Spectral Ranking and Unsupervised Feature Selection for Point, Collective and Contextual Anomaly Detection

by

Haofan Zhang

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2014

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Anomaly detection problems can be classified into three categories: point anomaly detection, collective anomaly detection and contextual anomaly detection [10]. Many algorithms have been devised to address anomaly detection of a specific type from various application domains. Nevertheless, the exact type of anomalies to be detected in practice is generally unknown under unsupervised setting, and most of the methods exist in literature usually favor one kind of anomalies over the others. Applying an algorithm with an incorrect assumption is unlikely to produce reasonable results. This thesis thereby investigates the possibility of applying a uniform approach that can automatically discover different kinds of anomalies. Specifically, we are primarily interested in Spectral Ranking for Anomalies (SRA) for its potential in detecting point anomalies and collective anomalies simultaneously. We show that the spectral optimization in SRA can be viewed as a relaxation of an unsupervised SVM problem under some assumptions. SRA thereby results in a bi-class classification strength measure that can be used to rank the point anomalies, along with a *normal* vs. *abnormal* classification for identifying collective anomalies. However, in dealing with contextual anomaly problems with different contexts defined by different feature subsets, SRA and other popular methods are still not sufficient on their own. Accordingly, we propose an unsupervised backward elimination feature selection algorithm BAHSIC-AD, utilizing Hilbert-Schmidt Independence Critirion (HSIC) in identifying the data instances present as anomalies in the subset of features that have strong dependence with each other. Finally, we demonstrate the effectiveness of SRA combined with BAHSIC-AD by comparing their performance with other popular anomaly detection methods on a few benchmarks, including both synthetic datasets and real world datasets. Our computational results jusitify that, in practice, SRA combined with BAHSIC-AD can be a generally applicable method for detecting different kinds of anomalies.

Acknowledgements

First and foremost, I would like to express my heartfelt gratitude to my supervisor, Professor Yuying Li, who have assisted me in every stage of completing this thesis. This thesis would not have been possible without her patience, insightful guidance, and gracious support. Thank you for believing in me.

I would also like to extend my appreciation and gratefulness to my readers, Professor Peter Forsyth and Professor Justin Wan, for taking their precious time to review this thesis. Lastly, my special thanks to every friend that I have met in the Scientific Computing Lab, Kai Ma, Eddie Cheung, Ke Nian, Aditya Tayal, Ken Chan, and Parsiad Azimzadeh for all the wonderful memories.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

The problem of *anomaly detection* is to find the data patterns that deviate from expected normal behavior in a given dataset [10]. The patterns that do not conform with normal pattern are generally referred to as *anomalies*, and the terms *outliers*, *novelties*, and *exceptions* are often used interchangeably in literature.

An enormous demand exists for anomaly detection mechanisms from a large variety of application domains, these include but not limited to detecting intrusion activities in network systems, identifying fraud claims in the health or automobile insurance, discovering malignant tumor in MSI image, and capturing suspicious human or vehicles from surveillance videos.

The economical value created by successful anomaly detection methods can also be significant. For instance, insurance fraud has been a severe problem in insurance industry for a considerably long time. While being difficult to estimate the exact loss due to insurance fraud, fraud cases are believed to account for around 10% of total adjustment expenses and incurred losses [27]. The situation is even more severe in certain subcategories. For automobile insurance, this figure goes up to 36% as reported in [14], however, only 3% among them are prosecuted. Since the fraud detection can be modeled as an anomaly de-

tection problem, substantial loss reduction can be achieved by effective anomaly detection algorithms.

Consider network intrusion detection system [40] [34] as another application of anomaly detection. Almost all contemporary web-based applications, and upper level facilities, require a secure networking infrastructure as their foundation. One important aspect of security is to prevent networking systems from malicious activities. The intrusions include any set of actions that threatens availability or integrity of networking resources. An effective anomaly detection method is evidently crucial for such a system, so that it can keep monitoring the network for possible dangerous misuse and abnormal activities. With anomalies being discovered, alarms can be raised for further actions.

Just as previous examples have shown, reasons for presence of anomalies are usually problem dependent. They can be pure noise introduced in data migration, or misrepresented information injected by people with malicious intension. However, despite their differences in actual causes, the main types of anomalies can be broadly categorized into three, i.e. *point anomalies*, *collective anomalies*, and *contextual anomalies* [10]. While many of ad-hoc methods proposed focus on a very specific problem, more studies focus on generic methods that can find a broad type of anomalies (e.g. point anomalies) instead. Although ad-hoc approaches can be more effective for a particular case, their success relies on a very good understanding about the nature of the problem. Since attempting to understand the cause of the anomalies, if not impossible, can pose additional complication to the study of the problem, the generic methods which can be applied to detection of different types of anomalies are thereby more desirable in general.

Most of existing anomaly detection methods adopt machine learning techniques, for the reason that machine learning methods are generally very powerful in terms of extracting useful data patterns from the problem with considerable size and complexity [24]. Depending on whether labels are required and how many labels are actually used, these machine learning methods can be further classified into supervised learning algorithms, semi-supervised learning algorithms, and unsupervised learning algorithms. Different from many other applications, where supervised learning normally plays the most important role, a large proportion of anomaly detection problems can only be formulated as unsupervised learning problems. This is primarily because of the practical difficulty in acquiring

labels for many real world applications. Although many unsupervised learning methods have been devised, we usually see strong assumptions made by these methods to detect only a specific type of anomaly. Under these assumptions, the results often favor one type of anomaly over the others. This makes it especially hard for users to choose appropriate unsupervised algorithm when the nature of problem to be addressed is not obvious.

Therefore, we are interested in a more general unsupervised learning method that can handle different kinds of anomalies at the same time. Based on the interpretation presented in [58], Spectral Ranking for Anomalies (SRA) proposed in [37] has the potential to tackle point anomalies and collective anomalies at the same time. Meanwhile, we notice how Hilbert-Schmidt Independence Criteria (HSIC) has the property of capturing arbitrary dependence relationships in a kernel space which can potentially be helpful in feature-contextual anomaly detection. Therefore, based on SRA and HSIC, this thesis proposes an unsupervised learning framework that has the flexibility to adapt to different types of anomaly detection problems with little tuning of parameters.

1.2 Thesis Contribution

This thesis first reviews anomaly detection problem in general by discussing three most common types of anomalies, namely point anomalies, collective anomalies and contextual anomalies. It then reviews prevailing machine learning approaches with a focus on unsupervised learning methods. Comments are made on advantages and limitations that are shared in common by these approaches.

The Spectral Ranking for Anomalies (SRA) proposed in [37] is investigated in greater details. In this thesis, we focus on the connection between SRA and unsupervised Support Vector Machine (SVM) as presented in [58]. We demonstrate how spectral optimization based on a Laplacian matrix can be viewed as a relaxation of the unsupervised SVM. Specifically, it can be interpreted, under reasonable assumptions, as a constant scaling-translation transformation of an approximate optimal bi-class classification function evaluated at given data instances. Based on this perspective, we justify how SRA has the potential to tackle point anomalies and collective anomalies at the same time by relating different settings of

SRA to different kinds of anomaly being detected.

We further observe limitations of SRA and other unsupervised methods in handling feature-contextual anomalies on their own. We thereby propose an unsupervised feature selection filter scheme, named BAHSIC-AD, based on Hilbert-Schmidt Independence Criteria (HSIC) for the purpose of identifying correct feature contexts of the contextual anomalies. By utilizing the property of HSIC, the proposed method can retain a subset of features that has strong dependence with each other in the implicit feature space. It thereby reconstructs the contexts for approaches like SRA to address the feature-contextual anomalies. With the insight we gain from unsupervised SVM and unsupervised feature selection, we discuss how SRA combined with BAHSIC-AD has the flexibility to handle all three kinds of anomalies with proper assumptions and appropriate problem formulations.

Computational results are presented to compare SRA and other approaches (with or without unsupervised feature selection) for different types of anomaly detection problems. Both synthetic data and real world dataset are utilized to evaluate the methods. We show that SRA can identify both point anomalies and collective anomalies simultaneously and HSIC helps reconstruct the contexts for detecting contextual anomalies. In addition, we take automobile insurance fraud detection as an example to illustrate how feature selection with HSIC also helps in improving the interpretability of the anomaly ranking results.

1.3 Thesis Organization

This thesis is organized as follows:

Chapter 2 provides the background about different types of anomaly detection problems and reviews the popular machine learning methods to address them.

Chapter 3 investigates the SRA algorithm with the perspective made in [58] which relates SRA with the unsupervised SVM problem. With the connection built with unsupervised SVM, it justifies how SRA has the potential in detecting both point anomalies and collective anomalies at the same time.

Chapter 4 proposes an unsupervised feature selection scheme based on HSIC to facilitate

SRA and other approaches in handling contextual anomalies with contexts being defined by feature subsets.

Chapter 5 compares the performance of SRA (with or without BAHSIC-AD) with other anomaly detection methods on both synthetic datasets and real world problems. We also justify how the algorithm improves the effectiveness and interpretability of the anomaly ranking results by studying its performance on an automobile insurance fraud detection dataset.

Chapter 6 concludes the thesis by highlighting the major contributions being made as well as potential directions for future exploration.

Chapter 2

Background

2.1 Types of Anomalies

Suppose we have a set of m training examples $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m\}$, where $\mathbf{x_i} \in \mathcal{X} \subseteq \mathbb{R}^d$, the goal of anomaly detection or anomaly ranking is to generate a ranking score $\mathbf{f} = \{f_1, f_2, \cdots, f_m\}$ for each example in $\mathcal{D}$ where higher value of f_i indicates the instance $\mathbf{x_i}$ more likely to be an anomaly.

As discussed in the survey of anomaly detection [10], the most common types of anomalies can be classified into three major categories, i.e. point anomalies, collective anomalies, and contextual anomalies. Point anomaly refers to the individual data instance that clearly deviates from the rest of the dataset. Collective anomalies, on the other hand, refer to the anomalous behavior revealed by a *group* of data instances. Point anomalies are the most common anomalies discussed and studied in anomaly detection literature whereas the collective anomalies is relatively less encountered but frequently emerged as a rare class classification problem. These two kinds of anomalies are discussed together in Section 2.1.1. Lastly, contextual anomaly refers to data instances that are anomalous in a certain context, and not otherwise. Note that, the definition of contextual anomaly requires a clear notion of "context" being defined and the definition of the contexts is crucial for anomalies to be identified. The contexts can be feature subset, data clusters etc. Also, being contextual

anomaly is not exclusive to other kinds of anomalies, as it is possible to have "contextual point anomalies" and "contextual collective anomalies". This thesis focuses on contextual anomalies with contexts being defined by feature subset, and we provide our discussions in Section 2.1.2.

2.1.1 Point Anomalies and Collective Anomalies

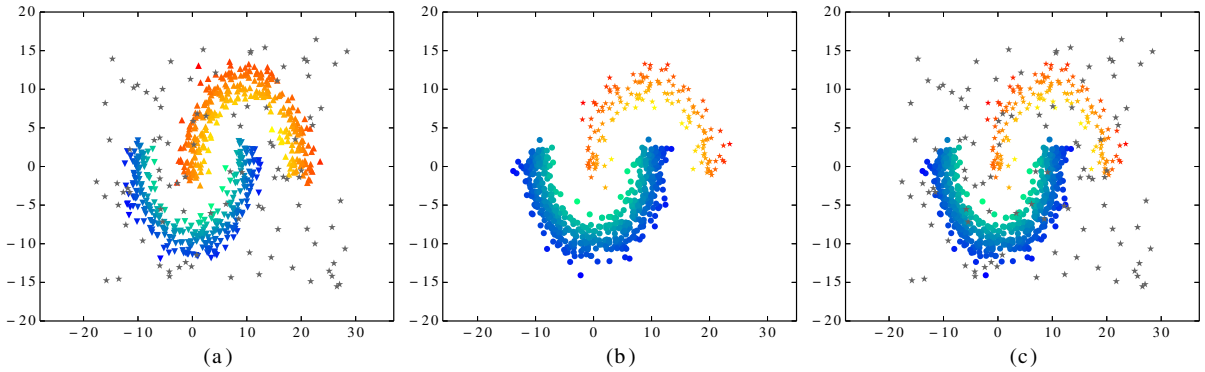

Figure 2.1: Examples of point anomalies (a), collective anomalies (b), and combination of both (c)

Examples of both point anomalies and collective anomalies are presented in Figure 2.1. Subplot (a) presents two balanced moon shape clusters, each consists of 500 points. There are additional 100 points (grey stars) uniformly scattered around the two moons which are clearly anomalies with respect to the two major patterns. Therefore, the grey star points can be treated as our examples of the point anomalies. Subplot (b), however, presents two *unbalanced* moon patterns. The lower moon (blue) consists of 1000 points in total and thereby has much higher mass and density compared with the upper moon (red), which is only of size 300. In this scenario, the lower moon forms the major pattern of the whole dataset. The individual points inside the red moon still lies inside the cluster, and thus cannot be treated as point anomalies. They however collectively form an anomalous pattern that deviates from the major pattern, i.e. the blue moon. This whole group of red points can then be treated as an example of collective anomalies. Subplot (c) shows a

combination of the two, where we have unbalanced patterns together with random scattered noise. The right subplot in the figure also shows the possibility for the presence of both kinds of anomalies in the same dataset.

2.1.2 Contextual Anomalies

Contextual anomalies is another type of anomalies that is frequently encountered in real world applications. Nevertheless, compared with point anomalies and collective anomalies, it is less studied in general because of the broad concept of "context". Within same dataset, different data instances can reveal distinctive anomalous behavior with different notion of "context". Indeed, a data cluster presents in the dataset can be a useful context and a specific feature subset can as well be a meaningful context. Therefore, a proper defined context is required if a reasonable anomaly ranking is expected. Most of successful approaches in literature indeed tended to be ad hoc or tailored for a particular kind of data such as time-series data [45] and spacial data [29] such that the notion of "context" is defined specific to the problem.

In this thesis, we focus on the *feature-contextual anomalies* with a reasonable assumption that different contexts of data correspond to different feature subset. These anomalies are also referred to as *conditional anomalies* in [51]. The feature-contextual anomalies actually emerge more frequent than people would normally expect. In many real world applications, when people construct the dataset, they normally tend to include features that are potentially relevant at the risk of introducing additional noise. However, this can compromise the performance of unsupervised anomaly detection algorithms when they simply treat all the features equally.

Consider the synthetic data presented in Figure 2.2 as an example of feature-contextual anomalies. Suppose we have the following data with three features as shown on the left side of Figure 2.2. The first two features are the noisy two moons which are very similar to the point anomaly dataset presented in Figure 2.1, whereas the third dimension is an additional noisy feature that we have injected into the original dataset. In this case, it is very difficult to identify red points as anomalies when we select subset $\{feature_1, feature_3\}$ or $\{feature_2, feature_3\}$, but they are clear anomalies when we only

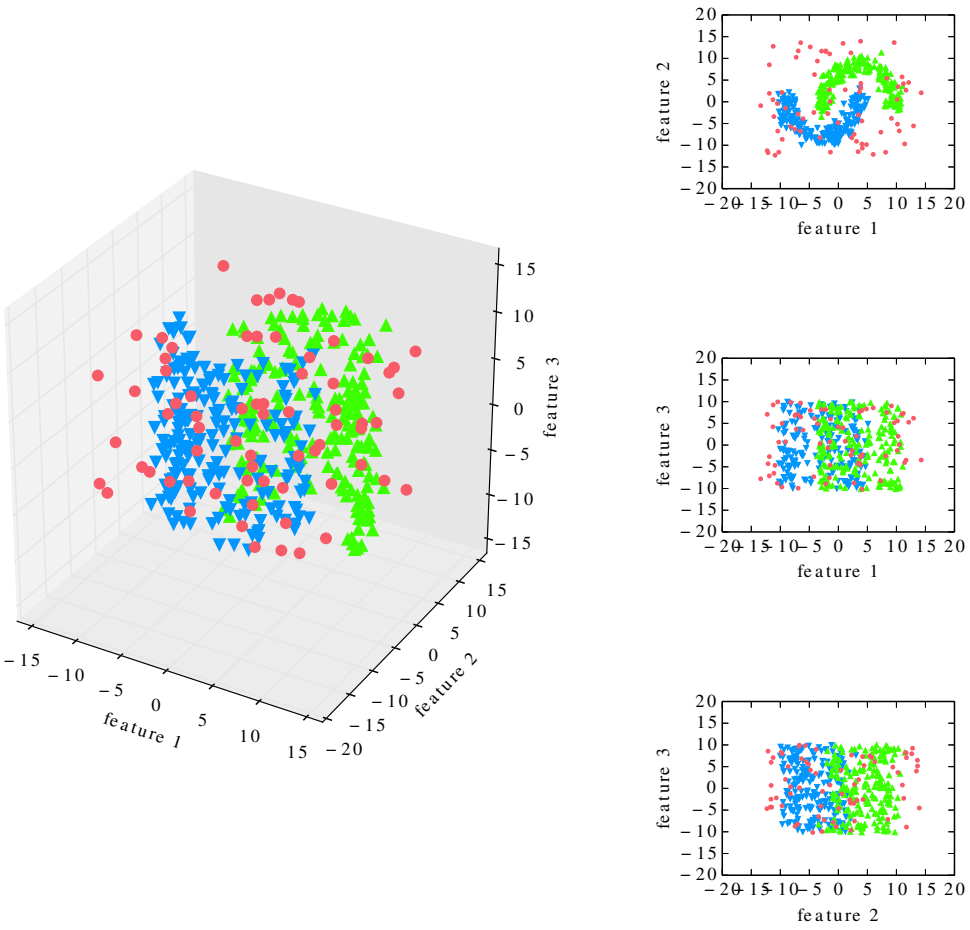

Figure 2.2: Example of feature-contextual anomalies defined by a feature subset: noisy two moons with an additional noisy feature

observe from $\{feature_1, feature_2\}$. Although the red points can still be identified with the full feature set, they are definitely not as clear when we observe from the first two dimensions. In this case, it is obvious that $feature_3$ adds no value in detecting the anomalies and the best feature contexts for anomaly detection is the subset features $\{feature_1, feature_2\}$.

2.2 Unsupervised Learning for Anomaly Detection

The existing machine learning approaches for anomaly detection in literature can be classified into three broad categories: supervised learning methods, unsupervised learning methods and semi-supervised learning methods. The difference among three categories lies in how many labeled training samples are utilized in the training process. Supervised learning usually requires a full labeled training set. Unsupervised learning, on the other hand, requires no labeled data instance in training. Lastly, semi-supervised operate on the dataset that has only limited number of labeled samples (e.g. only part of normal instances are labeled).

When the labels are actually available, it is generally preferable to apply supervised learning approaches, since the labels can provide additional information about the dependence relationship between features and the labels. Nevertheless, a very large number of anomaly detection problems are formulated as unsupervised learning problems instead of supervised learning problems. One important reason for the popularity of unsupervised learning is the implicit assumption made by most of anomaly detection methods [10]. Namely, the normal instances generally account for the majority of the dataset. Therefore, even without the labels, the pattern revealed by majority of the data can be considered as the normal class. Accordingly, many techniques designed for anomaly detection problems fall into the unsupervised learning category.

A more important reason for choosing unsupervised learning is the fact that the clean labeled training data are very scarce for many real world applications. The labels of the data can be difficult or even impossible to obtain due to practical limitations. Consider insurance fraud detection again as an example, people who commit fraud would normally deny their dishonest behavior unless strong evidence is presented. This also implies exis-

tence of a large portion of unidentified fraud cases in the historical data. Additionally, as time evolves, different types of anomaly emerge. Although supervised learning can best mimic human decisions, they however lack the capability in discovering novel patterns. This can potentially cause oversight of new kinds of fraud behavior. We thereby see the necessity of applying unsupervised learning techniques for these problems.

In the following subsection, we review different unsupervised learning approaches in Section 2.2.1 and the common problems they share in Section 2.2.2.

2.2.1 Existing Unsupervised Learning Methods

While there are numerous unsupervised learning methods designed for different tasks, here we review some of the most commonly used approaches along with their applications and assumptions behind them.

Nearest-Neighbor Based Methods

Nearest-Neighbor based methods are among most primitive methods to approach anomaly detection problems. The most basic example is the classical k-Nearest Neighbor (k-NN) global anomaly score. Given a set of training data, the k-NN algorithm finds the k data points that have the smallest distance to each of the data instance, and the score is assigned by either the average distance of the k nearest neighbors [59] [3] [6] or simply the distance to the k-th neighbor [9] [43]. The basic assumption is that, the data point with higher distance to its neighbors is more likely to be an anomaly and the normal instances generally lie closer to its neighbors. While being simple and intuitive, the effectiveness of k-NN methods however depend on the parameter k as well as an appropriate similarity or distance function. The choice of distance function is especially important to make k-NN feasible on the dataset with non-continuous features (e.g. nominal), and we note that several attempts [54] [38] have been made to address the issue .

Density Based Methods

Density based methods are very similar to nearest-neighbor based methods. They also rely on a notion of distance defined over the data and follow similar assumption as the nearest-neighbor based approaches that normal data instances lie in a dense neighborhood whereas

anomalous instances usually have a neighborhood with low density. However, instead of taking a global point of view as in nearest-neighbor based methods, density based methods generally only take local density into consideration.

The most commonly used density-based method is Local-Outlier Factor (LOF) as proposed in [8]. The local density of a data instance is calculated by first finding the volume of the smallest hypher-sphere that encompass its k-th nearest neighbors. The anomaly score is then derived by taking the average of the local density of its k-nearest neighbor and the local density of the instance itself. The instance in a dense region are assigned a lower score while the instances lie in the low density region will get higher score.

There are many variation of LOF methods that follow similar assumptions. The algorithm of Outlier Detection using In-degree Number (ODIN) simply assign the anomaly score as the inverse of the number of instances that have the given instance in their neighborhood [25]. An noticeable variation called Local Correlation Integral (LOCI) is proposed in [39]. LOCI claims to detect both point anomalies and a small cluster of anomalies at the same time. One final variation is the local outlier probabilities (LoOP)[30] which improves the interpretability of the ranking score by adopting a more statistically-oriented approach.

Clustering Based Methods

Clustering is one major stream of unsupervised learning research and many anomaly detection algorithms are built on top of existing clustering methods. The fundamental assumption behind most clustering based methods is that normal instances should form clusters while anomalies either do not belong to any cluster or lie far away from the closest cluster centroid. A few clustering methods have been proposed with the capability to exclude anomalies (noise) from clustering results, such as *Density-based spatial clustering of applications with noise* (DBSCAN) [18] and shared nearest neighbors (SNN) clustering [17]. They can also be applied to only identify the anomalies. However, since the methods are originally proposed for the purpose of clustering, they are generally not optimized for the purpose of anomaly detection.

Another clustering-based scheme is based on a two step process. It first applies an existing clustering algorithms (e.g. k-means [32], Self-Organizing Maps [28] or Hierarchical Clustering [35]) to obtain clusters in the data along with the calculated centroids, then the

anomaly scores are assigned as the distance to the closest cluster centroid.

One-Class Classification Based Methods

Anomaly detection problems can also be formulated as one-class classification problems. The basic assumption is that there exists only one class, i.e., the normal class, in the training set. The method then learns a boundary for the normal class, and classifies all the training instances outside the boundary as the anomalies. Examples of this category are the one-class Support Vector Machine (OC-SVM) [47] and one-class Kernel Fisher Discriminants [44], OC-SVM is especially popular for many applications. These methods usually utilize the kernel methods [48] so that they can be generalized to compute non-linear boundaries. Note however, it is not necessary for the training set to be truly one-class (every data instance comes from one class) for algorithms to produce reasonable results. For instance, after transforming the feature using kernel trick, the OC-SVM tries to find the smallest sphere enclosing the data in the space defined by kernel. The dissimilarity to the center of the sphere can then be utilized as the anomaly score.

2.2.2 Limitations of Existing Approaches

There are some common problems shared by the existing unsupervised learning methods in general. Successful unsupervised learning methods require a clear assumption made on the data. However, we see all the unsupervised approaches are based on the assumption that favors one kind of anomalies over the other, and most commonly, they favor towards the detection of point anomalies. This is especially true for most of clustering-based methods and density based methods. Assumptions of these methods generally ignore the possible existence of collective anomalies. Even methods, e.g. LOCI, that do take some special cases of collective anomalies into consideration, they are effective in cases that are specially addressed, such as micro-clusters formed by a very small group of anomalies.

Moreover, we notice that most unsupervised anomaly detection algorithms themselves are generally incomplete in dealing with feature-contextual anomalies. Consider again the example presented in Figure 2.2. Under unsupervised learning settings, the potential noisy feature can dramatically compromise the performance of these algorithms if they treated

all the features equally. In order to handle cases like this, it is necessary to introduce an unsupervised feature selection process whenever it is needed.

2.3 Receiver Operational Characteristic Analysis

Before we dive into the details of SRA, we review the Receiver Operational Characteristic (ROC) Analysis since this will be an important evaluation method for the forthcoming discussion in this thesis.

A ROC graph is a visualization tool for evaluating the performance of various classifiers. Since we are only interested in anomaly detection problems, we illustrate the concept under the settings for anomaly detection. We begin by considering an arbitrary anomaly detector $\mathcal{A}$. Essentially, $\mathcal{A}$ is a classifier that maps an input instance $\mathbf{x}$ to either positive class, being *anomaly*, or negative class, being *non-anomaly*. However, instead of output class membership directly, it is more often the case that $\mathcal{A}$ simply generate a continuous output (e.g., an estimate of the probability) indicating the likelihood of this instance being anomaly. Then a threshold is chosen to determine the class membership.

		Anomaly Detection Outcome		
		$\mathcal{A}(\mathbf{x}) = \mathbf{1}$	$\mathcal{A}(\mathbf{x}) = \mathbf{0}$	total
Actual Value	$\mathbf{y} = \mathbf{1}$	True Positive	False Negative	m^+
	$\mathbf{y} = \mathbf{-1}$	False Positive	True Negative	m^-

Figure 2.3: Example of confusion matrix

There are four possible outcomes with $\mathcal{A}$ and $\mathbf{x}$. More precisely, if $\mathbf{x}$ is anomaly, and

indeed classified as being anomaly, it is counted as *true positive*. If it is classified as non-anomaly, it is counted as *false negative*. Similarly, if $\mathbf{x}$ is not an anomaly, but mistakenly classified as an anomaly, it is counted as *false positive*. If it is correctly classified as non-anomaly, it is a *true negative*. These quantities are generally summarised in a confusion matrix as shown in the Figure 2.3.

We can then calculate the following metrics based on the classification result, i.e. *True Positive Rate (TP Rate)*,

$$\text{TP Rate} = \frac{\text{Anomalies correctly identified}}{\text{Total number of anomalies}}$$

and *False Positive Rate (FP Rate)*

$$\text{FP Rate} = \frac{\text{Anomalies incorrectly identified}}{\text{Total number of anomalies}}$$

If we vary the threshold, we can obtain different classification results. We thereby obtain a set of pairs of *TP Rate* and *FP Rate* correspond to different threshold values. By plotting the relationship between *TP Rate* and *FP Rate*, we obtain a ROC graph as shown in Figure 2.4.

For anomaly detection problems, if we change the threshold, we can include more data instances as anomalies, but at the risk of falsely including the normal cases. Therefore, we are usually interested in the trade-off between the benefits (higher *TP rate*) and costs (higher *FP rate*). This information can be obtained from the ROC graph, even without any prior knowledge about the actual costs due to misclassification. When we have all possible combinations of TP rate and FP rate, real world applications often require an optimal optimal operating point where we set the actual threshold in making the decision. This can be the point on the ROC curve that is closest to the ideal upper left-hand corner or simply the point corresponds to the maximum FP rate that we can possibly tolerate [52]. However, since selecting operating point is really problem dependent, we are more interested in a universal criterion to directly compare different ROCs.

In order to use a single scalar value to compare two or more ROCs generated by different anomaly detectors, it is natural to use the Area Under the ROC Curve (AUC) as the comparison criterion. Since the plot is on a unit square, the value of AUC will always

lie between 0 and 1.0, and a random guess will result in an 0.5 AUC. Although a ROC has a higher AUC is not necessarily better than one with a lower AUC in a certain region of the ROC plot, the value of AUC is generally a very reliable measure in practice. An important property of AUC is that, it is equivalent to the probability that the anomaly detector ranks a randomly chosen positive instance higher than a randomly chosen negative instance in the given dataset [19]. It is also equivalent to the U statistic in the Mann-Whitney U test as shown in [23].

The AUC will be our main evaluation criterion in Chapter 5, when we compare the performance of different anomaly detection methods.

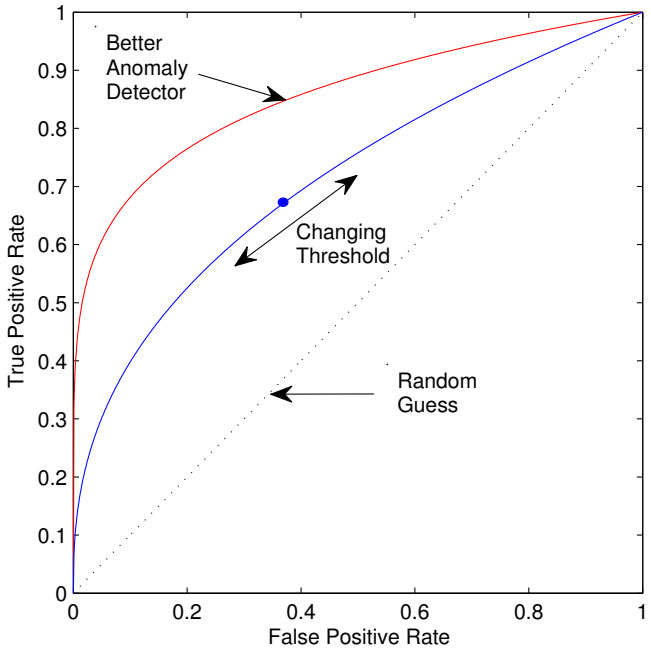

Figure 2.4: Examples of Receiver Operational Characteristic (ROC) curves

Chapter 3

Spectral Ranking for Point and Collective Anomalies

In this chapter, we analyze and discuss the algorithm of Spectral Ranking for Anomalies (SRA) as proposed in [37]. In Section 3.1, we present the SRA algorithm and discuss the motivation behind it. In Section 3.2, we analyze how spectral optimization based on the Laplacian matrix can be interpreted as a relaxation of an unsupervised SVM. Based on this connection between SRA and unsupervised SVM, we further justify effectiveness of SRA in handling point anomalies and collective anomalies in Section 3.3.

3.1 Spectral Ranking for Anomalies

3.1.1 Spectral Clustering

We start our discussion with a brief review on Spectral Clustering [53], which has motivated the SRA algorithm. Spectral clustering has gained its popularity in recent studies of clustering analysis. It has shown to be more effective than traditional clustering methods like k-means and hierarchical clustering. It is especially successful for applications like computer vision and information retrieval [49] [57] [15].

Suppose we have a set of m training examples $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m\}$, where $\mathbf{x_i} \in \mathcal{X} \subseteq \mathbb{R}^d$. The goal of spectral clustering is to group data instances into k groups so that data instances in each group are more similar to each other than to those in other groups. Successful spectral clustering relies on a notion of similarity defined over data instances which is provided in the form of a similarity matrix. We denote the given similarity matrix as $W \in \mathbb{R}^{m \times m}$ where W_{ij} is the similarity between instance $\mathbf{x}_i$ and instance $\mathbf{x}_j$. Note however that, the choices for the kernel and similarity measure is problem dependent and not the subject of this thesis. We refer interested readers to [37] for a more detailed discussion on these issues. For the convenience of later discussion, we also let the degree vector $\mathbf{d}$ be $\mathbf{d}_i = \sum_j W_{ij}, \;\; i = 1, 2, \cdots, m$, as well as D be the diagonal matrix with $\mathbf{d}$ on the diagonal.

The most important element for spectral clustering is the graph Laplacian matrix. There exist several variations of Laplacian matrices with different properties. The most popular ones include

- Unnormalized Laplacian [49]: $L = D - W$
- Random Walk Normalized Laplacian [12]: $L = I - D^{-1}W$
- Symmetric Normalized Laplacian [36]: $L = I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$

As discussed in [53], different variations of spectral clustering algorithms utilize different graph Laplacians. However, the main ideas of these algorithms are similar. Namely, they use graph Laplacians to change the representation of data so that it is easier to determine cluster membership in their new representations. In this thesis, we focus on the symmetric normalized Laplacian for majority of the discussion, and only briefly discuss unormalized Laplacian. Moreover, the symmetric normalized Laplacian is the primary graph Laplacian adopted by our SRA algorithm.

Following the spectral clustering algorithm in [36], an eigendecomposition is performed on the Laplacian matrix L. Assume that the derived eigenvectors are $\mathbf{g}_0^*, \mathbf{g}_1^*, \cdots, \mathbf{g}_{n-1}^*$ which are associated to the eigenvalues $\lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_{n-1}$ respectively. We then use the first k eigenvectors to construct a matrix $U \in \mathbb{R}^{n \times k}$ such that columns correspond to

the eigenvector $\mathbf{g}_0^*, \mathbf{g}_1^*, \cdots, \mathbf{g}_{k-1}^*$. After normalizing the rows of U to 1, we get a new set of representations of the original data instances. More specifically, the i-th row of normalized U is a new representation of $\mathbf{x}_i$ in the k dimensional eigenvector space. Finally, we can apply a traditional clustering algorithm, usually the k-means algorithm, to this new set of representation to figure out the cluster membership.

Note that, each non-principal eigenvector can be regarded as a solution to a relaxation of a normalized graph 2-cut problem. It finds a bi-class partition of the data in the space orthogonal to all previous $k-1$ eigenvector space. Therefore, spectral clustering can actually be interpreted as a k-step iterative bi-cluster classification method. A more rigorous and detailed discussion is provided in [53], along with other interpretations of spectral clustering.

3.1.2 Spectral Algorithm for Anomaly Detection

Inspired by spectral clustering, Spectral Ranking for Anomalies (SRA) has been proposed in [37] as a novel method to address anomaly detection problems. For practical applications like automobile insurance fraud detection where multiple patterns present as being normal, SRA has shown to be more effective than many traditional anomaly detection methods we have mentioned in Chapter 2, such as one class Support Vector Machine (OC-SVM), Local Outlier Factor (LOF), k-Nearest Neighbor(k-NN) etc. Same as spectral clustering, a similarity matrix is required to capture different characteristics of data and a symmetric normalized Laplacian is needed as the fundamental tool to generate the final ranking. We thereby follow the notation from previous section.

As mentioned in Chapter 2, the objective of anomaly ranking is to generate a ranking $\mathbf{f} = \{f_1, f_2, \cdots, f_m\}$ for each data instance in $\mathcal{D}$ where a higher value of f_i indicates the instance $\mathbf{x_i}$ more likely to be anomaly. Therefore, deciding the cluster membership is not as important as for clustering analysis and is insufficient for our purpose. However, as discussed in [37], we believe that the *first* non-principal eigenvector $\mathbf{g}_1^*$ actually has information beyond merely indicating memberships of data instances, and this information can be utilized for the purpose of anomaly ranking. Specifically, recall that spectral clustering can be interpreted as an iterative bi-cluster classification process. If we denote $\mathbf{z}^* = D^{\frac{1}{2}}\mathbf{g}_1^*$,

we can use $|\mathbf{z}_i^*|$ as a measure of how much data instance $\mathbf{x}_i$ contributes to the bi-class classification.

To better understand how the values of $|\mathbf{z}^*|$ can be helpful in the anomaly ranking, we consider the problem of getting the first non-principal eigenvector of L. It can be written as the following optimization problem:

$$\begin{aligned} \min_{\mathbf{g}\in\Re^n} \quad & \mathbf{g}^T L\mathbf{g} \\ \text{subject to} \quad & \mathbf{e}^T D^{\frac{1}{2}}\mathbf{g} = 0 \\ & \mathbf{g}^T\mathbf{g} = \upsilon \end{aligned} \tag{3.1}$$

where $\upsilon = \sum_{i=1}^n \mathbf{d}_i$.

Since $L = I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$ and if we now denote $\mathbf{z} = D^{\frac{1}{2}}\mathbf{g}$ and $K = D^{-1}WD^{-1}$, the objective function can be transformed in the following manner

$$\begin{aligned} \mathbf{g}^T L\mathbf{g} &= \mathbf{g}^T(I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}})\mathbf{g} \\ &= \mathbf{g}^T\mathbf{g} - (D^{\frac{1}{2}}\mathbf{g})^T(D^{-1}WD^{-1})(D^{\frac{1}{2}}\mathbf{g}) \\ &= \upsilon - \mathbf{z}^T K\mathbf{z} \end{aligned}$$

Therefore, if we ignore the constant υ, we have (3.1) in its equivalent form of

$$\begin{aligned} \min_{\mathbf{z}\in\Re^n} \quad & -\mathbf{z}^T K\mathbf{z} \\ \text{subject to} \quad & \mathbf{e}^T\mathbf{z} = 0 \\ & \mathbf{z}^T D^{-1}\mathbf{z} = \upsilon \end{aligned} \tag{3.2}$$

As discussed in [37], the objective function in (3.2) can be decomposed as

$$\text{sim}(\mathcal{C}_+) + \text{sim}(\mathcal{C}_-) - 2 \times \text{sim}(\mathcal{C}_+, \mathcal{C}_-)$$

where $\mathcal{C}_+ = \{j : \mathbf{z}_j \geq 0\}$, $\mathcal{C}_- = \{j : \mathbf{z}_j < 0\}$, $\text{sim}(\mathcal{C}) = \sum_{i,j\in\mathcal{C}} |\mathbf{z}_i||\mathbf{z}_j|K_{ij}$ measures similarity of instance in $\mathcal{C}$ ($\mathcal{C}$ can be either $\mathcal{C}_+$ or $\mathcal{C}_-$), and $\text{sim}(\mathcal{C}_+, \mathcal{C}_-) = \sum_{i\in\mathcal{C}_+, j\in\mathcal{C}_-} |\mathbf{z}_i||\mathbf{z}_j|K_{ij}$ measures similarity between $\mathcal{C}_+$ and $\mathcal{C}_-$. The value of the objective function can then be treated as a measure of the bi-class classification quality. Suppose the solution to (3.2) is

$\mathbf{z}^*$, the value of its i-th component $|\mathbf{z}_i^*|$ can be used as a strength measure for how much data instance $\mathbf{x}_i$ contributes to the quality of bi-class classification.

With the bi-class classification strength information provided by $|\mathbf{z}^*|$, we can generate the final rankings for anomaly depends on different scenarios which can possibly be encountered. The first case is when the data presents multiple major normal patterns. In this case, the data instances correspond to lower value of $|\mathbf{z}_i^*|$ are more likely to be anomalies since their memberships to different cluster are more ambiguous than others. Therefore, we can simply use $f(\mathbf{x}_i) = \max(|\mathbf{z}^*|) - |\mathbf{z}_i^*|$ as the ranking function for instance $\mathbf{x}_i$.

Another possible situation is when data instances are classified into two classes with normal class being actually clustered into one class and the rest data forms another class. This results in a *normal* vs. *abnormal* classification. In this case, the data instances that actually contribute most to the *abnormal* class are ranked higher. Depends on whether the number of data instances in $\mathcal{C}_+$ is higher than that of $\mathcal{C}_-$, we can use either $f(\mathbf{x}_i) = -|\mathbf{z}_i|$ or $f(\mathbf{x}_i) = |\mathbf{z}_i|$ to rank the anomalies.

To see the meaning of eigenvector more clearly, in next section we present a connection of spectral optimization (3.2) with unsupervised SVM.

3.2 SRA as a Relaxation of Unsupervised SVM

Before we illustrate how SRA can be used to detect point anomalies as well as collective anomalies, we further justify the use of eigenvector for anomaly ranking by illustration spectral optimization problem as an unsupervised SVM.

3.2.1 SVM Revisited

We first revisit the formulation of the standard *supervised* maximum margin SVM classifier. While there are other possible equivalent forms of SVMs, we mostly follow the formulation as in [46] and [60].

In a supervised bi-class classification problem, we are given a set of *labeled* training examples $\mathcal{D}' = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_n, y_n)\}$, where $\mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^d$ and $y_i \in \{+1, -1\}$. A

hyperplane in $\mathbb{R}^d$ is given by

$$h(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b = 0$$

A hyperplane is called a *separating hyperplane*, if there exists a c such that h satisfies

$$y_i(\mathbf{w}^T\mathbf{x} + b) \geq c \quad \forall i = 1, 2, \ldots, n$$

Moreover, by scaling $\mathbf{w}$ and b we can always get a *canonical separating hyperplane*, such that

$$y_i(\mathbf{w}^T\mathbf{x} + b) \geq 1 \quad \forall i = 1, 2, \ldots, n \tag{3.3}$$

Suppose two classes in the given dataset are perfectly separable by a hyperplane h, we then introduce the concept of *margin* (denoted as γ_h) of h, as twice the distance between h to its nearest data instance in $\mathcal{D}'$, i.e.

$$\gamma = 2 \times \min_{i=1,2,\ldots,n} y_i d_i \tag{3.4}$$

where d_i is the distance between data instance $\mathbf{x}_i$ to the hyperplane h. It can be easily shown that, the distance d_i is equal to

$$d_i = \frac{1}{\|\mathbf{w}\|}\left(\mathbf{w}^T\mathbf{x}_i + b\right)$$

where $\|\mathbf{w}\|$ is the Euclidean norm of $\mathbf{w}$. We can then rewrite the margin (3.4) as:

$$\gamma = 2 \times \min_{i=1,2,\ldots,n} y_i d_i = \frac{2}{||\mathbf{w}||}$$

A graphical illustration of margins, maximum margin and their corresponding hyperplanes is provided in Figure 3.1.

Intuitively, the best choice, among all hyperplanes that can separate two classes, is the one corresponds to the largest margin. Thereby, a linear *hard-margin* SVM tries to find the optimal hyperplane which corresponds to the maximal margin between two classes. This solves the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}\|\mathbf{w}\|^2, \\ \text{subject to} \quad & y_i\left(\mathbf{w}^T\mathbf{x}_i + b\right) \geq 1, \quad i = 1, ..., n, \end{aligned}$$

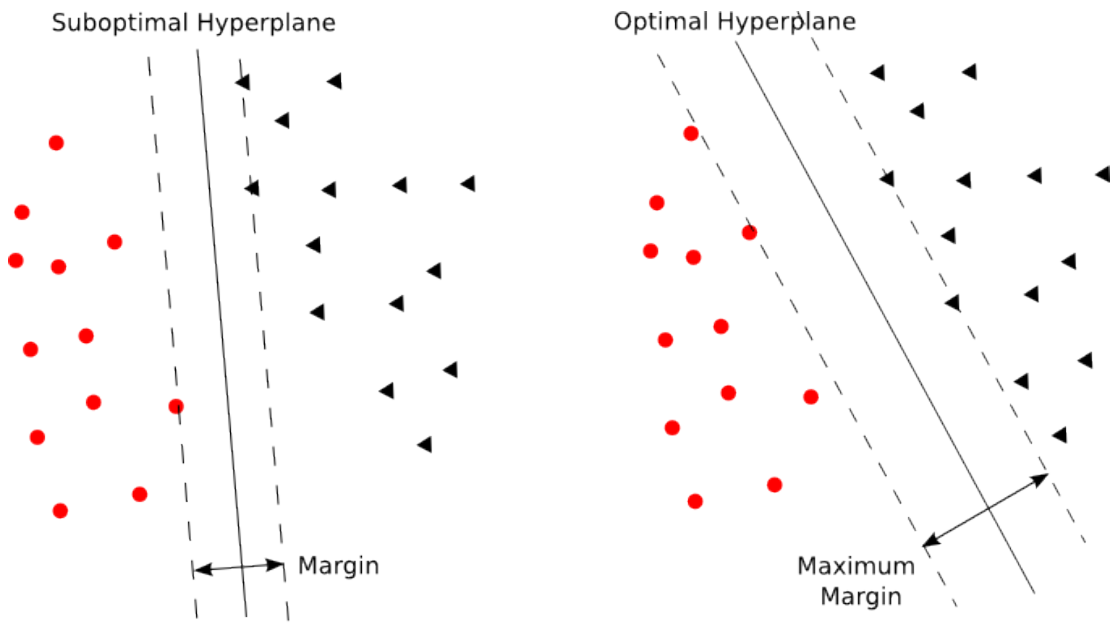

Figure 3.1: Example of margins and hyperplanes

However, a perfectly separable dataset is rare in practice. Therefore, we introduce slack variables ξ_i's to relax the separability condition in (3.3) when training instances are not linearly separable, and we have

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i, \\ \text{subject to} \quad & y_i\left(\mathbf{w}^T\mathbf{x}_i + b\right) \geq 1 - \xi_i, \quad i = 1, ..., n, \\ & \xi_i \geq 0, \quad i = 1, ..., n\,, \end{aligned} \tag{3.5}$$

where the regularization weight $C \geq 0$ is a penalty, associated with margin violations, which determines the trade-off between model accuracy and complexity. The optimal decision function then has the following form

$$h(\mathbf{x}) = \left(\sum_{j=1}^{n} y_j\alpha_j\mathbf{x}^T\mathbf{x}_j + b\right).$$

The SVM discussed so far is just a linear classifier, which has very limited power in many situations. The "kernel trick" is utilized to cope with more complicated cases. Suppose

we have $\phi : \mathcal{X} \mapsto \mathcal{F}$ which is a non-linear feature mapping from input space $\mathcal{X}$ to a (potentially infinite dimensional) feature space $\mathcal{F}$ derived from feature inputs. To find the optimal hyperplane in the feature space, we formulate a *kernel soft-margin* SVM, which solves the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i, \\ \text{subject to} \quad & y_i\left(\mathbf{w}^T\phi(\mathbf{x}_i) + b\right) \geq 1 - \xi_i, \quad i = 1, ..., n, \\ & \xi_i \geq 0, \quad i = 1, ..., n\,, \end{aligned} \tag{3.6}$$

with the optimal decision function

$$h(\mathbf{x}) = \left(\sum_{j=1}^{n} y_j \alpha_j^* \phi(\mathbf{x})^T \phi(\mathbf{x}_j) + b^*\right)$$

where (a^*, b^*) is a solution to (3.6).

Recall that the SVM problem (3.6) is a convex quadratic programming (QP) problem which satisfies the strong duality. This means that an optimal solution to (3.6) can be computed from its dual form

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & -\frac{1}{2}\sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) + \sum_{i=1}^{n}\alpha_i \\ \text{subject to} \quad & 0 \leq \alpha_i \leq C, \quad i = 1, \cdots, n \\ & \sum_{i=1}^{n}\alpha_i y_i = 0, \end{aligned} \tag{3.7}$$

By observing the dual problem (3.7), we notice that we can use the inner product $\phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$ in the objective function to solve the problem without explicityly knowing what ϕ is. In general, we can consider a *Kernel Function* $\mathcal{K} : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$ such that $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \langle\phi(\mathbf{x}_i), \phi(\mathbf{x}_j)\rangle = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j), \forall i, j = 1, 2, \cdots, n$. Accordingly, the n-by-n matrix K with $K_{ij} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$ is called a *Kernel Matrix*. Therefore, by simply utilizing different kernel, such as polynomial kernel

$$\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \left(\mathbf{x}_i^T\mathbf{x}_j + 1\right)^d$$

or Gaussian radial basis function (RBF) kernel

$$\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = e^{\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}}$$

we can find the optimal hyperplane in the implicit feature space induced by the corresponding kernel and thereby give SVM a lot more generality. Note that, the necessary and sufficient condition for $\mathcal{K}$ to be a valid kernel (also called a Mercer kernel) is that the corresponding kernel matrix K is symmetric positive semidefinite for any $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$ with any n [26]. We assume $\mathcal{K}$ is a valid kernel for all upcoming discussions.

Finally, we denote $Y = \text{DIAG}(\mathbf{y})$, and the dual problem of a SVM (3.7) with an (possibly) non-linear kernel can be rewritten into the following matrix form:

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & -\frac{1}{2}\boldsymbol{\alpha}^T Y K Y \boldsymbol{\alpha} + \mathbf{e}^T \boldsymbol{\alpha} \\ \text{subject to} \quad & 0 \leq \alpha_i \leq C, \quad i = 1, \cdots, n \\ & \mathbf{y}^T \boldsymbol{\alpha} = 0 \end{aligned} \tag{3.8}$$

3.2.2 Unsupervised SVM

For unsupervised SVM learning, we are given the data instances without labels. The goal then becomes finding the optimal label assignment for dataset such that the resultant hyperplane from supervised SVM has the maximal margin. Figure 3.2 gives an intuitive graphical illustration about how different label assignments can affect the maximum margin found by SVM.

Specifically, an unsupervised SVM is to find the labels $\mathbf{y}$ so that the objective value in (3.6) is minimum. Formally, this solves the following nested minimization problem:

$$\min_{y_i \in \{\pm 1\}} \left\{ \min_{w, \xi, b, y_i(\mathbf{w}^T \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \xi_i \geq 0} \frac{1}{2}\|\mathbf{w}\|_2^2 + C \sum_{i=1}^{n} \xi_i \right\} \tag{3.9}$$

Due to the integer constraints on y_i, we note that (3.9) is a NP-hard problem.

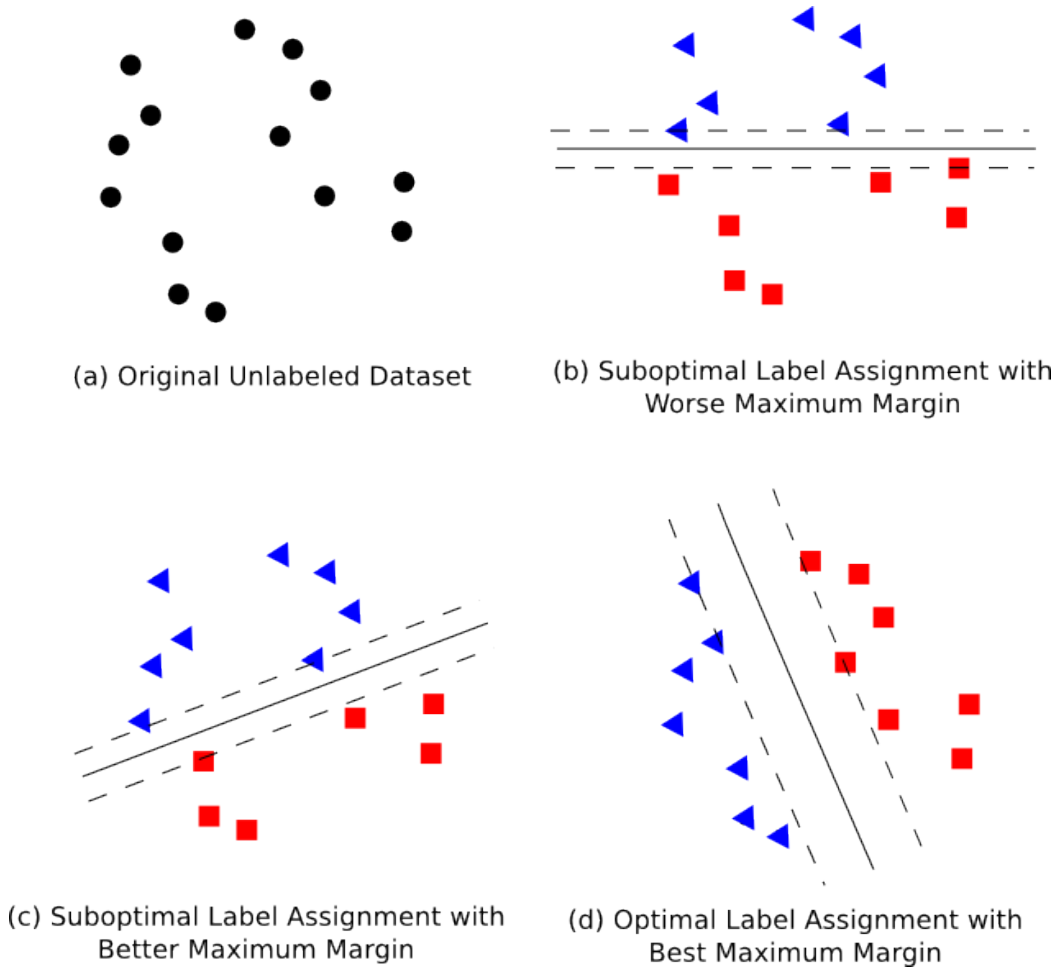

Figure 3.2: Example of different label assignments and resultant margins

Since we know the inner convex optimization problem satisfies strong duality, we can replace it by its dual problem and get the following equivalent minmax problem

$$\min_{y_i\in\{\pm 1\}} \max_{\substack{0\le\alpha_i\le C\\ \mathbf{y}^T\boldsymbol{\alpha}=0}} \quad -\frac{1}{2}\boldsymbol{\alpha}^T Y K Y \boldsymbol{\alpha} + \mathbf{e}^T\boldsymbol{\alpha} \tag{3.10}$$

Recall our discussion in previous section about supervised SVM, we now introduce another transformation of (3.8) as this will useful for the forthcoming discussions. If we introduce vector $\mathbf{z} \in \mathbb{R}^n$ such that

$$z_i = \alpha_i \cdot y_i, \qquad i = 1, \cdots, n$$

we have

$$\boldsymbol{\alpha}^T Y K Y \boldsymbol{\alpha} = \mathbf{z}^T K \mathbf{z}, \qquad \text{and} \qquad \mathbf{y}^T\boldsymbol{\alpha} = \mathbf{e}^T\mathbf{z}$$

Moreover, for any $\alpha_i \neq 0$, we have

$$y_i = \text{sign}(z_i), \qquad i = 1, \cdots, n \tag{3.11}$$

which also implies

$$\mathbf{e}^T\alpha = \mathbf{e}^T|\mathbf{z}|$$

Therefore, the optimization problem in (3.8) is also equivalent to

$$\begin{aligned} \max_{\mathbf{z}} \quad & \mathbf{e}^T|\mathbf{z}| - \frac{1}{2}\mathbf{z}^T K \mathbf{z} \\ \text{subject to} \quad & \mathbf{e}^T\mathbf{z} = 0, \\ & |\mathbf{z}| \le C \end{aligned} \tag{3.12}$$

We however notice the objective function in (3.12) is no longer concave, and it has many local maximizers. Since (3.12) is equivalent to the dual of the inner optimization problem in (3.10). Therefore, (3.10) can also be written as

$$\min_{y_i=\text{sign}(z_i)} \max_{\substack{\mathbf{e}^T\mathbf{z}=0\\ |\mathbf{z}|\le C}} \quad \mathbf{e}^T|\mathbf{z}| - \frac{1}{2}\mathbf{z}^T K \mathbf{z} \tag{3.13}$$

Now consider the following problem with a rectangular constraint

$$\begin{aligned} \min_{\mathbf{z}} \quad & -\frac{1}{2}\mathbf{z}^T K \mathbf{z} \\ \text{subject to} \quad & \mathbf{e}^T \mathbf{z} = 0, \\ & |\mathbf{z}| \leq C \end{aligned} \tag{3.14}$$

Assume K is positive definite in the space $\{\mathbf{z} : \mathbf{e}^T\mathbf{z} = 0\}$, and all local minimizers of (3.14) are at the boundary of $|\mathbf{z}| \leq C$. Also, assume all local maximizers of (3.14) have the same value for the term $\mathbf{e}^T|\mathbf{z}|$, then we can "simplify" the unsupervised SVM (3.13) to the minimization problem (3.14).

To better understand why relaxation (3.14) is reasonable, we consider following example with graphical illustrations. An examples of possible shapes of functions $\mathbf{e}^T|\mathbf{z}|$, $-\frac{1}{2}\mathbf{z}^T K\mathbf{z}$, and $\mathbf{e}^T|\mathbf{z}| - \frac{1}{2}\mathbf{z}^T K\mathbf{z}$ in two dimensional case are depicted in Figure 3.3 (a), (b), and (c) separately. For all plots, the x-axis and y-axis are the values of z_1 and z_2 separately. Recall the problem of unsupervised SVM, we are only interested in the label assignment of z_1 and z_2 such that we find the minimum of local maximums. In this two-dimensional case, we observe there are four local maximum as shown in Figure 3.3 (c). However, these actually correspond to only two cases, i.e. signs of z_1, z_2 are the same, or they are different. The case that $\text{sign}(z_1) = \text{sign}(z_2)$ corresponds to the upper right and lower left regions in the heatmap whereas $\text{sign}(z_1) \neq \text{sign}(z_2)$ corresponds to upper left and lower right regions. We note that the minimum of these local maximums is the case where $\text{sign}(z_1) = \text{sign}(z_2)$, namely, optimal choice of $\mathbf{z}$ should lie in the upper right and lower left regions to the origin. By observing the Figure 3.3 (b), we notice these are also the directions that function $-\frac{1}{2}\mathbf{z}^T K\mathbf{z}$ drops fastest. On the other hand, Figure 3.3 (a) shows that $\mathbf{e}^T|\mathbf{z}|$ elevates the values in same fashion on all four directions. These observations suggest that the best label assignments $\mathbf{y}$ for the minmax objective function (3.13) are simply the signs of $\mathbf{z}$ that decreases fastest in the objective function of (3.14). Therefore, we can simply ignore the label assignment and change our objective to finding the minimum of $-\frac{1}{2}\mathbf{z}^T K\mathbf{z}$ under the same constraints. In other words, we can change our objective function from (3.13) to (3.14) and simplify our problem in the aforementioned manner.

Note however, (3.14) remains an NP-hard problem since it is trying to find minimum of a concave objective function with rectangular constraint.

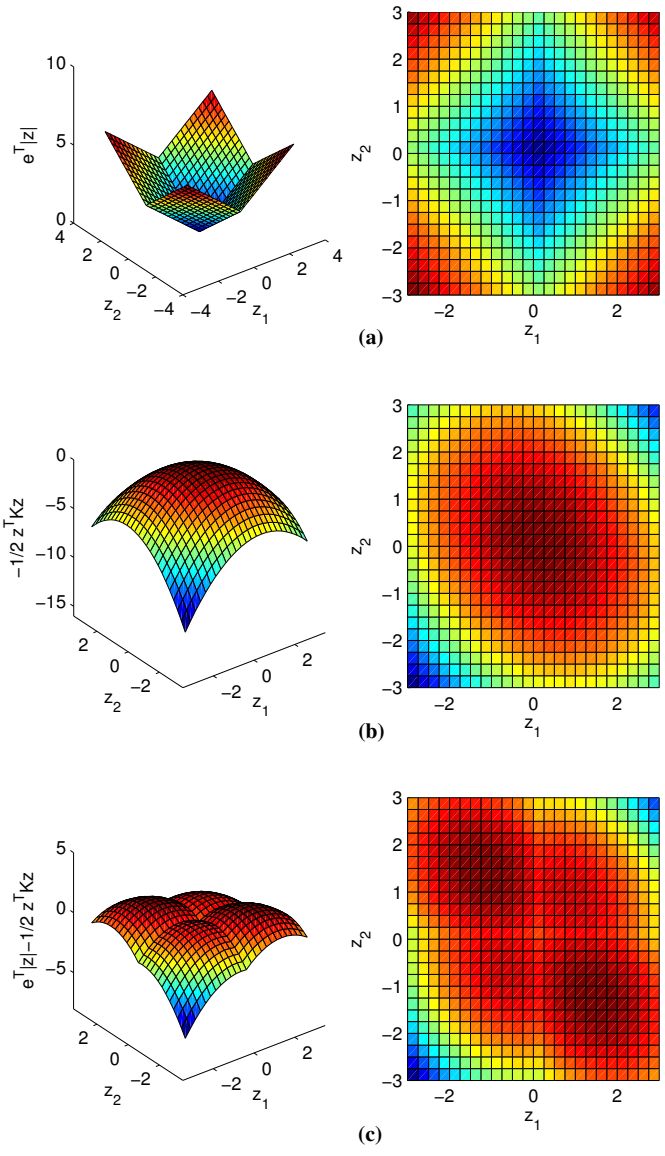

Figure 3.3: Graphical illustration of $\mathbf{e}^T|\mathbf{z}|$, $-\frac{1}{2}\mathbf{z}^T K\mathbf{z}$, and $\mathbf{e}^T|\mathbf{z}| - \frac{1}{2}\mathbf{z}^T K\mathbf{z}$

3.2.3 Connection between Spectral Optimization and Unsupervised SVM

Recall the optimization problem (3.1) for finding first non principal eigenvector is equivalent to

$$\begin{aligned} \min_{\mathbf{z}\in\Re^n} \quad & -\mathbf{z}^T K\mathbf{z} \\ \text{subject to} \quad & \mathbf{e}^T\mathbf{z} = 0 \\ & \mathbf{z}^T D^{-1}\mathbf{z} = \upsilon \end{aligned}$$

as presented in (3.2). Assuming K is positive definite, then we can replace the ellipsoidal equality constraint by an inequality constraint

$$\begin{aligned} \min_{\mathbf{z}\in\Re^n} \quad & -\mathbf{z}^T K\mathbf{z} \\ \text{subject to} \quad & \mathbf{e}^T\mathbf{z} = 0 \\ & \mathbf{z}^T D^{-1}\mathbf{z} \leq \upsilon \end{aligned} \tag{3.15}$$

because the ellipsoidal constraint in (3.15) should be active at a solution. Assume that we have $K = D^{-1}WD^{-1}$ and $C = \upsilon \cdot \mathbf{d}^{\frac{1}{2}}$, we notice the problem (3.15) can actually be considered as an approximation to the optimization problem (3.14) by approximating the rectangular constraint in (3.14) by the ellipsoidal constraint in (3.15).

This suggests that the normalized spectral optimization problem (3.1) can be regarded as an approximation to the unsupervised SVM problem (3.10) with the kernel $K = D^{-1}WD^{-1}$ and $C = \upsilon \cdot \mathbf{d}^{\frac{1}{2}}$.

Since the optimal separating hypothesis from the unsupervised SVM has the form

$$h(\mathbf{x}) = \left(\sum_{j=1}^{n} y_j^* \alpha_j^* \cdot K(\mathbf{x}, \mathbf{x}_j) + b^* \right)$$

and a non-principal eigenvector of the normalized spectral clustering $\mathbf{z}^*$ yields an approximation $|\mathbf{z}^*| \approx \alpha^*$ and $\text{SIGN}(\mathbf{z}^*) \approx y^*$, which are the coefficients of the bi-class separating optimal decision function, $|\mathbf{z}_j^*|$ provides a measurement of the strength of support from the

jth data point on the two class separation decision. We note however that, because of the use of the ellipsoidal constraint rather than rectangular constraints and other approximations, $\mathbf{z}^*$ is different from the exact SVM decision function coefficients. Specifically, the components of eigenvector are mostly nonzero which suggests every data instance provides certain level of support in this two clusters separation.

In addition, assume that $\mathbf{g}_1^*$ is the first non-principal eigenvector of a variation of unnormalized Laplacian $L = I - W$, with the eigenvalue λ_1. Then we have $K = W$ and $\mathbf{z}_1^* = \mathbf{g}_1^*$. Under this assumption, it can be easily verified that

$$K\mathbf{z}_1^* = (1 + \lambda_1)\mathbf{z}_1^*.$$

Consequently

$$(1 + \lambda_1)\mathbf{z}_1^* = K\mathbf{z}_1^* \approx \begin{pmatrix} f(\mathbf{x}_1) \\ f(\mathbf{x}_2) \\ \vdots \\ f(\mathbf{x}_n) \end{pmatrix} - b^*$$

Therefore $\mathbf{z}_1^*$ can as well be interpreted as a constant scaling-translation mapping of the approximate optimal bi-class separation function $f(\mathbf{x})$ evaluated at data instances. In this case, it is reasonable to use spectral optimization solution $\mathbf{z}^*$ as the ranking for the bi-cluster separation.

One remaining issue about SRA is to choose right ranking function based on the results of spectral optimization. As discussed in Section 3.1.2, two different rankings can be generated by SRA, i.e. $f(\mathbf{x}_i) = \max(|\mathbf{z}^*|) - |\mathbf{z}_i^*|$ for the case that multiple normal patterns present, and $f(\mathbf{x}_i) = |\mathbf{z}_i|$ for a normal vs. abnormal classification. To choose appropriate ranking, SRA simply introduces an input parameter χ as a user-defined upper bound of the ratio of anomaly. If the bi-class classification results in two very unbalanced clusters, it is very likely that we are facing the second scenario. We then report the ranking respect to a single major pattern and output an mFLAG = 0. On the other hand, if each class actually accounts for sufficient mass, it is more likely to be formed by other major normal patterns. Thereby, the ranking with respected to multiple major patterns is reported as in the first case with mFLAG set to 1.

A detailed description of SRA algorithm is provided in Algorithm 1.

Algorithm 1: Spectral Ranking for Anomalies (SRA)

Input: W: An m-by-m similarity matrix W.
χ: Upper bound of the ratio of anomaly

Output: $\mathbf{f}^* \in \Re^m$: A ranking vector with a larger value representing more abnormal
mFLAG : A flag indicating ranking with respect to multiple major patterns or a single major pattern

begin

Form Laplacian $L = I - D^{-1/2} W D^{-1/2}$;

Compute $\mathbf{z}^* = D^{\frac{1}{2}} \mathbf{g}_1^*$ where $\mathbf{g}_1^*$ is the 1st non-principal eigenvector for L ;

Let $\mathcal{C}_+ = \{i : \mathbf{z}_i^* \geq 0\}$ and $\mathcal{C}_- = \{i : \mathbf{z}_i^* < 0\}$;

if $min\{\frac{|\mathcal{C}_+|}{m}, \frac{|\mathcal{C}_-|}{m}\} \geq \chi$ **then**

mFLAG $= 1$, $\mathbf{f}^* = \max(|\mathbf{z}^*|) - |\mathbf{z}^*|$;

else if $|\mathcal{C}_+| > |\mathcal{C}_-|$ **then**

mFLAG $= 0$, $\mathbf{f}^* = -\mathbf{z}^*$;

else

mFLAG $= 0$, $\mathbf{f}^* = \mathbf{z}^*$;

end

end

3.3 Detecting Point Anomalies and Collective Anomalies with SRA

Although not specifically addressed, it has been demonstrated in [37] that SRA is capable of detecting point anomalies and collective anomalies at the same time. In this section, we further justify this fact and investigate the performance of SRA on different cases by taking the perspective based on its connection with the unsupervised SVM.

In order to examine the performance of SRA, we apply SRA to the two moon synthetic datasets presented in Figure 2.1 from the previous chapter, as they cover several typical scenarios of anomaly detection problems. In addition, the two moons are intuitive but non-trivial examples of bi-class classification problems. Therefore, by applying SRA on these datasets, we can see the performance of SRA as both an anomaly detection method as well as an unsupervised SVM classifier.

The results we obtained by applying SRA on these synthetic datasets are provided in Figure 3.4. The first row of the plots presents the information contained in the first and second non-principal eigenvectors of the normalized Laplacian matrices L's. It shows the relationship between $\mathbf{z}_1^* = D^{\frac{1}{2}}\mathbf{g}_1^*$ and $\mathbf{z}_2^* = D^{\frac{1}{2}}\mathbf{g}_2^*$ where $\mathbf{g}_1^*$ and $\mathbf{g}_2^*$ are the first and second non-principal eigenvectors, and the corresponding points are depicted with the same color as in Figure 2.1. It can be seen, in all three cases, how the points from two moons are separated by $x = 0$ on the x-axis which is in accordance with a bi-cluster separation in the unsupervised SVM. The points are classified into a positive class $\mathcal{C}_+$ and a negative class $\mathcal{C}_-$ which encapsulates the points of red moon and blue moon separately.

In order to illustrate different behavior of different kinds of anomalies in the ranking results, we consider only the 1st non-principal eigenvector and apply kernel density estimation (KDE) to the points corresponding to whole dataset (green shaded area) as well as only subsets of points corresponding to specific types of anomalies, i.e. the point anomalies (black curve) and collective anomalies (red curve). The results are given in second row of Figure 3.4. For all these cases, the score vector $\mathbf{z}_1^*$ derived from the 1st non-principal eigenvector presents a roughly multi-modal pattern with at least one noticeable peak on each side of the origin. We also notice that, the point anomalies are generally close to the

origin as the highest peak of its KDE is right around 0. This also conforms the intuition gained previously, as $|\mathbf{z}^*|$ provides a bi-class clustering strength measure and a smaller value suggests more ambiguity in terms of identification of the instance, therefore more likely to be the anomalies we are detecting. For the unbalanced case without additional noise, we notice how the 1st eigenvector perfectly separates the points and the curve corresponds to the positive class $\mathcal{C}_+$ perfectly aligns with the distribution of the rare class, i.e. the collective anomalies we defined. For the last case where both anomalies exist in the data, the general principal also holds, as the peaks of the green shaded area have their clear meaning: The highest peak of $\mathcal{C}_-$ corresponds to the majority pattern whereas the peak around 0 is related to the point anomalies, and the positive class still corresponds to the collective anomalies.

The above observations also relate different values of resultant mFLAG to different types of anomalies discovered. An output value of mFLAG = 0 would normally indicate the possible existence of the collective anomalies identified by SRA. Moreover, if both types of anomalies are present in the data at the same time, we notice the collective anomalies are ranked higher due to their stronger contribution to the "abnormal" class. This also suggests that mFLAG can be preset as an input to target a specific kind of anomaly. For instance, if we only want the ranking for point anomalies, we can simply set mFLAG = 0 and thereby ignore the ranking for collective anomalies. These observations justify that SRA has the capability of detecting collective anomalies and point anomalies simultaneously. It possesses the generality to detect different kinds of anomalies without the prior knowledge about the type of anomalies to be detected, and also retains the flexibility to let users determine what specific kind of anomalies they are interested in. This is especially valuable under unsupervised setting, as most other methods relies on the assumptions that only favor a specific kind of anomalies.

To justify the actual ranking quality obtained by SRA, we utilize the Receiver Operating Characteristic (ROC) curve as discussed in Section 2.3. The resultant ROC for each case is depicted on the last row of Figure 3.4. It can be seen that the collective anomalies can be perfectly tackled, as we can see a perfect *normal* vs. *abnormal* classification in the first non-principal eigenvector. The performance in terms of point anomalies is also remarkable considering the fact that the anomalies are not perfectly separable from normal

data instances due to the way they are generated. Finally, if we consider the case of targeting both at the same time, we can still obtain a nearly perfect overall ROC.

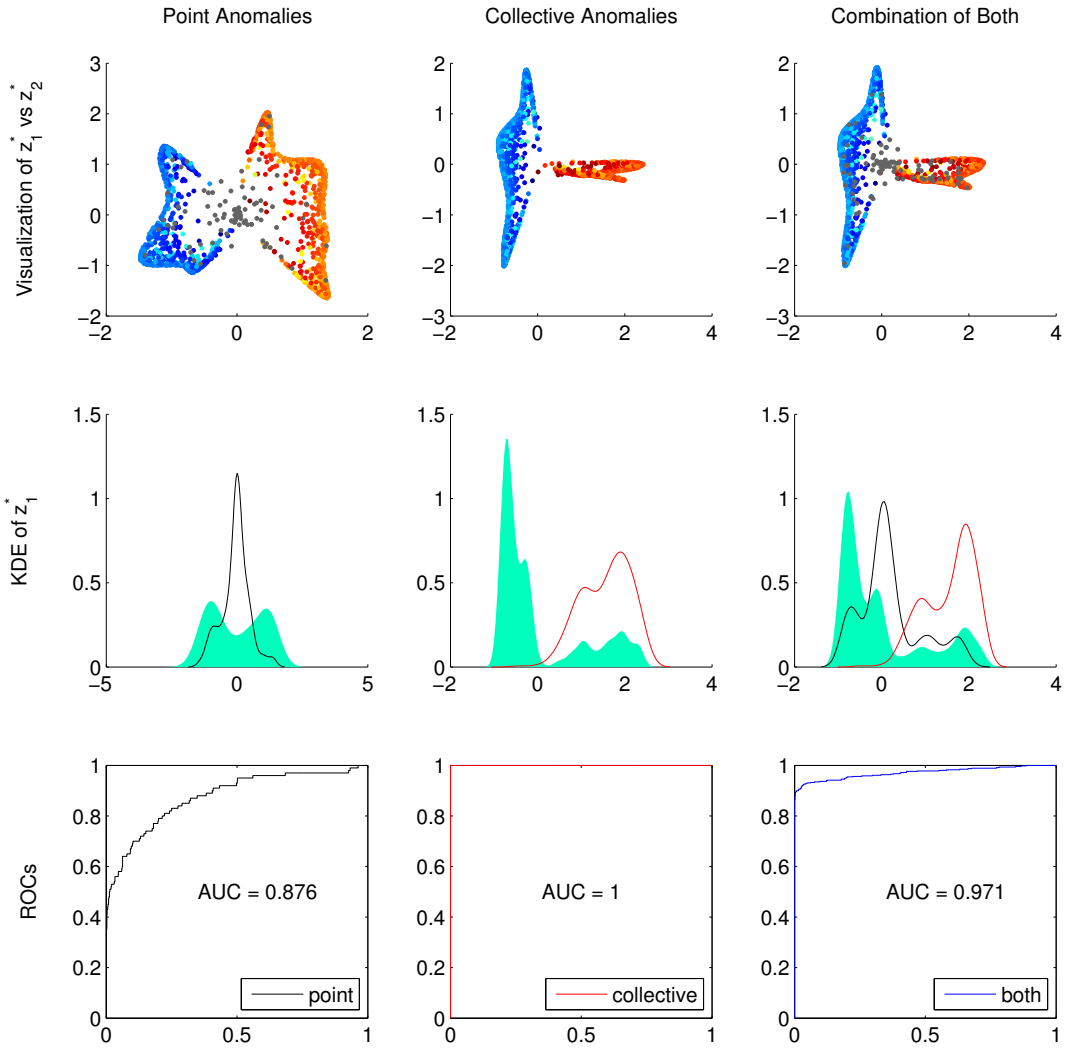

Figure 3.4: Result of SRA on the synthetic data shown in Figure 2.1. First row shows $\mathbf{z}_1^*$ (x-axis) and $\mathbf{z}_2^*$ (y-axis) based on first and second non-principal eigenvectors. Second row shows kernel density estimation of $\mathbf{z}_1^*$ over all dataset(green shaded area), point anomalies(black) and collective anomalies(red). Third row shows the ROC curves and corresponding AUCs.

Chapter 4

Unsupervised Feature Selection with HSIC to Detect Contextual Anomalies

In this chapter, we propose an unsupervised feature selection scheme based on the Hilbert-Schmidt Independence Criterion (HSIC) for the purpose of detecting feature-contextual anomalies. This chapter is divided into the following sections. In Section 4.1, we discuss how the feature selection for anomaly detection is different from other feature selection problems and the key assumption behind our proposed algorithm. In Section 4.2, we review the definition of HSIC and its application for the supervised feature selection. In Section 4.3, we present an unsupervised feature selection scheme based on HSIC that is useful for detecting contextual anomalies.

4.1 Feature Selection for Anomaly Detection

Extensive research has been conducted on the subject of supervised feature selection[22] [41] [31], and many attempts have been made for the unsupervised clustering as well [16]. In general, unsupervised feature selection can be very difficult because of the absence of

label information. With different selection criterion, the resultant feature subset can be significantly different and thereby greatly distort the performance of underlying algorithms. Moreover, most of existing unsupervised feature selection methods are not suitable for anomaly detection problems, as they mostly focus on searching the subset of features that results in best clustering quality, which is very different from the objective of anomaly detection.

Comparing with clustering analysis, feature selection can be even more challenging for anomaly detection problems due to the possible intervention from both unnecessary features and anomaly data instances. Additionally, intrinsic questions in real world problems inevitably inject uncertainty in the process of constructing features for training. Consider again the insurance fraud detection example discussed in Chapter 2, it is generally hard to target the exact relevant subset of features since adjusters or fraud experts tend to include more potential useful features at the risk of introducing noise. Consequently, we need a clear objective and reasonable assumptions to make feature selection possible for anomaly detections.

Recall different kinds of anomaly detection problems we have discussed so far, despite their differences, the abnormality are all defined over certain *normal property* that present in other data instances. This provides certain insight for us to approach the feature selection problem. Especially, when we are aware of the potential existence of noisy or unrelated features in the provided training dataset, we are most interested in the subset of features that can best reveal the structure of the data. In other words, we are interested in the subset of features the are actually useful for detecting anomalies. A reasonable assumption is that, a useful context is constructed by interactions among a subset of features and the interaction can be captured by a certain kind of dependence relationship among these features. For the noisy features, they should have no dependence with others, and the features that are not very helpful in constructing contexts should also have very limited dependence with other features. In summary, for the purpose of detecting feature-contextual anomalies, our objective is to reconstruct correct contexts for anomalies by eliminating the features that have little dependence relationship with others.

4.2 HSIC and supervised feature selection

To achieve the goal of effective feature selection for anomaly detection, we utilize Hilbert-Schmidt independence criterion (HSIC) as a fundamental tool in detecting dependence relationship among features. HSIC was proposed in [21] as a measure of statistical dependence and was first used for supervised feature selection in [50]. To prepare subsequent discussion, this section reviews the definition of HSIC and its useful properties that are helpful in feature selection. The presentation mainly follows [21] and [50].

Before we discuss detection of arbitrary dependence among data using HSIC, we first consider a simple case of detecting *linear dependence* among data. Following similar notations as previous chapters, assume that we have two feature domains $\mathcal{X} \subset \mathbb{R}^d$ and $\mathcal{Y} \subset \mathbb{R}^l$, and we have random variables (x, y) that are jointly drawn from $\mathcal{X}, \mathcal{Y}$. Then we denote the cross-covariance matrix of x, y as $\mathcal{C}_{xy}$, and we have

$$\mathcal{C}_{xy} = \mathbb{E}_{xy}\left[xy^T\right] - \mathbb{E}_x\left[x\right]\mathbb{E}_y\left[y\right]$$

We know that $\mathcal{C}_{xy}$ contains all the second order dependence between x and y, and the Frobenius norm of $\mathcal{C}_{xy}$ is defined as the trace of $\mathcal{C}_{xy}\mathcal{C}_{xy}^T$, namely

$$\|\mathcal{C}_{xy}\|^2_{\text{Frob}} = \text{tr}\left(\mathcal{C}_{xy}\mathcal{C}_{xy}^T\right)$$

which summarizes the degree of linear correlation between x and y. The value $||\mathcal{C}_{xy}||^2_{\text{Frob}}$ is zero if and only if there is no linear dependence between x and y, and this can thereby be utilized in detecting linear dependence between them. However, capturing only linear dependence is rather limited, especially when we are uncertain about the actual type of data we are dealing with, and the dependence relationship might not be captured by cross-covariance at all. Instead, we are interested in the flexibility of detecting arbitrary dependence, possibly nonlinear dependence, relationship between x and y. We thereby generalize the notion of cross-covariance to detect nonlinear relationship and to cope with different kinds of data.

In order to handle nonlinear cases, we introduce two feature mappings $\phi : \mathcal{X} \to \mathcal{F}$ and $\psi : \mathcal{Y} \to \mathcal{G}$ from original feature domain to their corresponding reproducing kernel

Hilbert spaces $\mathcal{F}$ and $\mathcal{G}$. The inner product between features can then be rewritten via their characteristic kernel functions

$$k(x, x') = \langle \phi(x), \phi(x') \rangle \text{ and } l(y, y') = \langle \psi(y), \psi(y') \rangle$$

Issues concerning of kernels are usually similar to the kernels selections for SVM as discussed in previous Chapter. Examples include polynomial kernel and Gaussian RBF kernel that map data to higher dimensional spaces. Following [20] and [5], we then generalize the idea of cross-covariance matrix and define a cross-covariance operator $\mathcal{C}_{xy} : \mathcal{G} \to \mathcal{F}$ between the feature maps such that

$$\mathcal{C}_{xy} = \mathbb{E}_{xy}\Big[\big(\phi(x) - \mathbb{E}_x[\phi(x)]\big) \otimes \big(\phi(y) - \mathbb{E}_y[\psi(y)]\big)\Big]$$

and $\otimes$ denotes the tensor product. Denote the distribution for sampling x and y as Pr_{xy}, HSIC is then defined as:

$$\text{HSIC}(\mathcal{F}, \mathcal{G}, Pr_{xy}) = \|\mathcal{C}_{xy}\|^2_{HS}$$

where $\|\cdot\|_{HS}$ is the Hilbert-Schmidt norm. The Hilbert-Schmidt norm is used here to extend the notion of Frobenius norm to operators, and similarly it has the form of $\text{tr}\left(\mathcal{C}_{xy}\mathcal{C}_{xy}^T\right)$. If we rewrite this measure in terms of kernel functions k and l, we have:

$$\begin{aligned}&\text{HSIC}(\mathcal{F}, \mathcal{G}, Pr_{xy}) = \\ &\mathbb{E}_{xx'yy'}[k(x, x')l(y, y')] + \mathbb{E}_{xx'}[k(x, x')]\mathbb{E}_{yy'}[l(y, y')] - 2\mathbb{E}_{xy}\big[\mathbb{E}_{x'}[k(x, x')]\mathbb{E}_{y'}[l(y, y')]\big] \quad (4.1)\end{aligned}$$

One advantage of HSIC is that it is very easy to estimate. Two most popular estimators are presented in [21] and [50] separately. With the chosen kernels and the set of observations $Z = (X, Y) = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ that are drawn $i.i.d$ from the joint distribution Pr_{xy}, we can then construct two kernel matrices $K, L \in \mathbb{R}^{m \times m}$, where $K_{ij} = k(x_i, x_j)$ and $L_{ij} = l(y_i, y_j)$. The one proposed in [21] has the following form

$$\tilde{\text{HSIC}}(\mathcal{F}, \mathcal{G}, Z) = (m-1)^{-2}\text{tr}\,(KHLH) \quad (4.2)$$

where $H = \mathbf{I} - m^{-1}\mathbf{e}\mathbf{e}^T$ with $\mathbf{e}$ being the vector of ones as before. This however is a biased estimate of $\text{HSIC}(\mathcal{F}, \mathcal{G}, Pr_{xy})$ with $\text{HSIC}(\mathcal{F}, \mathcal{G}, Pr_{xy}) - \tilde{\text{HSIC}}(\mathcal{F}, \mathcal{G}, Z) = O(m^{-1})$ as shown in [50].

In [50], an *unbiased* estimator for (4.1) is also proposed, which has the form:

$$\tilde{\text{HSIC}}(\mathcal{F}, \mathcal{G}, Z) = \frac{1}{m(m-3)} \left[\text{tr}(\tilde{K}\tilde{L}) + \frac{\mathbf{e}^T \tilde{K} \mathbf{e}\, \mathbf{e}^T \tilde{L} \mathbf{e}}{(m-1)(m-2)} - \frac{2}{m-2} \mathbf{e}^T \tilde{K} \tilde{L} \mathbf{e} \right] \tag{4.3}$$

where $\tilde{K}$ and $\tilde{L}$ are the matrices obtained by setting diagonal entries of K and L to zero. Though the unbiased estimator has relatively more complex form, both estimators are easy to compute and overall takes $O(m^2)$ time complexity. For upcoming discussions and empirical evaluations, we stick with the unbiased estimator (4.3).

As discussed in [50], with properly chosen kernels, HSIC can be used to detect arbitrary dependence between X and Y. The value of $\text{HSIC}(\mathcal{F}, \mathcal{G}, Pr_{xy}) = 0$ if and only if there are no dependence between x and y. We can thereby use $\tilde{\text{HSIC}}(\mathcal{F}, \mathcal{G}, Z)$ as a feature selection criteria. For supervised learning, if ψ is the kernel transformation corresponding to labels, it is reasonable to assume the best subset of features should correspond to the ones that maximize the dependence between features and labels.

Since finding the optimal feature subset with a given criteria is a typical NP-hard problem [55], a good approximation can be achieved by performing greedy backward elimination on the features which have least dependence with labels or forward appending the features that can increase the dependence most. Applying two different strategies leads to backward elimination (BAHSIC) and forward elimination HSIC (FOHSIC) respectively as detailed in [50]. Here we reiterate the algorithm of BAHSIC in Algorithm 2.

Note that, for the convenience of presentation, we override the notion of $\tilde{\text{HSIC}}_{kl}(\mathcal{S}, \mathcal{Y})$ to denote the estimated value of HSIC between data with selected feature set $\mathcal{S}$ and labels $\mathcal{Y}$. The kernels k and l are used respectively to construct K and L. Also, we use $\tilde{\text{HSIC}}_k(\mathcal{S}, \mathcal{S}')$ to denote the estimated value of HSIC between selected feature set $\mathcal{S}$ and $\mathcal{S}'$ with both K and L constructed using k as their kernels.

Algorithm 2: BAHSIC [50]

Input: k : kernel characteristic function for features
l : kernel characteristic function for labels
$\mathcal{S}$: full featureset
$Z = (X, Y)$: full dataset

Output: $\mathcal{S}^*$: The selected subset of features

begin
$\mathcal{S}_0 \leftarrow \mathcal{S}, \ \mathcal{Y} \leftarrow Y, \ i \leftarrow 0,$
while $|\mathcal{S}_i| > 0$ *and stopping criteria not satisfied* **do**
$i \leftarrow i + 1$
// removing features $\mathcal{I}_i$ results in maximum dependence with labels
$\mathcal{I}_i \leftarrow \arg\max_{\mathcal{I}} \sum_{I \in \mathcal{I}} \tilde{\text{HSIC}}_{kl}(\mathcal{S}_{i-1} \backslash \{I\}, \mathcal{Y}), \ \mathcal{I} \subset \mathcal{S}_{i-1}$
$\mathcal{S}_i \leftarrow \mathcal{S}_{i-1} \backslash \mathcal{I}_i$
$\mathcal{S}^* \leftarrow \mathcal{S}^* \cup \mathcal{I}_i$
end
$\mathcal{S}^* \leftarrow \mathcal{S}_i$
end

4.3 An unsupervised filter feature selection algorithm based on HSIC

4.3.1 BAHSIC-AD

Inspired by the application of HSIC in supervised feature selection, we propose an unsupervised *filter* algorithm BAHSIC-AD , with AD stands for Anomaly Detection, based on HSIC to better facilitate anomaly ranking by existing algorithms. The basic assumption follows the idea as discussed in Section 4.1. Namely, the goal is to eliminate the noisy features and keep the subset of features that has strong dependence with each other in the implicit feature space.

To accomplish this, we follow a greedy backward elimination procedure similar to BAH-

SIC. However, in each iteration, instead of estimating the dependence between features and labels, we estimate dependence among features. For each feature we calculate its dependence with the rest of features in our selected kernel space and we continue eliminating the feature that has the smallest dependence with the rest. The features get eliminated would most likely to be the least helpful ones in reconstructing the meaningful contexts we desired.

More specifically, assuming that we are at the ith iteration with the remaining feature set $\mathcal{S}_{i-1}$ from the previous iteration, and we want to eliminate another set of features $\mathcal{I}_i$, which is of the size p. Then we calculate $\tilde{\text{HSIC}}_k\big(\mathcal{S}_{i-1}\backslash\{I\}, \{I\}\big)$ for each feature $I \in \mathcal{S}\backslash\mathcal{I}$, and we get the $\mathcal{I}_i = \{I_1, I_2, \ldots, I_p\}$ such that $\sum_{I\in\mathcal{I}_i} \tilde{\text{HSIC}}_k\big(\mathcal{S}_{i-1}\backslash\{I\}, \{I\}\big)$ has the smallest value among all possible $\sum_{I\in\mathcal{I}} \tilde{\text{HSIC}}_k\big(\mathcal{S}_{i-1}\backslash\{I\}, \{I\}\big)$ for every $\mathcal{I} \subset \mathcal{S}_{i-1}$ that is of the size p. We keep removing the features following this manner, until certain stopping criteria is satisfied. This algorithm is summarised in Algorithm 3.

Note that, although we are interested in the subset of features that are dependent among each other, we are not interested in the features that are *perfectly correlated*. This is because the perfectly correlated features will add no information regardless applying either supervised or unsupervised learning. However, this is rarely the case in real world applications especially when anomalies are present.

4.3.2 A synthetic example

To demonstrate how the process of BAHSIC-AD affects the quality of our anomaly detection algorithm, we apply it on a synthetic dataset with 7 features, including 4 injected noisy features. The first three features $\{feature_1, feature_2, feature_3\}$ of the synthetic dataset are the only non-noisy features, and they are depicted in Figure 4.1. Two Gaussian mixture clusters are generated with mean $\mu_1 = (-1, 1, -1)$ and $\mu_2 = (3, -4, 3)$ separately, and simply using $\Sigma_1 = 2\mathbf{I}$, and $\Sigma_2 = \mathbf{I}$ as the covariance matrices where $\mathbf{I}$ is the identify matrix. The blue (left) cluster C_1 contains 400 points whereas the green (right) cluster C_2 contains 600 points. Additional 50 points are generated uniformly in $[-4, 3] \times [-3, 3] \times [-4, 3]$ as point anomaly targets. It is designed to have two major patterns, with one relatively dense but having fewer points, and the other more points but relatively sparse. Most importantly,

Algorithm 3: BAHSIC-AD: HSIC based unsupervised feature selection algorithm for anomaly detection

Input: k : kernel characteristic function for features
$\mathcal{S}$: full feature set
Z: full dataset

Output: $\mathbf{f}^* \in \Re^n$: A ranking vector with a larger value representing more abnormal
$\mathcal{S}^*$: The final subset of features that defined the context

begin

$\mathcal{S}_0 \leftarrow \mathcal{S}, \;\; i \leftarrow 0$

while $|\mathcal{S}_i| > 0$ *and stopping criteria not satisfied* **do**

$i \leftarrow i + 1$

//select features $\mathcal{I}_i$ that are least dependent with rest of features

$\mathcal{I}_i \leftarrow \arg\min \sum_{I \in \mathcal{I}} \tilde{\text{HSIC}}_k\big(\mathcal{S}_{i-1} \backslash \{I\}, \{I\}\big), \quad \mathcal{I} \subset \mathcal{S}_{i-1}$

$\mathcal{S}_i \leftarrow \mathcal{S}_{i-1} \backslash \mathcal{I}_i$

end

$\mathcal{S}^* \leftarrow \mathcal{S}_i$

Apply anomaly detection algorithm with respect to chosen $\mathcal{S}^*$ to get $\mathbf{f}^*$

end

we inject 4 more noisy features that contain pure noise generated by uniform distribution. Each dimension is then standardized subsequently to zero mean and unit variance. Note that, this dataset is designed for clear visualization for upcoming discussions.

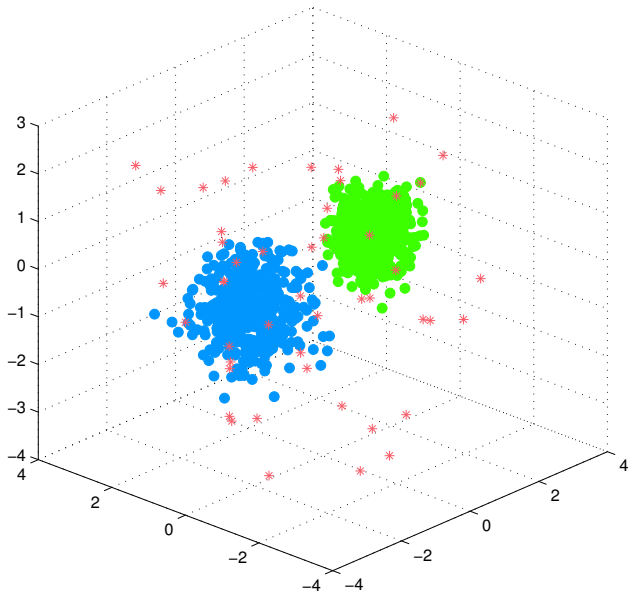

Figure 4.1: First two dimensions of toy dataset: two Gaussian clusters with anomalies

Similar to the example presented in Figure 2.2, where the first two dimensions are the context for contextual anomalies, the first three dimensions here apparently construct the context we are most interested in. Therefore, our goal is to first remove noisy features from the seven feature dataset to reconstruct the context and apply some anomaly detection algorithm, such as SRA, to identify anomalies.

We start with the dataset as the full feature set and trace down the feature selection process. Specifically, we are interested in how the quality of anomaly detection improves when the noisy feature gets eliminated and how the value $\tilde{\text{HSIC}}_k(\mathcal{S}_i \backslash \{I\}, \{I\})$, for any feature I, changes in each iteration i throughout the entire learning process. Here, we utilize the SRA algorithm we have discussed in Chapter 3 for the purpose of visualization

and comparisons. In each iteration, we apply SRA on the dataset with the remaining features, and for both SRA and BAHSIC-AD we use the Gaussian RBF kernel. The results are presented in Figure 4.3 with information in the eigenvector space and the corresponding ROC. The values of $\widetilde{\text{HSIC}}_k(\mathcal{S}_i \backslash \{I_i\}, \{I_i\})$ for every feature $I_i \in \mathcal{S}$ are also provided on the left subplot in Figure 4.2. Note that, if a feature gets eliminated at a specific iteraion, the corresponding line plot also terminates. For instance, feature 5 gest eliminated at the 4-th iteration, therefore the star black line plot that corresponds to feature five simply ends at iteration No.4.

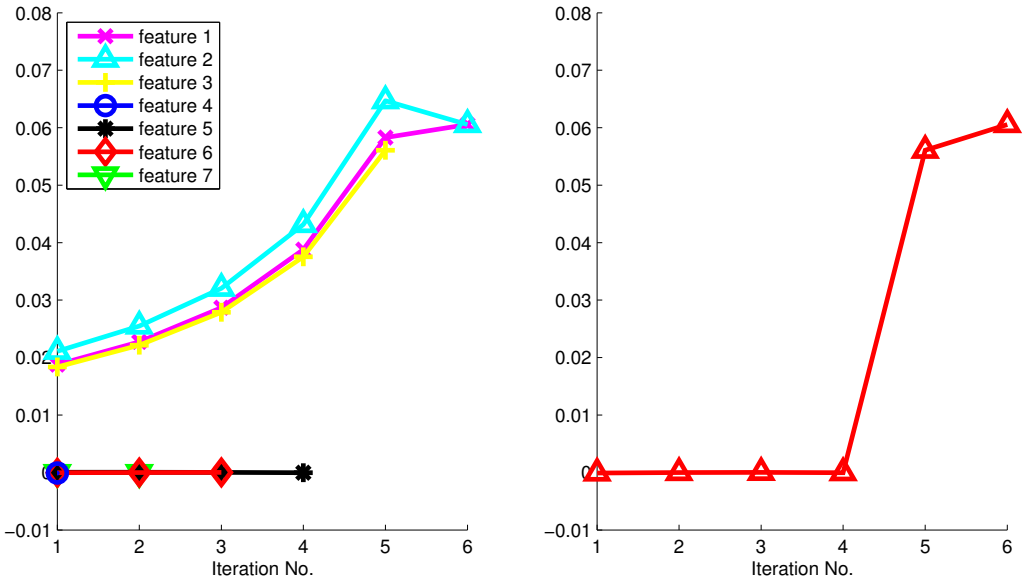

Figure 4.2: In each iteraion of feature elimination, the values of $\widetilde{\text{HSIC}}_k(\mathcal{S}_i \backslash \mathcal{I}, \mathcal{I})$ for each feature (left plot) and the value of $\min(\widetilde{\text{HSIC}}_k(\mathcal{S} \backslash \mathcal{I}, \mathcal{I}))$ (right plot)

From Figure 4.2, we notice how the noisy features are identified and eliminated from the beginning, as the first four features removed are $feature_4$, $feature_6$, $feature_5$ and $feature_7$, which match exactly the set of noisy features injected into the original dataset. While eliminating features, we can also observe, from Figure 4.3, how the two-class classification becomes more clear as the peaks of bi-modal pattern are further stretched when we have only two or three relevant features left. Also the red curves, which correspond

to anomalies, lie closer to the origin when we eliminate the noisy features and the AUC becomes significantly higher. This implies the BAHSIC-AD algorithm is very helpful in terms of revealing the context for anomaly detection. In the end, it correctly identifies the relevant feature subset, i.e. $\{feature_1, feature_2, feature_3\}$, in the second to the last iteration.

By observing the pattern of the data in the eigenvector space, we can also see how they better reveal the structure of the data in the original space. The values of $\tilde{\text{HSIC}}_k(\mathcal{S}_i \backslash \mathcal{I}, \mathcal{I})$ of relevant features also become more significant after the noisy ones get eliminated. Moreover, it also demonstrates how important proper context is when identifying the anomalies as the performance of SRA is significantly distorted when one of the useful feature (feature 3) gets eliminated. This is related to another important issue in the feature selection process, i.e. the stopping criteria. The stopping criteria is very important because we do not want the actual relevant features get eliminated while removing features.

The simplest way to stop the process is to set a fixed number k for the top k features. This is sometimes desirable in terms of interpretability of the results as many applications only require the knowledge about the top features that lead to the final ranking. Nevertheless, it is more often the case that there are lack of knowledge in the actual number of relevant features. While being an interesting research problem itself, there are rarely good solutions with respect to unsupervised learning problems, and effective supervised learning approaches like cross-validation are simply not applicable due to absence of labels.

The way we approach this problem is by observing the value $\tilde{\text{HSIC}}_k(\mathcal{S} \backslash \mathcal{I}, \mathcal{I})$ of the feature to be eliminated in each iteration. We notice that the minimum of $\tilde{\text{HSIC}}_k(\mathcal{S} \backslash \mathcal{I}, \mathcal{I})$ among all features is a good general stopping criteria, and the previous example also confirmed this point. The minimum value of $\tilde{\text{HSIC}}_k(\mathcal{S} \backslash \mathcal{I}, \mathcal{I})$ graphed in the right subplot in Figure 4.2. In this case, there are at most three relevant features and each time we eliminate a feature the value of $\min(\tilde{\text{HSIC}}_k(\mathcal{S} \backslash \mathcal{I}, \mathcal{I}))$ significantly increase until the feature set size is reduced to 3 which is the point where all features are relevant. A significant increase of $\min(\tilde{\text{HSIC}}_k(\mathcal{S} \backslash \mathcal{I}, \mathcal{I}))$ is therefore a good indicator to stop the feature elimination. In practice, there are several other possible scenarios. For instance, it is possible to see that $\min(\tilde{\text{HSIC}}_k(\mathcal{S} \backslash \mathcal{I}, \mathcal{I}))$ values drop from the very beginning of the feature elimination, this phenomenon strongly suggests all features are greatly dependent with others, and thereby

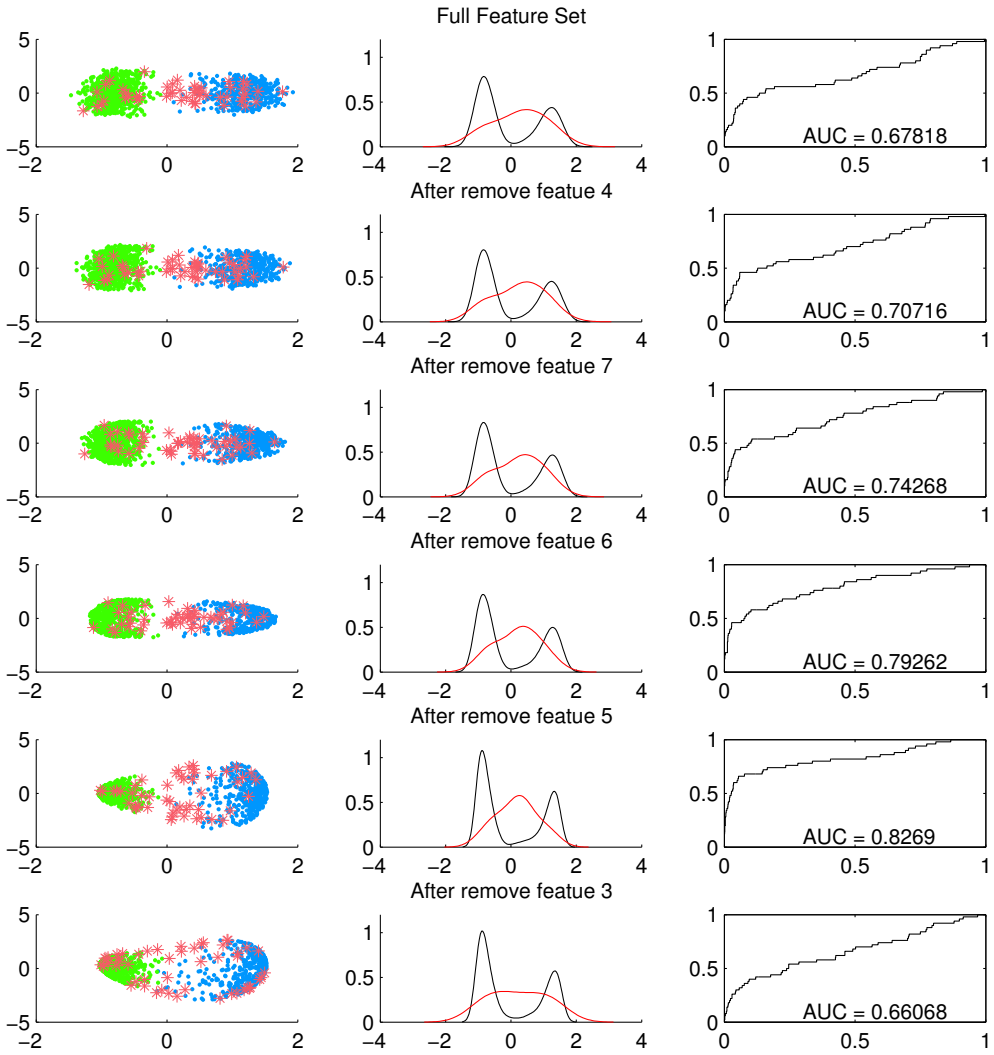

Figure 4.3: In the process of the feature selection, effect of feature selection on 1st and 2nd non-principal eigenvectors of Laplacian matrix(left), density of all data and anomalies on $\mathbf{z}_1 =$ (middle) and the ROC curve(right)

very important to the anomaly ranking. Empirically, this approach do not always guarantees the optimal stopping point, but as long as the data conforms with our assumption that useful features have strong dependence with each other, it generally provides a satisfactory result. The method described here is also the stopping criteria we used in the computational evaluations in the next chapter.

Chapter 5

Computational Results

In this chapter, we empirically evaluate the effectiveness of SRA combined with BAHSIC-AD, in dealing with different types of anomalies. We perform a comprehensive evaluation comparing its performance with some other prevailing anomaly detection methods on a series of benchmark datasets. Section 5.1 discusses the experiment settings and benchmark datasets which we apply in the evaluation. Section 5.2 presents the results. Finally, Section 5.3 uses automobile insurance fraud dataset as an example to discuss how feature selection with BAHSIC-AD can be helpful in terms of interpretation of the ranking results.

5.1 Benchmark Datasets and Experiment Settings

5.1.1 Synthetic Datasets

For the purpose of a comprehensive evaluation, we generate different synthetic datasets to simulate different common scenarios in anomaly detection. Similar to examples presented in the previous chapters, we mainly apply two mechanisms to generate synthetic examples, i.e., two moon clusters, and Gaussian clusters. A detailed description about the synthetic dataset is provided in the Table 5.1 and several examples are depicted in Figure 5.1.

Similar to the cases that we have discussed in Chapter 2, variations of two moon clusters

are included because they are conceptually easy problems for humans but generally hard for common classification based algorithms. We first simulate cases that only one kind of anomalies present, either point anomaly only or collective anomaly only. Figure 5.1 (a) presents the case when two major balanced moons are presented with random noise scattered around the major patterns. Figure 5.1 (b) and Figure 5.1 (c) simulate cases where only collective anomalies present.

In addition, we also generate multiple Gaussian clusters to simulate cases that both point anomalies and collective anomalies appear at the same time. Specifically, we generate different number of clusters other than the ideal bi-cluster case, as we are equally interested in scenarios when more than two noticeable patterns present. We also change the number of relevant features to see how different algorithms perform on datasets with more than 2 relevant features. The point anomalies are always noise deviated from any major pattern and the collective anomalies are the relative insignificant clusters among multiple clusters.

To test how algorithms perform in dealing with contextual anomalies, we also inject 5 or 10 noisy features to exam how they react to noisy features and whether BAHSIC-AD can reconstruct the original datasets.

An additional note about synthetic datasets in general is the fact that the way we generate the datasets particularly favors nearest-neighbor based approaches, like k-NN or weighted k-NN. Since the anomalies we defined here mostly conform with the assumption made by this set of methods. These synthetic examples are used to illustrate some typical cases, and we still need real world datasets to make a comprehensive evaluation.

5.1.2 Real World Datasets

Weal world datasets are also included to evaluate the performance of different algorithms for applications arises from practice. These datasets are mainly selected from UCI machine learning repository [4] and KEEL dataset repository [2], and they originated from various application domains, including life science, business, physics, and others. The automobile insurance dataset which has been utilized as a benchmark in [42] is also included for two reasons: Firstly, the insurance fraud detection is an important application

Table 5.1: Description of synthetic benchmark datasets

Name[1]	Source	Type[2]	Features[3]	m	m_+
Unbalanced Two Moons (0)	Synthetic	C	2C	1200	200
Unbalanced Two Moons (5)	Synthetic	C	7C	1200	200
Unbalanced Two Moons (10)	Synthetic	C	12C	1200	200
Close Unbalanced Moons (0)	Synthetic	C	2C	1200	200
Close Unbalanced Moons (5)	Synthetic	C	7C	1200	200
Close Unbalanced Moons (10)	Synthetic	C	12C	1200	200
Noisy Two Moons (0)	Synthetic	P	2C	1650	50
Noisy Two Moons (5)	Synthetic	P	7C	1650	50
Noisy Two Moons (10)	Synthetic	P	12C	1650	50
Gaussian (3,2,0)	Synthetic	C+P	2C	1400	400
Gaussian (3,2,5)	Synthetic	C+P	7C	1400	300
Gaussian (3,2,10)	Synthetic	C+P	12C	1400	400
Gaussian (4,4,0)	Synthetic	C+P	4C	1400	400
Gaussian (4,4,5)	Synthetic	C+P	9C	1400	400
Gaussian (4,4,10)	Synthetic	C+P	14C	1400	300

[1] The Gaussian (x, y, z) and Two Moons (z) are synthetic datasets, where x is the number of clusters, y is the number of relevant features and z is the number of injected noisy features.

[2] For the type of anomalies, "C" stands for collective anomalies, "P" for point anomalies and "C+P" for presence of both. The exact type of anomalies in real dataset is unknown.

[3] For the feature type of the data, "C" stands for continuous valued feature

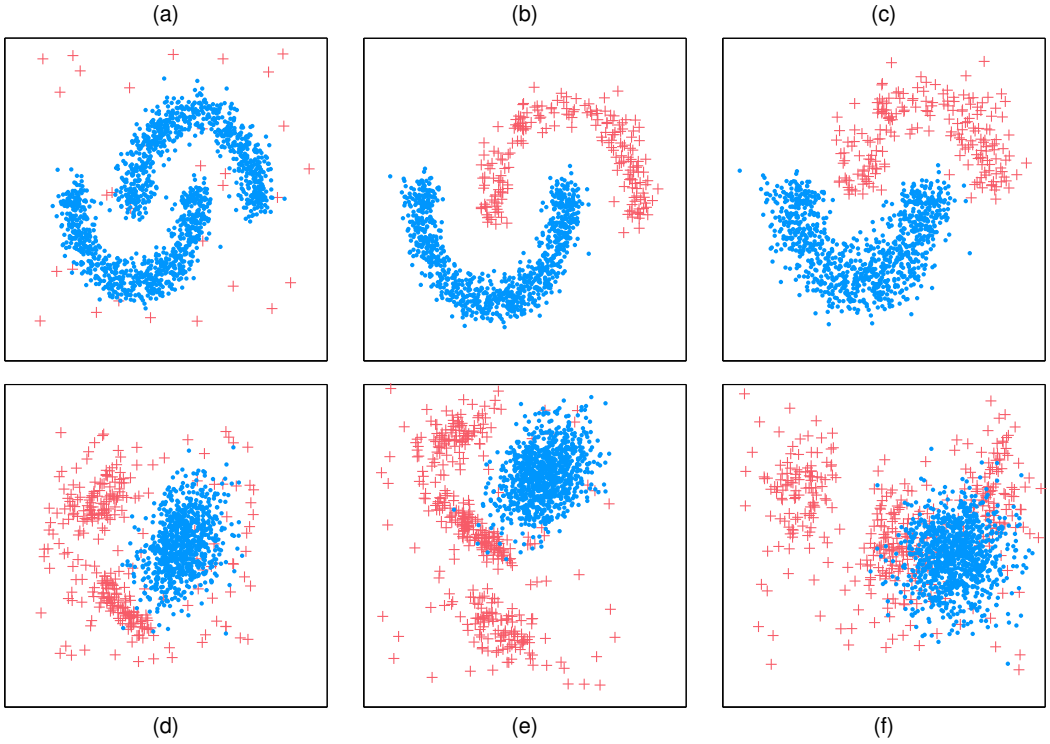

Figure 5.1: Examples of synthetic datasets, anomalies are marked by red "+" while normal patterns are depicted by blue dots. (a) Noisy Two Moons (b) Unbalanced Two Moons (c) Close Unbalanced Two Moons (d) Gaussian(3,2,0) (e) Gaussian(4,4,0) feature 1 vs feature 2 (f) Gaussian(4,4,0) feature 3 vs feature 4

for anomaly detection. Secondly, this is an example of data consists mainly of categorical (nominal) features. Among the benchmarks, all the bi-class datasets are highly unbalanced bi-classification problem, and we therefore treat the rare class as the anomaly class. For the datasets that are originally multi-class datasets, we treat the class that consists of the smallest number of instances as the anomaly class. In real world applications, the exact type of anomalies is usually unknown. Moreover, whether they present as contextual anomalies and whether feature selection is helpful are blind to users. We however stick with the BAHSIC-AD algorithm described previously, and see whether the feature selection can actually be helpful in all these cases. Detailed descriptions of benchmarks from real applications are provided in Table 5.2.

5.1.3 Experiment Settings and Evaluation Method

In addition to the SRA algorithm described in Chapter 3, we also select five exemplary but also prevailing methods for comparisons. These methods have been briefly discussed in Chapter 2, including kernel based approaches: One-Class SVM [33], density based approaches: Local Outlier Factor (LOF) and Local Outlier Probabilities (LoOP), approximate Local Correlation Integral (aLOCI) as well as the nearest neighbor approaches: k-NN (k-Nearest Neighbor) and weighted k-NN. For implementation of these algorithms, we use LibSVM [11] for the implementation of One-Class SVM, and ELKI [1] for LOF, LoOP, ALOCI, k-NN, and weighted k-NN.

For any kernel-based approach that requires a similarity or kernel defined over the data, such as OC-SVM, SRA, we use a consistent choice of the kernel. We apply RBF Guassian kernel $k(x, x') = \exp(-||x - x'||/2\sigma^2)$ for every dataset that consists of mainly numerical features. For the methods that require a distance function, we simply apply Euclidean distance.

Since the RBF kernel and Euclidean distance are only valid for continuous numerical features, we thereby need some preprocessing for certain datasets. For datasets that consist of a mixture of continuous numerical and nominal features such as Thyroid, we need to convert nominal features to continuous features. While there are a few technique designed for this kind of problem, here we apply one of the most commonly applied unsupervised

Table 5.2: Description of real world benchmark datasets

Name	Source	Features [1]	m	m_+
Shuttle0vs4	Keel	9C	1829	123
Satellite	UCI	36C	6435	626
Ecoli	UCI	7C	336	35
YeastME2	UCI	8C	1484	51
Thyriod	UCI	21N, 7C	3772	231
Glass4	Keel	13C	214	13
Libras	UCI	90C	360	24
Diabetes	UCI	8C	768	268
Survival	UCI	3C	306	81
Wine3	UCI	13C	178	48
Breast-Wisc	UCI	9C	699	241
Zoo	UCI	15B	101	4
Mushroom	UCI	22N	4508	300
Automobile Fraud	[42]	31N	15420	923

[1] For the feature type of the data, "C" stands for continuous numerical feature, "N" for nominal feature and "B" for binary feature

approach [56]. Namely, we transform the original nominal features into a set of binary features which can be treated as continuous. Specifically, we use $k-1$ binary features to represent a nominal feature which originally has k distinct values, with i-th binary feature set to 1 only when the original feature has its ith value. For example, suppose we have a nominal feature *season* which has 4 distinct values $\{spring, summer, autumn, winter\}$, we replace the feature set by three binary numerical features with values determined by $season = spring$, $season = summer$, and $season = autumn$. If a data instance has $season = spring$ in original dataset, then it has $(1, 0, 0)$ after transformation, and if another data instance has $season = winter$, it becomes $(0, 0, 0)$. Then we can treat the whole dataset as completely numerical. While there are other possible techniques for mapping the nominal values into numerical values, this technique can retain the information in the nominal feature without injecting the unnecessary ordinal information possessed by most numerical features. Note that, since dataset Zoo contains only binary features, we simply regard it as a numerical dataset.

In addition to the RBF Gaussian kernel, we are interested in whether a suitable kernel for a specific dataset can actually improve the results of kernel-based methods, we thereby purposely include two datasets with pure nominal features, i.e. Mushroom and Automobile Fraud datasets. For these two datasets, we compare binarizing the features as described above with applying the *Hamming distance kernel* directly on the original non-transformed dataset. Briefly speaking, a Hamming distance kernel is of the form $k(x, x') = \sum_{u \in \mathcal{D}^n} \theta_u(x)\theta_u(x')$ where $\mathcal{D}^n$ is an n-dimensional nominal feature space with $\mathcal{D}_i$ corresponds to i-th feature, and $\theta(x) = \lambda^{d^H(u,x)}$ with $d^H(x, x') = (1/n)\sum_{i=1}^{n}(\delta(x, x'))$ and λ being a damping parameter. δ is the overlapping similarity function such that $\delta(x, x') = 1$ when x and x' are identical, $\delta(x, x') = 0$ otherwise. The Hamming distance kernel is derived from a String Kernel, and specifically designed for datasets with pure nominal features. However, the detailed derivation of a Hamming distance kernel is not the subject of this thesis, we thereby refer interested readers to [13] and [37] for a more detailed discussion.

For consistency, we use the bandwidth $\sigma = \sqrt{n}$, with n being the number of features, for the Gaussian RBF kernel, and damping parameter $\lambda = 0.8$ for the Hamming distance kernel. For all methods except for SRA, OC-SVM and aLOCI, number of the nearest neighbor parameter k is required. Here we set $k = \min\{100, m/10\}$ where m is the total

number of data instances. The threshold parameter χ required by SRA is set as 35% for all the experiments. Additionally, we standardize all real world datasets to zero mean and unit variance i.e., $\mu = 0$, $\sigma^2 = 1$, before running any experiment.

Note that, the choice of parameters here can be suboptimal for a specific dataset. By fine tuning the parameters, we can observe certain level of improvement for a particular method. However, without labels being provided, parameter tuning under unsupervised setting can be dramatically harder than the supervised case. Therefore, we stick with a consistent choice of parameters here for the fairness of comparisons.

The Receiver Operating Characteristic (ROC) curve discussed in section 2.3 is applied as our primal evaluation method, and we only report the area under curve (AUC) as the performance comparison criterion.

5.2 Experiment Results

5.2.1 Results on the Synthetic Data

The computational results for the synthetic data are provided in Table 5.3 and Table 5.4. Table 5.3 presents the AUCs achieved by different algorithms on the synthetic datasets without any feature selection whereas Table 5.4 presents the results after feature selection by BAHSIC-AD .

In Table 5.3, we first focus on the performance of different algorithms on the cases *without* intervention of noisy features, namely, the point anomalies and collective anomalies do not present as contextual anomalies. The corresponding synthetic datasets are ones suffixed with (0) in Table 5.3.

The density based approaches, including LOF and LoOP, achieve top AUCs in detecting point anomalies on the Noisy Two Moon dataset. However, they significantly underperform other methods when dealing with contextual anomalies in Unbalanced Two Moon datasets and Gaussian datasets. This is expected considering that these methods assume the anomalies appear *only* in the low density region. For datasets with collective anomalies, even when the collective anomaly clusters clearly deviate from the normal pattern,

they however form clusters with sufficient density, which causes density based approaches less effective. It is also noticeable that since aLOCI was proposed to handle small anomaly clusters, it indeed outperforms LOF and LoOP in detecting collective anomalies. However it becomes much less effective in handling point anomalies.

Compared with density based approaches, the simplest nearest neighbor approaches perform much better for two moon datasets. Nevertheless, this is mainly due to the mechanism we generate two moons datasets actually favor these methods. The noticeable gap between two moons contributes significantly to their average distance to the nearest neighbors. However, they appear to be much less effective on Gaussian datasets with more than two major clusters present.

SRA in general produces more consistently better ranking among all methods under different scenarios. It can correctly identify the presence of collective anomalies while perform reasonably where in handling point anomalies. As a comparison among kernel based methods, we see OC-SVM is always dominated by SRA, especially in the cases when datasets with multiple patterns are :w present, such as Gaussian(4,4,0).

Now we observe how the algorithms are affected when the anomalies present themselves as contextual anomalies. For almost all algorithms, the injection of noisy features on the original dataset results in a significant performance degradation. An intuitive bar chart on how the performance is affected is depicted in Figure 5.2. When dealing with point anomalies, the methods utilized Gaussian kernels are especially susceptible to the noisy features, as both SRA and OC-SVM has a dramatic decrease in their AUCs. SRA however is relatively more robust with collective anomalies whereas OC-SVM performs consistently worse. One interesting observation we have on other methods is that, the density based methods can get a small boost of performance for collective anomalies when the noisy features are injected. This is mainly due to the fact that injected noisy features actually diluted the points that are originally incorrectly identified.

If we apply BAHSIC-AD first on the datasets with noisy features, and apply each anomaly detection algorithm on the dataset with selected features, their performance is presenting in Table 5.4. We notice that all the algorithms here have identical resultant AUCs as they achieved in the original datasets without noisy features. Also the stopping

iteration are exactly the same as the number of noisy features injected in each dataset. This suggests that BAHSIC-AD indeed correctly eliminates all the noisy features and correctly identify the best contexts for contextual anomalies. This property is especially valuable for methods like SRA which have the drawback of being more sensitive to noisy features. BAHSIC-AD thereby make these methods feasible to detect contextual anomalies that correspond to the feature subset.

Table 5.3: Computational results on synthetic datasets without feature selection

Name	SRA		OC-SVM	LOF	LoOP	k-NN	W k-NN	aLOCI
		mFlg[1]						
Unbalanced Moons (0)	0.9999	0	0.9965	0.8419	0.6089	0.9999	0.9999	0.9350
Unbalanced Moons (5)	0.6612	0	0.6661	0.8846	0.7860	0.9080	0.9317	0.8262
Unbalanced Moons (10)	0.4011	0	0.4715	0.8569	0.7583	0.8561	0.8680	0.4085
Close Unbalanced Moons (0)	0.9887	0	0.9556	0.8698	0.6289	0.9985	0.9958	0.9158
Close Unbalanced Moons (5)	0.9089	0	0.5093	0.8708	0.7772	0.8878	0.9095	0.8131
Close Unbalanced Moons (10)	0.7056	0	0.4918	0.8388	0.7526	0.8378	0.8476	0.4150
Noisy Moons (0)	0.9128	1	0.9118	0.9356	0.9264	0.9235	0.9223	0.6494
Noisy Moons (5)	0.6612	1	0.8069	0.7424	0.7531	0.7691	0.7659	0.5526
Noisy Moons (10)	0.4011	1	0.5155	0.7002	0.7170	0.7250	0.7153	0.4894
Gaussian (3,2,0)	0.9264	0	0.9123	0.5890	0.5865	0.9598	0.8601	0.9235
Gaussian (3,2,5)	0.7949	0	0.5615	0.8393	0.7184	0.8933	0.9013	0.7986
Gaussian (3,2,10)	0.7440	0	0.5725	0.8050	0.7170	0.8071	0.8222	0.4097
Gaussian (4,4,0)	0.9819	0	0.6923	0.5717	0.5197	0.9828	0.9561	0.9803
Gaussian (4,4,5)	0.9785	0	0.6923	0.8959	0.7241	0.9689	0.9695	0.6754
Gaussian (4,4,10)	0.9776	0	0.6653	0.8959	0.7412	0.9218	0.9307	0.3485

[1] mFlg stands for mFLAG produced by SRA

Table 5.4: Computational results on synthetic datasets with feature selection by BAHSIC-AD [2]

Name	BAHSIC-AD	SRA		OC-SVM	LOF	LoOP	k-NN	W k-NN	aLOCI
	sItr[1]		mFlg[1]						
Unbalanced Moons (5)	5	0.9999	0	0.9965	0.8419	0.6089	0.9999	0.9999	0.9350
Unbalanced Moons (10)	10								
Close Unbalanced Moons (5)	5	0.9887	0	0.9556	0.8698	0.6289	0.9985	0.9958	0.9158
Close Unbalanced Moons (10)	10								
Noisy Moons (5)	5	0.9128	1	0.9118	0.9356	0.9264	0.9235	0.9223	0.6494
Noisy Moons (10)	10								
Gaussian (3,2,5)	5	0.9264	0	0.9123	0.5890	0.5865	0.9598	0.8601	0.9235
Gaussian (3,2,10)	10								
Gaussian (4,4,5)	5	0.9819	0	0.6923	0.5717	0.5197	0.9828	0.9561	0.9803
Gaussian (4,4,10)	10								

[1] mFlg stands for mFLAG produced by SRA and sItr stands for number of iteration before stop in BAHSIC-AD

[2] note that, since the datasets in each category become identical after feature selection by BAHSIC-AD . The algorithms therefore have identical performance on datasets in each category, we therefore merge the computational results in that category.

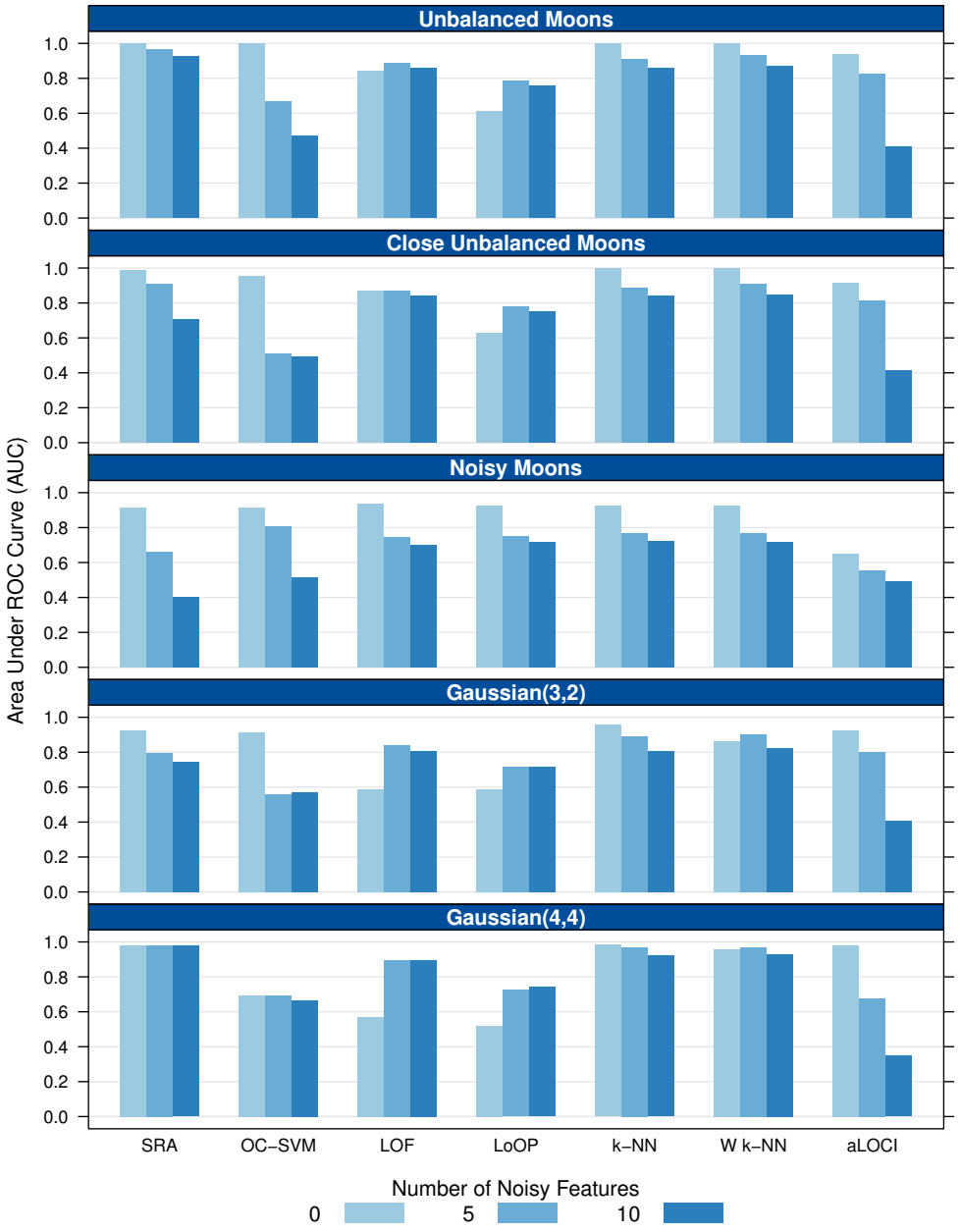

Figure 5.2: Effect of noisy features on the performance of different anomaly detection algorithms on synthetic dataset

5.2.2 Results on the Real World Data

The experiment results on the real world data datasets are provided in Table 5.5, and Table 5.6 presenting the performance of different algorithms on the datasets before and after the feature selection with BAHSIC-AD . Note that, Survival dataset is excluded in Table 5.6 as it contains only 3 features.

By observing the results in Table 5.5, we see different methods have vastly varied performance on different benchmarks. There are some interesting observations worth mentioning here. Similar to the results from the synthetic data, SRA still gives one of the best ranking quality for most of the datasets. This is because many problems being tested originated from a supervised classification problem, and thereby present themselves as a rare class detection problem. SRA with an output of mFLAG = 0 therefore performs a *normal* vs. *abnormal* classification and generates a ranking for the collective anomaly detection.

Compared with SRA, the performance of OC-SVM is always dominated by SRA and the gap is almost always noticeable. Similar to the synthetic cases where density based approaches are less effective in handling collective anomalies, they still suffer the drawback that they perform reasonably well on some of the datasets, such as Thyroid and Shuttle0vs4 while being significantly inferior to other methods on other datasets like Ecoli, Libras. This also confirms the point we made before that types of target anomalies do not always conform with the assumptions made by the density based methods. The nearest-neighbor approaches suffer similar problem, and they rarely provide the best results on the benchmarks. Finally, we notice that aLOCI almost always produces one of the worst ranking results. While it does deserve the merit of being parameter free, the ranking results are however far from being acceptable in general.

The results obtained after applying BAHSIC-AD are given in Table 5.6 and a plot that compares and contrasts the performance of different algorithms, with and without feature selection, is provided in Figure 5.3. In general, most anomaly detection algorithms can achieve better results, on the subsets of features selected by BAHSIC-AD, than their performance on the unfiltered dataset. Especially, we notice that some algorithm originally gives unsatisfactory results can achieve best results after the feature selection, such as LOF on Libras and SRA on Satellite. We believe this is because BAHSIC-AD indeed helps

identify the best contexts for these problems. Nevertheless, it is worth noting that in many cases the improvements are not as significant as the synthetic examples presented before, and there are several cases that BAHSIC-AD actually cause a marginal performance decrease. For example, for the Diabetes dataset, the algorithms that perform reasonably well on the full feature set actually become less effective on selected feature set, and the best AUC is achieved by SRA without feature selection. Nevertheless, except for rare cases like these, SRA can typically benefit from the selected feature set from BAHSIC-AD .

One final important observation we have is the improvement obtained from utilizing Hamming kernels on the nominal only feature set, i.e., Mushroom and Automobile Fraud. By utilizing a Hamming kernel, we see SRA can achieve significantly better results than ones achieved with RBF Gaussian kernel. Meanwhile, other algorithms can only achieve around 0.5 AUCs, which are almost equivalent to random guesses. While not significant in general, OC-SVM also gets a performance boost from using the Hamming kernel. It is especially noticeable when we use OC-SVM on the Mushroom dataset with the selected subset. These observations suggest the importance of introducing proper kernels in handling specific dataset, and simply preprocessing the nominal features by transforming them into binary features is not a good approach for the unsupervised anomaly detection.

In summary, the results are in accordance with the common belief about unsupervised learning that there is hardly a universal method which is applicable for every dataset. It is also important to introduce any prior knowledge by utilizing proper kernels or distance functions. It is thereby crucial for user to choose the right algorithm for a specific problem. However, lacking of the prior knowledge about the nature of the data and the specific type of anomalies to be detected, we observe SRA can handle most of the problems reasonably well, as other approaches fail in one or the other. This is especially true when we incorporate SRA with the proposed BAHSIC-AD. Additionally, it is noticeable that applying BAHSIC-AD can be beneficial in improving the performance of anomaly detection algorithms in general, as it helps to identify a correct context defined by a subset of features.

Table 5.5: Computational results on real world datasets without feature selection

Name	SRA		OC-SVM	LOF	LoOP	k-NN	W k-NN	aLOCI
		mFlg[1]						
Shuttle0vs4	0.9985	0	0.9983	0.9628	0.7355	0.9984	0.9979	0.9705
Satellite	0.4337	1	0.5760	0.4500	0.4672	0.4007	0.4032	0.4314
Ecoli	0.6708	0	0.4852	0.5280	0.4269	0.7019	0.6001	0.2433
YeastME2	0.7473	0	0.6201	0.6624	0.6147	0.7163	0.7092	0.5097
Thyriod	0.8111	0	0.8099	0.8759	0.8643	0.8297	0.8496	0.4715
Glass4	0.8396	0	0.7868	0.8236	0.8553	0.8112	0.8282	0.3810
Libras	0.7924	1	0.5005	0.6463	0.8095	0.6714	0.7352	0.3363
Diabetes	0.7695	0	0.5887	0.6947	0.6150	0.7273	0.7253	0.4724
Survival	0.7001	0	0.5809	0.6720	0.6449	0.6620	0.6645	0.5350
Wine3	0.9904	0	0.4212	0.7732	0.6135	0.6925	0.7144	0.4500
Breast-Wisc	0.9888	0	0.8779	0.6226	0.4962	0.9828	0.9730	0.8614
Zoo	0.9433	1	0.8093	0.9562	0.8299	0.7526	0.9304	0.4742
Mushroom[2]	0.4524	1	0.3528	0.5321	0.5169	0.5083	0.5145	0.4999
	0.8811	1	0.3501					
Automobile Fraud[2]	0.5407	1	0.5321	0.4993	0.5169	0.5083	0.5145	0.4999
	0.7441	1	0.5622					

[1] mFlg stands for mFLAG produced by SRA

[2] for the results of SRA and OC-SVM on Mushroom and Automobile Fraud datasets, first row presents result using RBF Gaussian kernel, and second row presents results using Hamming kernel

Table 5.6: Computational results on real world datasets with feature selection by BAHSIC-AD

Name	BAHSIC-AD	SRA		OC-SVM	LOF	LoOP	k-NN	W k-NN	aLOCI
	sItr[1]		mFlg[1]						
Shuttle0vs4	3	0.9954	0	0.9994	0.9620	0.7314	0.9994	0.9992	0.9669
Satellite	4	0.6567	1	0.5051	0.4505	0.4651	0.4181	0.4181	0.4806
Ecoli	3	0.9081	0	0.4804	0.5000	0.4533	0.6881	0.5882	0.3570
YeastME2	6	0.8506	0	0.7729	0.7780	0.7333	0.8476	0.8480	0.7082
Thyriod	16	0.9552	0	0.9571	0.8925	0.7809	0.9431	0.9319	0.9452
Glass4	2	0.8423	0	0.7891	0.8419	0.8802	0.8270	0.8446	0.3909
Libras	18	0.8084	1	0.8410	0.8762	0.8992	0.8413	0.8955	0.5751
Diabetes	5	0.6244	0	0.5177	0.6586	0.5849	0.6962	0.6840	0.5941
Wine3	6	0.9939	0	0.4678	0.8299	0.6548	0.7413	0.7213	0.5243
Breast-Wisc	4	0.9868	0	0.9188	0.4247	0.4085	0.9698	0.8888	0.9653
Zoo	8	1.0000	1	0.7629	0.8351	0.8144	0.7474	0.8763	0.4330
Mushroom[2]	19	0.8000	1	0.4678	0.5314	0.5678	0.5682	0.5680	0.5000
	13	0.9025	1	0.8743					
Automobile Fraud[2]	21	0.6424	1	0.6618	0.4993	0.4993	0.4993	0.4993	0.5000
	13	0.7526	1	0.5622					

[1] mFlg stands for mFLAG produced by SRA and sItr stands for number of iteration before stop in BAHSIC-AD

[2] for the results of SRA and OC-SVM on Mushroom and Automobile Fraud datasets, first row presents result using RBF Gaussian kernel, and second row presents results using Hamming kernel

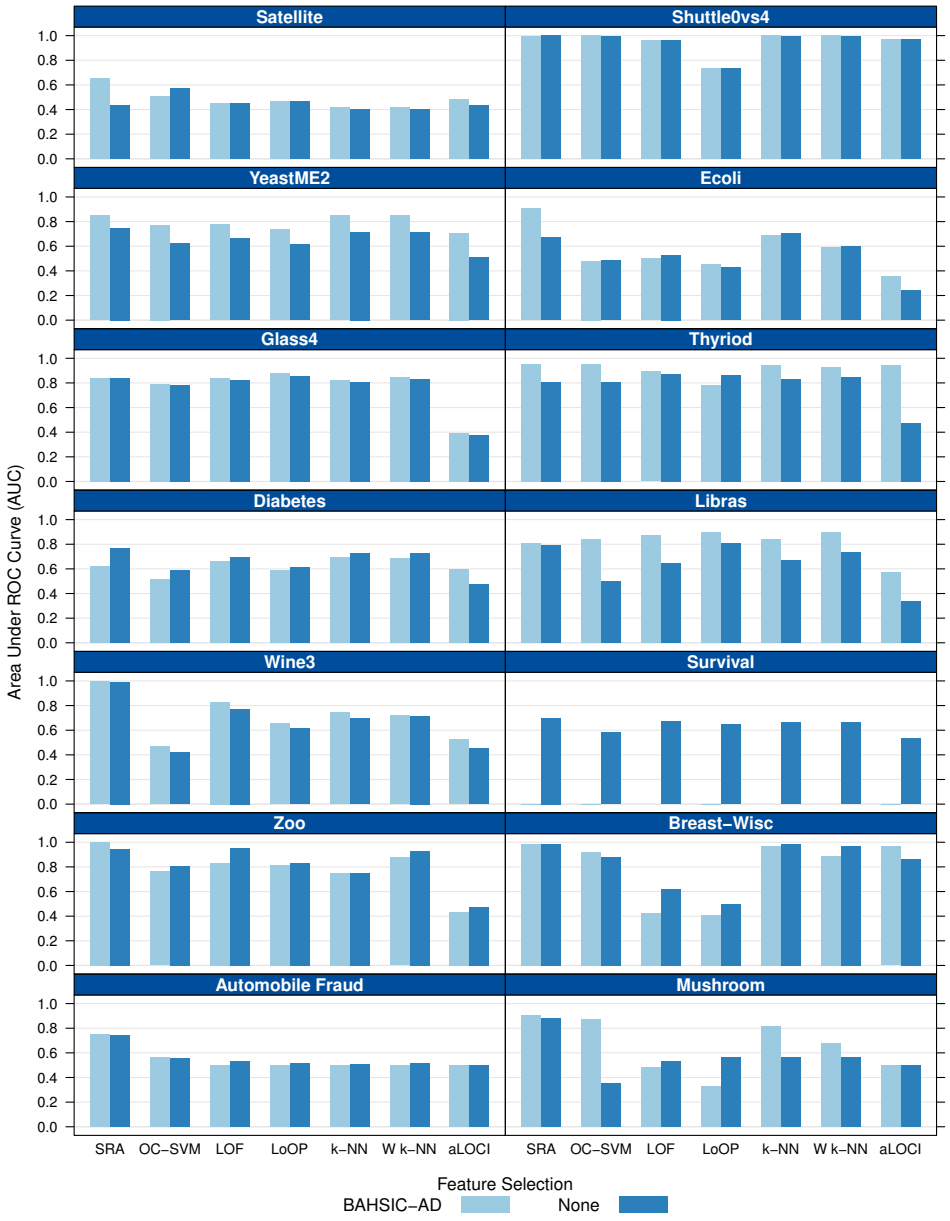

Figure 5.3: Effect of feature selection with BAHSIC-AD on the performance of different anomaly detection algorithms on real world dataset

5.3 Feature Ranking Facilitates Interpretation of Ranking Results

As mentioned in Chapter 2, insurance fraud detection is one of the most important applications of anomaly detection algorithms. This application generally requires highly interpretable results to help people make reasonable decisions, as more interpretable results can be more meaningful and convincing and thus significantly increase the value of the ranking results. For similar reasons, people occasionally sacrifice the accuracy in making prediction with simpler supervised methods like decision trees or logistic regressions, instead of adopting more sophisticated methods like Support Vector Machine or Neural Network, as the results generated by former methods are in general easier to interpret to humans.

In this section, we thereby focus on the automobile insurance fraud dataset that has been utilized as the benchmark in [42] for a more detailed discussion in terms of interpretability of the feature ranking result. We show how the rankings with BAHSIC-AD, can be helpful for interpretability even when methods like SRA already provide reasonable performance, and how the feature ranking quality can be on par with ones generated by supervised methods. From the results presented in Table 5.5, we notice that the feature selection with BAHSIC-AD does not significantly improve the ranking quality for this problem, as the AUC is almost the same as ones applied on the full set of feature. However, we illustrate that the results actually becomes more interpretable.

5.3.1 Feature Importance from Supervised Random Forest

We first consider whether the ranking of the features is reasonable. To find a reliable feature ranking comparison, we apply random forest [7] to generate a feature importance ranking as a trustworthy reference from supervised learning methods. Random Forest trains an ensemble of N_t decision trees, and the prediction of the ensemble is based on the aggregated prediction result of each decision tree. More precisely, we sample a subset X_b, Y_b of training instances form X, Y and train a decision tree f_b on X_b, Y_b. The final

prediction for a new sample u is then made by

$$f_r = \frac{1}{N_t} \sum_{b=1}^{N_t} f_b(u)$$

While training the random forest, an *out-of-bag error* is calculated by taking the mean prediction error on the training sample x_i using the trees which do not have x_i as a training sample. In order to measure the feature importance of the ith feature, the ith is perturbed feature by replacing it with random noise and re-calculate the out-of-bag error on the perturbed dataset. The average difference between out-of-bag error before and after the perturbation is the feature importance measure. In other words, a larger increase of the out-of-bag error suggests a more important feature and the feature thereby gets ranked higher.

5.3.2 Feature Ranking Comparison

Now we compare supervised random forest and BAHSIC-AD in generating the feature ranking for automobile insurance dataset. For BAHSIC-AD, we do not terminate the feature elimination process until the last feature gets eliminated, and the ones get eliminated later ranks higher. The top ranked features from both methods are provided in Table 5.7. Among all 31 features, we notice that the top ranked features significantly overlap with each other, which strongly suggests that the feature selection with HSICs provides a meaningful ranking even without the labels provided. The top ranked features from HSIC can accordingly help fraud investigators to determine a more useful feature subset.

Utilizing SRA on the selected feature subset, we can examine how the top ranked features affect the formation of the clusters in the eigenspace. The information presented in the first and second non-principal eigenvectors of Laplacian, constructed with the full feature set and the selected feature subset are both depicted in Figure 5.4. We notice that the clusters closer to origin have much higher fraud ratios comparing with the clusters that lie further away from the origin. Furthermore, we can utilize this visualization to explore the useful information revealed by the subset of features. Since the dataset consists of only nominal features, we can observe a more concise and succinct representation of the clusters

Ranking	**Random Forest**	**BAHSIC-AD**
1st	base policy	base policy
2nd	party at fault	vehicle category
3rd	vehicle category	past no. of claims
4th	incidence time month	party at fault
5th	claimed time month	age of vehicle
6th	age of policy holder	age of policy holder

Table 5.7: Top ranked features from supervised random forest and HSIC among 31 features of car insurance dataset

with the selected feature subset. This also provides a useful method in identifying what values actually form a suspicious cluster as shown in Figure 5.4, majority of points close to the origin are the cases with collision as the base policy, sedan as the vehicle type and the at-fault party are usually policy holder. This is helpful in justifying how the potential anomalies (ones close to origin) in this scenario correspond to the fraud cases.

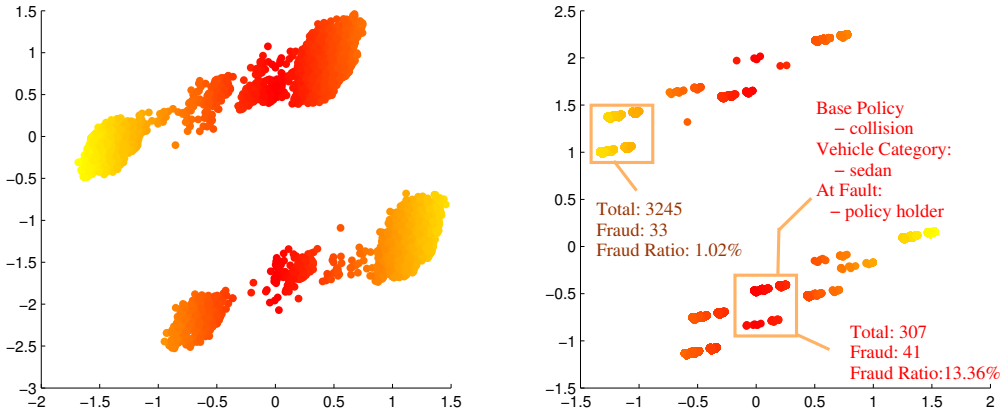

Figure 5.4: $\mathbf{z}_1^*$ and $\mathbf{z}_2^*$ based on the first and second non-principal eigenvectors of Laplacian for Automobile Insurance dataset with the full features (left) and selected feature set (right).

Chapter 6

Conclusions and Future Work

6.1 Conclusion

Anomaly detection has been an active and challenging research area with tremendous practical values in a wide variety of application domains. While the problem formulation can be problem dependent, anomaly detection problems can be roughly classified into three categories: point anomaly detection, collective anomaly detection, and contextual anomaly detection. Many existing methods exist in the literature have been devised to address different anomaly detection problems. Nevertheless, the assumptions made by most of the prevailing approaches usually emphasis on one type of anomalies over the other. Since the exact type of anomalies to be discovered in real world applications is often unknown to users, we want to develop a more general approach that can automatically discover different kinds of anomalies at the same time. This is one of the main motivations behind our work on SRA and unsupervised feature selection with BAHSIC-AD.

In this thesis, we first discuss and analyze the SRA algorithm in greater detail by focusing on its connection with unsupervised SVM. We realize that, with proper assumptions, the spectral optimization in SRA can be viewed as a relaxation of unsupervised SVM problem. Taking this perspective, we observe SRA has the potential to tackle point anomalies and collective anomalies at the same time. Specifically, it provides a bi-class classification

strength measure that can be used to rank the point anomalies and to generate a *normal* vs. *abnormal* classification for identifying the collective anomalies.

For feature-contextual anomaly detection problems with different contexts correspond to different feature subsets. We explore the possibility of utilizing dependence between features as the feature selection criteria and propose a backward elimination filter algorithm BAHSIC-AD. The main assumption of BAHSIC-AD is that the anomalies present as anomalies in the subset of features that has strong dependence with each other. By utilizing HSIC, we can estimate the dependence among features in the space defined by the selected kernel.

We evaluate the effectiveness of SRA by comparing its performance with other popular anomaly detection methods on a collection of benchmarks, including both synthetic datasets and real world datasets. The synthetic datasets simulate different common scenarios of anomaly detection problems and the real world datasets are taken from various application domains. The results confirm that most other popular methods do favor certain types of anomalies over the other, while SRA can deliver a satisfactory results consistently, even when the exact type of anomalies to be targeted is unknown. By detecting contextual anomalies with the help of BAHSIC-AD, the results also demonstrate that BAHSIC-AD are generally helpful in reconstructing the contexts for anomaly detections.

6.2 Possible Future Work

There are several directions which can be further explored. The SRA algorithm solely utilizes the first non-principal eigenvector in generating anomaly ranking. However, as multiple eigenvectors are utilized in spectral clustering, it will be interesting to explore how we can make use of the information present in the additional non-principal eigenvectors.

With respect to the unsupervised feature selection for anomaly detection, especially the application of BAHSIC-AD, it will be interesting to come up with a better stopping criteria to terminate the feature elimination process. While the current strategy do provide certain level of improvement in general, it can be suboptimal. Another interesting direction is to utilize HSIC for parameter selection. Since we see the dependence among features in the

kernel space can be helpful in feature selection, it is also reasonable to assume it can be used for optimizing the parameter selection.

References

[1] Elke Achtert, Hans-Peter Kriegel, Erich Schubert, and Arthur Zimek. Interactive data mining with 3D-parallel-coordinate-trees. In *SIGMOD Conference*, pages 1009–1012, 2013.

[2] J Alcalá, A Fernández, J Luengo, J Derrac, S García, L Sánchez, and F Herrera. Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing*, 17:255–287, 2010.

[3] Fabrizio Angiulli and Clara Pizzuti. Fast outlier detection in high dimensional spaces. In *Principles of Data Mining and Knowledge Discovery*, pages 15–27. Springer, 2002.

[4] K. Bache and M. Lichman. UCI machine learning repository, 2013.

[5] Charles R Baker. Joint measures and cross-covariance operators. *Transactions of the American Mathematical Society*, 186:273–289, 1973.

[6] Richard J Bolton and David J Hand. Statistical fraud detection: A review. *Statistical Science*, pages 235–249, 2002.

[7] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[8] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. LOF: identifying density-based local outliers. In *ACM Sigmod Record*, volume 29, pages 93–104. ACM, 2000.

[9] Simon Byers and Adrian E Raftery. Nearest-neighbor clutter removal for estimating features in spatial point processes. *Journal of the American Statistical Association*, 93(442):577–584, 1998.

[10] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3):15, 2009.

[11] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[12] Fan RK Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.

[13] Julia Couto. Kernel k-means for categorical data. In *Advances in Intelligent Data Analysis VI*, pages 46–56. Springer, 2005.

[14] Richard A Derrig. Insurance fraud. *Journal of Risk and Insurance*, 69(3):271–287, 2002.

[15] Inderjit S Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 269–274. ACM, 2001.

[16] Jennifer G Dy and Carla E Brodley. Feature selection for unsupervised learning. *The Journal of Machine Learning Research*, 5:845–889, 2004.

[17] Levent Ertöz, Michael Steinbach, and Vipin Kumar. *Finding topics in collections of documents: A shared nearest neighbor approach*. Springer, 2004.

[18] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[19] Tom Fawcett. An introduction to ROC analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[20] Kenji Fukumizu, Francis R Bach, and Michael I Jordan. Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. *The Journal of Machine Learning Research*, 5:73–99, 2004.

[21] Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *Algorithmic learning theory*, pages 63–77. Springer, 2005.

[22] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.

[23] James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1):29–36, 1982.

[24] Trevor Hastie, Robert Tibshirani, Jerome Friedman, T Hastie, J Friedman, and R Tibshirani. *The elements of statistical learning*, volume 2. Springer, 2009.

[25] Ville Hautamäki, Ismo Kärkkäinen, and Pasi Fränti. Outlier detection using k-nearest neighbour graph. In *ICPR (3)*, pages 430–433, 2004.

[26] Thomas Hofmann, Bernhard Schölkopf, and Alexander J Smola. Kernel methods in machine learning. *The annals of statistics*, pages 1171–1220, 2008.

[27] Insurance Information Institute. Insurance fraud. Available at: http://www.iii.org/issue-update/insurance-fraud, march 2014. Accessed: 2014-Jun-23.

[28] Teuvo Kohonen. Self-organization and associative memory. *Self-Organization and Associative Memory, 100 figs. XV, 312 pages.. Springer-Verlag Berlin Heidelberg New York. Also Springer Series in Information Sciences, volume 8*, 1, 1988.

[29] Yufeng Kou, Chang-Tien Lu, and Dechang Chen. Spatial weighted outlier detection. In *SDM*. SIAM, 2006.

[30] Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek. Loop: local outlier probabilities. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1649–1652. ACM, 2009.

[31] Huan Liu and Lei Yu. Toward integrating feature selection algorithms for classification and clustering. *Knowledge and Data Engineering, IEEE Transactions on*, 17(4):491–502, 2005.

[32] Stuart Lloyd. Least squares quantization in PCM. *Information Theory, IEEE Transactions on*, 28(2):129–137, 1982.

[33] Larry M Manevitz and Malik Yousef. One-class svms for document classification. *the Journal of machine Learning research*, 2:139–154, 2002.

[34] Biswanath Mukherjee, L Todd Heberlein, and Karl N Levitt. Network intrusion detection. *Network, IEEE*, 8(3):26–41, 1994.

[35] Julio F Navarro, Carlos S Frenk, and Simon DM White. A universal density profile from hierarchical clustering. *The Astrophysical Journal*, 490(2):493, 1997.

[36] Andrew Y Ng, Michael I Jordan, Yair Weiss, et al. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2:849–856, 2002.

[37] Ke Nian, Haofan Zhang, Aditya Tayal, Thomas Coleman, and Yuying Li. Auto insurance fraud detection using unsupervised spectral ranking for anomaly. *Submitted to Journal of Risk and Insurance*, 2014.

[38] Matthew Eric Otey, Amol Ghoting, and Srinivasan Parthasarathy. Fast distributed outlier detection in mixed-attribute data sets. *Data Mining and Knowledge Discovery*, 12(2-3):203–228, 2006.

[39] Spiros Papadimitriou, Hiroyuki Kitagawa, Phillip B Gibbons, and Christos Faloutsos. Loci: Fast outlier detection using the local correlation integral. In *Data Engineering, 2003. Proceedings. 19th International Conference on*, pages 315–326. IEEE, 2003.

[40] Vern Paxson. Bro: a system for detecting network intruders in real-time. *Computer networks*, 31(23):2435–2463, 1999.

[41] Hanchuan Peng, Fulmi Long, and Chris Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(8):1226–1238, 2005.

[42] Clifton Phua, Damminda Alahakoon, and Vincent Lee. Minority report in fraud detection: classification of skewed data. *ACM SIGKDD Explorations Newsletter*, 6(1):50–59, 2004.

[43] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. In *ACM SIGMOD Record*, volume 29, pages 427–438. ACM, 2000.

[44] Volker Roth. Kernel fisher discriminants for outlier detection. *Neural computation*, 18(4):942–960, 2006.

[45] Stan Salvador, Philip Chan, and John Brodie. Learning states and rules for time series anomaly detection. In *FLAIRS Conference*, pages 306–311, 2004.

[46] Bernhard Schölkopf, Christopher JC Burges, and Alexander J Smola. *Advances in kernel methods: support vector learning*. MIT press, 1999.

[47] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

[48] John Shawe-Taylor and Nello Cristianini. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[49] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.

[50] Le Song, Alex Smola, Arthur Gretton, Justin Bedo, and Karsten Borgwardt. Feature selection via dependence maximization. *The Journal of Machine Learning Research*, 98888(1):1393–1434, 2012.

[51] Xiuyao Song, Mingxi Wu, Christopher Jermaine, and Sanjay Ranka. Conditional anomaly detection. *Knowledge and Data Engineering, IEEE Transactions on*, 19(5):631–645, 2007.

[52] Arian R Van Erkel and Peter M Th Pattynama. Receiver operating characteristic (roc) analysis: basic principles and applications in radiology. *European Journal of radiology*, 27(2):88–94, 1998.

[53] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[54] Li Wei, Weining Qian, Aoying Zhou, Wen Jin, and X Yu Jeffrey. Hot: Hypergraph-based outlier test for categorical data. In *Advances in Knowledge Discovery and Data Mining*, pages 399–410. Springer, 2003.

[55] Jason Weston, André Elisseeff, Bernhard Schölkopf, and Mike Tipping. Use of the zero norm with linear models and kernel methods. *The Journal of Machine Learning Research*, 3:1439–1461, 2003.

[56] Ian H Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[57] Stella X Yu and Jianbo Shi. Multiclass spectral clustering. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 313–319. IEEE, 2003.

[58] Haofan Zhang, Ke Nian, Thomas Coleman, and Yuying Li. Spectral ranking and unsupervised feature selection for point, collective and contextual anomalies. *Unpublished manuscript*, 2014.

[59] Ji Zhang and Hai Wang. Detecting outlying subspaces for high-dimensional data: the new task, algorithms, and performance. *Knowledge and information systems*, 10(3):333–355, 2006.

[60] Mu Zhu. Kernels and ensembles. *The American Statistician*, 62(2), 2008.