set with labeled instances for the normal as well as the outlier class. In such techniques, predictive models are built for both normal and outlier classes. Any unseen data instance is compared against the two models to determine which class it belongs to. There are two major issues in supervised learning algorithms. First, training data is imbalanced because the anomalous instances are far fewer compared to the normal instances. Second, obtaining accurate labels is usually challenging. Other than these two issues, the supervised anomaly detection problem is similar to building predictive models.

Most supervised anomaly detection algorithms are classification based. In the training phase, a classifier is learned using the available labeled training data. In the testing phase, a test instance is classified as normal or anomalous using the classifier. Popular classifiers include neural network, Bayesian network, Support Vector Machines and ruled based. A neural network is trained on the normal training data to learn the different normal classes and then each test instance is provided as an input to the neural network. If the network accepts the test input, it is normal and if the network rejects a test input, it is an anomaly [25, 40]. Bayesian networks have also been widely used. A basic technique using a naive Bayesian network estimates the posterior probability of observing a class label (from a set of normal class labels and the anomaly class label), given a test data instance. The class label with largest posterior is chosen as the predicted class for the given test instance [54, 55]. Support Vector Machines (SVMs) have been applied to anomaly detection in the one-class setting since 1995. Such techniques use one class learning techniques for SVM and learn a region that contains the training data instances (a boundary). If a test instance falls within the learnt region, it is declared as normal, else it is declared as anomalous [46, 60]. Rule based anomaly detection techniques learn rules that capture the normal behavior of a system. A test instance that is not covered by any such rule is considered as an anomaly [56, 50].

An unsupervised outlier detection technique makes no assumption about the availability of labeled training data. Thus, these techniques are more widely applicable. Several unsupervised techniques make the basic assumptions such that the majority of the instances in the data set are normal. Nearest neighbor based techniques, clustering, statistical model and spectrum anomaly detection are most popular unsupervised anomaly detection techniques. Nearest neighbor based techniques based on an assumption that normal data instances occur in dense neighborhoods, while anomalies occur far from their closest neighbors. Basic nearest neighbor anomaly detection techniques can be broadly grouped into two categories: anomaly score is the distance of data instance to its kth nearest neighbor, or computed as the density of the data instance. Clustering based techniques usually consist of two steps. First, the data is clustered using a clustering algorithm such as k-means, Expectation Maximization and DBSCAN. In the second step, for each data instance, its distance to its closest cluster centroid is calculated as its anomaly score [15, 35, 17]. Statistical techniques usually have some assumption on the distribution of given data. By applying statical inference test we determine if an unseen instance is normal or abnormal. Normal instances occur in high probability regions of the statistical model while anomalies occur in a low probability regions of the stochastic model [18, 52]. Spectral anomaly detection techniques try to find a lower dimensional subspace in which normal instances and anomalies appear significantly different. Principal Component Analysis is the most widely used. Principal Components capture the normal and abnormal behaviors underlying the data and projection of data instances on principal components is used to make detection decision [21].

From another point of view, based on the data used in detection procedure, anomaly detection from network data streams are divided into two categories: those at the source level and those at the network level. The source level anomaly detection approaches embed detection algorithm at each stream source, resulting in a fully distributed anomaly detection system [19, 34, 44]. Detection is based on individual data and decision is made for each source. The major problems of these approaches are two folds: some source level anomalies

may not be indicative of network level anomalies due to the ignorance of the rest of the network [21], and there may not be available space to perform anomaly detection in each source. To improve source level anomaly localization methods, several algorithms have been recently proposed to anomaly at the network level. The network level anomaly detection approaches take the whole network into consideration. Since the decision is made based on the entire network, the network level anomaly detection approaches are not as knowledgeable about any source specifics. This leads to one of the major restrictions: they usually fail to pinpoint which sources should be responsible for the anomalies, that is, anomaly localization.

## 2.2 Current Anomaly Localization Techniques

Source level anomaly detection embeds detection algorithm at each stream source and makes decision for each source. Hence anomaly detection and anomaly localization are finished in one step. For network level anomaly detection, anomaly localization is an additional step after anomaly detection. More specifically, network level anomaly detection is a binary decision such that whether the whole network is normal or abnormal. If the network is abnormal, we need to go one step further to determine which sources are responsible for the observed anomaly.

Some algorithms have been recently proposed to localize anomaly at the network level. Brauckhoff [3] 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.*[30] 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 [53]. 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 [38]. 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.

Most related to our work, Ide *et al.*[22, 23] 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.*[20] 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. In [28], we proposed a two step approach that first computed normal subspace from ordinary PCA and then derived a sparse abnormal subspace on the residual data subtracted from the original data.

## 2.3 Applications of Anomaly Detection and Anomaly Localization

Applications of anomaly detection could be found in computer related system [59], sensor network streams [5], social networks [51], cloud computing [44], and finance networks [24] among others.

Detection of malicious activity in computer related system refers to intrusion detection. The malicious activities include flood-type attack, break-ins, and other forms of computer abuse. These attacks are different from the normal behavior of the computer system, and hence anomaly detection techniques are applicable in intrusion detection domain. There

are multiple data sources for intrusion detection and the common ones are at host level and network level. Based on the sources, Intrusion Detection System (IDS) are grouped into Host-Based IDS and Network-Based IDS. These intrusion detection systems were responsible for the security of an individual (host) machine instead of the security of the network as a whole. In contrast to a host-based IDS, a network-based IDS monitors and protects the network as a whole. The key challenge for intrusion detection is the large volume of data. Such data usually involves thousands of connections so the anomaly detection techniques need to be computationally efficient to handle these large sized inputs.

Fraud detection is another anomaly detection application which is applicable to many industries including banking and financial sectors, insurance, credit card companies, stock market and more [47, 45]. The fraud cases have to be detected from the available huge data sets such as the logged data and user behaviors. The types of frauds mostly discussed in recent papers are credit card frauds, mobile phone frauds, and insurance claim fraud. The most important requirement of anomaly detection techniques in this domain is to detect fraud in an online manner and as early as possible.

Anomaly detection and localization involving image data are are either interested in motion detection and localization (changes in an image over time) or in abnormal regions detection and localization on the static image [39, 6]. Image data has spatial as well as temporal characteristics, hence anomaly analysis has to be done in both spatial and temporal domain. One of the key challenges in this domain is the large size of the input. When dealing with video data, online techniques are required.

When applied to sensor network, anomaly detection and localization are usually responsible to detect faulty sensor from sensor network or detect events that are interesting for analysis. For instance, anomaly detection is a critical step in nature disaster monitoring including flooding and forest fire monitoring. Due to severe sensor resource constraints, the anomaly detection and localization techniques need to be power efficient. Another challenge is data is collected in a distributed fashion, and hence a distributed data mining approach

is required to analyze the data [7, 57].

Anomaly detection has also been applied to several other domains such as detecting novel topics or events a collection of documents or news articles, detecting anomalies in biological data, detecting users whose behavior deviates from the usual behavior in a social network.

# Chapter 3

# Preliminaries

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

## 3.1 Notation

We use uppercase calligraphic 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 $\lambda_1, \lambda_2$ 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 $i$th row and $j$th column. We use $x_i$ to represent the $i$th 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 $\mathbf{X} \in \mathcal{R}^{n \times p}$, $\|\mathbf{X}\|_{1,q} = \sum_{i=1}^{n} \|\tilde{\mathbf{x}}_i\|_q$ is the $l_1/l_q$ norm of the matrix $\mathbf{X}$, where $\tilde{\mathbf{x}}_i$ is the $i$th row of $\mathbf{X}$ in column vector form. Unless stated otherwise, all vectors are column vectors. In Table 3.1, we summarize the notations in our paper.

Table 3.1: Notations in the paper.

| Symbol | Notation |
|---|---|
| $\mathcal{S}$ | a set $\mathcal{S}$ |
| $\mathbf{X}$ | matrix $X$ |
| $x_{ij}$ | the entry of the $i$th row and the $j$th entry of matrix $X$ |
| $\mathbf{x}$ | a column vector $x$ |
| $x_i$ | the $i$th entry of the vector $x$ |
| $\mathbf{x}_i$ | the $i$th column of the matrix $X$ |
| $\tilde{\mathbf{x}}_i$ | $i$th row of $X$ in column vector form |

## 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 [10], *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 $X = (x_{i,j})$ 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 $t_i$.

## 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 [32, 21, 33]. 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:

$$\mathbf{X} = \mathbf{X}\mathbf{V}^{(1)}\mathbf{V}^{(1)^T} + \mathbf{X}\mathbf{V}^{(2)}\mathbf{V}^{(2)^T} = \mathbf{X}_n + \mathbf{X}_a \tag{3.1}$$

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 [21, 32]. 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 [48], lies in the fact that there is no direct mapping between two subspaces $\mathbf{V}_{(1)}$, $\mathbf{V}_{(2)}$ and the data sources. Specifically, let $\mathbf{V}_{(2)} = [\mathbf{v}_{k+1}, \cdots, \mathbf{v}_p]$ be the abnormal subspace spanned by the last $p - k$ PCs, $\mathbf{X}_a$ is essentially an aggregated operation that performs linear combination of all the data sources, as follows:

$$\begin{aligned} \mathbf{X}_a &= \mathbf{X}\mathbf{V}_{(2)}\mathbf{V}_{(2)}^T \\ &= \left[ \sum_{j=1}^{p} \mathbf{x}_j \tilde{\mathbf{v}}_j^T \tilde{\mathbf{v}}_1, \cdots, \sum_{j=1}^{p} \mathbf{x}_j \tilde{\mathbf{v}}_j^T \tilde{\mathbf{v}}_i, \cdots, \sum_{j=1}^{p} \mathbf{x}_j \tilde{\mathbf{v}}_j^T \tilde{\mathbf{v}}_{p-k} \right] \end{aligned} \tag{3.2}$$

where $\mathbf{x}_j$ is the data from the $j$th source and $\tilde{\mathbf{v}}_j$ is the $j$th row of $\mathbf{V}_2$ in column vector form. Considering the $i$th column of $\mathbf{X}_a$ with the value $\sum_{j=1}^{p} \mathbf{x}_j \tilde{\mathbf{v}}_j \tilde{\mathbf{v}}_i^T$, there is no correspondence between the original $i$th column of $\mathbf{X}$ and $i$th 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 necessary 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 $\forall t \in \mathcal{S}$, $\mathbf{x}_t$ has no projection on the abnormal subspace. In other words, these sources have no contribution to the abnormal behavior. Consider the squared prediction

error $SPE = tr(\mathbf{X}_a^T \mathbf{X}_a)$ and plug equation 3.2 in:

$$\begin{aligned} tr(\mathbf{X}_a^T \mathbf{X}_a) &= tr(\mathbf{X}_a \mathbf{X}_a^T) \\ &= tr(\mathbf{V}_2^T X^T X \mathbf{V}_2) \\ &= tr((\textstyle\sum_{j=1}^{p} \mathbf{x}_j \tilde{\mathbf{v}}_j^T)^T (\sum_{j=1}^{p} \mathbf{x}_j \tilde{\mathbf{v}}_j^T)) \\ &= \sum_{i=1}^{p} \sum_{j=1}^{p} tr(\tilde{\mathbf{v}}_i \mathbf{x}_i^T \mathbf{x}_j \tilde{\mathbf{v}}_j^T) \\ &= \sum_{i \notin \mathcal{I}} \sum_{j \notin \mathcal{I}} (\mathbf{x}_i^T \mathbf{x}_j \tilde{\mathbf{v}}_j^T \tilde{\mathbf{v}}_i). \end{aligned} \tag{3.3}$$

From equation (3.3), it is clear that $\forall i \in \mathcal{I}$, the data $\mathbf{x}_i$ from source $i$ has no projection on abnormal subspace and hence would be excluded from the statistics used for anomaly detection. We call such a pattern with an entire row with zeros *"joint sparsity"*.

Unfortunately ordinary PCA does not guarantee any sparsity in PCs. Sparse PCA is a recently developed algorithms where each PC is a sparse linear combination of the original sources [62]. 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 show the following example of anomaly detection and anomaly localization in network data streams. This example will be used in the following chapters as well.

We plot the normalized stock index streams of eight countries over a period of three months in Figure 3.1. 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.

we plotted the entries of each PC for ordinary PCA and for sparse PCA (figure 3.2) for the stock data set shown in figure 3.1. White blocks indicate zero entries and the darker color indicates a larger absolute loading. Sparse PCA produces sparse entries but that alone

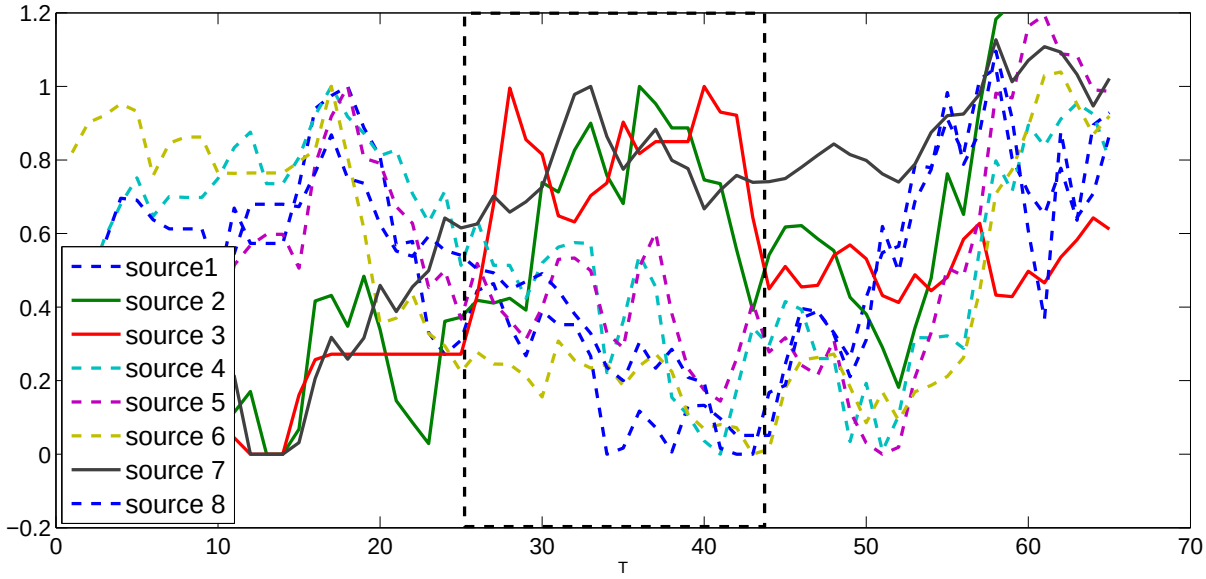


Figure 3.1: Illustration of time-evolving stock indices data

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.

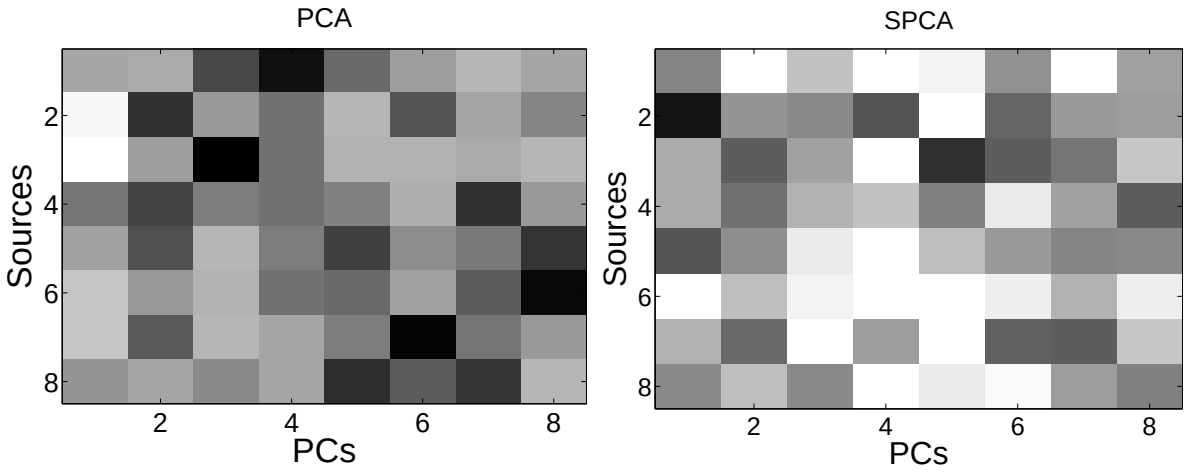


Figure 3.2: Comparing PCA and Sparse PCA.

# Chapter 4

# Sparse PCA for Anomaly Localization

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. Then we consider the network topology in the original data and incorporate such topology into JSPCA and develop an approach named Graph JSPCA (GJSPCA). We also 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.

## 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:

$$\begin{aligned} \min_{\mathbf{V}^{(1)},\mathbf{V}^{(3)}} \quad & \frac{1}{2}||\mathbf{X} - \mathbf{X}\mathbf{V}^{(1)}{\mathbf{V}^{(1)}}^T - \mathbf{X}\mathbf{V}^{(3)}{\mathbf{V}^{(3)}}^T||_F^2 + \lambda||\mathbf{V}^{(3)}||_{1,2} \\ \text{s.t.} \quad & \mathbf{V}^T\mathbf{V} = I_{p\times p}. \end{aligned} \tag{4.1}$$

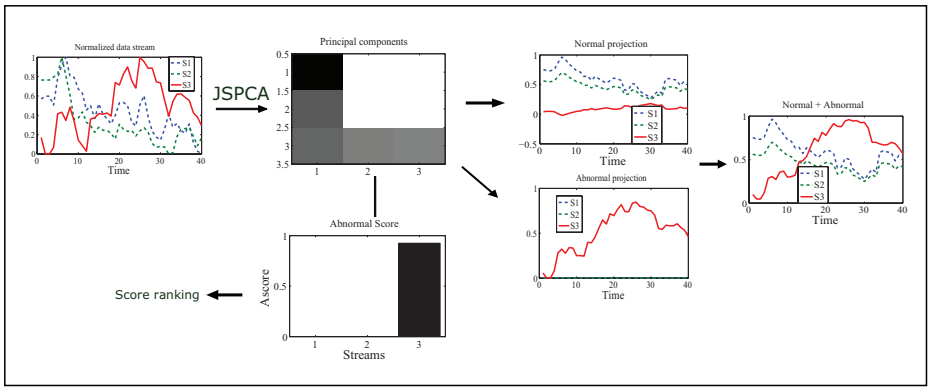


Figure 4.1: Demonstration of JSPCA on three network data streams with one anomaly (solid line) and two normal streams (dot lines).

Using one variable $\mathbf{V}$, we simplify equation (4.1) as:

$$\begin{aligned} \min_{\mathbf{V}} \quad & \frac{1}{2}||\mathbf{X} - \mathbf{X}\mathbf{V}\mathbf{V}^T||_F^2 + \lambda||\mathbf{W} \circ \mathbf{V}||_{1,2} \\ \text{s.t.} \quad & \mathbf{V}^T\mathbf{V} = I_{p\times p}. \end{aligned} \tag{4.2}$$

Here $\circ$ is the *Hadamard product* operator (entry-wise product), $\lambda$ 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:

$$\tilde{\mathbf{w}}_j = [\underbrace{0, \cdots, 0}_{k}, \underbrace{1, \cdots, 1}_{p-k}]^T, \quad j = 1, \cdots, p. \tag{4.3}$$

The regularization term $\|\mathbf{W} \circ \mathbf{V}\|_{1,2}$ is a $L_1/L_2$ penalty which enforces joint sparsity for each source across in the abnormal subspace spanned by the remaining $p-k$ principal components.

The major disadvantage of equation (4.2) 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 [62] with elastic net penalty [61], in which two variables and an alternative optimization algorithm were introduced. Here we share the first least square loss term but with a different regularization term. Motivated

by [62], we consider a relaxed version:

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

Where $\mathbf{A}, \mathbf{B} \in \mathcal{R}^{p\times p}$. The advantage of the new formalization is two folds: first, equation (4.4) is convex to each subproblem when fixing one variable and optimizing the other. As asserted in [62] disregarding the Lasso penalty, the solution of equation (4.4) 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 [26], which is similar to our formalization. But their structure is defined on groups and cannot be directly applied for anomaly localization.

Figure 4.3 demonstrates the principal components generated from JSPCA for the stock market data shown in figure 3.1. 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 3.1.

## 4.2 Anomaly Scoring

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

**Definition 4.2.1** *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 $i$th row*

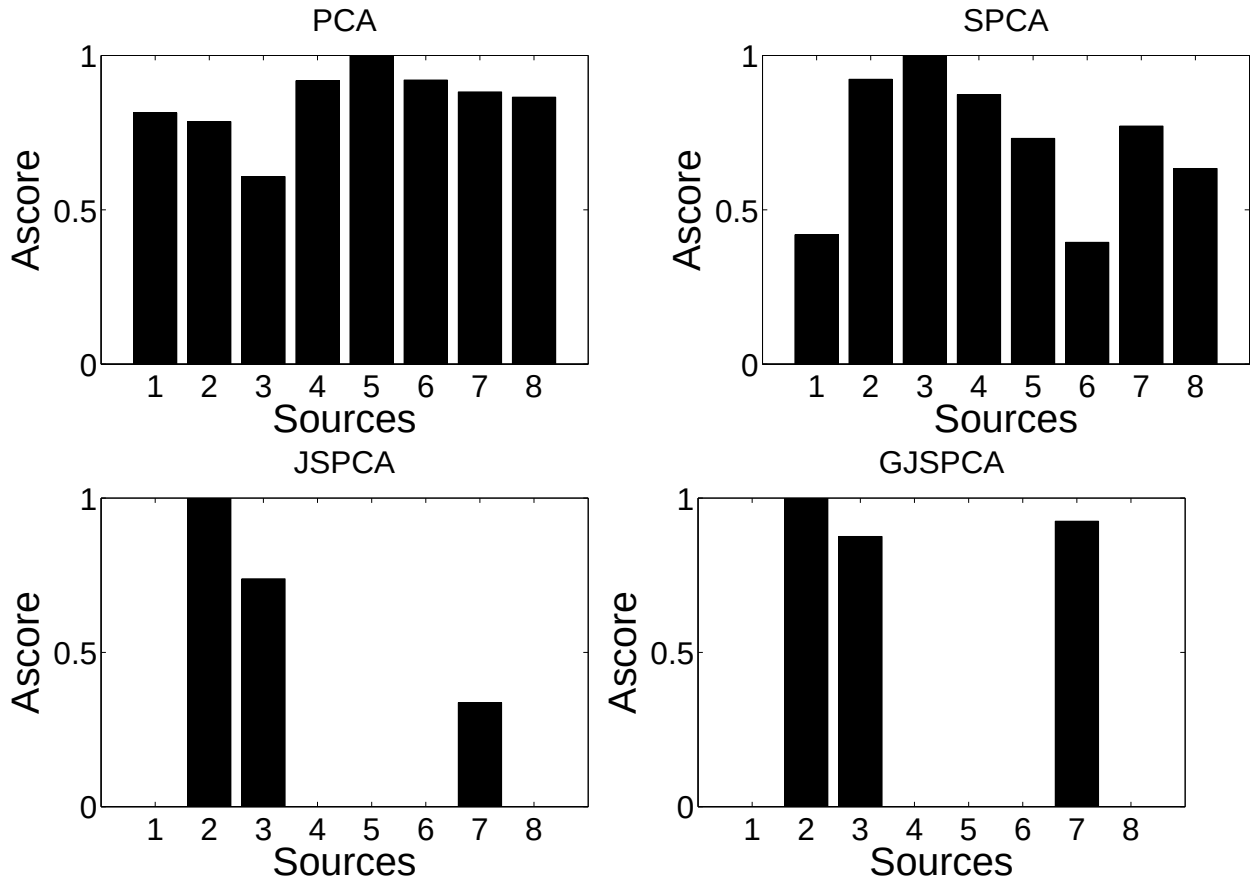


Figure 4.2: Comparing different anomaly localization methods. From left to right: PCA, sparse PCA, JSPCA, and GJSPCA.

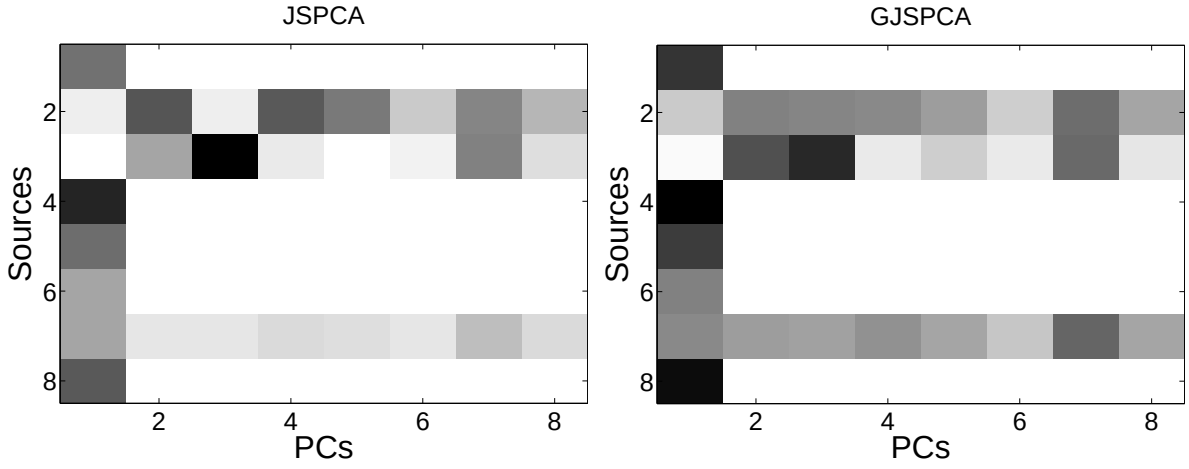


Figure 4.3: Comparing *joint sparse PCA* (JSPCA) and *graph joint sparse PCA* (GJSPCA).

*of* $\mathbf{V}^{(3)}$*, divided by the size of the row:*

$$\zeta_i = \frac{\sum_{j=k+1}^{p} |\tilde{v_{ij}}|}{p-k}, \tag{4.5}$$

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

*For each input data matrix $\mathbf{X}$, (4.5) 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 4.2. 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 1** Anomaly Localization with JSPCA

1: Input: $\mathbf{X}$, $k$ and $\lambda_1$.
2: Output: anomaly scores.
3: Calculate a set of PCs $\mathbf{V} = [\mathbf{V}^{(1)}, \mathbf{V}^{(3)}]$ (matrix $\mathbf{B}$ in equation (4.4)), $\mathbf{V}^{(1)}$ is normal subspace, $\mathbf{V}^{(3)}$ is abnormal subspace with joint sparsity;
4: Compute abnormal score for each source by the definition (4.2.1);

## 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 3.1, 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 Figure 4.3, 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 [12]. With this addition, we obtain the following optimization problem:

$$\begin{aligned} \min_{\mathbf{A},\mathbf{B}} \quad & \frac{1}{2}||\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}^T||_F^2 + \lambda_1\|\mathbf{W} \circ \mathbf{B}\|_{1,2} + \\ & \tfrac{1}{2}\lambda_2 tr((\mathbf{W} \circ \mathbf{B})^T L(\mathbf{W} \circ \mathbf{B})) \\ \text{s.t.} \quad & \mathbf{A}^T\mathbf{A} = I_{p\times p}, \end{aligned} \tag{4.6}$$

,

where $L$ is the *Laplacian* of a graph that captures the correlation structure of sources [12].

In Figure 4.3 we show the comparison of applying JSPCA and GJSPCA on the data shown in figure 3.1. 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 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 4.2. 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

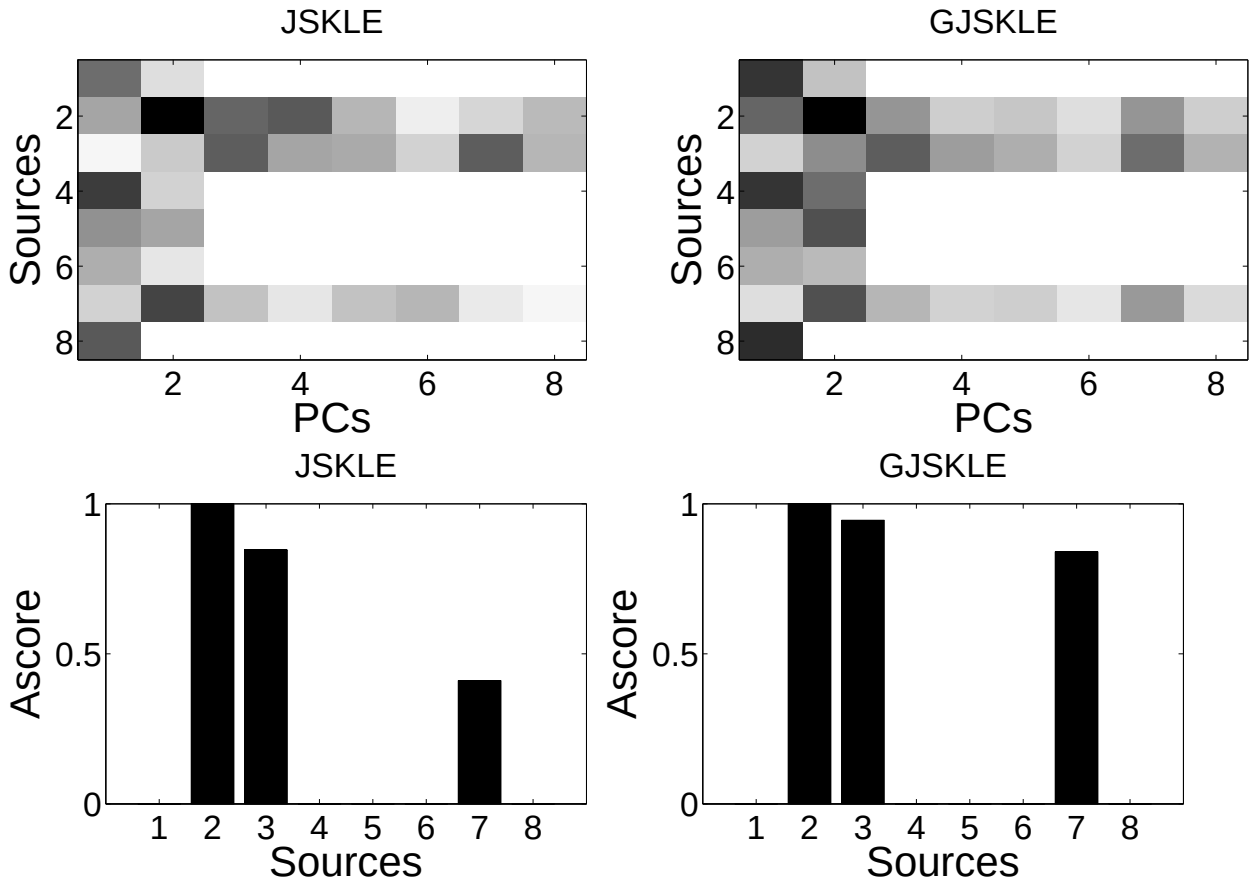


Figure 4.4: From left to right: PC space for JSKLE and GJSKLE, abnormal score for JSKLE, and GJSKLE.

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 Extension with Karhunen Loève Expansion

In this section, we extend our previous work with multi-dimensional discrete KLE. KLE was first considered as a representation of a stochastic process on an infinite linear combination of orthogonal functions [16], and usually named as continuous KLE. Later on, discrete KLE was then given [31] and the its one dimensional version (PCA) has been successfully applied to a broad domain of applications [32, 11]. The advantage of KLE over PCA is that KLE

takes both spatial and temporal correlation into consideration while PCA only considers the spatial correlation.

In [4], Brauckhoff et al. claimed that by extending PCA to KLE, they stabilized the anomaly detection performance and solved the sensitivity problem of PCA when changing the number of principal components representing the normal subspace [48]. Since JSPCA and GJSPCA are based on PCA, they both involve the same problem proposed in [48]. Therefore, we extend our regularization framework to KLE, called JSKLE and GJSKLE respectively towards the goal of stabilizing localization performance, which is illustrated in our experimental studies.

Generalize PCA to KLE amounts for expanding the original data matrix $\mathbf{X} \in \mathcal{R}^{n \times p}$ to $\mathbf{X}' \in \mathcal{R}^{(n-N+1) \times pN}$ in both spatial and temporal domain as follows:

$$\mathbf{X}'^{T} = \begin{bmatrix} x_1(1) & \cdots & x_1(t) & \cdots & x_1(n-N+1) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_1(N) & \cdots & x_1(t+N-1) & \cdots & x_1(n) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_p(1) & \cdots & x_p(t) & \cdots & x_p(n-N+1) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_p(N) & \cdots & x_p(t+N-1) & \cdots & x_p(n) \end{bmatrix} \tag{4.7}$$

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 [49]:

$$x(t) = \sum_{i=1}^{\infty} \alpha_i \psi_i(t) \tag{4.8}$$

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

orthogonal deterministic functions such that

$$\int_D \psi_i(t)\psi_j(t)dt = \delta_{ij}$$
$$\delta_{ij} = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases} \tag{4.9}$$

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

$$\int_a^b K_x(t,s)\psi_j(s)ds = \lambda_i\psi_i(t) \tag{4.10}$$

The uncorrelated random coefficients $\alpha_i$ 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

$$\mathbf{x} = [x(1), x(2) \dots x(n)]^T \tag{4.11}$$

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

$$\Gamma_{xx} = E(\mathbf{x}\mathbf{x}^T) \tag{4.12}$$

To estimate the covariance matrix $\Gamma_{xx}$, we use sliding window averaging algorithm as the covariance estimator [41]. 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 4.4.1** *Given a scaler time series* $\mathbf{x}$*, the estimate of covariance matrix* $\Gamma_{xx}$ *using a sliding window approach is defined as:*

$$\Gamma_{xx} = \sum_{i=1}^{n-N+1} \mathbf{x_i} \mathbf{x_i}^T \tag{4.13}$$

*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* $\Gamma_{xx}$*.*

The summation function in (4.13) 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 (4.11):

$$\mathbf{X^T} = \begin{bmatrix} x(1) & x(2) & \ldots & x(n-N+1) \\ x(2) & x(3) & \ldots & x(n-N+2) \\ \vdots & \vdots & \ddots & \vdots \\ x(N) & x(N+1) & \ldots & x(n) \end{bmatrix} \tag{4.14}$$

The integral equation (4.10) 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 $\psi_i$ 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 [49], 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 $i$th component $\mathbf{x}_i$ from the $i$th source takes the form in (4.11). Followed the equation (4.12), covariance matrix is defined as:

$$\Gamma_{XX} = E(\mathbf{X}\mathbf{X}^T) \tag{4.15}$$

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 (4.14) and its corresponding data matrix format in (4.14), we have the data matrix $X'$ for multi-dimensional KLE defined in (4.7). 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 (4.4) and (4.6) to expanded data matrix $X'$ 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 $B = [B_1^T, B_2^T, \cdots, B_p^T]^T$ where $B_i$ is a $N$ by $pN$ matrix. The top $k$ PCs of $B$ 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 $B$. More specifically, we consider the following optimization problem similar to the objective of JSKLE:

$$\begin{aligned} \min_{\mathbf{A},\mathbf{B}} \quad & \frac{1}{2}||\mathbf{X}' - \mathbf{X}'\mathbf{B}\mathbf{A}^T||_F^2 + \lambda_1 \sum_{j=1}^{p} ||\mathbf{W}_j \circ \mathbf{B}_j||_F \\ \text{s.t.} \quad & \mathbf{A}^T\mathbf{A} = I_{pN\times pN}, \end{aligned} \tag{4.16}$$

where $\mathbf{W}_j \in \{0,1\}^{N\times pN}$ is the $j$th matrix block of $\mathbf{W}^T = [\mathbf{W}_1, \mathbf{W}_2, \cdots, \mathbf{W}_p]$ similar to (4.3) 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:

$$\begin{aligned} \min_{\mathbf{A},\mathbf{B}} \quad & \frac{1}{2}||\mathbf{X}' - \mathbf{X}'\mathbf{B}\mathbf{A}^T||_F^2 + \lambda_1 \sum_{j=1}^{p} ||\mathbf{W}_j \circ \mathbf{B}_j||_F \\ & \tfrac{1}{2N}\lambda_2 tr((\mathbf{W} \circ \mathbf{B})^T P^T L P (\mathbf{W} \circ \mathbf{B})) \\ \text{s.t.} \quad & \mathbf{A}^T\mathbf{A} = I_{pN \times pN}, \end{aligned} \tag{4.17}$$

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 4.4, 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 (4.16) or (4.17)), the anomaly score for source $i, \quad i = 1 \cdots p$ is

$$\zeta_i = \frac{||\mathbf{V}_i^{(3)}||_1}{(p-k)N} \tag{4.18}$$

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

Abnormal scores computed by JSKLE and GJSKLE are shown in Figure 4.4. 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.

## 4.5 Optimization Algorithms

We present our optimization technique to solve equations (4.4), (4.6), (4.16) and (4.17) based on accelerated gradient descent [43] and projected gradient scheme [2]. Since (4.16) and (4.17) are similar to (4.4) and (4.6), our following discussion will focus on (4.4) and (4.6). The solutions for (4.16) and (4.17) can be obtained by the same procedure with only minor changes on calculating gradient and gradient projection.

Although equations (4.4) and (4.6) 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.

**A given B**: If $\mathbf{B}$ is fixed, we obtain the optimal $\mathbf{A}$ analytically. Ignoring the regularization part, equation (4.4) and equation (4.6) degenerate to

$$\begin{array}{ll} \underset{\mathbf{A}}{min} & \frac{1}{2}||\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}^T||_F^2 \\ \text{s.t.} & \mathbf{A}^T\mathbf{A} = I_{p\times p}. \end{array} \tag{4.19}$$

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

$$\begin{aligned} \mathbf{GB} &= \mathbf{U}\mathbf{D}\mathbf{V}^T \\ \hat{\mathbf{A}} &= \mathbf{U}\mathbf{V}^T. \end{aligned} \tag{4.20}$$

Solution in the form of Procrustes Rotation is widely discussed, see [62] for example for a detailed discussion.

**B given A**: If $\mathbf{A}$ is fixed, we consider equation (4.6) only since equation (4.4) is a special

case of equation (4.6) when $\lambda_2 = 0$, Now the optimization problem becomes:

$$\begin{aligned} \min_{\mathbf{A},\mathbf{B}} \quad & \frac{1}{2}||\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}^T||_F^2 + \lambda_1 \|\mathbf{W} \circ \mathbf{B}\|_{1,2} + \\ & \tfrac{1}{2}\lambda_2 tr((\mathbf{W} \circ \mathbf{B})^T L(\mathbf{W} \circ \mathbf{B})). \end{aligned} \tag{4.21}$$

Equation (4.21) 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(4.21)

$$f(\mathbf{B}) = \frac{1}{2}||\mathbf{X}' - \mathbf{X}'\mathbf{B}\mathbf{A}^T||_F^2 + \frac{1}{2}\lambda_2 tr((\mathbf{W} \circ \mathbf{B})^T L(\mathbf{W} \circ \mathbf{B})) \tag{4.22}$$

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 (4.22) is a convex and smooth function over $\mathbf{B}$ with Lipschitz continuous gradient 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 $1/L_i$:

$$\mathbf{B}_{i+1} = \mathbf{B}_i - \frac{1}{L_i}\nabla f(\mathbf{B}_i). \tag{4.23}$$

In [1, 43], it has shown that the gradient step equation (4.23) 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

$$f_{L_i}(\mathbf{B}, \mathbf{B}_i) = f(\mathbf{B}_i) + \langle \mathbf{B} - \mathbf{B}_i, \nabla f(\mathbf{B}_i) \rangle + \frac{L_i}{2}\|\mathbf{B} - \mathbf{B}_i\|_F^2 \tag{4.24}$$

Based on the relationship, we combine equations (4.24) and $R(B)$ together to formalize the

*generalized gradient update step*:

$$\begin{aligned} Q_{L_i}(\mathbf{B}, \mathbf{B}_i) &= f_{L_i}(\mathbf{B}, \mathbf{B}_i) + \lambda_1 ||\mathbf{W} \circ \mathbf{B}||_{1,2} \\ q_{L_i}(\mathbf{B}_i) &= \underset{\mathbf{B}}{\operatorname{argmin}}\ Q_{L_i}(\mathbf{B}, \mathbf{B}_i). \end{aligned} \tag{4.25}$$

The insight of such a formalization is that by exploring the structure of regularization $R(.)$ we can easily solve the optimization in equation (4.25), then the convergence rate is the same as that of gradient decent method. Rewriting the optimization problem in equation(4.25) and ignoring terms that do not depend on $B$, the objective can be expressed as:

$$q_{L_i}(\mathbf{B}_i) = \underset{\mathbf{B} \in \mathcal{M}}{\operatorname{argmin}} \left(\frac{1}{2}\|\mathbf{B} - (\mathbf{B}_i - \frac{1}{L_i}\nabla f(\mathbf{B}_i))\|_F^2 + \frac{\lambda_1}{L_i}||\mathbf{W} \circ \mathbf{B}||_{1,2}\right). \tag{4.26}$$

With ordinary first order gradient method for smooth problems, the convergence rate is $O(1/\sqrt{\epsilon})$ [43] where $\epsilon$ is the desired accuracy. In order to have a better convergence rate, we apply the Nestrerov accelerated gradient descent method [43] with $O(1/\sqrt{\epsilon})$ convergence rate, and solve the *generalized gradient update step* in equation (4.25) for each gradient update step. Such a procedure has demonstrated scalability and fast convergence in solving various sparse learning formulations [9, 27, 36]. 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. $10^{-5}$).

Now we focus on how to solve the generalized gradient update in equation (4.26). Let $\mathbf{C} = \mathbf{B}_i - \frac{1}{L_i}\nabla f(\mathbf{B}_i)$ and $\bar{\lambda} = \lambda_1 / L_i$, equation (4.26) can be represented as:

$$\begin{aligned} q_{L_i}(\mathbf{B}_i) &= \underset{\mathbf{B}}{\operatorname{argmin}} \left(\tfrac{1}{2}||\mathbf{B} - \mathbf{C}||_F^2 + \bar{\lambda}||\mathbf{W} \circ \mathbf{B}||_{1,2}\right) \\ &= \underset{\tilde{\mathbf{b}}_1, \cdots, \tilde{\mathbf{b}}_p}{\operatorname{argmin}} \ \textstyle\sum_{j=1}^{p} \left(\tfrac{1}{2}||\tilde{\mathbf{b}}_j - \tilde{\mathbf{c}}_j||_2^2 + \bar{\lambda}||\tilde{\mathbf{w}}_j \circ \tilde{\mathbf{b}}_j||_2\right) \end{aligned} \tag{4.27}$$

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 $j$th row of matrices $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{W}$. By the additivity of equation (4.27), we decompose equation (4.27) into $p$ subproblems. For

**Algorithm 2** 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. (4.26);
    **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 $\mathbf{B}$;

each subproblem, we ignore the row index $j$:

$$\min_{\mathbf{b}} \quad \frac{1}{2}||\mathbf{b}-\mathbf{c}||_2^2 + \bar{\lambda}||\mathbf{w}\circ\mathbf{b}||_2. \tag{4.28}$$

The following theorem provides the analytical solution of equation (4.28).

**Theorem 4.5.1** *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 (4.28)* $\mathbf{b}^* = [\mathbf{b}_1^{*T}, \mathbf{b}_2^{*T}]^T$ *is given by:*

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

*and*

$$\mathbf{b}_2^* = \begin{cases} (1-\frac{\bar{\lambda}}{||\mathbf{c}_2||_2})\mathbf{c}_2 & ||\mathbf{c}_2||_2 > \bar{\lambda} \\ 0 & \textit{otherwise.} \end{cases} \tag{4.29}$$

**Proof 4.5.1** *By the definition of the* $l_2$ *norm, the equation (4.28) can be rewritten as:*

$$\min_{\mathbf{b}_1,\mathbf{b}_2} \frac{1}{2}||\mathbf{b}_1-\mathbf{c}_1||_2^2 + \frac{1}{2}||\mathbf{b}_2-\mathbf{c}_2||_2^2 + \bar{\lambda}||\mathbf{b}_2||_2 \tag{4.30}$$

*where* $\mathbf{b} = [\mathbf{b}_1^T, \mathbf{b}_2^T]^T$. *The solution can be found by decomposing (4.30) 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* $\mathbf{b}_1^*$, *(4.30) degenerates to*

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

*The analytical solution of equation (4.31) is given in equation (4.29) and can be found by forming Lagrangian dual. A detailed proof can be found in [36].*

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 $\mathbf{B}^* = [\mathbf{B}_1^*, \cdots, \mathbf{B}_p^*]^T$ of (4.16) and

(4.17) given $\mathbf{A}$ is obtained:

$$\mathbf{B}_i^* = \begin{cases} (1 - \frac{\bar{\lambda}}{\sqrt{tr(\mathbf{C}_i\mathbf{C}_i^T)}})\mathbf{C}_i & \sqrt{tr(\mathbf{C}_i\mathbf{C}_i^T)} > \bar{\lambda} \\ 0 & \text{otherwise} \end{cases} \tag{4.32}$$

where $\mathbf{C}_i$ is the $i$th 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 (4.25), (4.27) in an extended data matrix and principal components.

**Algorithm 3** Graph Joint Sparse PCA (GJSPCA)

Input: $\mathbf{X}$, $k, \lambda_1, \lambda_2$ 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 Algorithm 2;
  Compute $\mathbf{A}$ given $\mathbf{B}$ via (4.20);
  **if** (Converge) **then**
    break;
  **end if**
**end for**
return $\mathbf{B}$;

We summarize what is briefly discussed previously for GJSPCA in the algorithm below. 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 (4.16), (4.17) and projected gradient given by (4.32).

Given data matrix $\mathbf{X} \in \mathcal{R}^{n\times p}$ and the number of PCs representing normal subspace $k$ and regularization parameters $\lambda_1, \lambda_2$, 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.

# Chapter 5

# Evaluation

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 of two state-of-the-art anomaly localization methods at the network level: stochastic nearest neighbor (SNN) [23] and eigen equation compression (EEC) [20] 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.

## 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 $\delta$ in that if the correlation between two sources is greater than $\delta$, 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 5.1, 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 5.1: 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 for anomaly localization, $Indices$: starting point and ending point of the intervals for anomaly localization, $W$: total number of data windows for anomaly localization, $W2$: total number of data windows for anomaly detection $L$: sliding window size, -: not applicable.

| D | $T$ | $p$ | $I$ | $Indices$ | $W1$ | $W2$ | $L$ |
|---|---|---|---|---|---|---|---|
| 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 | 1099 | 20 |
| D3 | 3000 | 20 | 1 | [1-1500] | 29 | 119 | 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 [8] where small turbulence and co-movements of most markets are considered as normal, dramatic price changes or 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 [29]. 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. Hence A data matrix with size $1500 \times 20$ are used for anomaly localization. For anomaly detection, we concatenate the data matrix for anomaly localization with all the 20 normal series to make a new data matrix with size $3000 \times 20$.

**KDDCup 99 Intrusion Detection Data Set:** The KDDCup99 intrusion detection

data set is obtained from UCI Repository [13]. The 10% training data set consisting of 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.

## 5.2 Experimental Protocol

### 5.2.1 Localization Model Construction

For each data set, a sliding window with length $L$ and offset $L/2$ is used to create multiple data windows. Each data window is a data matrix with size $L \times p$, in which $p$ is the dimensionality of network data streams. On Each data matrix, localization algorithms are run to generate an abnormal score vector $\tilde{\zeta} = [\tilde{\zeta}_1, \cdots, \tilde{\zeta}_p]^T$ with the size $1 \times p$. The $i$th entry of the score vector corresponds to the abnormal score of the $i$th source and represents the probability that the source is abnormal.

With the sliding windows moving forward, we generated $W1$ data windows and an abnormal score matrix with the size $W1 \times p$ was obtained as well finally. $W1$ is the number of data window for localization and $p$ is the number of sources. By comparing with a a cut-off threshold between $[0, 1]$, a localization prediction matrix with size $W1 \times p$ was obtained.

Each entry in the prediction matrix is 0 or 1 to indicate whether the source is normal or abnormal.

### 5.2.2 Detection Model Construction

For anomaly detection, we only tested on sensor dataset and motorcurrent dataset. Manually labeling the whole stock market dataset is a huge amount of work therefore we only zoomed in four abnormal intervals (as shown in table 5.1) and tested anomaly localization algorithms on them.

One should notice that when labeling the abnormal intervals for localization tests, each source is labeled as 1 or 0 in each time window, indicating the specific source is abnormal or normal. However, when testing the anomaly detection performance, each time window is labeled as 0 or 1, indicating the entire time window is normal or abnormal. This is because PCA based methods is used to detect whether there is anomaly existing in the whole network. The time window is labeled as 1 if there is one (or more than one) source(s) acts abnormal.

For anomaly detection test, a sliding window with length $L$ and offset $L/2$ is also used but in the entire dataset to create multiple data windows. In each data window, JSPCA is used to calculate a residual, which will be discussed in next section. For the entire dataset, a residual vector with size $W2$ is generated and then compared with a threshold to determine if these specific time windows are normal or abnormal.

### 5.2.3 Model Evaluation

We used the standard ROC curves and area under ROC curve (AUC) to evaluate anomaly localization and detection performance.

For anomaly localization, comparing the localization prediction matrix with the ground truth matrix 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 $W1$ windows. By changing the threshold from 0 to 1, we obtained the ROC curve and the AUC value.

To obtain ROC curve for anomaly detection, we changed the threshold from 0 to the maximum of residual and compared the threshold with the residual vector to get a series of FPR and TPR.

For network traffic data set, we evaluated our method in a qualitative way 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 category 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 on network traffic data set, we also compare our results with those obtained from other feature selection methods such as performance based ranking method (PBRM) and Support Vector Decision Function Ranking Method (SVDFRM) [42].

### 5.2.4 Parameter Selection

We have several parameters to tune when doing anomaly localization. However, the change of parameters has no effect on the performance of anomaly detection (except the number of principal components $k$ representing normal subspace) because the normal subspace has no regularization. Since the emphasis of our work is on anomaly localization, we do not analyze the sensitivity of detection results on $k$ but just choose the number $k$ which is best to do anomaly localization.

To do anomaly localization, we have two parameters to tune in JSPCA: $\lambda_1$: controlling the sparsity, and $k$: the dimension of normal subspace. GJSPCA has two more parameters: $\lambda_2$: controlling the smoothness, and $\delta$, the correlation threshold to construct the correlation graph. JSKLE and GJSKLE introduce one more parameter: the temporal offset $N$. 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

Table 5.2: Optimal parameters combinations on three data sets. J:JSPCA, GJ: GJSPCA, JK:JSKLE, GJK: GJSKLE. The best temporal offset is 2 for all data sets

| | $\lambda_1$ | | | | $k$ | | | | $\lambda_2$ | | $\delta$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data set | J | GJ | JK | GJK | J | GJ | JK | GJK | GJ | GJK | GJ | GJK |
| Stock | $2^{-3}$ | $2^{-4}$ | $2^{-3}$ | $2^{-4}$ | 1 | 1 | 2 | 2 | $2^{-4}$ | $2^{-4}$ | 0.6 | 0.5 |
| Sensor | $2^{-7}$ | $2^{-5}$ | $2^{-6}$ | $2^{-5}$ | 1 | 1 | 2 | 2 | $2^{-6}$ | $2^{-6}$ | 0.6 | 0.7 |
| Motor | $2^{-2}$ | $2^{-2}$ | $2^{-2}$ | $2^{-2}$ | 5 | 5 | 7 | 8 | $2^{-8}$ | $2^{-8}$ | 0.5 | 0.6 |

parameters and then compared the performance.

For each data set, we tuned $\lambda_1$, $\lambda_2$ within $\{2^{-8}, 2^{-7}, \cdots, 2^8\}$, $\delta$ 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. , $N$ was tuned from 2 to 5 for KLE based methods. 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 5.2 lists the best parameter combination for JSPCA, GJSPCA, JSKLE and GJSKLE. 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 $2 \sim 4$.

## 5.3 Anomaly Detection Performance

In this section, anomaly detection performance of JSPCA was evaluated on sensor dataset and motorCurrent dataset. The result is shown in figure 5.1. JSPCA does anomaly detection in the same way as PCA. Both of them use the first $k$ principal components as normal subspace. As we mentioned in previous chapter, in the normal subspace we did not add any regularization and hence JSPCA has the same normal subspace as ordinary PCA. Therefore its anomaly detection performance is the same (closed) as that of PCA.

JSPCA first calculated the principal component matrix: the first $k$ columns as ordinary

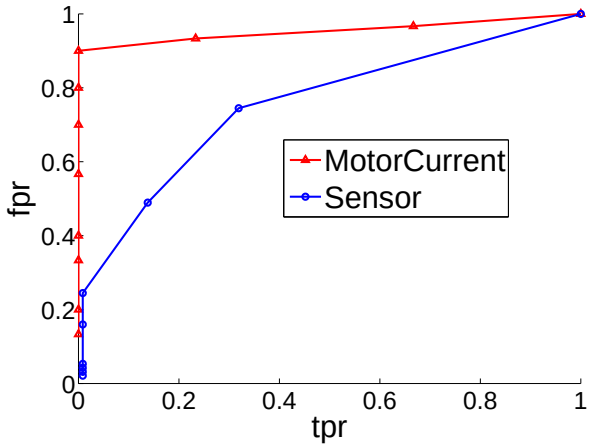


Figure 5.1: ROC curve for anomaly detection on sensor dataset and motorCurrent dataset. AUC for sensor dataset is 0.7832, for motorCurrent dataset is 0.9688

normal subspace and the left $n - k$ columns as joint sparse abnormal subspace. Then the normal subspace was used for anomaly detection . By projecting data onto normal subspace, the normal (modeled) component of data was extracted. The difference between original data and normal part of data is called the residual, corresponding to the anomalies and noise. Frobenius norm of residual(also called square prediction error (SPE)) was then calculated and taken as a useful statistic for detecting abnormal changes. A larger SPE indicates the higher probability there is an anomaly. Different thresholds were used to generate ROC curve, shown in figure 5.1.

GJSPCA constructs the same normal subspace as JSPCA, and hence the anomaly detection performance is the same and doesn't show here.

## 5.4 Anomaly Localization Performance

In Figure 5.2, we show the performances for 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 $[0.5 \sim 0.6]$. The first figure shows the four ROC curves for stock indices data. JSPCA and GJSPCA are comparable although GJSPCA is slightly better. SNN performs worst on

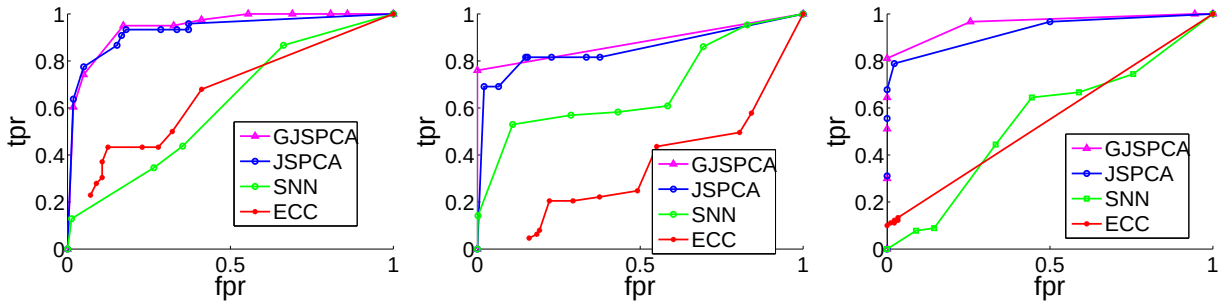


Figure 5.2: ROC curves and AUC for different methods on three data sets. From left to right: ROC for the stock indices data, ROC for the sensor data, ROC curve for MotorCurrent data

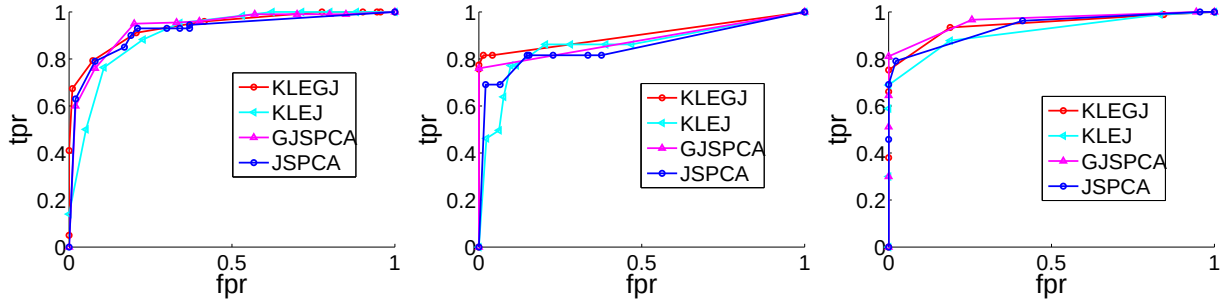


Figure 5.3: ROC curve for KLE extension methods on three data sets. From left to right: ROC for the stock indices data, ROC for the sensor data, ROC curve for MotorCurrent data

this dataset with around 0.5 AUC value. ECC outperforms SNN at first, but SNN catches up when FPR equals 0,63 and then stays better afterward. For the sensor dataset,SNN outperforms ECC with a 0.33 AUC difference. On the last dataset, motorcurrent dataset, SNN are ECC are comparable.

Both SNN and ECC are calculating abnormal score based on the deviation of neighborhood graph and the graph is constructed from covariance matrix. The reason they perform different in different dataset, in my mind, is ECC based on the assumption that the data is from Gaussian distribution and abnormal score is calculated from distribution while SNN has no assumption. When the specific dataset is approximately Gaussian distributed, ECC is able to have a better performance and wise versa.

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 | ECC | JSPCA | GJSPCA | JSKLE | GJSKLE |
|---|---|---|---|---|---|---|
| Stock Indices | 0.6119 | 0.6510 | 0.9216 | 0.9457* | 0.9277 | 0.9405 |
| Sensor | 0.6783 | 0.3491 | 0.8527 | 0.8798 | 0.8542 | 0.8883* |
| MotorCurrent | 0.5444 | 0.5515 | 0.9273 | 0.9601* | 0.9021 | 0.9494 |

Figure 5.4: AUC for different methods on three data sets

| F Value | JSPCA | GJSKLE |
|---|---|---|
| JSKLE | 0.03 | 1.19 |
| GJSPCA | 0.63 | 0.01 |

| p Value | JSPCA | GJSKLE |
|---|---|---|
| JSKLE | 0.8637 | 0.3363 |
| GJSPCA | 0.4721 | 0.9408 |

Figure 5.5: pairwise ANOVA testing

We also test our KL extension of localization methods: JSKLE, GJSKLE. In Figure 5.3, we show the performance of JSKLE and GJSKLE in comparison with JSPCA and GJSPCA with $N = 2$.

With **ANOVA** (analysis of variance) test on JSPCA and GJSPCA (figure 5.5), we found the F-ratio is 0.03 and p value is 0.8637. **ANOVA** test on GJSPCA and GJSKLE returns a F value as 0.01 and a p value as 0.9408. Both of the p values are not small enough to conclude statistical significance. Hence we conclude that KLE extension does not outperform PCA based methods on anomaly localization. However, KLE extension stabilizes localization performance, which will be shown in the section 5.6.

For the pairs with and without graph guided (JSPCA and GJSPCA, JSKLE and GJSKLE), both p value are smaller than 0.5. The probability that JSKLE and GJSKLE are different reaches 66% and that of JSPCA and GJSPCA is also above half. We conclude that structure information of network data streams improves localization performance to some degree.

As mentioned earlier, anomaly localization on the KDDCUP 99 intrusion detection data

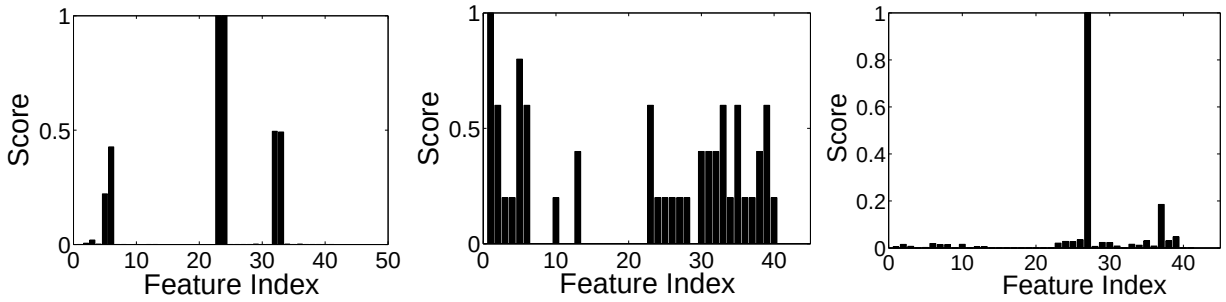


Figure 5.6: Anomaly Localization Performance of GJSPCA, Stochastic Nearest Neighborhood, Eigen-Equation Compression on Network Intrusion Data Set(DoS Attack)

| | Selected in common | Selected by JSPCA | Selected by PBRM | Selected by SVDFRM |
|---|---|---|---|---|
| DoS | 5,6,23,24,33 | NA | 1,3,8,19,25~28,32 35,36,38~41 | 1,25,26,32,36, 38,39 |
| Probe | 5 | NA | 3,6,23,24,32,33 | 1~4,6,23, 24,29,32,33 |
| U2R | 5,6,32,33 | 1 | 15,16,18 | 1~4,12,23,24 |
| R21 | 3,5,6,32,33 | 1 | 24 | 1 |

Figure 5.7: Most relevant features selected for different attacks

set performs as a feature relevant analysis. Localizing abnormal data streams amounts to identifying 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 5.3.

In Figure 5.6, 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 figure 5.6. 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 5.7, we summarize the most relevant features for each attack obtained by GJSPCA, and two feature ranking algorithms described [42]: performance based ranking method (PBRM) and Support Vector Decision Function Ranking Method

Table 5.3: 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 |

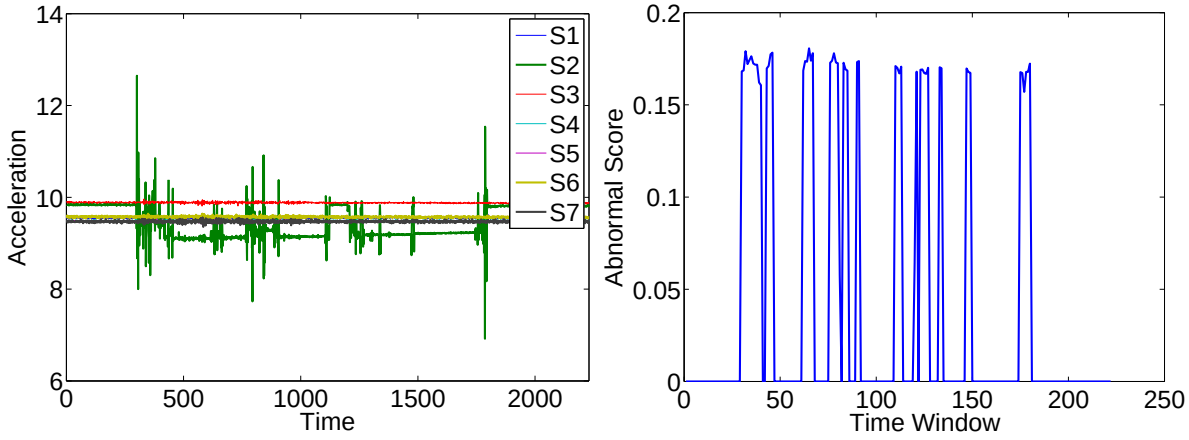


Figure 5.8: Left: original data in time interval $[2001, 3300]$ in sensor dataset. Right: time series of abnormal score calculated from left figure (with window size 20 and offset 10)

(SVDFRM). GJSPCA and the two baseline methods produce large consistent results.

## 5.5 Trend Analysis on Abnormal Score

Trend estimation is a statistical technique to aid interpretation of data. When a series of measurements of a process are treated as a time series, the trend can be used to make and justify statements about tendencies in the data. In particular, it is useful to determine if measurements exhibit an increasing or decreasing trend which is statistically distinguished

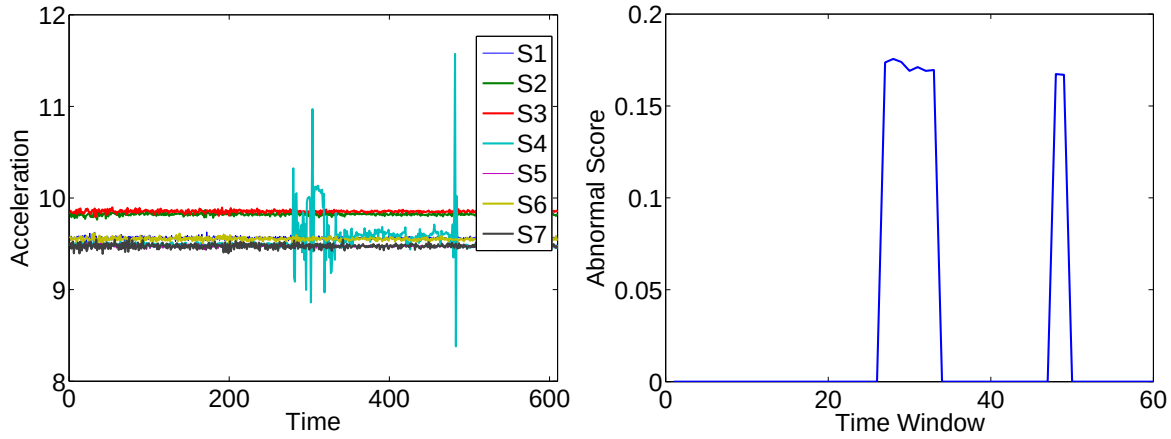


Figure 5.9: Left: original data in time interval $[7391, 8000]$ in sensor dataset. Right: time series of abnormal score calculated from left figure (with window size 20 and offset 10)

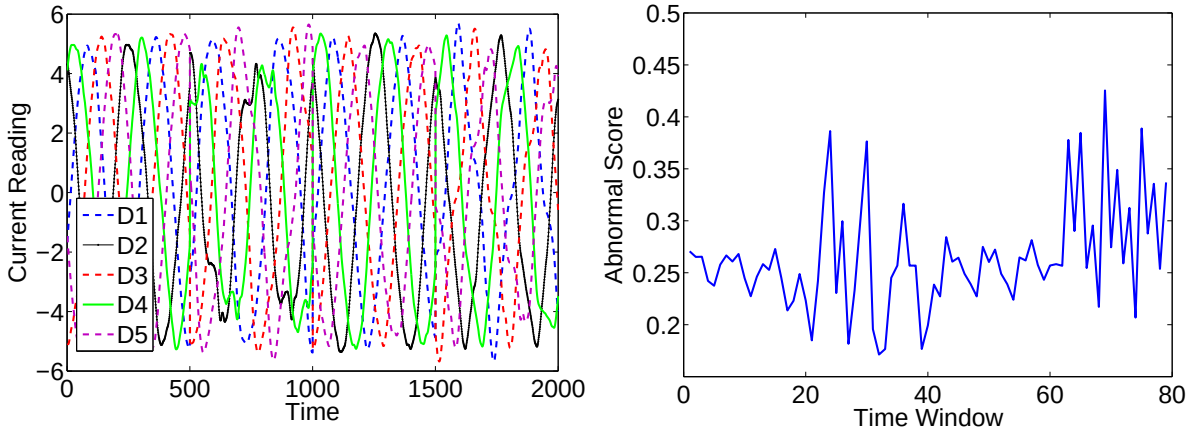


Figure 5.10: Left: original data in motorcurrent dataset. Right: time series of abnormal score calculated from left figure (with window size 50 and offset 25)

from random behavior.

By the definition of abnormal score in section 4.4, there is an abnormal score for each window. As the window moving forward, we have a set of abnormal scores. It is natural to view the abnormal score with different moving windows as a time series, in which each time point corresponds to a window. An interesting question is to identify the trend of abnormal score so that we can more deeply understand the abnormal events in data streams.

In Figure 5.8, we show the sensor data in time interval $[2001, 3300]$ ($S7$ is the abnormal source) and the abnormal score plot. From the Figure, we observe that when no anomalies happen, the abnormal score is always 0; however, once anomalies happen, the score suddenly

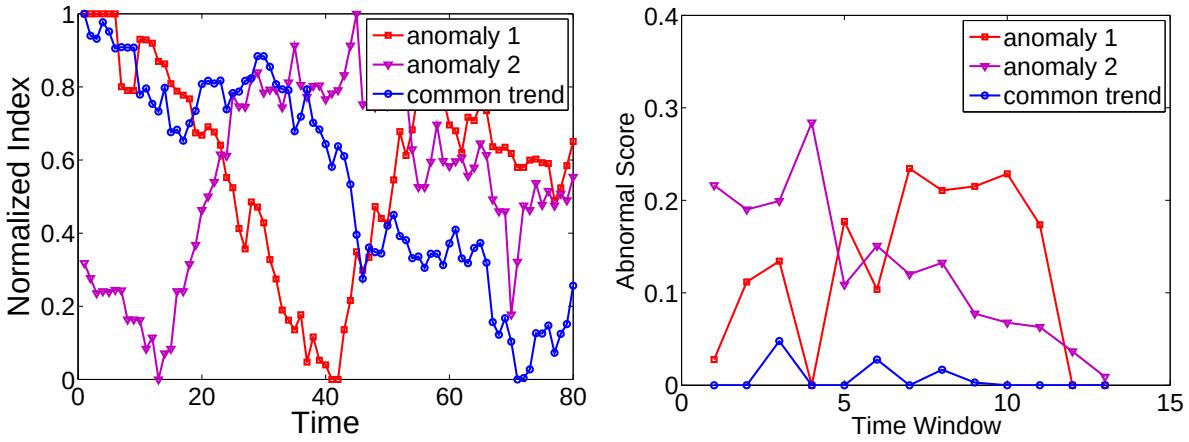


Figure 5.11: Left: original data in time interval $[341, 420]$ on stock market dataset. Right: time series of abnormal score calculated from left figure (with window size 20 and offset 5)

jumps to a high value. We carefully examined the trend in right panel of Figure 5.8 and found no significant increasing or decreasing trend. A possible reason is that the abnormal events that we simulated (e.g. removing a sensor out or flip a sensor) significantly deviate from normal patterns. All the abnormal events generate large abnormal scores and there is no clear trend. Similarly, we show another abnormal score plot in Figure 5.9 where $S_4$ is the anomaly. There is still no obvious trend on the abnormal score values.

In Figure 5.10, we demonstrate the original motor current data and the abnormal score plot. To better demonstrate the change of abnormal score, we rearranged the normal and abnormal time series to include both normal intervals and abnormal intervals. The abnormal intervals are $[500, 1000]$ and $[1500, 2000]$ and the abnormal sources are D2 and D4. From the Figure, we observe that the abnormal score is decreasing gradually and then increasing excluding a few spikes in the abnormal interval. An interesting pattern is that there is a sharp decreasing and then sharp increasing in the abnormal intervals. The explanation is that the abnormal events happened only at peak points (as shown in the left figure) and the rest readings are normal.

We show the similar study on stock index dataset. In Figure 5.11, the left is the original data and we use one representative normal index as common trend and the rest two are anomalies. The right panel shows the abnormal score. Obviously, when the other indices

are co-evolving as the common trend evolves; the abnormal score is low, otherwise high. In general, the trend of anomaly 2 is decreasing since it becomes more and closer to common trend while the trend for anomaly 1 is increasing though there is a sharp decreasing at the last point.

## 5.6 Sensitivity of Parameter

The performances of different methods depend on the 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 $\lambda_1$ and $\lambda_2$, $\delta$, $k$. Similar results are observed on the other two data sets.

In Figure 5.12, we show the stability by changing $\lambda_1$ in GJSPCA. We observe that AUC is quite stable over a wide range of $\lambda_1$. A similar phenomenon is also observed when changing $\lambda_2$. On the middle part of figure 5.12, we performed sensitivity analysis on parameter $\delta$. 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 $\delta$ 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 is $k$, the number of PCs spanning the normal subspace. In [48], Ringberg *et al.*claimed that the anomaly detection performance was very sensitive to $k$. From the right part of figure 5.12, it is clear that GJSPCA is still sensitive to the dimension of normal subspace. More specifically, the overall

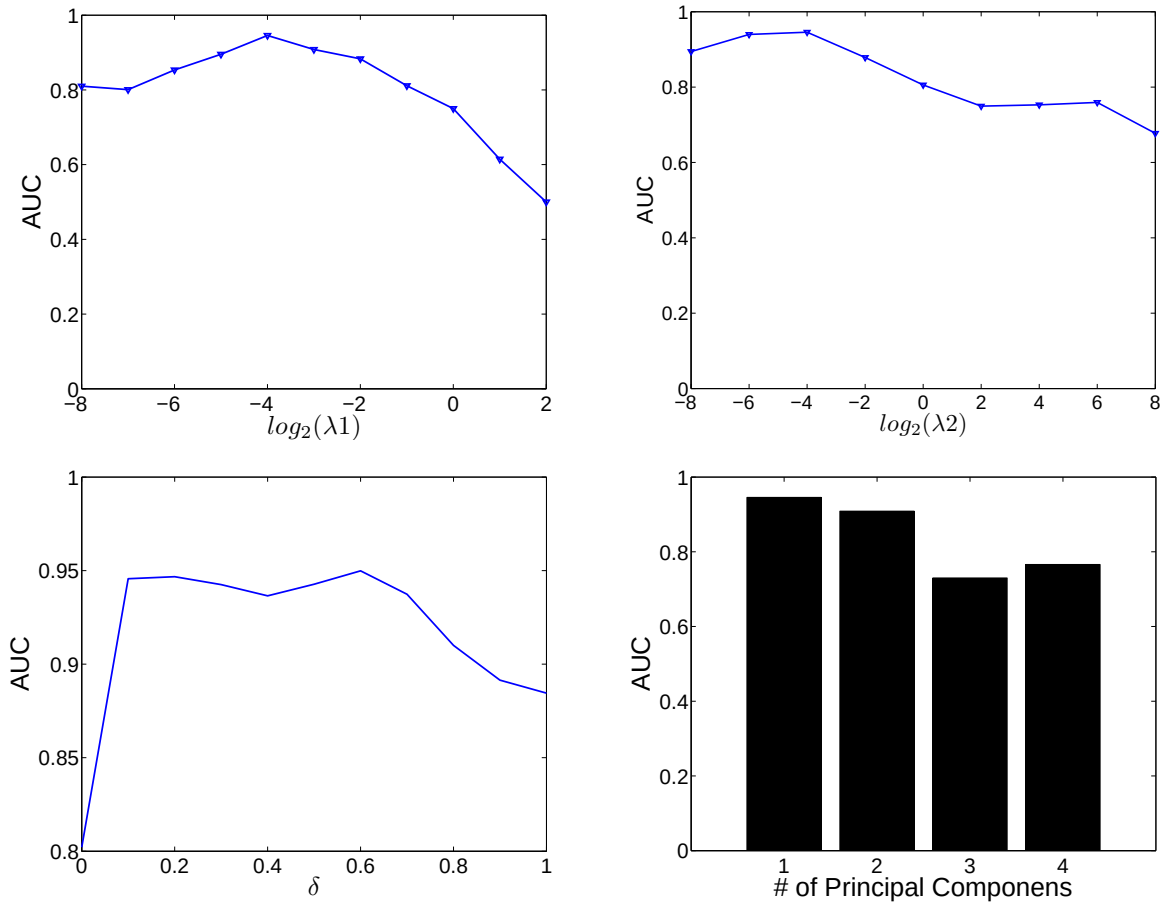


Figure 5.12: From left to right, sensitivity analysis of GSPCA on $\lambda_1$, $\lambda_2$, $\delta$, and the dimension of the normal subspace.

AUC gradually decreases from 0.96 to 0.72 as $k$ changes from 1 to 3 and then increases to 0.77 at $k = 4$. However even in the worst case $k = 3$ it still has a good performance with AUC= 0.73.

Figure 5.13 shows the parameters sensitivity of KL extension and demonstrates that the effectiveness of KLE to stabilize performance.. The most noticeable improvement is the sensitivity of $k$ shown in the last figure. Compared with PCA based method, GJSKLE successfully stabiles the localization performance when $k$ changes. For $k \in [1, 4]$, AUC remains above 0.9. Specifically, For $k = 2$, AUC has a 5% increase to 0.94, compared with the worst case 0.89 for $k = 4$. KL extension also has a considerable stabilizing effect on sensitivity with $\delta$ changing. When $\delta$ changes from 0 to 1 with step 0.1, AUC increases to

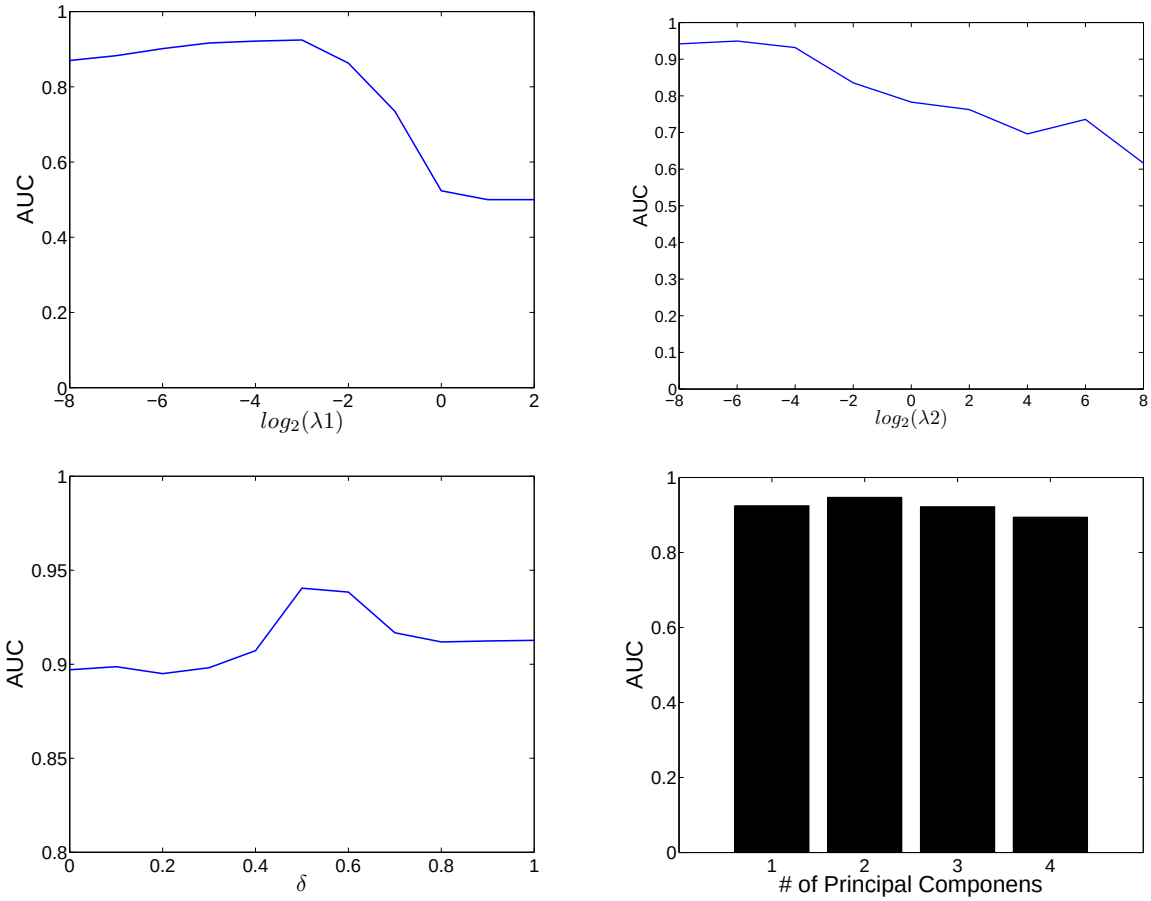


Figure 5.13: From left to right, sensitivity analysis of GJSKLE on $\lambda_1$, $\lambda_2$, $\delta$, and the dimension of the normal subspace.

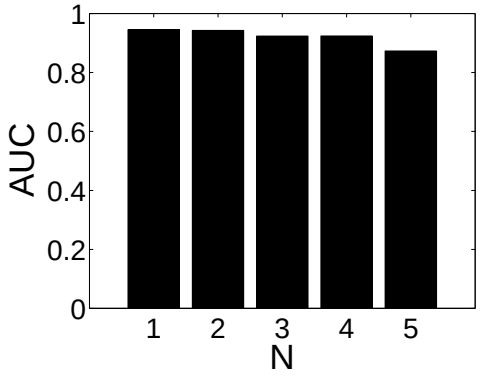


Figure 5.14: Sensitivity analysis of GJSKLE on N.

its optimum 0.94 at $\delta = 0.5$, and then decreases 3% to its minimum 0.91 at $\delta = 1$. The sharp decrease in the range $[0, 0, 1]$ and $[0.6, 1]$ in the last figure of 5.12 becomes much more

moderate.

JSKLE and GJSKLE involves one more parameter: the temporal correlation range $N$. To test the sensitivity of $N$, we repeated the experiments of KLE with different $N$ 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 5.14. With the changing 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 $\delta$ and dimension of normal space, as we can see the difference in last two figures of 5.13.

# References


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

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

[3] 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.

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

[5] 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.

[6] A. B. Chan, V. Mahadevan, and N. Vasconcelos. Generalized stauffer-grimson background subtraction for dynamic scenes. *Mach. Vis. Appl.*, 22(5):751–766, 2011.

[7] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):15:1–15:58, July 2009.

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

[9] X. Chen, W. Pan, J. T. Kwok, and J. G. Carbonell. Accelerated gradient method for multi-task sparse learning problem. In *ICDM*, pages 746–751, 2009.

[10] 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.

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

[12] 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.

[13] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[14] C. Franke and M. Gertz. Orden: outlier region detection and exploration in sensor networks. In *SIGMOD Conference*, pages 1075–1078, 2009.

[15] Z. Fu, W. Hu, and T. Tan. Similarity based vehicle trajectory clustering and anomaly detection. In *ICIP (2)*, pages 602–605, 2005.

[16] K. Fukunaga. *Introduction to statistical pattern recognition (2nd ed.)*. Academic Press Professional, Inc., San Diego, CA, USA, 1990.

[17] S. R. Gaddam, V. V. Phoha, and K. S. Balagani. K-means+id3: A novel method for supervised anomaly detection by cascading k-means clustering and id3 decision tree learning methods. *IEEE Trans. on Knowl. and Data Eng.*, 19(3):345–354, Mar. 2007.

[18] J. Gao, F. Liang, W. Fan, C. Wang, Y. Sun, and J. Han. On community outliers and their efficient detection in information networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, pages 813–822, New York, NY, USA, 2010. ACM.

[19] X. Gu and H. Wang. Online anomaly prediction for robust cluster systems. In *ICDE*, pages 1000–1011, 2009.

[20] 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, 2009.

[21] 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.

[22] T. Idé, A. C. Lozano, N. Abe, and Y. Liu. Proximity-based anomaly detection using sparse structure learning. In *SDM*, pages 97–108, 2009.

[23] 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.

[24] 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.

[25] S. Jakubek and T. Strasser. Fault-diagnosis using neural networks with ellipsoidal basis functions. In *Proceedings of the American Control Conference*, volume 5, pages 3846–3851, 2002.

[26] 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.

[27] 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.

[28] 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.

[29] E. Keogh and T. Folias. The ucr time series data mining archive. Website, 2002. `http://www.cs.ucr.edu/eamonn/TSDMA/index.html`.

[30] 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.

[31] D. Kosambi. Statistics in function space. 7:76–88, 1943.

[32] A. Lakhina, M. Crovella, and C. Diot. Diagnosing network-wide traffic anomalies. In *In ACM SIGCOMM*, pages 219–230, 2004.

[33] A. Lakhina, M. Crovella, and C. Diot. Mining anomalies using traffic feature distributions. In *In ACM SIGCOMM*, pages 217–228, 2005.

[34] J.-G. Lee, J. Han, and X. Li. Trajectory outlier detection: A partition-and-detect framework. In *ICDE*, pages 140–149, 2008.

[35] E. Leon, O. Nasraoui, and J. Gomez. Anomaly detection based on unsupervised niche clustering with application to network intrusion. In *In Proceedings of the congress of evolutionary Computation*, pages 502–508. Press, 2004.

[36] J. Liu, S. Ji, and J. Ye. Multi-task feature learning via efficient l2,1-norm minimization. In *Conference on Uncertainty in Artificial Intelligence (UAI) 2009*, 2009.

[37] 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.

[38] 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.

[39] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos. Anomaly detection in crowded scenes. In *CVPR*, pages 1975–1981, 2010.

[40] L. M. Manevitz and M. Yousef. Document classification on neural networks using only positive examples (poster session). In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '00, pages 304–306, New York, NY, USA, 2000. ACM.

[41] S. L. Marple. *Digital Spectral Analysis With Applications.* Prentice Hall, Australia, Sydney, 1987.

[42] 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.

[43] Y. Nesterov. Introductory lectures on convex optimization: A basic course. 2003.

[44] H. Nguyen, Y. Tan, and X. Gu. Pal: Propagation-aware anomaly localization for cloud hosted distributed applications. In *PST*, Cascais, Portugal, 2011.

[45] S. Pandit, D. H. Chau, S. Wang, and C. Faloutsos. Netprobe: a fast and scalable system for fraud detection in online auction networks. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 201–210, New York, NY, USA, 2007. ACM.

[46] R. Perdisci and G. Gu. Using an ensemble of one-class svm classifiers to harden payload-based anomaly detection systems. In *In Proceedings of the IEEE International Conference on Data Mining (ICDM06*, pages 488–498. IEEE Computer Society, 2006.

[47] C. Phua, V. Lee, K. Smith-Miles, and R. Gayler. A comprehensive survey of data mining-based fraud detection research. 2005.

[48] 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.

[49] C. A. Schenk and G. I. Schuëller. *Uncertainty Assessment of Large Finite Element Systems*, volume 24. Springer-Verlag, Berlin/Heidelberg/New York, 2005.

[50] R. Sekar, A. Gupta, J. Frullo, T. Shanbhag, A. Tiwari, H. Yang, and S. Zhou. Specification-based anomaly detection: A new approach for detecting network intrusions, 2002.

[51] 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.

[52] T. Stibor, J. Timmis, and C. Eckert. A comparative study of real-valued negative selection to statistical anomaly detection techniques. In *Proceedings of the 4th International Conference on Artificial Immune Systems, volume 3627 of LNCS*, pages 262–275. Springer, 2005.

[53] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319 – 2323, 2000.

[54] M. Tuba, D. Bulatovic, O. Miljkovic, and D. Simian. Specific attack adjusted bayesian network for intrusion detection system. In *Proceedings of the 9th WSEAS International Conference on Mathematics & Computers In Biology & Chemistry*, MCBC'08, pages 107–111, Stevens Point, Wisconsin, USA, 2008. World Scientific and Engineering Academy and Society (WSEAS).

[55] G. C. M. W. Weng-Keen Wong, Andrew Moore. Bayesian network anomaly pattern detection for disease outbreaks. In T. Fawcett and N. Mishra, editors, *Proceedings of the Twentieth International Conference on Machine Learning*, pages 808–815, Menlo Park, California, August 2003. AAAI Press.

[56] W.-K. Wong, A. Moore, G. Cooper, and M. Wagner. Rule-based anomaly pattern detection for detecting disease outbreaks. In *In Proceedings of the 18th National Conference on Artificial Intelligence*, pages 217–223. MIT Press, 2002.

[57] M. Xie, S. Han, B. Tian, and S. Parvin. Anomaly detection in wireless sensor networks: A survey. *J. Netw. Comput. Appl.*, 34(4):1302–1325, July 2011.

[58] N. Xu, S. Rangwala, and et al. A wireless sensor network for structural monitoring. In *IN SENSYS*, pages 13–24, 2004.

[59] 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.

[60] R. Zhang, S. Zhang, S. Muthuraman, and J. Jiang. One class support vector machine for anomaly detection in the communication network performance data. In *Proceedings of the*

*5th conference on Applied electromagnetics, wireless and optical communications*, ELECTROSCIENCE'07, pages 31–37, Stevens Point, Wisconsin, USA, 2007. World Scientific and Engineering Academy and Society (WSEAS).

[61] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society B*, 67:301–320, 2005.

[62] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, 2006.