Anomaly detection based on zero appearances in subspaces

by

Guansong Pang

Thesis

Submitted by Guansong Pang

for fulfillment of the Requirements for the Degree of

Master of Philosophy (Information Technology) (3337)

Supervisor: Professor Kai-Ming Ting

Associate Supervisor: Dr. David Albrecht

Clayton School of Information Technology
Monash University

April, 2015

Copyright Notices

Notice 1

Notice 2

To my wife Lisa and my son Louis

Contents

List of tables

List of figures

Anomaly detection based on zero appearances in subspaces

Guansong Pang
[redacted]
Monash University, 2015

Supervisor: Professor Kai-Ming Ting
[redacted]
Associate Supervisor: Dr. David Albrecht
[redacted]

Abstract

Anomaly detection is regarded as one of the most important tasks in data mining due to its wide application in various domains, such as finance, information security, healthcare and earth science. With advancements in data collection techniques, the volume and dimensionality of anomaly detection data sets increase explosively, and diverse attribute types occur within these data sets. Also, in many data sets, anomalies can be detected in some attributes only, while other attributes are irrelevant to anomaly detection. All these characteristics pose new challenges to existing anomaly detection techniques. Motivated by this fact, this research aims to design an anomaly detection method which can scale up to large and high dimensional data, is able to identify anomalies in data sets with different types of attributes, and tolerates irrelevant attributes.

This thesis posits that anomalies are instances with low probabilities in subspaces in a data set. So, in a random subset of the data set, anomalies have higher probabilities of having zero appearances in the subspaces than normal instances. Based on this property, this thesis proposes a novel anomaly detection method called ZERO++ which employs the number of zero appearances in subspaces to detect anomalies. ZERO++ is the only anomaly detector based on zero appearances in subspaces, as far as we know. It is unique in that it works in regions of subspaces that are not occupied by data; whereas other methods work in regions occupied by data. Utilising the anti-monotone property: 'if an instance has zero appearances in a subspace, it must also have zero appearances in subspaces containing this subspace', we show that only a small number of subspaces with low dimensionality needs to be considered to identify anomalies effectively. ZERO++ is an efficient algorithm with linear time complexity with respect to data size and data dimensionality, and it can work effectively in data sets with different types of attributes, and a low percentage of relevant attributes.

Anomaly detection based on zero appearances in subspaces

Declaration

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Guansong Pang
April 27, 2015

Acknowledgments

I would like to express my deepest appreciation and thanks to my supervisors, Professor Kai-Ming Ting and Dr. David Albrecht, for their tremendous support and patience, and for their fruitful and excellent advice. Their guidance helped me a lot through all the time of researching on the master project and writing of this thesis.

Besides my supervisors, I would like to thank the rest of my candidature confirmation committee: Professor Bala Srinivasan and Dr. Nayyar Zaidi, for their encouragement and insightful comments.

I would like to thank Dr. Janice Miller for her excellent editing and proofreading of this thesis.

I would like to express my special thanks to Dr. Huidong Jin, who, as a collaborator and a friend, was always happy to help and provided insightful suggestions to my research.

I would also like to thank my colleagues Ye Zhu, Fei Tony Liu, Tharindu Bandaragoda, Sunil Aryal and Jonathan Wells for their support and help in my research and daily life.

I wholeheartedly thank my parents for eternal love and support throughout my life. I thank my beloved wife Lisa. She was always there with me through the good times and the bad times.

Guansong Pang

Monash University
April 2015

Chapter 1

Introduction

Anomalies are data patterns that are rare and exceptional compared to the majority of data. Detecting anomalies is attractive and valuable because finding such patterns often uncovers either underlying treasures or potential hazards.

Anomaly detection generally refers to the process of finding anomalies. It is regarded as one of the most important tasks in data mining due to its wide application in various domains, such as finance, information security, healthcare and earth science. A key challenge in anomaly detection is to identify anomalies accurately and efficiently in ever growing complex data sets, e.g., very large and high-dimensional data sets with different types of attributes. In recent years, a number of techniques have been proposed to handle this challenge with varying degrees of success. In this research, we break down this key challenge into four components, and propose a novel anomaly detection method to meet all the four challenges.

This chapter provides an introduction to the research subject and research motivation in Sections 1.1 and 1.2, respectively, and states our contributions in Section 1.3. The organisation of this thesis is then presented in Section 1.4.

1.1 Research subject

Anomaly detection refers to the process of identifying abnormal instances [1] in data. Abnormal instances, or anomalies, may have different meanings in different application domains. Finding anomalies are very important for all these domains, because it may uncover new treasures because of the discovery of rare patterns, or prevent catastrophic consequences of anomalous events. Research on anomaly detection dates back to as early as the 19th century (Chandola et al., 2009). Anomaly detection has been intensively studied in recent years, and it is regarded as one of the four most important tasks in data mining, besides classification, cluster analysis, and association analysis (Tan, Steinbach and Kumar, 2006). This section presents a brief introduction to anomaly detection in terms of the definition of anomaly, types of anomaly, anomaly detection techniques and their applications.

1.1.1 Definition of anomaly

Anomalies are referred to as outliers, exceptions, aberrations, abnormalities, novelties, deviants and discordants in different domains (Aggarwal, 2013a; Chandola et al., 2009). Anomalies and outliers are the two most widely used terms and are often interchangeable in the data mining community. One classic definition of anomaly is given by Hawkins

[1] Instances are often referred to as points and records in the computer science community, and samples and observations in the statistics community.

(Hawkins, 1980) as "*An outlier is an observation that differs so much from other observations as to arouse suspicion that it was generated by a different mechanism*", but there are different definitions given from different perspectives. From a geometrical perspective, Johnson et al. (1998) and Kriegel and Zimek (2008) assume anomalies are at the boundaries in the data space, whereas Knorr and Ng (1997), Breunig et al. (2000) and He et al. (2003) generally assume that anomalies lie in regions with low density. A more recent definition is from the isolation concept which is motivated by the fact that anomalies are susceptible to isolation, i.e., anomalies can be isolated using significantly fewer partitions than those for normal instances (Liu et al., 2012).

The above definitions are based on numeric data (i.e., data sets with numeric attributes only), and they apply to numeric domain only. For categorical data (i.e., data sets with categorical attributes only), one widely used assumption is that anomalies occur infrequently (or rarely) in the feature space (Ghoting et al., 2004; Koufakou and Georgiopoulos, 2010; He, Xu, Huang and Deng, 2005). As far as we know, there is no widely used definition of anomaly in mixed data (i.e., data sets with both numeric and categorical attributes). A big challenge in dealing with mixed data is that it requires the capture of the definitions of anomaly from both the numeric domain and categorical domain, and also the interaction between these two heterogeneous domains (Ghoting et al., 2004; Koufakou and Georgiopoulos, 2010; Zhang and Jin, 2011)

In this research, an anomaly is defined as an instance that occurs rarely in a categorical data set. To apply to numeric or mixed data sets, we utilise a discretisation method to convert numeric attributes to categorical attributes before applying the proposed anomaly detection method.

1.1.2 Types of anomaly

Anomalies can generally be divided into three categories: *point anomalies*, *conditional* (or *contextual*) *anomalies* and *collective anomalies* (Chandola et al., 2009). Point anomalies and conditional anomalies refer to individual instances only, while collective anomalies are based on a collection of instances. An instance is considered as a point anomaly if it is anomalous compared to other instances in a data set. Conditional anomalies have a similar definition as point anomalies except that they are defined with some conditions. The conditions vary in different applications e.g., location and time are commonly used as a condition to define anomalies in spatial data and time series data, respectively. A collective anomaly is a collection of data instances where each instance by itself appears normal but together exhibit anomalous behaviour. Such anomalies often occur in sequence data, graph data and spatial data.

Point anomalies can be further classified into *global anomalies* and *local anomalies* based on the view of neighbourhood, and *scattered anomalies* and *clustered anomalies* based on their distributions. Global anomalies are anomalous instances which are far away from both sparse and dense normal clusters in the feature space, while local anomalies are instances located near dense normal clusters but far away from sparse normal clusters. Scattered anomalies refer to anomalies having a scattered distribution. In contrast, clustered anomalies are anomalies which are very close to each other and form a small cluster.

It should be noted that the concepts of global and local anomalies, and scattered and clustered anomalies rely on the key characteristic of numeric data, i.e., the notion of ordering. As far as we know, these concepts are not well defined in categorical data because there is no ordering in categorical attribute values. Existing research on anomaly detection for categorical data (Ghoting et al., 2004; Koufakou and Georgiopoulos, 2010; Zhang and Jin, 2011; Das et al., 2008; Wu and Wang, 2013) focuses on point anomalies, which are

simply defined as instances with low frequency in the feature space. This research also focuses on point anomalies only.

1.1.3 Anomaly detection and its applications

Anomaly detection has been studied in several research communities, e.g., statistics, data mining, machine learning and information theory. Numerous anomaly detection methods have been proposed over the years, including statistical test based methods (Barnett and Lewis, 1994), depth-based methods (Johnson et al., 1998), angle-based methods (Kriegel and Zimek, 2008), distance-based methods (Knorr and Ng, 1997), density-based methods (Breunig et al., 2000) and clustering-based methods (He et al., 2003). Ensemble methods for anomaly detection have been explored in recent years (Aggarwal, 2013b; Zimek et al., 2012). We introduce these methods in detail in Chapter 2.

Anomaly detection techniques have wide application in various domains. Examples of application are presented as follows in terms of the notion of anomaly and its implications (Chandola et al., 2009; Aggarwal, 2013a).

- **Intrusion detection**. In intrusion detection tasks, anomalies refer to malicious activities in a network or a computer system. Anomaly detection techniques help monitor and analyse network or computer system events for intrusions.
- **Fraud detection**. Anomalies generally refer to frauds in this domain, including credit card transaction frauds, insurance claim frauds and insider trading. Detection of such anomalies can prevent related organisations from huge financial loss.
- **Healthcare**. In the healthcare domain, anomalies often refer to unusual conditions of patients (indicating certain diseases), or disease outbreaks. Early detection of such anomalies allows more time for treatment or prevention of the spread of disease.
- **Fault detection**. In this domain, anomalies often refer to faults in mechanical components such as motors, turbines and engines. Early detection of these faults can prevent catastrophic events such as aircraft crashes.
- **Image and video processing**. Examples of anomalies are irregularities in images and unusual changes in videos over time. Typical application scenarios are mammography image analysis, satellite image analysis and video surveillance.

1.2 Research motivation

Compared to normal instances, anomalies typically account for only a very small portion of a data set. Identifying anomalies is like 'finding a needle in a haystack'. With advancements in data collection and storage techniques, data sets have become more and more complex, e.g., large data size, high dimensionality, different types of attributes and data noise. This makes anomaly detection much more challenging. Particularly, this research is motivated by the following four challenges in anomaly detection:

1. **Ability to handle data sets with different types of attributes**. Diverse data types exist in many real-world anomaly detection applications, such as numeric pixel attributes derived from images and videos, boolean-value based or multiple-label [2] based categorical attributes in medical data, mixed attributes in demographic data and network intrusion data. This results in data sets with different types of attributes, i.e., data sets with numeric attributes only, data sets with categorical attributes only, and data sets with mixed attributes.

[2] Unordered labels are referred to as categorical attribute values in this thesis.

2. **High detection accuracy**. High detection accuracy is an essential requirement in anomaly detection. This is because false negative detections bear very high costs in many real-world applications, such as fraudulent transaction detection and early detection of cancer diseases.

3. **Scale up to very large data size and high dimensionality**. With advancements in data collection techniques, anomaly detectors are often required to be able to detect anomalies in very large and high dimensional data quickly. High detection accuracy in many detectors comes at a cost to computational efficiency. These detectors often cannot scale up well in terms of data size or data dimensionality.

4. **Tolerant to irrelevant attributes**. Anomalies are often only detectable in some attributes. Other attributes, which are irrelevant attributes to anomaly detection tasks, often mask anomalies. This is particularly true for high dimensional data (Zimek et al., 2012). Many anomaly detection applications are high dimensional domains, e.g., over 100 dimensions are used to describe instances in credit card fraudulent transaction detection (Pham and Pagh, 2012) and disease diagnostics (Guvenir et al., 1997).

A number of techniques have been proposed to handle these challenges with varying degrees of success. There are a few methods (Ghoting et al., 2004; He, Xu, Huang and Deng, 2005; Koufakou and Georgiopoulos, 2010; Zhang and Jin, 2011) that are proposed to handle categorical or mixed data, but their time complexity is at least quadratic in terms of data dimensionality or data size. Many other existing anomaly detection methods are numeric data oriented methods, including statistical test based methods (Aggarwal, 2013a; Barnett and Lewis, 1994), depth-based methods (Tukey, 1977; Johnson et al., 1998), distance-based methods (Knorr and Ng, 1997; Knox and Ng, 1998), density-based methods (Breunig et al., 2000; Papadimitriou et al., 2003), clustering-based methods (He et al., 2003; Jiang et al., 2006) and isolation-based methods (Liu et al., 2010, 2012). Also, widely used methods like ϵ-neighbourhood (Knox and Ng, 1998), kNN (k-th Nearest Neighbour) distance (Ramaswamy et al., 2000) and LOF (*Local Outlier Factor*) (Breunig et al., 2000) have at least $O(n^2)$ [3] time complexity. Although it can be reduced to $O(n \log n)$ if an indexing scheme such as R^*-tree (Beckmann et al., 1990) is employed, most indexing methods work in low dimensional data sets only, and they break down in high dimensionality. Moreover, many existing methods (Angiulli and Pizzuti, 2002; Angiulli and Fassetti, 2009; Knox and Ng, 1998; Bay and Schwabacher, 2003; Breunig et al., 2000; He et al., 2003; Ramaswamy et al., 2000) use full dimensionality to define anomalies and thus fail to detect anomalies in data sets with high percentages of irrelevant attributes due to the *curse of dimensionality* (Zimek et al., 2012).

Anomaly detection using ensemble techniques is an emerging research direction (Aggarwal, 2013b; Zimek, Campello and Sander, 2013). There are mainly two types of anomaly detection ensembles, i.e., subspace-based methods (Lazarevic and Kumar, 2005; Keller et al., 2012) and subsampling-based methods (Zimek, Campello and Sander, 2013; Sugiyama and Borgwardt, 2013). These ensembles are often based on conventional anomaly detection methods, such as LOF, and thus share similar defects, e.g., they are unable to handle data sets with different types of attributes effectively, and are unable to scale up with data size or data dimensionality.

Some ensemble-based methods are based on both subspace-based methods and subsampling-based methods, such as isolation-based methods (Liu et al., 2012). They build models on randomly selected attribute subsets and subsamples, and have linear time complexity in

[3] In this thesis, n and d denote data size and data dimensionality, respectively

terms of data size and dimensionality, but they are also numeric data oriented methods and are sensitive to irrelevant attributes.

This situation has motivated us to design a novel method which can provide a solution to all these four challenges.

1.3 Contributions

This research aims to produce an anomaly detection method that will meet all the four challenges stated in Section 1.2, i.e., a highly accurate detection method which can: scale up to very large and high dimensional data, tolerate irrelevant attributes, and effectively handle data sets with different types of attributes. We aim to demonstrate the effectiveness and efficiency of our proposed method in both theoretical and empirical analyses.

To this end, we will focus on subsampling-based ensemble methods, which have favourable scalability in terms of both data size and data dimensionality, as reported in Ting et al. (2013), Liu et al. (2012), and Sugiyama and Borgwardt (2013). Also, we will explore subspace-based anomaly scoring functions, which are insensitive to irrelevant attributes (Kriegel, Kröger, Schubert and Zimek, 2009; Keller et al., 2012) and able to handle mixed attributes effectively (Ghoting et al., 2004; Koufakou and Georgiopoulos, 2010).

The contributions of this thesis are as follows:

- This thesis proposes a categorical data based anomaly detection method which identifies anomalies based on zero appearances in subspaces. A statistical justification is provided to explain why our proposed method works.
- Two discretisation methods are examined to extend our proposed method to numeric data and mixed data.
- A series of experiments is conducted to evaluate the effectiveness and efficiency of our proposed method. It is shown that our proposed anomaly detector is able to detect anomalies more accurately and efficiently than existing state-of-the-art anomaly detectors.
- An empirical evaluation of existing state-of-the-art anomaly detectors is conducted on data sets with different types of attributes.

1.4 Organisation

The rest of this thesis is organised as follows.

Chapter 2 provides a review of related achievements in this research area and discusses their strengths and limitations. We first discuss two types of conventional anomaly detection methods, including extreme value analysis based methods and proximity-based methods. We then review relatively new established anomaly detection methods, namely ensemble methods for anomaly detection. Finally, we discuss techniques for categorical and mixed data.

Chapter 3 presents a novel anomaly detection method which is based on zero appearances in subspaces. We first present our motivation and statistical justification of the anomaly score used in our proposed method. We then discuss how our categorical data based method can be extended to handle numeric and mixed data. Finally, we explain the characteristics of our method and provide a conceptual comparison with related anomaly detectors.

Chapter 4 provides an empirical evaluation of our proposed method. We examine the detection performance, ability to tolerate irrelevant attributes, scalability and sensitivity

of our proposed method using a range of data sets. We also apply our proposed method for identifying anomalies in data sets with unknown ground truth.

The thesis is concluded in Chapter 5.

Chapter 2

Literature review

A wide range of methods have been proposed for anomaly detection over the years. Based on the extent to which the methods use class labels (i.e., labels assigned to each instance as being either *normal* or *anomalous* in a given data set), they can be generally categorised into supervised methods, semi-supervised methods and unsupervised methods. Supervised methods employ labelled instances of both the normal class and the anomalous class to train detection models. Some examples of these methods are Support Vector Machines (SVMs) and Neural Networks (Mukkamala et al., 2002). Semi-supervised methods require labelled instances of the normal class only, in order to train their detection models, e.g., one-class SVMs (Ma and Perkins, 2003). Unsupervised methods do not require labelled instances. Examples of unsupervised methods are statistical test based methods, distance-based methods, density-based methods and clustering-based methods (Aggarwal, 2013a). All methods make (explicit or implicit) assumptions on behaviours of normal instances or abnormal instances and detect anomalies by examining how instances conform to the behaviours.

Compared to supervised methods and semi-supervised methods, unsupervised methods are more widely used in industry, because obtaining accurate labelled data for anomaly detection often has a very high cost (Chandola et al., 2009). Particularly, collecting accurate labelled data often requires substantial effort to manually assign the labels, and obtaining labelled abnormal data is prohibitively expensive in many application domains such as early detection of catastrophic events (e.g., spread of epidemic diseases, terrorist activities and aircraft faults). Also, it is difficult to collect all types of anomalies as new types of anomalies might emerge in new data.

This research focuses on unsupervised methods, and we review unsupervised methods for anomaly detection only. Surveys of semi-supervised and supervised methods can be found in Chandola et al. (2009) and Görnitz et al. (2014).

This chapter provides a literature review of conventional anomaly detection techniques in Section 2.1, including extreme value analysis based methods and proximity-based methods. We review a newly established technique for anomaly detection, i.e., ensemble learning methods in Section 2.2, followed by methods for categorical and mixed data in Section 2.3. This chapter is then summarised in Section 2.4.

2.1 Conventional anomaly detection techniques

Following Kriegel, Kröger and Zimek (2009) and Aggarwal (2013a), traditional anomaly detection methods can be broadly divided into extreme value analysis based methods and proximity-based methods.

2.1.1 Extreme value analysis based methods

Statistical test based methods, depth-based methods and angle-based methods are generally based on the assumption that anomalies are points with extreme values in the feature space. Based on this definition, we denote these methods as extreme value analysis based methods.

Statistical test based methods

Statistical test based methods (Barnett and Lewis, 1994; Aggarwal, 2013a) assume that all data instances are generated by a certain type of statistical distribution, such as *Gaussian*. Statistical test methods, e.g., t-test and χ^2, are then used to determine the probabilities of data values along with a statistical significance level. Instances lying at the tail (i.e., extreme values) of the given distribution are identified as anomalies. To quantify the lower and upper probabilistic tail bounds of the distribution, a number of tail inequalities can be used, such as *Hoeffding Inequality* and *Chernoff Inequality* (Aggarwal, 2013a).

These methods have well established probabilistic and statistical properties to interpret anomaly detection results. Also, such methods can be used in the final stage of other anomaly detection methods to report anomalies with a statistical significance level (Das et al., 2008). However, these methods are parametric methods that make assumptions on data distributions. They are also very sensitive to noise and anomalies. For example, the mean and standard deviation estimation of *Gaussian* distribution can be severely biased by noise and anomalies.

Depth-based methods

Depth-based methods (Tukey, 1977; Johnson et al., 1998) conventionally define instances lying on the outer layers of a convex hull (Jarvis, 1973) as anomalies. These methods operate in an iterative way to obtain the anomaly scores of instances: all instances located at the corners of the convex hull are removed iteratively until the data set becomes empty. An instance has $depth = k$ if it is removed in the k-iteration. Instances with depth less than a threshold r are considered as anomalies.

Depth-based methods share a similar methodology as statistical test based methods, but it should be noted that depth-based methods are non-parametric methods that do not assume any data distribution. One typical limitation of these methods is their high time complexity in convex hull computation. The brute force convex hull computation method has $O(n^5)$. It can be reduced to $O(n \log n)$ for data sets with two and three dimensions by using a *divide-and-conquer* technique, but it increases exponentially with data dimensionality (Preparatat and Shamos, 1985; Knox and Ng, 1998).

Angle-based methods

The basic assumption in angle-based methods (Kriegel and Zimek, 2008; Pham and Pagh, 2012) is that anomalies lie at the boundaries of the data space. Therefore, compared to normal instances in the inner regions, anomalies have smaller angles to pairs of instances in the data set. Given a data set D and a test instance $\mathbf{x}$, its anomaly score is the variance over angles between $\mathbf{x}$ to any pairs of instances in D. Instances with higher angle variances are more likely to be anomalies.

The first angle-based method ABOD (Angle-Based Outlier Detection) was proposed by Kriegel and Zimek (2008) and is dedicated for anomaly detection in high dimensional data. The time complexity of these methods is determined by the computation cost of the angle variance between $\mathbf{x}$ to pairs of instances. The brute force method computes the angles between $\mathbf{x}$ to all pairs of instances in the data set. This has $O(n^3)$ time complexity.

An approximate method proposed in Kriegel and Zimek (2008) uses the variance over angles between $\mathbf{x}$ to pairs of instances from the k nearest neighbours to approximate the original variance. This reduces the time complexity to $O(n^2k)$. A further near linear time approximation method was proposed in Pham and Pagh (2012). Angle-based methods are free of parameters, which is one big advantage over many existing methods. It has been reported in Kriegel and Zimek (2008) that these methods could alleviate the effects of the curse of dimensionality compared to anomaly detection methods using a distance concept. However, it should be noted that, as discussed in Aggarwal (2013a), angle-based measures such as *cosine* are influenced by concentration effects (Zimek et al., 2012) and irrelevant attributes in high dimensional data.

Strength and weakness. Extreme value analysis based methods have good statistical or geometrical interpretation of anomalies, and they can obtain high detection accuracy when anomalies contain extreme values. However, in many real-world applications, anomalies can be surrounded by normal clusters. In such cases, the detection performance of these methods will decrease substantially.

2.1.2 Proximity-based methods

Distance-based methods, density-based methods and clustering-based methods are popular anomaly detection methods because of their simplicity and intuitive interpretation. The basic assumption in these methods is that anomalies lie in regions with low density.

Distance-based methods

Distance-based methods make use of the distance of an instance to its nearest neighbours to define proximity. Instances with large nearest neighbour distances have sparse proximity, and thus can be reported as anomalies. Seminal work on distance-based methods is $DB(\pi, \epsilon)$-$Outliers$, where π is a fraction of a data set D and ϵ is a distance threshold. The method was proposed in Knorr and Ng (1997), in which an instance $\mathbf{x}$ is considered as an anomaly if at least π percent of instances in D have distance to $\mathbf{x}$ greater than ϵ.

Alternatively, the definition can be interpreted as: $\mathbf{x}$ is reported as an anomaly if at most $(1-\pi)$ percent of instances in D have distance to $\mathbf{x}$ smaller than ϵ. This alternative definition facilitates the ϵ-neighbourhood method proposed in Knox and Ng (1998), in which it proposes to use indexing techniques such as k-d trees (Bentley, 1975) to conduct a range search with radius ϵ. An instance is considered as an anomaly if no more than M instances are found in the ϵ-neighbourhood. Excluding the time complexity of indexing techniques, the ϵ-neighbourhood method has $O(n^2d)$ time complexity. A nested-loop pre-process method was also proposed in Knox and Ng (1998) in order to avoid the expensive indexing construction time cost for some application contexts (e.g., high dimensional data), but the nested-loop based ϵ-neighbourhood anomaly detection method still has $O(n^2d)$ time complexity. A cell-based pre-process method was also proposed in Knox and Ng (1998), in which it built grids such that any two instances within the same cell have at most ϵ distance to each other. This method reduces the time complexity of the ϵ-neighbourhood method, to be linear with respect to n but exponential with d.

Ramaswamy et al. (2000) simplifies the ϵ-neighbourhood anomaly definition and identifies anomalies based on the distance of an instance to its k-th nearest neighbour. Instances with large k-th nearest neighbour distance are reported as anomalies. This reduces the number of parameters from two to one, i.e., k. In order to search for kNN efficiently, Ramaswamy et al. (2000) also introduces a partition-based pre-process method, which uses linear-time clustering methods to partition instances into disjoint subsets and pruned redundant instances. A number of other techniques (Bay and Schwabacher, 2003; Angiulli

and Pizzuti, 2002; Angiulli and Fassetti, 2009) have been proposed to speed up the distance computation in kNN search, which can reduce the time complexity of kNN search to near-linear time in some contexts, but can break down when dealing with large and high dimensional data.

Density-based methods

Density-based methods are generally based on the assumption that anomalies lie in regions of relatively low density. An instance is considered as an anomaly if the ratio of its density to that of its local neighbourhood is small. Since neighbourhood distance based methods use a single distance threshold (e.g., k-th distance) to measure anomalousness, they fail to detect anomalies in data sets with normal clusters of varying densities. Motivated by this fact, Breunig et al. (2000) proposes the Local Outlier Factor (LOF) method, which can detect anomalies with the above-mentioned data characteristic.

The LOF of an instance is computed as the mean ratio of its average reachability distance to that of its neighbours. The Reachability Distance (RD) of an instance $\mathbf{x}$ with respect to $\mathbf{y}$ is defined as:

$$RD(\mathbf{x}, \mathbf{y}) = max\{d_k(\mathbf{y}), dist(\mathbf{x}, \mathbf{y})\}$$

where $d_k(\mathbf{y})$ is the kth nearest neighbour distance to $\mathbf{y}$ and $dist(\mathbf{x}, \mathbf{y})$ denotes the distance between $\mathbf{x}$ and $\mathbf{y}$. It is evident that $RD(\mathbf{x}, \mathbf{y})$ is not symmetric between $\mathbf{x}$ and $\mathbf{y}$ because the kth nearest neighbour distance to $\mathbf{x}$ and $\mathbf{y}$ may be different. This asymmetric property helps highlight $\mathbf{x}$ when $\mathbf{x}$ locates in regions with relatively low density. The Local Reachability Distance (LRD) of $\mathbf{x}$ is inverse of the Average Reachability Distance (ARD) of $\mathbf{x}$ to its k nearest neighbours, as defined below:

$$LRD(\mathbf{x}) = \frac{1}{ARD(\mathbf{x})} = \frac{1}{\frac{\sum_{\mathbf{y} \in kNN(\mathbf{x})} RD(\mathbf{x}, \mathbf{y})}{Card(kNN(\mathbf{x}))}} = \frac{Card(kNN(\mathbf{x}))}{\sum_{\mathbf{y} \in kNN(\mathbf{x})} RD(\mathbf{x}, \mathbf{y})}$$

where $kNN(\mathbf{x})$ denotes the set of k nearest neighbours of $\mathbf{x}$ and $Card(kNN(\mathbf{x}))$ is the cardinality of $kNN(\mathbf{x})$. The LOF of $\mathbf{x}$ is then defined as:

$$LOF(\mathbf{x}) = \frac{\sum_{\mathbf{y} \in kNN(\mathbf{x})} \frac{LRD(\mathbf{y})}{LRD(\mathbf{x})}}{Card(kNN(\mathbf{x}))}$$

Instances with $LOF \approx 1$ are located within a cluster; while instances with $LOF \gg 1$ are considered to be anomalies. The only parameter in LOF, k, plays a crucial role in its performance. This parameter acts as a smoothing factor in computing the anomaly scores. Larger values of k lead to greater smoothing. The detection performance of LOF is dependent on the choice of the k value. In practice, a range of k values is employed to compute $LOF(\mathbf{x})$ and the maximum LOF value is used as the anomaly score of $\mathbf{x}$. Motivated by the success of LOF, a variety of LOF variants has been proposed, such as Connectivity-based Outlier Factor (COF) (Tang et al., 2002) and LOcal Correlation Integral (LOCI) (Papadimitriou et al., 2003). COF improves LOF by giving a different treatment to isolated instances and instances in regions of low density. LOCI replaces k nearest neighours with ϵ-neighbourhood and uses multiple granularities of ϵ-neighbourhood to define the anomaly factor. In LOCI, ϵ can be automatically determined, and thus the method does not require parameter tuning.

Clustering-based methods

Many clustering-based anomaly detection methods proceed in a two-phase fashion. Instances are normally clustered into disjoint groups in the first phase. Some criteria based on the clustering results are then used to identify anomalies. One typical criterion is the cluster size. Jiang et al. (2001) first employs modified k-means clustering to partition all instances into clusters, and then reports instances belonging to small clusters as anomalies. Compared to the original k-means clustering, a cluster splitting-and-merging procedure is added into the modified version in Jiang et al. (2001) and allows the final number of clusters to be larger than k. This results in many small and medium sized clusters. Instances contained in the small clusters are considered as anomalies. This method can detect clustered anomalies but can also report small normal clusters as anomalies.

Another straight-forward criterion is the distance of a given instance to its closest cluster centroid. The larger the distance is, the more likely the instance is to be an anomaly (Aggarwal, 2013a). This type of method is able to find isolated anomalies, but it might fail to detect clustered anomalies, because clustered anomalies can be very close to the anomaly cluster and their distance to the cluster is rather small.

Instead of using a single criterion, He et al. (2003) makes use of both cluster size and the distance of the instance to its closest cluster centroid to define anomalies. In another method, instead of deriving an instance outlier factor based on the clustering results, Jiang et al. (2006) employs the distances between clusters to design a cluster outlier factor, and then labelled clusters as either normal or abnormal using a threshold. Instances are considered as anomalies if the class label of their closest cluster is abnormal.

In general, clustering-based methods are more suitable for sparse data than distance-based and density-based methods, because clusters are aggregated representations which can well represent the sparse data. Most clustering-based methods cannot provide an anomalous degree of an instance, because they only produce a binary class label about whether instances are anomalies or not.

Strength and weakness. Proximity-based methods are straight-forward and easy-to-implement, and thus they are widely used methods. However, distance computation is an essential component within these methods. Such methods do not work effectively in high dimensional data due to the curse of dimensionality, and they are also sensitive to irrelevant attributes because they use the full dimensionality to define distance (Zimek et al., 2012). Another major issue for this type of method is that the distance computation requires $O(n^2)$ time complexity, and thus cannot scale up to very large data sets. Although the distance computation can be reduced to $O(n \log n)$ when instances are preprocessed by indexing methods such as R^*-tree (Beckmann et al., 1990), most indexing methods work in low dimensional data sets only, and they break down in high dimensionality.

2.2 Ensemble methods for anomaly detection

Ensemble learning is a well established research area and has wide application in classification and clustering (Dietterich, 2000), but it has been rarely applied in anomaly detection (Aggarwal, 2013b; Zimek, Campello and Sander, 2013). Dozens of anomaly detection ensembles proposed in recent years have shown promising improvement in the detection performance of traditional anomaly detection methods. These ensembles can be divided into subspace-based methods and subsampling-based methods. Subspace-based methods build a set of anomaly detectors on the full data set with subsets of attributes, while subsampling-based methods build the anomaly detectors using data subsets with all the attributes. Very little work has been done using both techniques, i.e., build detectors on data subsets with subsets of attributes.

2.2.1 Subspace-based methods

Subspace-based anomaly detection methods are motivated by the fact that anomalies are detectable using some subsets of attributes only. This is particularly true for high dimensional data. Therefore, most subspace-based methods were proposed to identify anomalies in high dimensional data. *FeatureBagging* (Lazarevic and Kumar, 2005) is seminal work exploring how to combine results from different detection models built on attribute subsets. Specifically, Lazarevic and Kumar (2005) first employs a traditional anomaly detection method, i.e., LOF, to construct a set of models on data with randomly selected attribute subsets (subspaces). It then investigates two different strategies, including *breadth-first* and *cumulative-sum*, to combine anomaly scores from different models. In the breadth-first strategy, instances are simply assigned with the highest anomaly score from all models, while instances are assigned with the sum of all the anomaly scores from the models in the cumulative-sum strategy. Unlike Lazarevic and Kumar (2005) who uses a single detector on the attribute subsets, Nguyen et al. (2010) examines the effectiveness of using heterogeneous detectors on different randomly selected attribute subsets. One major limitation in Lazarevic and Kumar (2005) and Nguyen et al. (2010) is that randomly selected attribute subsets might contain irrelevant attributes; in the worst case, all the selected attributes are irrelevant attributes. Current research in this direction mainly focuses on how to select informative subspaces.

Kriegel, Kröger, Schubert and Zimek (2009) selects informative attributes for an instance based on the variance of instances in a reference set of the instance. Given an instance $\mathbf{x}$ and its reference set $Ref(\mathbf{x})$, i.e., a local neighbourhood of $\mathbf{x}$, the method aims to find a subspace hyperplane $\mathcal{H}$ spanned by $Ref(\mathbf{x})$, where the variance of instances in $Ref(\mathbf{x})$ is high; while in its perpendicular subspace $\mathcal{S}$, the variance of the instances in $Ref(\mathbf{x})$ is low. The instance $\mathbf{x}$ is considered as an anomaly if it deviates significantly from its reference instances in the subspace hyperplane $\mathcal{H}$. Such a deviation is captured by the average *Euclidean* distance from $\mathbf{x}$ to the centre of each attribute in the perpendicular subspace $\mathcal{S}$. The deviation is called Subspace Outlier Degree (SOD), which is defined as:

$$SOD_{Ref(\mathbf{x})}(\mathbf{x}) = \frac{\sqrt{\sum_{A_i \in \mathcal{S}} (\mathbf{x}_i - \mu_i)^2}}{|\mathcal{S}|}$$

where A_i denotes a specific attribute in the subspace $\mathcal{S}$, $\mathbf{x}_i$ is the attribute value of $\mathbf{x}$ in A_i, μ_i is the mean value of all the instances in $Ref(\mathbf{x})$ in A_i, and $|\mathcal{S}|$ denotes the number of attributes in $\mathcal{S}$. The critical component in SOD is to find a meaningful reference set for a given instance. In order to reduce the effect of the curse of dimensionality in high dimensional data, Kriegel, Kröger, Schubert and Zimek (2009) adopts the *Shared Nearest Neighbours* (SNN) (Houle et al., 2010) measure to identify the reference set, because "*even though all points are almost equidistant to a given point p, a nearest neighbour ranking of the data objects is usually still meaningful*", as argued in Kriegel, Kröger, Schubert and Zimek (2009). Specifically, for the instance $\mathbf{x}$, let $kNN(\mathbf{x})$ denotes the k nearest neighbours with respect to the Euclidean distance, the SNN similarity between $\mathbf{x}$ and $\mathbf{y} \in D$ is $Sim_{SNN}(\mathbf{x}, \mathbf{y}) = Card(kNN(\mathbf{x}) \cap kNN(\mathbf{y}))$, and the reference set of $\mathbf{x}$ consists of l most similar instances with respect to Sim_{SNN}. The SNN computation for each instance is a time-consuming process, which has $O(dn^2)$ time complexity. When $k \ll n$ and $l \ll n$, the total time complexity of SOD is $O(dn^3)$. Thus, the effectiveness of SOD comes at a high computational time cost. Also, SOD only considers anomalousness on a single dimension basis, in order to reduce computational cost, and thus cannot detect anomalies exhibited in subspaces with two or more dimensions.

Muller et al. (2011) proposes the subspace-based method OUTERS, which detects anomalies in subspaces with any number of dimensions. However, OUTERS can only work on small data sets with very low dimensionality, because its time complexity is exponential to data dimensionality and quadratic to data size. The method called HiCS (Keller et al., 2012) combines conditional *Probability Density Function* (PDF) and *Welch*'s t-test to find *high contrast subspaces*, which are defined as subspaces where anomalies can be clearly distinguished from other instances, based on a well defined notion of anomalousness. LOF is then used to identify anomalies based on the high contrast subspaces. HiCS is able to find anomalies in subspaces with varying numbers of dimensions. Since HiCS uses the Apriori-like mechanism (Agrawal et al., 1996) to generate candidate subspaces, it is more scalable to dimensionality then OUTERS, but its time complexity is still quadratic to data size.

Strength and weakness. The first and simple approach is to randomly select some attribute subsets to construct the subspaces. These methods have comparable time complexity to proximity-based methods. However, since the process of attribute subset selection is random, irrelevant attributes can be selected. Therefore, these methods do not work well in data sets with a large percentage of irrelevant attributes. The second approach aims to search informative subspaces in a preprocessing step before employing anomaly detection methods. These methods can overcome the sensitivity to irrelevant attributes but they often have expensive time computation, e.g., at least quadratic to data size or data dimensionality.

2.2.2 Subsampling-based methods

Compared to subspace-based methods, less work has been done in subsampling-based methods. The work by Zimek, Gaudet, Campello and Sander (2013) is one of the early attempts to investigate how subsampling techniques could be used to improve detection efficiency and effectiveness over a single local anomaly detector [1]. It uses a traditional local anomaly detection method such as LOF, as a base method to build a set of models on a set of subsamples derived from the full data set. Given a data set D and the base detector LOF, the anomaly score of an instance $\mathbf{x}$ is average over all the scores from all models of the ensemble, as defined below:

$$EnLOF(\mathbf{x}|D) = \frac{1}{t}\sum_{i=1}^{t} LOF(\mathbf{x}|\mathcal{D}_i)$$

where t is the ensemble size, i.e., the number of models built in the ensemble, $\mathcal{D}_i$ is a subsample with randomly selected r percent of instances from the full data set D, and $LOF(\mathbf{x}|\mathcal{D}_i)$ denotes the LOF score of $\mathbf{x}$ based on $\mathcal{D}_i$. It is argued in Zimek, Gaudet, Campello and Sander (2013) that subsampling can help distinguish anomalies and normal instances as it increases the gap between anomalies and normal instances in the anomaly ranking, and it is able to induce diversity into the ensemble. In Wu and Jermaine (2006) and Sugiyama and Borgwardt (2013), the authors theoretically and empirically demonstrate how subsampling techniques could be used to enhance global anomaly detection methods in terms of both effectiveness and efficiency. Global methods such as ϵ-neighbourhood and kNN-distance can be used as base methods in this ensemble method.

[1] *Local* anomaly detection methods consider the relative densities as anomaly scores, which are ratios of the density of an instance to the densities of its neighbourhood, whereas *global* methods, such as $DB(\pi, \epsilon)$-*Outliers*, ϵ-neighbourhood and kNN distance, compute the anomaly score for each instance based on the global neighbourhood.

Strength and weakness. These subsampling-based methods are based on traditional anomaly detection methods, so they would inevitably inherit the weaknesses of the traditional methods, e.g., sensitivity to irrelevant attributes and the curse of dimensionality in high dimensional data. The time complexity of these methods is strongly dependent on the ensemble size and subsampling size. Subsampling can obtain favourable speed-up over the base method if small ensemble size and small subsampling size are employed. However, since traditional methods like LOF require a sufficiently large number of instances to approximate the neighbourhood of an instance, the subsampling size is required to be fairly large, e.g., 10 percent of the data set. Also, the ensemble size needs to be set large enough, e.g., at least 10, to introduce diversity. Building the ensemble, using $t = 10$ and 10% of D instances, will take about the same runtime as building a single model using the same base method on the full data set. In Zimek, Gaudet, Campello and Sander (2013), a fixed number of 25 models on subsamples with 10% of D instances is used by default, and this ensemble has higher time complexity than the base method, by a factor of roughly 2.5 times.

2.2.3 Using both subspace-based and subsampling-based methods

Very limited work has been done on ensemble methods based on both subsamples and subspaces. iForest (Isolation Forest)(Liu et al., 2012) is seminal work in this field. iForest utilises the property, that anomalies are susceptible to isolation, to build isolation trees to identify anomalies. Each tree is grown using a subsample until every instance is isolated, where the attribute and cut-point at each node are randomly selected. To score a test instance, the path length traversed from the root to a leaf node by the test instance is then used as the anomaly score. Because anomalies can be isolated using significantly fewer partitions than normal instances, anomalies have a shorter path length than normal instances. Given an instance $\mathbf{x}$, the anomaly score is defined as follows:

$$Score(\mathbf{x}) = 2^{-\frac{E(h(\mathbf{x}))}{c(\psi)}}$$

where $h(\mathbf{x})$ denotes the path length, $E(h(\mathbf{x})) = \frac{1}{t}\sum_{i=1}^{t} h_i(\mathbf{x})$ is the average path length of $\mathbf{x}$ from a set of t isolation trees, $c(\psi)$ is the expected average path length given the subsample size ψ and can be estimated by $ln(\psi) + 0.5772156649$ (Euler's constant).

MassAD (Mass-based Anomaly Detection) (Ting et al., 2013) utilises mass estimation techniques to detect anomalies. Mass is simply the number of instances in a region, which is formed by axis-parallel splits using a subsample with randomly selected attributes. Instances, which fall in sparse regions frequently, would have low mass values and are considered as anomalies. In addition to the methodology, iForest and MassAD also share some other features, e.g., they both use the average of anomaly scores from all the models as the final score. It is worthwhile noting that the path length used in iForest is a proxy to mass, as discussed in Ting et al. (2013).

Strength and weakness. iForest and MassAD require no distance computation and have linear time complexity in terms of data size and data dimensionality. On the other hand, they have some common weaknesses, e.g., they are very sensitive to irrelevant attributes because they work on a few randomly selected attributes in each subsample.

2.3 Methods for categorical and mixed data

All the methods in previous sections are numeric data oriented [2]. Compared to methods for numeric data, less work has been done for categorical and mixed data. In order to identify anomalies in categorical or mixed data, one way is to convert categorical attributes into numeric attributes, and then employ numeric data oriented methods; another way is to directly design methods based on the characteristics of categorical or mixed data.

2.3.1 Categorical-to-numeric transformation

Existing research focuses on embedding a transformation from categorical attributes to numeric attributes into a distance definition, because it facilitates the evaluation of different transformation methods.

Diverse methods have been proposed from this perspective. Occurrence frequency based methods assign higher weight to frequent categorical values, while inverse occurrence frequency based methods assign less weight to these values (Lin, 1998). A comparative study between these methods was conducted in Boriah et al. (2008). The results show no single distance measure can obtain consistent superiority over other measures. In other words, such transformation methods are application context dependent; and it is thus difficult to find a universally effective method for different data sets. For mixed data, another major challenge is how to effectively combine the distance computation results in mixed attributes (Huang, 1997).

These transformation methods can be well integrated into proximity-based anomaly detection methods, but they are not applicable for other types of methods, such as extreme value analysis based methods and isolation-based methods (Aggarwal, 2013a; Liu et al., 2012).

One commonly used transformation method in the data mining and machine learning community is to convert categorical attributes into binary attributes using the 1-of-ℓ transformation method (Hall et al., 2009; Aggarwal, 2013a; Zhang and Jin, 2011). In this method, a ℓ-label attribute is first converted into ℓ binary attributes. The binary attributes are then regarded as numeric attributes, along with the original numeric attributes, to be further processed. A major limitation of this method is that the number of attributes in the converted data would be much larger than that in its original form if the categorical attributes contain many labels. This may render detectors less effective due to the curse of dimensionality (Aggarwal, 2013a). The advantage of this method is that it can be easily used by different types of detectors.

2.3.2 Categorical or mixed data oriented methods

Most anomaly detection methods for categorical data are pattern based methods, including normal pattern based methods and anomaly pattern based methods. FPOF (Frequent Pattern based Outlier Factor) (He, Xu, Huang and Deng, 2005), a well known method dedicated for categorical data, employs the *Apriori* method (Agrawal et al., 1993) to generate frequent itemsets as normal patterns. If instances satisfy few or none of the frequent itemsets, they are considered as anomalies. Let $FPS(D, \delta)$ be the frequent itemsets with *support* no less than a given minimum support δ in the data set D. For a test instance $\mathbf{x}$, its anomaly score is computed as follows:

$$FPOF(\mathbf{x}) = \frac{\sum\limits_{g \subseteq \mathbf{x} \wedge g \in FPS(D,\delta)} support(g)}{|FPS(D, \delta)|}$$

[2] More details about the ability of existing detectors to handle specific data types will be presented in the analysis of strength and weakness at the end of this section.

where g is a frequent itemset, $g \subseteq \mathbf{x}$ denotes g satisfies $\mathbf{x}$, and $support(g)$ returns the support of g. FPOF has been widely used and reported as one of the most effective methods (Koufakou et al., 2007; Wu and Wang, 2013). Its time complexity is linear to data size but at least quadratic to dimensionality size.

In contrast to FPOF, some methods search for anomaly patterns to detect anomalies. These patterns can be infrequent itemsets (Ghoting et al., 2004; Koufakou and Georgiopoulos, 2010) or Bayesian Network rules (Das et al., 2008). These methods perform comparably to FPOF, but they also cannot scale up with dimensionality. There has been some information-theoretic based methods (He, Deng and Xu, 2005; Wu and Wang, 2013) for categorical data, which formalised anomaly detection as an optimisation problem to minimise the uncertainty in a data set by using some information-theoretic measures, such as entropy. Instances are considered as anomalies if removing these instances can minimise the uncertainty of the data set. These methods often work as top-k anomaly detectors, which return the k top ranked anomalies.

For categorical data oriented detectors, in order to deal with numeric or mixed data, numeric attributes are first discretised into multiple bins, and the discretised attributes, along with the original categorical attributes, are then further processed by the detectors. Some widely used discretisation methods are the equal-frequency and equal-width methods (Hall et al., 2009), but it should be noted that different detectors have different requirements on discretisation granularity, so their performance is often sensitive to the number of bins predefined in discretisation methods. However, compared to categorical-to-numeric transformation, discretisation is a simpler process because no ordering information is required for categorical attributes, and it is a well established research area (Liu et al., 2002).

LOADED (Link-based Outlier and Anomaly Detection in Evolving Data sets) (Ghoting et al., 2004) is seminal work dedicated to anomaly detection in mixed data. For categorical data, LOADED searches infrequent itemsets, which consist of categorical values in distinct attributes. The anomaly score of a test instance is inverse to the length of infrequent itemsets appearing in the instance. For mixed data, LOADED uses correlations of numeric attributes on an itemset basis to measure the anomalousness of the test instance. This helps to capture dependencies between two types of attributes. Though an approximation scheme is employed, LOADED has high time complexity, which is quadratic to the number of numeric attributes and is exponential to the number of categorical attributes.

ODMAD (Outlier Detection for Mixed Attribute Datasets) (Koufakou and Georgiopoulos, 2010) also searches infrequent patterns in order to compute the anomaly score in terms of categorical attributes. For numeric attributes, ODMAD first generates centroids of instances containing a specified categorical value, and then employs cosine similarity between test instances and the centroids to identify anomalies. Results in Koufakou and Georgiopoulos (2010) show that ODMAD performed better than LOADED in terms of both effectiveness and efficiency. The time complexity of ODMAD is linear to data size and the number of numeric attributes, but it still increases exponentially with the number of categorical attributes.

Strength and weakness. Most existing anomaly detection methods are numeric data oriented. Categorical attributes are required to be transformed into numeric attributes in order to make these methods applicable for categorical and mixed data, but their performance is application context dependent. Limited anomaly detectors have been proposed to deal with categorical and mixed data directly. To treat mixed data, categorical data oriented detectors can be employed with discretisation methods, but their detection performance is often sensitive to the number of bins used in the discretisation methods.

Table 2.1: A summary of the ability of existing anomaly detection methods to meet the four challenges stated in Section 1.2. The four challenges include the ability to handle data sets of different types of attributes (A), high detection accuracy (B), scale up to very large data size and high dimensionality (C) and tolerant to irrelevant attributes (D). The mark "×" denotes the methods generally cannot address a particular challenge, while "√" indicates the methods can often meet the challenge.

Methods	A	B	C	D
Extreme value analysis based	×	√	×	×
Representative: ABOD	×	√	×	×
Proximity-based	×	√	×	×
Representative: LOF	×	√	×	×
Subspace-based	×	√	×	√
Representative: SOD	×	√	×	√
Subsampling-based	×	√	×	×
Representative: EnLOF	×	√	×	×
Subspace and subsampling based	×	√	√	×
Representative: iForest	×	√	√	×
Categorical or mixed data oriented	×	√	×	√
Representative: FPOF	×	√	×	√

For mixed data oriented methods, in order to capture the interaction between numeric and categorical attributes, their treatments of categorical (numeric) attributes are based on the results of handling numeric (categorical) attributes. In data sets with categorical (numeric) attributes only, they do not have results from numeric (categorical) attributes to assist the processing of categorical (numeric) attributes, and thus they are unable to detect anomalies directly. Also, most existing categorical or mixed data oriented methods can handle low dimensional data effectively, but their computation time quadratically increases with data dimensionality.

2.4 Chapter summary

Anomaly detection is an important research area in data mining and has been studied intensively in recent years. A variety of methods have been proposed, including extreme value analysis based methods and proximity-based methods. Anomaly detection using ensemble learning techniques is an emerging research direction due to its advantages in dealing with high dimensional data, and its potential effectiveness and efficiency benefits over single detectors. It should be noted that ensemble based methods often use extreme value analysis based methods and proximity-based methods as base methods. A summary of the ability of existing anomaly detection methods to meet the four challenges stated in Section 1.2 is presented in Table 2.1. It shows that existing methods can handle the four challenges with varying degrees of success, but none of the existing methods can meet all the four challenges. Motivated by this fact, this research aims to design a novel method to cope with all four challenges in a unified framework.

Chapter 3

ZERO++: A novel anomaly detection method

In this chapter, we propose a novel anomaly detection method, which employs the number of zero appearances in subspaces to detect anomalies. Our proposed method is called ZERO++ because its anomaly score involves the sum of the number of ZERO appearances in subspaces over a set of subsamples (i.e., a double summation ++).

Most existing anomaly detection methods rely on the key characteristic of numeric data, i.e., the notion of ordering. For example, extreme value analysis based anomaly detection methods employ the ordering information to identify anomalies with extremely large or small values. For proximity-based methods, the ordering information is used to define neighbourhood and identify anomalies that lie in regions of low density. However, these methods cannot handle categorical data, which is inherently unordered or lacks continuity in attribute values. Though techniques in transforming categorical data into numeric data allow these anomaly detectors to treat categorical data, their detection performance is often context dependent. Therefore, these methods can generally work well in numeric data, but they often fail to obtain favourable detection performance in categorical data and mixed data.

In contrast, ZERO++ is based on categorical data. To handle numeric data or mixed data, numeric attributes are discretised into categorical attributes prior to employing our proposed method. Discretisation is a well established field (Liu et al., 2002) and it is a process simpler than the one which requires a reverse conversion in most existing methods, because no ordering information is required for categorical attributes. As such, ZERO++ is in a better position to treat mixed data.

Based on the property that anomalies have a higher probability of having zero appearances in subspaces and in subsamples than normal instances, ZERO++ aims to use the number of zero appearances in subspaces over a set of subsamples to identify anomalies. A major challenge in this motivation is that the number of subspaces is exponential to data dimensionality, so it is inapplicable for high dimensional data. However, zero appearances in subspaces follow *the anti-monotone property*, which states that 'if an instance has zero appearances in a subspace, it must also have zero appearances in subspaces containing this subspace'. Utilising this property, an efficient and effective approximation method is proposed by using a small set of low dimensional subspaces only.

This chapter presents the intuition of our proposed method in Section 3.1, followed by the introduction of our method in Section 3.2. Next, we discuss how ZERO++ can be extended to handle numeric and mixed data in Section 3.3. We then present an analysis of the characteristics of ZERO++ and its algorithmic framework in Sections 3.4 and 3.5, respectively. A comparison between ZERO++ and anomaly detection methods in most

related work is provided in Section 3.6. This chapter ends with a chapter summary in Section 3.7. The symbols and notations used are provided in Table 3.1.

Table 3.1: Symbols and notations

D	A data set with d attributes, where $\|D\| = n$
$\mathcal{D}$	A subsample of D, where $\|\mathcal{D}\| = \psi$
$\mathcal{A}$	A product set of all the attributes in D
$\mathcal{S}$	A subspace having a product set of m attributes
R	Set of all the subspaces in D
R_m	Set of all the m-dimensional subspaces
R'_m	A random subset of R_m
$\mathbf{y}$	An instance in D
$r(\mathbf{y})$	Frequency count of $\mathbf{y}$ in D, i.e., $r(\mathbf{y}) = \|\{\mathbf{x} \in D : \mathbf{x} = \mathbf{y}\}\|$
$r_{\mathcal{S}}(\mathbf{y})$	Frequency count of $\mathbf{y}$ in $\mathcal{S}$ and in D
$P_{\mathcal{S}}(\mathbf{y}\|\mathcal{D})$	The probability of $\mathbf{y}$ in $\mathcal{S}$ given $\mathcal{D}$
$\mathcal{Z}_{\mathcal{S}}(\mathbf{y})$	A binary variable: whether $\mathbf{y}$ has zero appearances in $\mathcal{S}$ given $\mathcal{D}$, or not
t	The number of subsamples

3.1 Intuition

In a categorical data set, anomalies are rare instances, i.e., those instances which have combinations of values that are rare. Furthermore, in a random subsample, the probability of having no instances in the subsample with the same values as a given test instance, on any attribute subset, increases monotonically with a decrease in the frequencies of the values in the full data set. Therefore, anomalies are likely to have zero appearances in small subsamples, and also have a higher probability of having zero appearances than normal instances in subsamples of any size (see Definition 2 in Section 3.2.1 for the formal definition of zero appearances). Based on this property, we propose to employ the number of subspaces having zero appearances in subsamples to identify anomalies. Instances with a high number of zero appearances in subspaces will have a high anomaly score. To demonstrate this intuition, we provide two examples: one using univariate data and another using multivariate data.

Given a univariate categorical data set with 1,000 instances, and there exists an anomaly with 10 appearances and a normal instance with 100 appearances. When randomly subsampling eight instances without replacement from the data set, the probability of the anomaly having zero appearances in the subsample is $\frac{\binom{1000-10}{8}}{\binom{1000}{8}} \approx 0.9225$, whereas that of the normal instance is $\frac{\binom{1000-100}{8}}{\binom{1000}{8}} \approx 0.4291$ only. Therefore, in a set of subsamples, anomalies are likely to have a higher number of zero appearances than normal instances.

In multivariate data sets, the rarity and exception characteristics of anomalies are reflected in subspaces. Therefore, compared to normal instances, anomalies are likely to have a larger number of zero appearances in subspaces. A big challenge in examining zero appearances of instances in subspaces is its time and space complexities exponentially increasing with data dimensionality, i.e., the number of all the subspaces is $\sum_{m=1}^{d} \binom{d}{m} = 2^d - 1$ in total for D. This number could be too big to store in memory, e.g., $d = 50$ requires about one petabyte of main memory, and the runtime for examining zero appearances in all these subspaces is prohibitive for only a few dozen instances.

However, zero appearances in subspaces follow the anti-monotone property, which states that 'if an instance has zero appearances in a subspace, it must also have zero appearances in subspaces containing this subspace'. Utilising this property, we provide an effective and efficient approximation to identify anomalies by considering a small set of low dimensional subspaces only.

An application of the anti-monotone property for a simple two-dimensional subspace is shown in Figure 3.1. Let $\mathcal{S}_{Ai}$ be the region $A = i$ in the one-dimensional subspace of attribute A.

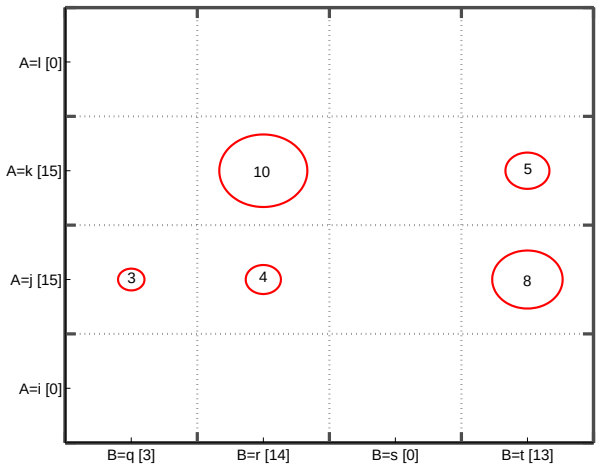

Figure 3.1: A two-dimensional categorical data subset with 30 instances, where the size of the circle indicates the number of instances in a region; and the number in [ ] indicates the number of instances in the region of one-dimensional subspace. Labels for A are $\{i, j, k, l\}$ and labels for B are $\{q, r, s, t\}$

.

Given the data distribution shown in Figure 3.1, regions of zero appearances in one-dimensional subspaces are $\mathcal{S}_{Ai}$, $\mathcal{S}_{Al}$ and $\mathcal{S}_{Bs}$ only. Any test instance having either $A = i$, $A = l$ or $B = s$ is more likely to be an anomaly. Since instances must have zero appearances in higher-dimension subspaces of either $A = i$, $A = l$ or $B = s$, we only need to examine regions in these one-dimensional subspaces.

In high dimensional data sets, the anti-monotone property substantially reduces the number of subspaces that need to be examined. Although using zero appearances in low dimensional subspaces to approximate zero appearances in all the subspaces may lose some accuracy, it gains significant reduction in time and space complexities, i.e., the time and space complexities are reduced from 2^d to d, and in Chapter 4 we will empirically show that our approximation can identify anomalies more effectively than state-of-the-art anomaly detectors in a wide range of real-world and synthetic data sets.

3.2 ZERO++: The anomaly detection method

In ZERO++, based on our argument in Section 3.2.1, which states that anomalies have a higher probability of having zero appearances in subspaces and in subsamples than normal instances, in Section 3.2.2, our anomaly score computation method is introduced based on the number of zero appearances in subspaces over a set of subsamples. Since the time and space complexities of examining zero appearances in all subspaces are prohibitive

for high dimensional data sets, we provide an efficient approximation for anomaly score computation using a small set of low dimensional subspaces only in Section 3.2.3.

3.2.1 Zero appearances in subspaces

In this thesis, a subspace refers to a m-attribute subspace in a categorical space and it is defined as a product set of m attributes. Formally, let D be a set of i.i.d. instances $\{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ with d categorical attributes, and $\mathcal{A} = A_1 \times A_2 \times \ldots \times A_d$ be a product set of the attributes; and $\mathbf{y} = [y_1, \ldots, y_d]$.

Definition 1 *A subspace is a product set of m attributes*

$$\mathcal{S} = A_{k_1} \times A_{k_2} \times \ldots \times A_{k_m}, \tag{3.1}$$

where $1 \leq k_1 < k_2 \ldots < k_m \leq d$.

Let $\mathcal{D}$, with ψ randomly selected instances (sampling without replacement), be a random subset of D; and $I_{\mathcal{S}}(\mathbf{x} = \mathbf{y})$ be an indicator function, which is 1 if instance $\mathbf{x}$ is identical to $\mathbf{y}$ in subspace $\mathcal{S}$, and 0 otherwise.

Definition 2 *An instance $\mathbf{y}$ has zero appearances in $\mathcal{S}$ given $\mathcal{D}$, if $\forall \mathbf{x} \in \mathcal{D}$, $I_{\mathcal{S}}(\mathbf{x} = \mathbf{y}) = 0$, i.e., $\{\mathbf{x} \in \mathcal{D} : [x_{k_1}, \ldots, x_{k_m}] = [y_{k_1}, \ldots, y_{k_m}]\} = \emptyset$.*

For example, assume A_1 and A_2 are two attributes of D, and A_1 contains three values a_1, a_2, a_3 and A_2 contains two values b_1, b_2, then $S = A_1 \times A_2 = \{(a, b) : a \in A \& b \in B\} = \{(a_1, b_1), (a_1, b_2), (a_2, b_1), (a_2, b_2), (a_3, b_1), (a_3, b_2)\}$. In a subsample $\mathcal{D}$ of D, suppose only $\{(a_1, b_1), (a_1, b_2), (a_2, b_1), (a_2, b_2)\}$ occur, then (a_3, b_1) and (a_3, b_2) have zero appearances in S given $\mathcal{D}$.

In a given data set, anomalies are instances with low probabilities in subspaces. So, in a random subset of the data set, anomalies are likely to have zero appearances in the subspaces. This is demonstrated in the following theorem [1] and its corollaries.

Definition 3 *$\mathcal{Z}_{\mathcal{S}}(\mathbf{y}) = 1$ if $\mathbf{y}$ has zero appearances in $\mathcal{S}$ given $\mathcal{D}$, and $\mathcal{Z}_{\mathcal{S}}(\mathbf{y}) = 0$ otherwise.*

Theorem 1 *The probability of $\mathcal{Z}_{\mathcal{S}}(\mathbf{y})$ is equal to its expected value $E(\mathcal{Z}_{\mathcal{S}}(\mathbf{y})) = \frac{\binom{n - r_{\mathcal{S}}(\mathbf{y})}{\psi}}{\binom{n}{\psi}}$.*

Based on Theorem 1, we have the following three properties and their implications in anomaly detection:

(*i*) If $r_{\mathcal{S}}(\mathbf{y}) < r_{\mathcal{S}}(\mathbf{x}) \leq n - \psi$, then $E(\mathcal{Z}_{\mathcal{S}}(\mathbf{x})) < E(\mathcal{Z}_{\mathcal{S}}(\mathbf{y}))$. Anomalies are rare, and thus they have smaller $r_{\mathcal{S}}(\cdot)$ compared to normal instances. As anomalies have smaller $r_{\mathcal{S}}(\cdot)$ than normal instances, they have a higher probability of having zero appearances in a given subspace.

(*ii*) If $\mathbf{y}$ has a large number of identical instances in $\mathcal{S}$, such that $r_{\mathcal{S}}(\mathbf{y}) > n - \psi$, then $E(\mathcal{Z}_{\mathcal{S}}(\mathbf{y})) = 0$, i.e., $\mathbf{y}$ must not have zero appearances in the subspace. In such cases, $\mathbf{y}$ is considered as a normal instance because it conforms to a major behaviour in the subspace.

[1] Proofs of theorems are provided in Appendix A

(*iii*) If $\mathbf{y}$ is a previously unseen instance, i.e., $r_{\mathcal{S}}(\mathbf{y}) = 0$, then $E(\mathcal{Z}_{\mathcal{S}}(\mathbf{y})) = 1$. Such instances are considered as anomalies because they conform to an unusual behaviour. Also, the probability of $\mathbf{y}$ having zero appearances in the subspace approaches 1 when $r_{\mathcal{S}}(\mathbf{y})$ becomes very small and ψ is small. Given the rarity and exception nature, anomalies often have very small $r_{\mathcal{S}}(\cdot)$, so in a small subsample they have very high probability of having zero appearances in a given subspace.

3.2.2 Anomaly score

Based on Theorem 1 and its properties in Section 3.2.1, we define anomalies as follows:

Definition 4 *Anomalies are instances having zero appearances in a large number of subspaces over a set of subsamples.*

Definition 5 *The probability of instance* $\mathbf{y}$ *in subspace* $\mathcal{S}$ *given* $\mathcal{D}$ *is defined as:*

$$P_{\mathcal{S}}(\mathbf{y}|\mathcal{D}) = \frac{\sum_{\mathbf{x}\in\mathcal{D}} I_{\mathcal{S}}(\mathbf{x} = \mathbf{y})}{|\mathcal{D}|} \tag{3.2}$$

If $\mathbf{y}$ has zero appearances in $\mathcal{S}$ given $\mathcal{D}$, i.e., $\{\mathbf{x} \in \mathcal{D} : [x_{k_1}, \ldots, x_{k_m}] = [y_{k_1}, \ldots, y_{k_m}]\} = \emptyset$, then $P_{\mathcal{S}}(\mathbf{y}|\mathcal{D})$ will be equal to 0. ZERO++ employs the number of zero appearances in subspaces as an anomaly score. Given $\mathcal{D}$ and R, the anomaly score for $\mathbf{y}$ is defined as follows:

Definition 6 *The anomaly score for* $\mathbf{y}$ *is defined as the number of zero appearances in* $\mathcal{D}$ *and* R*:*

$$score(\mathbf{y}|\mathcal{D}, R) = \sum_{\mathcal{S}\in R} I(P_{\mathcal{S}}(\mathbf{y}|\mathcal{D}) = 0) \tag{3.3}$$

where $I(P_{\mathcal{S}}(\mathbf{y}|\mathcal{D}) = 0)$ *is an indicator function, which is 1 if* $P_{\mathcal{S}}(\mathbf{y}|\mathcal{D}) = 0$*, and 0 otherwise.*

The anomaly score is bounded by $[0, |R|]$. Based on the first property in Theorem 1, compared to normal instances, anomalies have a higher probability of having zero appearances in a given subspace. Therefore, in a set of subspaces, anomalies are likely to have a larger number of zero appearances than normal instances, and thus have a larger anomaly score.

Theorem 2 *If* $\mathcal{Z}_{\mathcal{S}}(\mathbf{y})$ *are independent, then*

$$0 \leq E(score(\mathbf{y}|\mathcal{D}, R)) \leq |R| \left(1 - \left(1 - \frac{\binom{n-r(\mathbf{y})}{\psi}}{\binom{n}{\psi}}\right)^{\frac{1}{|R|}}\right) \tag{3.4}$$

Moreover, if $E(\mathcal{Z}_{\mathcal{S}}(\mathbf{y}))$ *are identical for every* $\mathcal{S} \in R$*, then*

$$E(score(\mathbf{y}|\mathcal{D}, R)) = |R| \left(1 - \left(1 - \frac{\binom{n-r(\mathbf{y})}{\psi}}{\binom{n}{\psi}}\right)^{\frac{1}{|R|}}\right) \tag{3.5}$$

Note that here R is a fixed set, so the expected anomaly score of $\mathbf{y}$ is subject to $\mathcal{D}$ only. Suppose the assumptions in Theorem 2 hold, then given a test instance $\mathbf{y}$ and a fixed subsampling size ψ,

(*i*) if $\mathbf{y}$ has $r(\mathbf{y})$ such that $r(\mathbf{y}) \leq n - \psi$, then based on Equation (3.4) its anomaly score is upper bounded by

$$|R| \left(1 - \left(1 - \frac{\binom{n-r(\mathbf{y})}{\psi}}{\binom{n}{\psi}} \right)^{\frac{1}{|R|}} \right).$$

It should be noted that the independence of zero appearances in subspaces is a strong assumption, so it may not be a tight upper bound. However, Equation (3.4) shows that the anomaly score of $\mathbf{y}$ is inversely proportional to its appearances in the full data set, and provides an explanation as to why our proposed anomaly score can be used to identify anomalies effectively.

(*ii*) if $\mathbf{y}$ is a frequent instance, e.g., such that $r(\mathbf{y}) > n - \psi$, then it has the smallest anomaly score 0 and should be considered as a normal instance.

(*iii*) $\mathbf{y}$ will have the largest anomaly score $|R|$, when D does not contain any instance which is identical to $\mathbf{y}$, e.g., a previously unseen anomaly.

(*iv*) Equation (3.5) is built on a very strong assumption that $E(\mathcal{Z}_{\mathcal{S}}(\mathbf{y}))$ are identical for every $\mathcal{S} \in R$. When this assumption holds, anomalies must have higher anomaly scores than normal instances since they have smaller $r(\cdot)$.

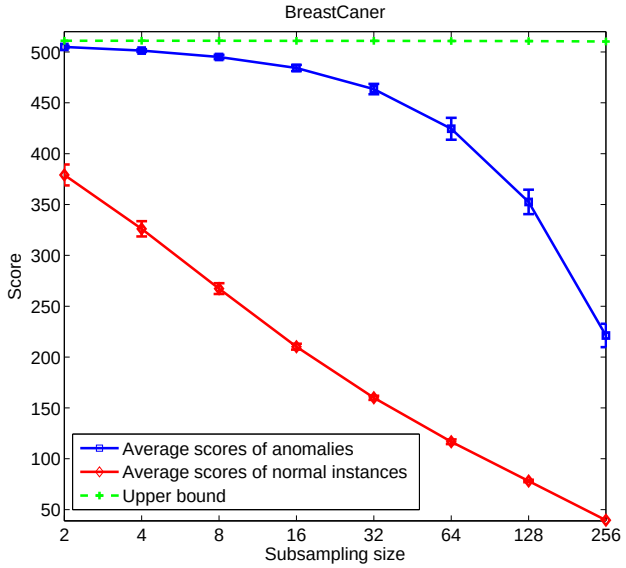

Figure 3.2: Average anomaly scores and two standard deviations of anomalies and normal instances in *BreastCancer* using different subsampling sizes. The average anomaly score is derived as follows: we first compute the anomaly score for each anomaly (normal instance), and the sum of anomaly scores for anomalies (normal instances) is then divided by the total number of anomalies (normal instances). We obtain the standard deviations based on the average scores over 10 runs.

We use the data set *BreastCancer* (Asuncion and Newman, 2007), which contains 699 instances and 9 categorical attributes, to demonstrate the implication of Theorem 2 in Figure 3.2. The data set contains 444 'benign' instances and 241 'malignant' instances. Following Hawkins et al. (2002) and He, Xu, Huang and Deng (2005), to transform this classification data set for anomaly detection tasks, we selected the first 39 'malignant' instances as anomalies against all 'benign' instance. The total number of subspaces in R is equal to $\sum_{m=1}^{9} \binom{9}{m} = 2^9 - 1 = 511$, and thus the anomaly score is bounded by $[0, 511]$.

Figure 3.2 presents a comparison of anomaly scores of anomalies and normal instances in *BreastCancer* with the subsampling sizes of 2, 4, 8, 16, 32, 64, 128 and 256. The average score and two standard deviations over 10 runs are presented. We also visualise the upper bound of the anomaly score with different subsample sizes. The averages over the anomaly scores of both anomalies and normal instances approach 0 as the subsampling size increases, but the average over anomaly scores of anomalies decrease at a much slower rate, and they are generally much larger than those of normal instances. As $r(\cdot)$ is as small as 1, the upper bound of the anomaly score is $511 \times \left(1 - \left(1 - \frac{483-\psi}{483}\right)^{\frac{1}{511}}\right)$, which decreases slowly with increasing subsample sizes. It is interesting to note that small subsamples have only a few dozen instances, so the independence of $\mathcal{Z}_{\mathcal{S}}(\mathbf{y})$ is likely to hold, and as a result, the upper bound is rather tight using small subsamples.

In order to obtain a more accurate estimation of the anomaly score for each instance, we use a set of subsamples to compute anomaly scores.

Definition 7 *For a test instance* $\mathbf{y}$ *given* $\mathcal{D}_i$, $i = 1, 2, \cdots, t$, *and* R, *the anomaly score of* $\mathbf{y}$ *is defined as follows:*

$$score(\mathbf{y}) = \sum_{i=1}^{t} score(\mathbf{y}|\mathcal{D}_i, R) \tag{3.6}$$

In Equation (3.6), according to the Law of Large Numbers (Etemadi, 1981), for a given $\mathbf{y}$, $score(\mathbf{y})$ will converge and have a stable value when t is sufficiently large. Figure 3.3 shows the average anomaly scores and two standard deviations over 10 runs for anomalies and normal instances in *BreastCancer* using $\psi = 8$ and different t values. The score converges very quickly and it becomes very stable when $t \geqslant 50$. We will present more experiments on examining the sensitivity of ψ and t in Chapter 4 and Appendix C.

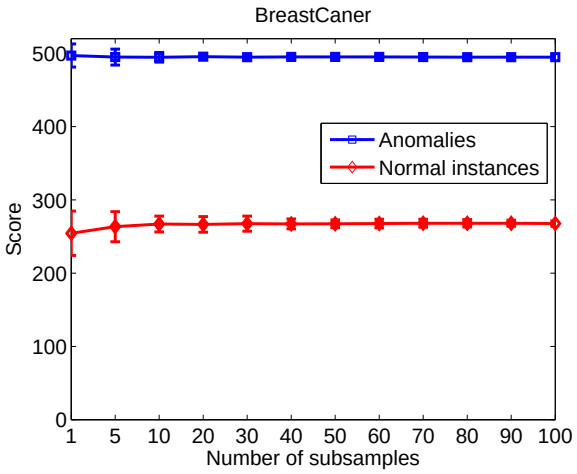

Figure 3.3: Average anomaly scores and two standard deviations over 10 runs for anomalies and normal instances in *BreastCancer* using $\psi = 8$ and different numbers of subsamples.

3.2.3 Approximation

The number of subspaces in R is $\sum_{m=1}^{d} \binom{d}{m} = 2^d - 1$ in total for D, which leads to exponential time and space complexities with respect to data dimensionality. For high dimensional data, examining the zero appearances of a given instance in all the subspaces is prohibitive in terms of both time and space complexities. For example, a 50-dimensional data set

requires about one petabyte of main memory to store all the subspaces, and for each test instance, we need to examine zero appearances in $2^{50} - 1$ subspaces, so the time and space complexities are prohibitive for data with only a few dozen instances. Therefore, ZERO++ with R is inapplicable in many real-world data sets, which have hundreds of dimensions and thousands of instances.

Definition 8 *Suppose we have two subspaces* $S = A_{k_1} \times A_{k_2} \times \ldots \times A_{k_m}$ *and* $\mathcal{S}' = A_{k_1} \times A_{k_2} \times \ldots \times A_{k_q}$*, if* $q > m$*, then* $\mathcal{S}'$ *is a higher-dimension subspace containing* $\mathcal{S}$*.*

In this research, utilising the anti-monotone property of zero appearances in subspaces, we introduce an efficient approximation to the proposed anomaly score with R as shown in Equation (3.6).

The **anti-monotone property** states that:
given a subspace $\mathcal{S}$*, let* $\mathcal{S}'$ *be a higher-dimension subspace containing* S*, if* $P_{\mathcal{S}}(\mathbf{y}|\mathcal{D}) = 0$*, then* $P_{\mathcal{S}'}(\mathbf{y}|\mathcal{D}) = 0$.

Based on this property, if a test instance has zero appearances in a subspace, it must also have zero appearances in the higher-dimension subspaces of this subspace. Thus, examining zero appearances in low dimensional subspaces is often sufficient to distinguish anomalies and normal instances.

The anti-monotone property enables us to approximate our anomaly score by using R_m with a small m only. The simplest case is to replace R with R_1 in Equation (3.6). In such a case, there are d subspaces, with each subspace spanned by a single attribute. However, it is very easy for anomalies to mask themselves by having the same attribute value as normal instances in single attributes, and using zero appearances in R_1 fails to work when the zero appearances are dependent on multiple attributes. An example of this case is demonstrated by a two dimensional occupation-salary artificial data set in Figure 3.4, where attribute A stands for income level with h, i, j, k and l corresponding to five respective levels *very low*, *low*, *medium*, *high* and *very high*, and B stands for occupation with a, b, c, d and e corresponding to five respective occupations *cleaner*, *premier*, *software engineer*, *astronaut* and *CEO*. In the left panel, $A = h$ (having very low salary), $A = l$ (having very high salary), $B = b$ (being a premier), $B = d$ (being an astronaut), and $B = e$ and $A = i$ (a low-salary CEO) are rare attribute values, and instances having these values are considered as anomalies, so there are five anomalies in this data set. For the subsample in the right panel, any instance having either $A = h$, $A = l$, $B = b$ or $B = d$ has zero appearances in the one-dimensional subspaces, and four anomalies having $A = l$ or $B = d$ can be detected. However, it cannot identify the fifth anomaly with $B = e$ and $A = i$ by working on one-dimensional subspaces only, as the zero appearance of this anomaly is dependent on both attributes.

Also, the time and space complexities of using R_m with $m \geq 2$ are prohibitive for large and high-dimensional data sets. For example, the time and space complexities of examining zero appearances in all the subspaces in R_2 are quadratic to data dimensionality, and that for R_m with $3 \leq m \leq \lceil \frac{d}{2} \rceil$ are at least cubic to data dimensionality. For computational efficiency, we focus on using a small random subset of R_m, denoted by R'_m, to replace R_m.

In this research, ZERO++ employs the zero appearances in subspaces in R'_2 to identify anomalies. R'_2 is a subset of R_2 having $|R'_2| = d$ such that every attribute must appear exactly twice in R'_2. This is to ensure that R'_2 covers all the attributes and every attribute has an equal chance to be considered in R'_2. The coverage of attributes in R'_2 enables ZERO++ to tolerate irrelevant attributes (A discussion about this can be found in Section 3.4.).

R'_2 is generated randomly as follows: A random order of d attributes $A_{i_1}, A_{i_2}, \ldots, A_{i_d}$ is first generated. Then, d attribute-pairs are formed by chaining the consecutive pair of attributes circularly until each attribute appears exactly twice, yielding $R'_2 = \{\mathcal{S}_{i_1 i_2}, \mathcal{S}_{i_2 i_3},$

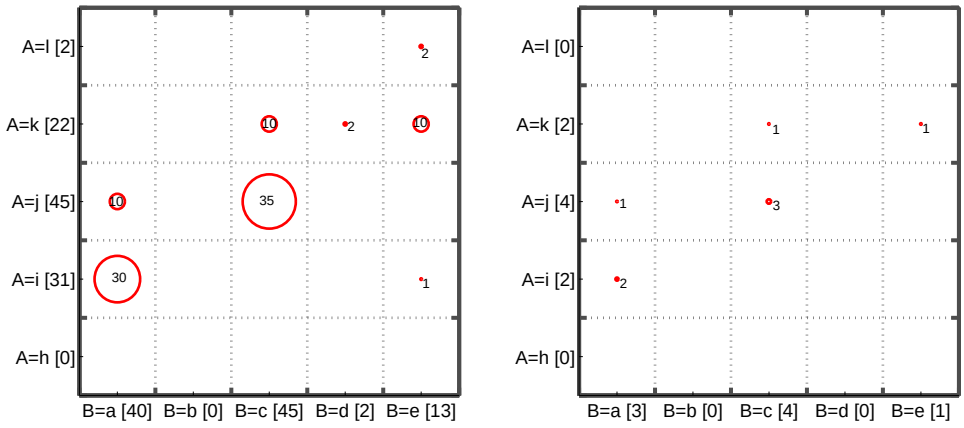

Figure 3.4: A two-dimensional occupation-salary data set with 100 instances, where the size of the circle indicates the number of instances in a region; and the number in [ ] indicates the number of instances in the region of one-dimensional subspace. The left panel is for the full data set; the right panel is a result of subsampling eight instances from the data set.

$\ldots, \mathcal{S}_{i_{(d-1)}i_d}, \mathcal{S}_{i_d i_1}\}$, where $\mathcal{S}_{ij} = A_i \times A_j$. Note that to produce an anomaly score for $\mathbf{y}$ as shown in Equation (3.6), R'_2 is generated randomly t times. In contrast, R_2 is a unique set.

Our anomaly score using R'_m is an efficient approximation to that using R shown in Equation (3.6), and m can be any value within the range $[1, d]$ [2]. The reasons for the use of R'_2 are as follows:

- Using R'_2 is a trade-off between the use of R_1 and R'_m with $3 \leq m \leq d$ in terms of detection performance in data sets with different attribute dependences.

  In data sets where anomalies exhibit abnormal behaviours based on multiple attributes, in order to obtain favourable detection performance, zero appearances in subspaces spanned by two or more attributes, e.g., subspaces in R'_m with $2 \leq m \leq d$, need to be examined. Using R_1 fails to detect these anomalies because subspaces in R_1 cannot capture dependence of abnormal behaviours on multiple attributes.

  In data sets where attributes are independent, e.g., abnormal behaviours are not dependent on multiple attributes, using R'_m with a larger m works less effectively, e.g., using R_1 works best in such data sets. This is because data subspace becomes sparser with increasing dimensionality sizes, and normal instances also become rare instances in subspaces in R'_m with a larger m, and as a result, normal instances can be incorrectly reported as anomalies.

  To demonstrate the above two situations, we provide two real-world examples: one using *Mushroom* where abnormal behaviours are dependent on multiple attributes, and another using *Shuttle* where attributes are independent [3].

  In *Mushroom*, many poisonous mushroom cases can be detected only when examining behaviours in two or more attributes (Duch et al., 1996). Therefore, using R'_m

[2]The process of generating R'_2 can also be used to generate R'_m with other m values, but it should be noted that R'_m with $m = 1$ or $m = d$ has a different property as R'_m with $2 \leq m \leq d-1$, i.e., every attribute appears once only in R'_1 and R'_d. R'_1 has the same subspace set as R_1, and R'_d contains one subspace only spanned by all the attributes.

[3]Descriptions of the data sets can be found in Appendix B. The empirical results are based on $\psi = 8$ and $t = 50$, which are the default settings for the two parameters in our experiments in Chapter 4.

with $m \geq 2$ is expected to obtain better detection performance than using R_1. The AUC [4] and two standard errors over 10 runs using R'_m with a different m in *Mushroom* is presented in Figure 3.5. It shows that our anomaly score using R'_m with $2 \leq m \leq 22(d = 22)$ outperforms that using R_1 significantly, and the AUC performance using R'_m with $2 \leq m \leq 10$ increases with m.

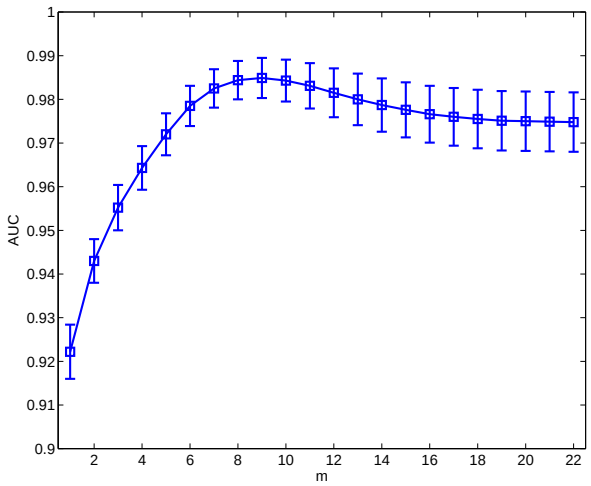

Figure 3.5: AUC performance and two standard errors over 10 runs using R'_m with a different m in *Mushroom*.

In *Shuttle* [5], most anomalies can be detected by examining behaviours in the first or the seventh attributes. In such data sets, our anomaly score using R_1 is expected to perform better than that using R'_m with $2 \leq m \leq d$. The result in *Shuttle* is provided in Figure 3.6. It shows that the AUC performance decreases quickly with increasing m, and using R'_m with a small m is able to outperform that with a large m significantly. Note that AUC performance using R'_2 is very closed compared to that using R_1, i.e., the AUC difference is 0.0001 only.

Different data sets have different dependences of abnormal behaviours on attributes, and it is often difficult to obtain the dependence information in advance and then use R'_m with a proper m value. We employ R'_2 as a trade-off between the use of R_1 and R'_m with $3 \leq m \leq d$ to deal with data sets having different attribute dependences.

- Time and space complexities are prohibitive in order to capture dependence of abnormal behaviours on different numbers of attributes, i.e., subspaces in R'_m with different m values.

  Subspaces in R'_m with different m values needed to be examined in order to capture the dependence of abnormal behaviours on different numbers of attributes. An example of this is provided in Table 3.2, where a data subset contains twelve instances and four attributes. Elements of (a_1, b_2, c_2) appear in every one- or two-dimensional subspaces spanned by any subset of A_1, A_2 and A_3, but it has zero appearances in the three-dimensional subspace spanned by these three attributes. For (a_1, b_2, c_1, d_3), its elements exist in all subspaces with less than four dimensions, but it has zero

[4] AUC is the area under the curve of Receiver Operating Characteristic (ROC) which is a plot of the true positive rate against the false positive rate at various threshold settings. Higher AUC indicates better detection performance. More detail of this is presented in Section 4.1.2.

[5] *Shuttle* is a numeric data set. This data set was discretised by the $\overline{x} \pm 3s$ rule discretisation method introduced in Section 3.3 before applying our proposed method to it. All the nine attributes in this data set are independent.

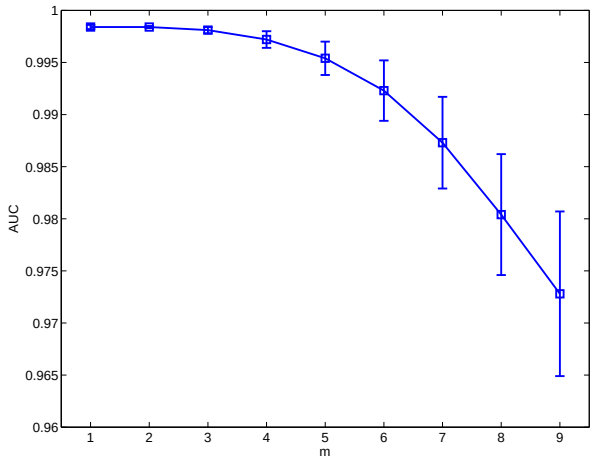

Figure 3.6: AUC performance and two standard errors over 10 runs using R'_m with a different m in *Shuttle*.

appearances in the full dimensionality. In order to capture all these types of zero appearances, it requires remembering appearance information in subspaces in R'_m with different m values, i.e., R'_2, R'_3, $\cdots$, and R'_d, which has time and space complexities increasing exponentially with d, and thus is inapplicable for high dimensional data.

Table 3.2: A toy example: Zero appearances occur in three or higher dimensional subspaces only. Each attribute contains three labels, i.e., $A_1 = \{a_1, a_2, a_3\}$, $A_2 = \{b_1, b_2, b_3\}$, $A_3 = \{c_1, c_2, c_3\}$ and $A_4 = \{d_1, d_2, d_3\}$.

Instances	A_1	A_2	A_3	A_4
$\mathbf{y}_1$	a_3	b_2	c_2	d_3
$\mathbf{y}_2$	a_3	b_2	c_2	d_1
$\mathbf{y}_3$	a_2	b_2	c_1	d_3
$\mathbf{y}_4$	a_2	b_2	c_2	d_1
$\mathbf{y}_5$	a_1	b_2	c_1	d_2
$\mathbf{y}_6$	a_2	b_2	c_1	d_3
$\mathbf{y}_7$	a_1	b_2	c_1	d_1
$\mathbf{y}_8$	a_1	b_2	c_1	d_1
$\mathbf{y}_9$	a_1	b_3	c_2	d_1
$\mathbf{y}_{10}$	a_1	b_1	c_1	d_3
$\mathbf{y}_{11}$	a_1	b_1	c_1	d_3
$\mathbf{y}_{12}$	a_1	b_2	c_2	d_3

Compared to the use of R_2, our anomaly score using R'_2 with a sufficiently large number of subsamples can provide a good approximation to that using R_2, while at the same time reducing time and space complexities from a quadratic to a linear level. For example, for low dimensional data sets, e.g., with $d \leq 30$, Equation (3.6) with R_2 needs to examine the zero appearances of $\mathbf{y}$ in $|R_2| = \frac{d(d-1)}{2} \leq 14.5d$ subspaces, while if Eqn.(3.6) uses a sufficiently large t number of R'_2, e.g., $t \geqslant 50$, it will examine at least $t|R'_2| \geq 50d$ subspaces. In such a case, Equation (3.6) with R'_2 can provide an effective approximation to that using R_2. For higher dimensional data sets, in many real-world data sets, many attributes are irrelevant to anomaly detection tasks, so numerous subspaces in R_2 are spanned by irrelevant attributes. Assume a 100-dimensional data set with only 10% relevant attributes,

then the total number of relevant subspaces in R_2 is $\frac{10\times9}{2} = 45$. However, in R'_2, there will be $\frac{10}{2} \times 50 = 250$ relevant subspaces for $t = 50$. Therefore, Equation (3.6) with R'_2 is still able to provide an effective approximation to that using R_2. It should be noted that R'_2 is generated randomly for each subsample, so there may exist the same subspace in R'_2 in different subsamples. This allows ZERO++ to examine whether the zero appearances in the same subspaces occur by chance.

Compared to the use of R or R'_m with different m values, our anomaly score using R'_2 might lose some accuracy, but time and space complexities have been reduced from at least $O(2^d)$ to $O(d)$.

ZERO++ is employed with R'_2 by default hereafter. We will empirically show that ZERO++ with R'_2 can identify anomalies more effectively than state-of-the-art anomaly detectors in a wide range of real-world and synthetic data sets in Chapter 4.

3.3 Extensions to numeric and mixed data

For numeric and mixed data, as ZERO++ is based on categorical attributes, numeric attributes are discretised to become categorical attributes. This is a process simpler than the one which requires a reverse conversion because no ordering information is required for categorical attributes. We examine ZERO++ with two discretisation methods, i.e., the equal-width method and the $\overline{x} \pm 3s$ rule discretisation method.

A number of discretisation methods have been proposed, but they were mainly dedicated for supervised learning techniques (Liu et al., 2002). Two commonly used unsupervised discretisation methods include equal-frequency and equal-width methods. The equal-frequency method divides instances into bins of the same number of instances in each attribute. Since our interest is to find zero appearances of infrequent attribute values in subsamples, the equal-frequency method is inapplicable because values in each attribute will have the same frequency after using this method.

The equal-width method divides instances into bins of equal width in each attribute. Formally, it works as follows. For a given attribute A and a user-defined number of bins N_{bin}, we first find the maximum and minimum values in A, denoted by $max(A)$ and $min(A)$; the bin width is then obtained by $w = \frac{max(A)-min(A)}{N_{bin}}$; and the N_{bin} bins are finally generated by $(N_{bin} - 1)$ cut points $min(A) + i \times w$ where $i = 1, 2, \cdots, N_{bin} - 1$. Different widths will lead to varying binning results, which will in turn result in unstable anomaly detection performance.

In this research, we also examine a simple preprocessing method which converts a numeric attribute into a categorical attribute with two labels as follows. For each subsample, we compute the mean $\overline{x}$ and the standard deviation s for each attribute. If a numeric value falls within the range $[\overline{x} - 3s, \overline{x} + 3s]$, it is assigned a label 'y'; otherwise label 'n' is assigned.

The intuition of the $\overline{x} \pm 3s$ rule method is demonstrated using a synthetic data set in Figure 3.7. We visualise results of the discretisation in a Gaussian distribution with 10,000 instances in a two-dimensional numeric feature space shown in Figure 3.7, where each rectangle area indicates a discretised partition result of a subsample, bounded by $[\overline{x} - 3s, \overline{x} + 3s]$ in each dimension. Each rectangle is generated based on 64 randomly selected instances from which $\overline{x}$ and s are computed. An instance that falls outside the rectangle area has zero appearances in this region. The figure shows that anomaly **o** does not appear in all 50 regions created from 50 subsamples; while normal instance **x** appears in all 50 regions. Figure 3.8 shows the average probabilities of having zero appearances in the region outside the rectangle with respect to increasing ψ values. The probability of anomaly **o** having zero appearances is substantially higher than that of **x**, and it converges very quickly with increasing subsampling sizes.

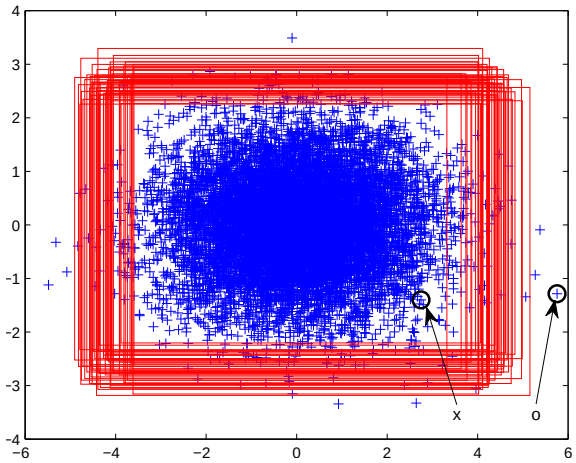

Figure 3.7: A data set of 10,000 instances generated from a Gaussian distribution. The 50 rectangles are 2-D subspaces generated from 50 subsamples, each having 64 instances.

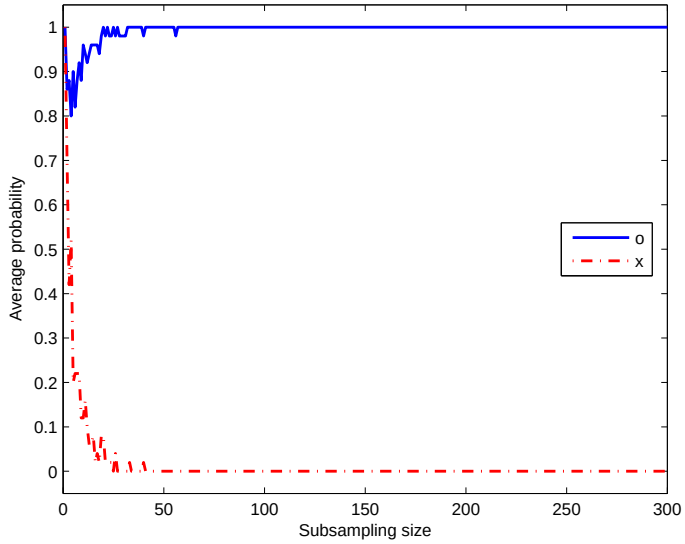

Figure 3.8: Average probabilities of having zero appearances for **x** and **o** with respect to different subsampling sizes.

The $\overline{x} \pm 3s$ rule is known to be not robust, i.e., $\overline{x}$ and s are easily influenced by anomalies. However, $\overline{x}$ and s are derived from each subsample which is less likely to contain anomalies. In addition, the multiple models used in ZERO++ also reduce the impact of biased $\overline{x}$ and s.

This discretisation method is based on an underlying assumption that the normal instances follow uni-modal distributions. Therefore, once the uni-modal distribution assumption is violated, more advanced discretisation methods may be required in order to obtain favourable detection performance. We will show in Chapter 4 that ZERO++ using the $\overline{x} \pm 3s$ rule method is able to handle many real-world numeric data sets and mixed data sets effectively.

3.4 Characteristics of ZERO++

ZERO++ has the following three characteristics:

1. ZERO++ works well with a small subsample size.

Let $p = \frac{r_{\mathcal{S}}(\mathbf{y})}{n}$ be the probability of $\mathbf{y}$ occurring in subspace $\mathcal{S}$ given the full data set D. $E(\mathcal{Z}_{\mathcal{S}}(\mathbf{y}))$ can then be computed as follows:

$$E(\mathcal{Z}_{\mathcal{S}}(\mathbf{y})) = (1-p) \times \frac{n(1-p)-1}{n-1} \times \cdots \times \frac{n(1-p)-(\psi-1)}{n-(\psi-1)}$$

For a large n, it can be simply approximated as follows:

$$E(\mathcal{Z}_{\mathcal{S}}(\mathbf{y})) \approx (1-p)^{\psi} \tag{3.7}$$

Based on Equation (3.7), given a small subsample $\mathcal{D}$, if $\mathbf{y}$ is a rare instance in D, i.e., p is very small, then the probability of $\mathbf{y}$ having zero appearances in $\mathcal{D}$ is very high. Figure 3.9 presents $E(\mathcal{Z}_{\mathcal{S}}(\mathbf{y}))$ with respect to different ψ, i.e., 2, 4, 8, 16, 32, 64, 128, 256 [6], given different p, including 0.1, 0.05, 0.01, 0.005. It shows that a small subsample size, e.g., $\psi \leqslant 64$, can generally ensure rare instances bearing high probability (> 0.5) of having zero appearances in subsamples. Particularly, for $p \leqslant 0.05$, $E(\mathcal{Z}_{\mathcal{S}}(\mathbf{y})) > 0.65$ if $\psi \leqslant 8$. For a relatively large p, e.g., $p = 0.1$, a smaller ψ, e.g., 2 or 4, should be taken in order to ensure $E(\mathcal{Z}_{\mathcal{S}}(\mathbf{y}))$ within the range $(0.5, 1.0]$. Considering the percentage of anomalies is normally less than 5%, a small subsample size, e.g., $\psi \leqslant 64$, is preferred in order to ensure anomalies having a sufficiently large number of zero appearances in the subspaces. In this research, $\psi = 8$ is used as the default setting in our experiments.

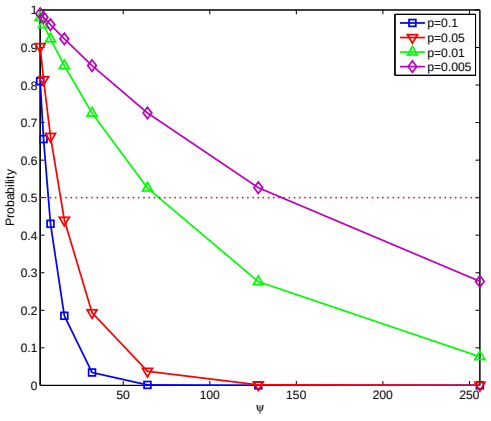

Figure 3.9: Probability of having zero appearances in subsamples with respect to different subsample sizes, given instances with different p.

2. ZERO++ is able to work on data sets with a low percentage of relevant attributes.

   For anomaly detection tasks, anomalies do not exhibit abnormal behaviours in irrelevant attributes, i.e., anomalies and normal instances share the same behaviour in such attributes. Therefore, in ZERO++, both normal instances and anomalies have equivalent anomaly scores in subspaces spanned by those attributes. However, in subspaces spanned by relevant attributes where anomalies have rare attribute values, anomalies are more likely to have zero appearances in subsamples than normal instances, and as a result, anomalies will have higher anomaly scores than normal instances. Therefore, even if the data sets have a very low percentage of relevant attributes, e.g., 1%, anomalies are still likely to have a larger number of zero appearances in the subspaces compared to normal instances.

[6] In ZERO++, these values are used as a search range for best performance by default in our experiments.

It should be noted that R'_2 in ZERO++ covers every attribute in each subsample, whereas other subspace-based methods, such as FeatureBagging (Lazarevic and Kumar, 2005), iForest (Liu et al., 2012) and MassAD (Ting et al., 2013), work on subspaces spanned by a randomly selected attribute subset only. If data sets contain a high percentage of irrelevant attributes, these methods are very likely to work on subspaces spanned by irrelevant attributes only, and as a result, they perform poorly in such data sets.

3. Considering interactions within attributes is an integral component in ZERO++.

   For ZERO++ working with R'_2, instances are considered as anomalies when they exhibit abnormal behaviours in one or two attributes of a given subspace, so it is able to identify anomalies which have abnormal behaviours depending on two attributes. ZERO++ can be adapted to cases which are required to capture dependences between three or more attributes by simply replacing R'_2 with a higher dimensional subspace set, e.g., R'_3. Attribute independence is often violated in many real-world data sets (Ghoting et al., 2004; Webb et al., 2005). The ability to capture dependences between numeric and categorical attributes is often required for detecting anomalies in mixed data sets (Ghoting et al., 2004; Zhang and Jin, 2011). ZERO++ with R'_2 captures the interaction between one numeric attribute and one categorical attribute in a seamless manner.

3.5 The algorithm

Given a data set with categorical attributes, ZERO++ builds a model in the training stage, and the model can then be used to score every instance in the testing stage. The procedures of these two stages are given below.

Training. In the training stage, ZERO++ builds a probability table from each subsample for R'_2. The probability table consists of probabilities of instances occurring in each subspace in R'_2, as defined in Equation (3.2). As there are d subspaces in R'_2, the width of the probability table is equal to d. The procedure to generate the probability tables is presented in Algorithm 1. Note that we are interested in entries of the subspaces having zero probabilities only, and the non-zero entries in the probability table are only useful in so far as to identify zero entries; the actual probabilities are immaterial.

Algorithm 1 *ProbabilityTable(D, t, ψ)*

Input: D - input data, t - the number of subsamples, ψ - subsample size
Output: Ω - a set of probability tables

```
Initialise Ω as an empty set
for i = 1 to t do
    Initialise probability table ω_i
    D_i ← Randomly select ψ instances without replacement from D
    Generate a randomised subspace set R'_2
    Build ω_i for R'_2 from D_i.
    Ω ← ω_i
end for
return Ω
```

Testing. To score a test instance $\mathbf{y}$, ZERO++ computes the number of zero appearances in the subspaces in the probability table ω_i, as defined in Equation (3.2), as the anomaly score for $\mathbf{y}$. The higher the score is, the more likely $\mathbf{y}$ is an anomaly. This procedure is presented in Algorithm 2.

Algorithm 2 *ZERO++(*$\mathbf{y}$*)*

Input: $\mathbf{y}$ - a test instance
Output: z - the number of zero appearances in subspaces of $\mathbf{y}$
1: $z \leftarrow 0$
2: **for** $i = 1$ to t **do**
3: $r \leftarrow$ number of zero appearances in subspaces in $\omega_i(\mathbf{y})$.
4: $z \leftarrow z + r$
5: **end for**
6: **return** z

Complexity analysis. In the training stage, ZERO++ builds t d-sized probability tables, each using a subsample of ψ instances. Thus, ZERO++ has time complexity $O(td\psi)$. During the testing stage, for a test instance, ZERO++ needs to look up t probability tables, where each table look up takes $O(d)$. To score n instances in a data set, ZERO++ has time complexity $O(ntd)$. Since n is normally far larger than ψ, the time complexity of ZERO++ is $O(ntd)$.

In terms of space complexity, ZERO++ needs to store t d-sized probability tables for every subsample. Let ℓ be the average number of labels per attribute, so for each probability table, $O(d\ell^2)$ is required to store values in its subspaces. Therefore, ZERO++ has space complexity $O(td\ell^2)$.

A comparison of time and space complexities between ZERO++, FPOF (He, Xu, Huang and Deng, 2005), iForest (Liu et al., 2012), LOF (Breunig et al., 2000) and SOD (Kriegel, Kröger, Schubert and Zimek, 2009) is provided in Table 3.3. Both ZERO++ and iForest have linear time complexity with respect to both data size and dimensionality, and constant space complexity with respect to data size. The state-of-the-art anomaly detector for categorical data FPOF, density-based detector LOF, and subspace-based detector SOD have much higher time and space complexities than ZERO++ and iForest. The time complexity of FPOF is linear to data size but quadratic to dimensionality, and it is affected by the length of the itemsets considered and the minimum *support* threshold. Though the time complexity of LOF and SOD can be reduced to $O(n \log(n)\ d)$ and $O(n^2 d)$ respectively when using some indexing scheme such as R^*-tree (Beckmann et al., 1990), most indexing schemes only work on low-dimensional numeric data and do not work on data with high dimensionality or with categorical attributes.

Table 3.3: time and space complexities between ZERO++, FPOF, iForest, LOF and SOD.

Methods	Time complexity	Space complexity
ZERO++	$O(ntd)$	$O(td\ell^2)$
FPOF	$O(n2^d)$	$O(2^d)$
iForest	$O(nt)$	$O(t\psi)$
LOF	$O(n^2 d)$	$O(nd)$
SOD	$O(n^3 d)$	$O(nd)$

Note that we have ignored the time and space requirements for the preprocessing step. For ZERO++ or FPOF, converting numeric attributes to categorical attributes is only required for data sets having numeric attributes. The time complexity of the equal-width discretisation method is dominated by searching for maximum and minimum values, which can be done in linear time. Only the bin width, and maximum and minimum values for each attribute are required to be stored, which is negligible. For the $\overline{x} \pm 3s$ rule discretisation method, time complexity is dominated by the computation of standard

deviation for which a linear time complexity algorithm can be found in (Donald, 1999). The space required is to store the means and standard deviations for each attribute, which is also negligible. For methods based on numeric data such as iForest, LOF and SOD, a preprocessing step is required to convert categorical attributes to numeric attributes for data sets having categorical attributes. The time and space requirements in this step are small as well.

3.6 Comparison to detectors in most related work

Most existing categorical data motivated methods, including ZERO++, frequent patterns based detectors (e.g., FPOF (He, Xu, Huang and Deng, 2005)) and infrequent patterns based detectors (e.g., LOADED (Ghoting et al., 2004)), are based on a general assumption that anomalies are instances with rare attribute values. FPOF aims to capture normal behaviours using frequent patterns and report instances as anomalies when they do not conform to the patterns. In contrast, both ZERO++ and LOADED focus on capturing abnormal behaviours, but they use different anomaly scores and algorithmic frameworks. ZERO++ identifies anomalies based on the number of zero appearances in the subspaces while LOADED is based on the inclusion of infrequent itemsets. Also, ZERO++ builds a set of models on a set of subspaces and subsamples, whereas LOADED builds a single model on an entire data set.

ZERO++ has much lower time and space complexities than FPOF and LOADED. FPOF and LOADED need to search for frequent patterns or infrequent patterns to detect anomalies, which have time and space complexities at least quadratic to data dimensionality; while ZERO++ detects anomalies in a small set of randomised two-dimensional subspaces, which has time complexity linear to data dimensionality and data size. The anti-monotone property is applied in FPOF and LOADED to reduce the search space, whereas ZERO++ applies the same property to consider low dimensional subspaces only and no search is required in ZERO++.

ZERO++ uses a similar algorithmic framework as iForest (Liu et al., 2012) and MassAD (Ting et al., 2013), i.e., construct detection models over subspaces and subsamples, and have similar time and space complexities, but they have significant differences in terms of motivation, working principles and anomaly scores, as summarised in Table 3.4. ZERO++ is based on categorical data and easily extended to numeric data and mixed data, while iForest and MassAD are numeric data oriented methods and there are no good solutions to extend them to handle categorical attributes thus far. Also, compared to iForest and MassAD, ZERO++ uses a totally different anomaly score and considers a lot more subspaces, as a result, ZERO++ converges much faster.

3.7 Chapter summary

We introduce a novel anomaly detection method ZERO++ which is the only anomaly detector based on zero appearances in subspaces, as far as we know. It is unique in that it works in regions of subspaces that are not occupied by data; whereas existing methods work in regions occupied by data. ZERO++ works well with small subsample sizes, and it is able to identify anomalies in data sets with a low percentage of relevant attributes and capture the dependence of abnormal behaviours on attributes.

ZERO++ is based on categorical data, but it can be easily extended to handle numeric data and mixed data by using discretisation methods. ZERO++ is an efficient and scalable detector, which has linear time complexity in terms of data size and dimensionality and constant space complexity. A series of empirical results is presented in Chapter 4 to

Table 3.4: Conceptual differences between ZERO++, iForest and MassAD

Motivation	ZERO++	It is a categorical data oriented method and can be easily extended to numeric and mixed data by using existing discretisation methods.
	iForest	It is a numeric data oriented method. Ordering information for each categorical attribute is required to extend iForest to categorical domain, and there are no good solutions to order the categorical values for iForeest thus far.
	MassAD	It is also a numeric data oriented method. It has the same limitation as iForest in handling data sets with categorical attributes.
Working principle	ZERO++	Its working principle is based on zero appearances in subspaces. The subspaces used in each subsample cover all the attributes.
	iForest	It is based on susceptibility to isolation, implemented in a tree structure. Its working regions in each subsample are based on a few randomly selected attributes only.
	MassAD	It models the data in terms of mass estimation, implemented using trees. Its working regions in each subsample are also based on a few randomly selected attributes only.
Anomaly score	ZERO++	Number of zero appearances in subspaces. In each subsample, ZERO++ examines zero appearances in d subspaces, and so the anomaly score is based on the number of zero appearances in td subspaces in t subsamples, which considers a lot more subspaces than iForest and MassAD, leading to a faster convergence.
	iForest	Path length in isolation trees. In each subsample, a path length is derived from a subspace of a few randomly selected attributes, and so it only considers t subspaces in t subsamples.
	MassAD	Mass values in half-space trees. MassAD has a similar working process as iForest and its anomaly score is only based on t subspaces in t subsamples.

demonstrate the effectiveness of ZERO++ in handling data sets with different types of attributes, its favourable scalability and its ability to tolerate irrelevant attributes.

Chapter 4

Experiments

In this chapter, after an introduction to the experiment settings in Section 4.1, a series of empirical results is presented from Section 4.2 to 4.4 to illustrate how ZERO++ can meet the four challenges stated in Section 1.2:

- Although ZERO++ is based on categorical data, it can be easily extended to handle numeric and mixed data by using discretisation techniques, as discussed in Section 3.3. In Section 4.2, we empirically compare ZERO++ with four well-known anomaly detectors, i.e., FPOF (He, Xu, Huang and Deng, 2005), iForest (Liu et al., 2012), LOF (Breunig et al., 2000) and SOD (Kriegel, Kröger, Schubert and Zimek, 2009), on 19 real-world data sets and a synthetic data set. There are seven mixed data sets, four categorical data sets, and nine numeric data sets. We first show the results of ZERO++ and its contenders in data sets with categorical attributes only in Section 4.2.1, and then the results in data sets with numeric attributes only and mixed data sets in Section 4.2.2.

- We argue in Section 3.4 that ZERO++ is able to work well in data sets with a low percentage of relevant attributes. Section 4.3 provides empirical results of examining the ability of ZERO++ to tolerate irrelevant attributes on a set of synthetic data sets.

- A complexity analysis in Section 3.5 demonstrates that the time complexity of ZERO++ is linear to the data size and data dimensionality. In Section 4.4, we empirically evaluate the scalability of ZERO++ with respect to data dimensionality and data size, with the four contenders as baselines. A set of synthetic data sets with dimensions from 10 up to 1,000 was used in the scaleup test with respect to dimensionality. Seven subsets of the largest data set used in our experiments were employed to evaluate the scalability of ZERO++ with respect to data size. The smallest subset contains 1,000 instances, and subsequent subsets are increased by a factor of four, until the largest subset which contains 4,096,000 instances.

Subsample size ψ and ensemble size t are the only two parameters in ZERO++. In Section 4.5, we examine the sensitivity of ZERO++ with respect to its two parameters in all the 20 data sets used in Section 4.2.

We also tested ZERO++ on data sets without ground truth. Section 4.6 presents anomalies identified by ZERO++ when it was applied in these data sets. One numeric data set and two categorical data sets were used.

In Section 4.7, there is a discussion over the empirical results, and a summary of all the empirical results is presented in Section 4.8.

4.1 Experiment settings

Four well-known anomaly detection methods were selected as the contenders of ZERO++. A series of experiments was conducted to compare ZERO++ with its contenders on a wide range of real-world and synthetic data sets in terms of effectiveness and efficiency.

4.1.1 Contenders and their parameter settings

We compared ZERO++ with FPOF (He, Xu, Huang and Deng, 2005), iForest (Liu et al., 2012), LOF (Breunig et al., 2000) and SOD (Kriegel, Kröger, Schubert and Zimek, 2009) . The contenders are the representative methods of four categories of anomaly detectors, i.e., methods for categorical or mixed data, methods based on both subspace and subsampling, proximity-based methods, and subspace-based methods, as illustrated in Table 2.1 in Section 2.4. These methods were selected as contenders because:

- FPOF is a state-of-the-art categorical data based method. It has been reported as one of the most effective methods and widely used as performance contender in previous literature on anomaly detection for categorical data (Koufakou et al., 2007; Wu and Wang, 2013). Also, FPOF is a closely related work to ZERO++, as discussed in Section 3.6. Another closely related work is LOADED (Ghoting et al., 2004). Compared to LOADED, which is proposed for mixed data sets, FPOF is more related to ZERO++ in the sense that FPOF and ZERO++ are based on categorical data, so we chose FPOF over LOADED.

  Compared to numeric data oriented methods, relatively less work has been done for categorical data. An effective information-theoretic based method was recently proposed in Wu and Wang (2013). However, this method is aimed at identifying anomalies among an anomaly candidate set only, i.e., a data subset, and therefore may result in low true positive rate regardless of its high detection precision. Also, only the instances in the candidate set have anomaly scores. Therefore, we are unable to obtain its overall detection performance, measured in terms of the **A**rea **U**nder the ROC **C**urve (AUC) (Hand and Till, 2001). We examine the ability of detectors in identifying anomalies in terms of ROC, so this method is inconsistent with our examination objective.

- iForest is a state-of-the-art ensemble method for anomaly detection, and it is also a closely related work to ZERO++ in terms of methodology, i.e., both ZERO++ and iForest are ensemble methods based on both subspace and subsampling.

- LOF has been recognised as a state-of-the-art method in handling numeric data sets, and it is widely used as a performance baseline in the literature (Chandola et al., 2009).

- SOD is a well-known subspace-based method. SOD computes anomaly scores using selected informative subspaces while ZERO++ scores instances using randomly selected subspaces and does not have a procedure to select informative subspaces. It is interesting to compare such two different subspace-based methods.

- Anomaly detection methods from other categories, such as extreme value analysis based methods and subsampling-based methods, are less effective than these four methods in addressing the four challenges stated in Section 1.2, as discussed in Section 2.4. Therefore, we focus on comparing ZERO++ with FPOF, iForest, LOF and SOD.

In order to have a thorough evaluation, ZERO++ and its four contenders were examined with default and tuned parameter settings. Specifically,

- **ZERO++ and iForest**. Both ZERO++ and iForest employed $t = 50$ as the default settings. The subsampling size ψ was 8 by default in ZERO++. As recommended in (Liu et al., 2012), iForest employed $\psi = 265$ as the default setting. For both methods, the best ψ was searched over the range 2, 4, 8, 16, 32, 64, 128, 256.

- **FPOF**. Following He, Xu, Huang and Deng (2005), FPOF employed the minimum *support* threshold $\delta = 0.1$ and 5 as the maximum length of itemsets by default. We searched δ over the range 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 and 0.9, and report the best results.

- **LOF**. In order to obtain favourable detection performance, the lower bound of the neighbourhood size k in LOF is 10, as discussed in Breunig et al. (2000). Also, k is an important factor for the computation time of LOF, e.g., its runtime increases quickly with k. LOF employed $k = 20$ as the default setting to have a reasonable trade-off between effectiveness and efficiency. We also searched k over the range 10, 20, 40, 60, 80, 150, 250, 300, 500, 1000, 2000, 3000 and 4000, and report the best results.

- **SOD**. Since the reference set in SOD works similarly to the neighbourhood in LOF, SOD used a similar search range and default setting as in LOF. (k, l) were searched with the difference $k - l = 100$, i.e., (110, 10), (120, 20), (140, 40), (160, 60), (180, 80), (250, 150), (350, 250), (400, 300), (600, 500), (1100, 1000), (2100, 2000), (3100, 3000) and (4100, 4000). This implies that the top l SNN-based reference set selection is always built upon a larger kNN set, and the size l of the reference set in SOD is equal to the size of neighbourhood used in LOF. k and l were 120 and 20 by default in SOD, respectively.

All the methods were implemented in JAVA. We implemented ZERO++ in WEKA (Hall et al., 2009). FPOF was implemented using the *Apriori* algorithm in WEKA. iForest is already in the WEKA platform, and LOF and SOD in the ELKI platform (Achtert et al., 2013) were used in the experiments. R^*-tree indexing (Beckmann et al., 1990) was used by default in LOF and SOD. All the experiments were performed as a single-thread job processed at 2.27 GHz in a Linux cluster with 40GB memory.

4.1.2 Datasets and detection performance measure

Experiments were conducted on 19 real-world data sets from the UCI repository(Asuncion and Newman, 2007) and one synthetic data set generated by the Mulcross data generator (Rocke and Woodruff, 1996). We only considered data sets with over 1,000 instances or over 100 dimensions in order to avoid empirical bias derived from small and low-dimensional data sets. A summary of the data sets is given in Table 4.1. Further detail of these data sets can be found in Appendix B.

The True Positive rate (denoted by TP), False Positive rate (denoted by FP), and AUC are widely used quantitative measures for anomaly detection performance (Chandola et al., 2009). TP refers to the percentage of the number of correctly detected anomalies, while FP refers to the percentage of the number of normal instances incorrectly identified as anomalies. AUC is a measure integrating TP and FP, which is the area under the ROC curve that is a plot of the TP against the FP at various threshold settings. We used AUC as the detection performance measure in our experiments. Higher AUC indicates better detection performance. We also recorded the run time to compare their efficiency. The AUC and runtime results were averaged over 10 runs for all randomised methods,

Table 4.1: A summary of data sets used. #num and #cate denote the number of numeric and categorical attributes respectively. The *Anomaly class* column presents the anomaly class selected and its percentage in each data set. #binary is the total number of categorical values contained in all the categorical attributes. It is also the total number of binary attributes produced from the 1-of-ℓ transformation which converts categorical attributes to binary attributes. Horizontal lines are used to separate data sets with different types of attributes.

Data set	n	d	#num	#cate	Anomaly class	#binary
Linkage	5749132	9	4	5	match(0.36%)	5
Census	299285	40	7	33	50K+(6.20%)	493
CoverType	286048	54	10	44	class 4 (0.96%)	44
Probe	64759	41	34	7	attack(6.43%)	83
U2R	60821	41	34	7	attack(0.37%)	83
AnnThyroid	7200	21	6	15	class 1,2(7.42%)	15
Arrhythmia	452	**279**	206	73	8 smallest classes(14.60%)	73
Nursery	4648	8	0	8	very_recom (7.06%)	26
Chess	4580	6	0	6	zero(0.59%)	39
Mushroom	4429	22	0	22	poisonous(5.00%)	121
SolarFlare	1066	10	0	10	flare X(0.47%)	29
Http	567497	3	3	0	attack(0.39%)	0
Mulcross	262144	4	4	0	2 clusters(10.00%)	0
Smtp	95156	3	3	0	attack(0.03%)	0
Shuttle	49097	9	9	0	classes 2,3,5,6,7 (7.15%)	0
Mammo	11183	6	6	0	class 1(2.32%)	0
HAR	7032	**561**	561	0	class 3(20.00%)	0
Satimage	6435	36	36	0	crop(10.92%)	0
Isolet	730	**617**	617	0	class Y (1.37%)	0
Mfeat	410	**649**	649	0	digit 0 (2.44%)	0

i.e., ZERO++ and iForest. In this research, confidence intervals are based on the average AUC over 10 runs and its two standard errors, so the statistical significance statement is at 95% confidence level.

We employed a commonly used performance evaluation method for unsupervised anomaly detection techniques (Aggarwal, 2013a). Specifically, we trained and evaluated detection models on the same data set, but it is assumed that class labels are unavailable in the training stage. The class labels are only used to compute the detection performance measure AUC in the evaluation stage.

4.2 Detection performance in different types of data sets

This section aims to compare the detection performance of ZERO++ and its four contenders on data sets with different types of attributes: (i) categorical attributes only; (ii)numeric attributes only; and (iii) both numeric and categorical attributes. In our experiments, there are four categorical data sets, nine numeric data sets, and seven mixed data sets. In the experiments on categorical data, mixed data sets were used with categorical attributes only. Likewise, these mixed data sets were also used in experiments on numeric data by removing categorical attributes. We compared the detection performance of ZERO++ with its contenders using the default settings and the best parameters obtained from the search range specified in Section 4.1. The method that produces the best performance for each data set is underlined in the detection performance comparison tables throughout this section.

Note that iForest, LOF and SOD are numeric data oriented methods. To run these algorithms on data sets with categorical attributes, the attributes were first converted into binary attributes using the 1-of-ℓ transformation method [1], i.e., a ℓ-label attribute is converted into ℓ binary attributes. The binary attributes were then regarded as numeric attributes to be further processed by these three methods.

4.2.1 Categorical data sets

Table 4.2 shows the detection performance of ZERO++, FPOF, iForest, LOF and SOD with the default settings on categorical data. The average AUC of ZERO++ over 10 runs is the best in 6 out of 11 data sets, with three close to the best (having the difference in AUC less than 0.01). ZERO++ outperforms FPOF significantly in five data sets, and has comparable performance to FPOF in four data sets. We cannot obtain the result of FPOF in *Arrhythmia* due to an out-of-memory exception error. ZERO++ outperforms iForest, LOF and SOD significantly in most data sets. Note that LOF and SOD were unable to process large categorical data sets because the indexing methods did not work in the data sets, *Linkage*, *Census*, *CoverType*, *Probe* and *U2R* [2].

Table 4.3 presents the runtime results of the five detectors on the categorical data sets. It shows that ZERO++ is significantly faster than FPOF by a factor of more than 100 and 1,000 in the two large data sets with high dimensions, *Census* and *CoverType*, respectively. ZERO++ is slower than iForest by a factor of between 10 and 40 in the same

[1] As discussed in Section 2.3.1, this transformation method can be employed by different types of detectors, which ensures a fair empirical comparison between the detection performance of different detectors, so we selected it as a preprocessing method to enable three different types of detectors to treat categorical or mixed data.

[2] For LOF and SOD, R^*-tree or other indexing methods in ELKI do not work on large categorical data because there are too many identical values in the attributes. Also, since data is pre-indexed by default in ELKI, LOF and SOD still cannot work on those data sets even though they do not use R^*-tree or other indexing methods throughtout this section.

Table 4.2: AUC performance comparison between ZERO++, FPOF, iForest, LOF and SOD with the default settings on categorical data.

	ZERO++	FPOF	iForest	LOF	SOD
	$\psi = 8$	$\delta = 0.1$	$\psi = 256$	$k = 20$	$l = 20$
Linkage	0.9973±0.0001	0.9972	0.9790±0.0041	n/a	n/a
Census	0.6420±0.0056	0.6148	0.5449±0.0172	n/a	n/a
CoverType	0.9946±0.0020	0.9965	0.9773±0.0043	n/a	n/a
Probe	0.9802±0.0020	0.9867	0.9776±0.0022	n/a	n/a
U2R	0.9891±0.0009	0.9156	0.9729±0.0073	n/a	n/a
AnnThyroid	0.4401±0.0012	0.4357	0.4386±0.0008	0.4875	0.4766
Arrhythmia	0.6588±0.0093	n/a	0.6878±0.0024	0.6114	0.6205
Mushroom	0.9430±0.0047	0.9218	0.9182±0.0117	0.7609	0.7738
Nursery	1.0000±0.0000	1.0000	0.9986±0.0010	0.7866	0.7002
SolarFlare	0.9750±0.0052	0.9791	0.9325±0.0070	0.5785	0.8884
Chess	0.9774±0.0101	0.9122	0.8606±0.0566	0.7304	0.9179
ZERO++ vs. (#wins/losses/draws)		5/1/4	8/1/2	5/1/0	5/1/0

two data sets, and it also runs slower than FPOF in the largest data set, *Linkage*, which has only five dimensions.

Table 4.3: Runtime (in seconds) comparison between the five detectors on categorical data.

	ZERO++	FPOF	iForest	LOF	SOD
	$\psi = 8$	$\delta = 0.1$	$\psi = 256$	$k = 20$	$l = 20$
Linkage	468.51	74.21	70.00	n/a	n/a
Census	172.03	89168.49	14.88	n/a	n/a
CoverType	225.89	1044166.03	6.17	n/a	n/a
Probe	7.37	1.85	1.11	n/a	n/a
U2R	6.96	1.78	0.80	n/a	n/a
AnnThyroid	1.92	16.59	0.18	37.07	62.48
Arrhythmia	0.78	n/a	0.07	0.51	0.74
Mushroom	1.86	102.57	0.25	11.61	21.12
Nursery	0.64	0.35	0.13	0.50	13.17
SolarFlare	0.20	0.35	0.07	0.61	1.08
Chess	0.49	0.30	0.18	6.64	16.27

The best detection performance of the five detectors and their best parameters are shown in Tables 4.8 and 4.9, respectively. ZERO++ is better than FPOF, having four wins, only one loss and five draws. ZERO++ outperforms iForest significantly in 8 out of 11 data sets, with three draws but no loss. ZERO++ obtains the best performance in most data sets with $\psi \leqslant 64$, while iForest often requires a larger ψ, e.g., $\psi = 256$. The performance of LOF and SOD has been improved using the best parameters, but is required to search for the parameters in a wide range of values.

Summary. Compared to FPOF, ZERO++ performs significantly better or comparably in most data sets, and it runs two to three orders of magnitude faster in large data sets with high dimensions, such as *Census* and *CoverType*. Note that FPOF cannot work in the high dimensional data set, *Arrhythmia*, due to its high space complexity, even though

Table 4.4: AUC performance comparison between ZERO++, FPOF, iForest, LOF and SOD with the best parameter on categorical data.

	ZERO++	FPOF	iForest	LOF	SOD
	best ψ	best δ	best ψ	best k	best l
Linkage	0.9976±0.0002	0.9978	0.9790±0.0041	n/a	n/a
Census	0.6465±0.0053	0.6148	0.5544±0.0286	n/a	n/a
CoverType	0.9954±0.0021	0.9965	0.9773±0.0043	n/a	n/a
Probe	0.9820±0.0025	0.9867	0.9776±0.0022	n/a	n/a
U2R	0.9910±0.0010	0.9156	0.9729±0.0073	n/a	n/a
AnnThyroid	0.4735±0.0143	0.4868	0.4429±0.0020	0.4965	0.5515
Arrhythmia	0.6905±0.0012	n/a	0.6878±0.0024	0.6295	0.7033
Mushroom	0.9842±0.0023	0.9218	0.9182±0.0117	0.9770	0.9815
Nursery	1.0000±0.0000	1.0000	0.9995±0.0008	1.0000	1.0000
SolarFlare	0.9784±0.0011	0.9791	0.9641±0.0057	0.9778	0.9695
Chess	0.9981±0.0010	0.9122	0.8606±0.0566	0.9948	0.9989
ZERO++ vs. (#wins/losses/draws)		4/1/5	8/0/3	3/1/2	2/2/2

Table 4.5: Parameter settings for the best performance of ZERO++, FPOF, iForest, LOF and SOD on categorical data.

	ZERO++	FPOF	iForest	LOF	SOD
	ψ	δ	ψ	k	l
Linkage	4	0.5	256	n/a	n/a
Census	32	0.1	16	n/a	n/a
CoverType	16	0.1	256	n/a	n/a
Probe	2	0.1	256	n/a	n/a
U2R	16	0.1	256	n/a	n/a
AnnThyroid	64	0.9	64	300	4000
Arrhythmia	4	n/a	256	150	60
Mushroom	128	0.1	256	150	500
Nursery	8	0.1	128	80	1000
SolarFlare	4	0.1	64	500	500
Chess	64	0.1	256	40	250

it has no more than 500 instances. Although ZERO++ runs slower than iForest, it outperforms iForest significantly in most data sets, either with the default setting or the best parameter. LOF and SOD do not work on most categorical data sets. ZERO++ performs significantly better than or comparably to LOF and SOD in most data sets in which LOF and SOD can obtain the results, and it is significantly faster than LOF and SOD by a factor of more than 10 in moderate large data sets, such as *AnnThyroid* and *Mushroom*. The runtime gap between ZERO++ and LOF, SOD would be much larger on large data sets. This can be observed in the next section.

It is interesting to note that ZERO++ can obtain favourable detection performance using very small subsample sizes, such as 2 and 4, and it obtains the best performance in 10 out of 11 data sets using $\psi \leqslant 64$.

4.2.2 Extensions to numeric and mixed data

ZERO++ and FPOF are based on categorical data. Two discretisation methods, i.e., the equal-width method and the $\overline{x} \pm 3s$ rule method, were used to enable them to work in numeric and mixed data. ZERO++ with the equal-width (EW) method and the $\overline{x} \pm 3s$ rule (MS) method are denoted by ZERO++(EW) and ZERO++(MS), respectively. For the EW discretisation method, numeric attributes are discretised into 10 bins by default.

We have also tested FPOF with both EW and MS. Note that MS used in ZERO++ is based on the $\overline{x}$ and s from each subsample, which is not directly applicable for FPOF because FPOF works on the full data set. MS was adapted to FPOF by using the μ and σ from the full data set. Our results showed that the detection performance of FPOF with EW was much better than that using MS. This is because the EW method discretised data in a much smaller granularity than the MS method. Therefore, FPOF with EW obtains significantly more frequent patterns than that using MS, which helps to capture the normal behaviours more effectively [3]. Also, the full data set based MS discretisation can perform poorly, since the μ and σ derived from the full data set are sensitive to anomalies. In this research, we report the best results of FPOF, i.e., FPOF with EW.

Numeric data sets

Table 4.6 shows the AUC performance of ZERO++(MS), ZERO++(EW), FPOF, iForest, LOF and SOD with the default settings on numeric data sets. ZERO++(MS) works best, which obtains the best performance in 10 out of 16 data sets. It outperforms FPOF and iForest significantly in nine data sets, and outperforms both LOF and SOD significantly in 13 data sets. ZERO++(EW) works less effectively than ZERO++(MS). It has 6 and 10 losses in comparison to FPOF and iForest respectively, though it has nine wins in comparison to LOF and SOD.

Note that we cannot obtain the results of FPOF in high dimensional data sets, including *HAR*, *Isolet*, *Mfeat* and *Arrhythmia*, because there are too many frequent itemsets generated in these data sets, leading to out-of-memory exception errors. Also, the runtime for LOF and SOD is prohibitive in the two largest data sets (i.e., *Http* and *Linkage*) and we cannot obtain the results in two weeks.

Table 4.7 presents the runtime comparison between ZERO++(MS), ZERO++(EW), FPOF, iForest, LOF and SOD on numeric data sets. ZERO++(MS) and ZERO++(EW) run two to three orders of magnitude faster than LOF and SOD in most data sets, and they are also two to three orders of magnitude faster than FPOF in two medium-sized 34-dimensional data sets, i.e., *Probe* and *U2R*. ZERO++(MS) runs faster than

[3] As shown in Section 4.2.1 and 4.2.2, FPOF often works best using a small minimum support, e.g., $\delta = 0.1$, which implies that FPOF often needs to work with a larger number of frequent patterns.

Table 4.6: AUC performance comparison between ZERO++, FPOF, iForest, LOF and SOD with the default settings on numeric data.

	ZERO++(MS)	ZERO++(EW)	FPOF	iForest	LOF	SOD
	$\psi = 8$	$\psi = 8$	$\delta = 0.1$	$\psi = 256$	$k = 20$	$l = 20$
Linkage	0.8772±0.0243	0.6390±0.0233	0.5469	0.9974±0.0000	n/a	n/a
Census	0.7890±0.0171	0.7049±0.0101	0.7531	0.6659±0.0072	0.6292	0.6367
CoverType	0.9198±0.0058	0.6882±0.0075	0.5793	0.8726±0.0173	0.5262	0.7202
Probe	0.9900±0.0003	0.9924±0.0003	0.9943	0.9652±0.0110	0.5338	0.5851
U2R	0.9863±0.0005	0.9774±0.0012	0.9795	0.9860±0.0015	0.5471	0.9708
AnnThyroid	0.9128±0.0066	0.6125±0.0061	0.6224	0.8317±0.0136	0.7064	0.7697
Arrhythmia	0.8137±0.0026	0.8102±0.0032	n/a	0.7962±0.0104	0.7707	0.7924
Http	0.9981±0.0012	0.9975±0.0001	0.9973	0.9997±0.0001	n/a	n/a
Mulcross	0.9980±0.0009	0.6517±0.0632	0.9494	0.9533±0.0085	0.6019	0.1396
Smtp	0.8879±0.0082	0.5892±0.0000	0.5892	0.8834±0.0058	0.6514	0.6975
Shuttle	0.9984±0.0001	0.9862±0.0006	0.9751	0.9957±0.0008	0.5243	0.7292
Mammo.	0.8386±0.0080	0.8554±0.0028	0.8537	0.8484±0.0069	0.7396	0.7970
HAR	0.9985±0.0060	0.9995±0.0001	n/a	0.9815±0.0022	0.3218	0.9876
Satimage	0.9856±0.0012	0.9677±0.0061	0.9756	0.9804±0.0028	0.5191	0.6291
Isolet	1.0000±0.0000	0.9987±0.0011	n/a	0.9997±0.0003	0.9999	0.9982
Mfeat	0.9499±0.0060	0.9190±0.0146	n/a	0.9401±0.0124	0.9542	0.9688
ZERO++(MS) vs. (#wins/losses/draws)			9/2/1	9/2/5	13/0/1	13/1/0
ZERO++(EW) vs. (#wins/losses/draws)			4/6/2	3/10/3	9/4/1	9/4/1

ZERO++(EW) because ZERO++(MS) works on 2-bin data while ZERO++(EW) works on 10-bin data. ZERO++(MS) and iForest have similar runtime results.

The best performance of ZERO++(MS), ZERO++(EW), FPOF, iForest, LOF and SOD is presented in Table 4.8. Tuned ZERO++(MS) has similar performance to ZERO++(MS) with the default setting. It has 11, 7 and 11 wins against FPOF, iForest and SOD, respectively. ZERO++(MS) has six wins and six losses compared to LOF. ZERO++(EW) performs better than SOD (with eight wins, five losses and one draws), but it works less effectively than FPOF, iForest and LOF overall.

The best parameter used by each method is reported in Table 4.9. The results show that ZERO++(MS) and ZERO++(EW) obtain the best performance in 13 out of 16 data sets using a small subsample size ($\psi \leqslant 64$) regardless of the diverse characteristics in those data sets (e.g., data size and dimensionality). In contrast, LOF and SOD require a much wider range of parameter searches (range from 10 to 4,000) than ZERO++ in order to obtain the best AUC performance. iForest and FPOF can often perform best using $\psi = 265$ and $\delta = 0.1$, respectively.

Summary. ZERO++ using the MS discretisation method obtains the most favourable detection performance. ZERO++ (MS) with the default setting outperforms FPOF, iForest, LOF and SOD significantly in most data sets. For the results using the best parameter, it has similar superiority to that with the default setting over FPOF, iForest and SOD, and it has comparable detection performance to LOF (six wins and six losses). However, it should be noted that LOF requires a wide range parameter search in order to obtain preferable AUC performance, whereas ZERO++(MS) often obtains the best performance using a small subsample size ($\psi \leqslant 64$), and for ZERO++(MS) there is normally a small gap between the best performance and the performance using the default setting. ZERO++ (EW) performs less effectively than ZERO++(MS), but it is still able to achieve quite comparable detection performance to the other four detectors.

Table 4.7: Runtime comparison between ZERO++, FPOF, iForest, LOF and SOD with the default settings on numeric data.

	ZERO++(MS)	ZERO++(EW)	FPOF	iForest	LOF	SOD
Linkage	79.59	190.72	17.69	140.57	n/a	n/a
Census	6.96	31.34	4.67	9.46	6676.55	35134.71
CoverType	9.01	41.92	4.80	6.07	1214.91	50984.72
Probe	4.32	39.82	13562.03	3.44	981.85	2594.77
U2R	4.08	37.22	12502.24	2.69	844.64	2358.70
AnnThyroid	0.11	0.72	0.48	0.14	0.74	33.46
Arrhythmia	0.56	3.45	n/a	0.03	0.06	0.96
Http	6.75	25.23	1.08	14.30	n/a	n/a
Mulcross	3.87	15.60	1.21	6.31	503.62	59690.21
Smtp	2.03	4.38	0.51	2.64	274.50	6795.19
Shuttle	1.38	6.85	2.87	2.04	124.23	1948.35
Mammography	0.13	1.14	0.46	0.31	15.54	32.02
HAR	12.65	117.67	n/a	3.05	90.26	116.56
Satimage	0.66	4.60	2.57	0.17	8.99	22.64
Isolet	1.55	14.22	n/a	0.03	2.11	3.10
Mfeat	0.73	9.00	n/a	0.01	1.01	1.52

Table 4.8: AUC performance comparison between ZERO++, FPOF, iForest, LOF and SOD with the best parameter on numeric data.

	ZERO++(MS)	ZERO++(EW)	FPOF	iForest	LOF	SOD
	best ψ	best ψ	best δ	best ψ	best k	best l
Linkage	0.9708±0.0246	0.8755±0.0196	0.5469	0.9974±0.0000	n/a	n/a
Census	0.8247±0.0082	0.7476±0.0057	0.79	0.7906±0.0247	0.6690	0.7120
CoverType	0.9598±0.0011	0.9514±0.0027	0.6982	0.8726±0.0173	0.9781	0.8428
Probe	0.9964±0.0000	0.9938±0.0003	0.9954	0.9747±0.0081	0.6321	0.9670
U2R	0.9913±0.0001	0.9782±0.0006	0.9846	0.9860±0.0015	0.8873	0.9747
AnnThyroid	0.9157±0.0082	0.6268±0.0081	0.6460	0.8717±0.0287	0.7170	0.7800
Arrhythmia	0.8146±0.0030	0.8157±0.0018	n/a	0.8164±0.0106	0.8296	0.8000
Http	0.9990±0.0001	0.9975±0.0001	0.9973	0.9997±0.0001	n/a	n/a
Mulcross	0.9992±0.0008	0.7204±0.0638	0.9494	0.9979±0.0012	0.6035	0.2667
Smtp	0.9018±0.0042	0.5892±0.0000	0.5892	0.8834±0.0058	0.9500	0.8214
Shuttle	0.9988±0.0002	0.9872±0.0005	0.9751	0.9962±0.0007	0.9809	0.9919
Mammography	0.8412±0.0052	0.8574±0.0031	0.8555	0.8557±0.0101	0.8644	0.8100
HAR	0.9986±0.0090	0.9998±0.0000	n/a	0.9996±0.0001	0.9988	0.9900
Satimage	0.9884±0.0027	0.9683±0.0033	0.9756	0.9836±0.0027	0.9934	0.9800
Isolet	1.0000±0.0000	0.9999±0.0001	n/a	0.9997±0.0003	1.0000	1.0000
Mfeat	0.9643±0.0016	0.9559±0.0068	n/a	0.9401±0.0124	0.9800	0.9700
ZERO++(MS) vs. (#wins/losses/draws)			11/1/0	7/3/6	6/6/2	11/1/2
ZERO++(EW) vs. (#wins/losses/draws)			4/6/2	4/8/4	6/7/1	8/5/1

Table 4.9: Parameter settings for the best performance of ZERO++, FPOF, iForest, LOF and SOD on numeric data.

	ZERO++(MS)	ZERO++(EW)	FPOF	iForest	LOF	SOD
	ψ	ψ	δ	ψ	k	l
Linkage	2	256	0.1	256	n/a	n/a
Census	4	2	0.4	8	80	250
CoverType	128	256	0.5	256	3000	1000
Probe	128	2	0.2	128	4000	4000
U2R	128	2	0.8	256	500	500
AnnThyroid	16	2	0.4	16	40	10
Arrhythmia	4	64	n/a	128	80	500
Http	16	8	0.1	256	n/a	n/a
Mulcross	4	4	0.1	16	40	300
Smtp	2	8	0.1	256	1000	500
Shuttle	16	16	0.1	128	4000	4000
Mammography	16	4	0.2	64	150	60
HAR	4	2	n/a	8	4000	4000
Satimage	32	16	0.1	64	2000	2000
Isolet	8	32	n/a	256	20	20
Mfeat	64	128	n/a	256	80	40

In terms of runtime, ZERO++(MS) and ZERO++(EW) are two to three orders of magnitude faster than LOF and SOD in most data sets, and they run significantly faster than FPOF in data sets with slightly higher dimensions (i.e., *Probe* and *U2R*) by a factor of more than 100 and 1,000, respectively. ZERO++(MS) runs faster than ZERO++(EW), and has similar runtime results as iForest.

Mixed data sets

Table 4.10 presents the detection performance of ZERO++, FPOF, iForest, LOF and SOD with the default settings on mixed data. It shows that ZERO++(MS) achieves the best performance in five out of seven data sets, with the other two close to the best. ZERO++(MS) outperforms FPOF and iForest significantly in four data sets, and outperforms LOF and SOD in six data sets. ZERO++(EW) has similar advantages to ZERO++(MS) over FPOF, LOF and SOD. Note that we cannot obtain the results of FPOF in *CoverType* and *Arrhythmia* because the space complexity of FPOF is prohibitive in these two data sets and it results in out-of-memory exception errors. Also, the runtime of LOF and SOD in *Linkage* is too expensive and we cannot get the results in two weeks.

Runtime results for each method in each data are presented in Table 4.11. Both versions of ZERO++ are significantly faster than FPOF, LOF and SOD by a factor more than 100 or 1,000 in most data sets. They run slower than iForest.

The best performance of each detector in each data set is presented in Table 4.12. ZERO++(MS) with the best parameter achieves similar performance to that using the default setting. It performs significantly better, in five data sets than FPOF and iForest, and in six out of seven data sets than LOF and SOD.

The best parameter used by each method is reported in Table 4.13. ZERO++(MS) obtains the best performance in all the seven data sets using a small subsample size ($\psi \leqslant 64$), and requires a smaller search range than ZERO++(EW), iForest, LOF and SOD. The best

Table 4.10: AUC performance comparison between ZERO++, FPOF, iForest, LOF and SOD with the default settings on mixed data.

	ZERO++(MS)	ZERO++(EW)	FPOF	iForest	LOF	SOD
	$\psi = 8$	$\psi = 8$	$\delta = 0.1$	$\psi = 256$	$k = 20$	$l = 20$
Linkage	0.9997±0.0001	0.9779±0.0059	0.9715	0.9999±0.0001	n/a	n/a
Census	0.7710±0.0043	0.6634±0.0051	0.6564	0.5712±0.0174	0.5479	0.6746
CoverType	0.9780±0.0073	0.9432±0.0045	n/a	0.9436±0.0235	0.5363	0.7216
Probe	0.9965±0.0002	0.9965±0.0003	0.9971	0.9954±0.0014	0.5475	0.6505
U2R	0.9876±0.0003	0.9837±0.0007	0.9807	0.9832±0.0015	0.5827	0.9405
AnnThyroid	0.8436±0.0114	0.5923±0.0053	0.5801	0.6486±0.0126	0.6683	0.5850
Arrhythmia	0.8119±0.0040	0.8097±0.0021	n/a	0.8026±0.0085	0.6522	0.7807
ZERO++(MS) vs. (#wins/losses/draws)			4/1/0	4/0/3	6/0/0	6/0/0
ZERO++(EW) vs. (#wins/losses/draws)			4/1/0	1/2/4	5/0/1	5/0/1

Table 4.11: Runtime comparison between ZERO++, FPOF, iForest, LOF and SOD with the default settings on mixed data.

	ZERO++(MS)	ZERO++(EW)	FPOF	iForest	LOF	SOD
Linkage	310.39	315.76	171.14	200.80	n/a	n/a
Census	102.16	107.81	174559.87	16.03	106648.16	265268.48
CoverType	44.17	132.94	n/a	10.23	2056.70	83013.78
Probe	27.71	22.89	174745.96	3.35	1916.51	2383.72
U2R	29.57	21.77	157964.46	3.10	1413.63	3851.64
AnnThyroid	1.04	1.47	238.84	0.33	8.95	37.74
Arrhythmia	6.34	4.05	n/a	0.02	0.76	0.63

Table 4.12: AUC performance comparison between ZERO++, FPOF, iForest, LOF and SOD with the best parameter on mixed data.

	ZERO++(MS)	ZERO++(EW)	FPOF	iForest	LOF	SOD
	best ψ	best ψ	best δ	best ψ	best k	best l
Linkage	0.9998±0.0001	0.9939±0.0010	0.9975	0.9999±0.0001	n/a	n/a
Census	0.7711±0.0049	0.6678±0.0027	0.6680	0.5712±0.0174	0.5745	0.6746
CoverType	0.9866±0.0035	0.9891±0.0013	n/a	0.9436±0.0235	0.7051	0.9551
Probe	0.9982±0.0000	0.9970±0.0001	0.9972	0.9971±0.0008	0.6260	0.9662
U2R	0.9909±0.0002	0.9880±0.0006	0.9857	0.9832±0.0015	0.9205	0.9854
AnnThyroid	0.8718±0.0177	0.5964±0.0032	0.5828	0.7818±0.0320	0.6905	0.6344
Arrhythmia	0.8148±0.0034	0.8134±0.0015	n/a	0.8127±0.0122	0.6522	0.7978
ZERO++(MS) vs. (#wins/losses/draws)			5/0/0	5/0/2	6/0/0	6/0/0
ZERO++(EW) vs. (#wins/losses/draws)			2/2/1	3/2/2	5/1/0	4/2/0

parameter of FPOF ranges from 0.2 up to 0.8.

Table 4.13: Parameter settings for the best performance of ZERO++, FPOF, iForest, LOF and SOD on mixed data.

	ZERO++(MS)	ZERO++(EW)	FPOF	iForest	LOF	SOD
	ψ	ψ	δ	ψ	k	l
Linkage	16	64	0.7	256	n/a	n/a
Census	4	128	0.4	256	10	20
CoverType	16	64	n/a	256	2000	1000
Probe	32	2	0.2	128	1000	500
U2R	64	64	0.8	256	500	250
AnnThyroid	4	16	0.4	16	10	2000
Arrhythmia	4	64	n/a	32	20	40

Summary. ZERO++(MS), with either the default setting or the best parameter, performs consistently and significantly better than FPOF, iForest, LOF and SOD in most data sets. ZERO++(MS) obtains the best performance in all the data sets using $\psi \leqslant 64$. Although ZERO++(EW) performs less effectively than ZERO++(MS), its performance is comparable to FPOF and iForest, and is superior to LOF and SOD.

Both ZERO++(MS) and ZERO++(EW) run two to three orders of magnitude faster than FPOF, LOF and SOD in most data sets. They are slower than iForest.

4.3 Ability to tolerate irrelevant attributes

In many real-world anomaly detection tasks, anomalies are only visible in some attributes. Other attributes are irrelevant to anomaly detection and degrade detection performance of anomaly detectors which cannot tolerate irrelevant attributes. In this section, we examined the ability of ZERO++ to handling data sets with irrelevant attributes. We focused on using synthetic numeric data sets in this experiment, because it is much easier and more straight-forward to define irrelevant attributes and create anomalies compared to categorical or mixed data.

A Gaussian cluster with 100-dimensional 10,000 instances was generated where anomalies could only be detected with r percentage of attributes which are relevant. The other $(1-r)$ percentage of attributes are irrelevant attributes with uniform random noise. From the 10,000 instances, 2% were randomly selected and applied an offset to create anomalies, which were randomly generated outside the range $[\mu-2\sigma, \mu+2\sigma]$ in the relevant attributes. This is the same method as used in (Zimek et al., 2012). Note that the relevant attributes of each anomaly are randomly selected, and thus each anomaly has different relevant attributes. We examined r in the range 1%, 2%, 3%, ..., 30%. At each r value, an average result is reported from 10 runs using 10 generated data sets.

Figure 4.1 shows that ZERO++(MS) works best in data sets having a low percentage of relevant dimensions: it performs better than iForest, LOF and SOD in data sets having 1 to 15 relevant dimensions out of the 100 dimensions; and it performs significantly better than FPOF in data sets having one to five relevant dimensions. Note that with only 1% relevant dimensions, ZERO++(MS) can perform at the level of AUC close to 1, whereas the other four detectors have AUC about 0.55 to 0.65, which are nearly equivalent to random ranking. ZERO++(EW) performs less effectively than ZERO++(MS) in data sets with relevant attributes lower than 5%, but it performs better than the other four detectors.

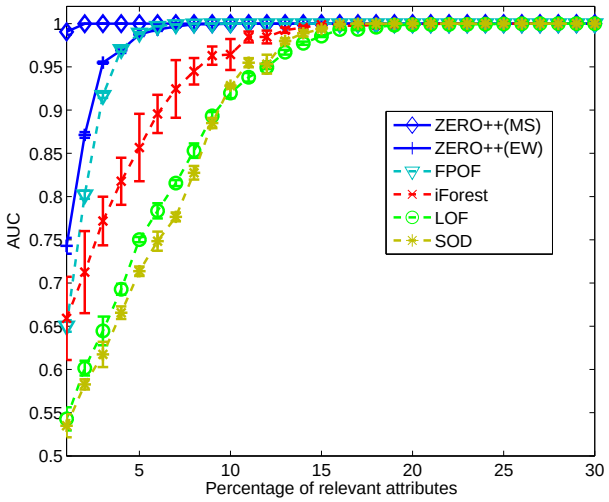

Figure 4.1: AUC performance of ZERO++ with increasing percentage of relevant dimensions, using FPOF, iForest, LOF and SOD as baselines.

4.4 Scalability examination

This section investigates the scalability of ZERO++ with respect to data dimensionality and data size, using FPOF, iForest, LOF and SOD as baselines.

Section 4.2 showed that ZERO++(MS) generally worked much better than ZERO++(EW) in numeric and mixed data sets. Also, in Section 4.3, ZERO++(MS) performed well in data sets with a very low percentage of relevant attributes. Therefore, in the following experiments, we use ZERO++(MS) in handling numeric and mixed data sets by default.

4.4.1 Dimensionality

We examine the scalability of ZERO++ with respect to dimensionality using seven synthetic data sets. The data sets contain the same number of instances, i.e., 10,000 instances, but have different dimensions, ranging from 10 dimensions up to 1,000 dimensions. The results are shown in Figure 4.2. The results show that both ZERO++ and iForest have runtime linear to the data dimensionality, and run two orders of magnitude faster than LOF and SOD, and three orders of magnitude faster than FPOF. Note that the space complexity of FPOF increases quickly with increasing dimensions, and FPOF runs out-of-memory when the dimension reaches 500.

4.4.2 Data size

We examine the scalability of ZERO++ with respect to data size using seven subsets of the largest data set *Linkage*. The smallest data subset contains 1,000 instances, and other subsets increase by a factor of four, the largest subset contains 4,096,000 instances. In this experiment, both numeric and categorical attributes in *Linkage* are used. All the categorical attributes in *Linkage* are boolean attributes. Therefore, the discretised version and the categorical-to-numeric converted version of *Linkage* have the same number of attributes. This ensures that categorical data oriented methods and numeric data oriented methods work on data sets with the same dimensionality.

The scaleup test results are reported in Figure 4.3. It shows that ZERO++, FPOF and iForest are linear to data size. ZERO++ is comparably fast to FPOF and iForest, and runs significantly faster than LOF and SOD by a factor of more than 1,000.

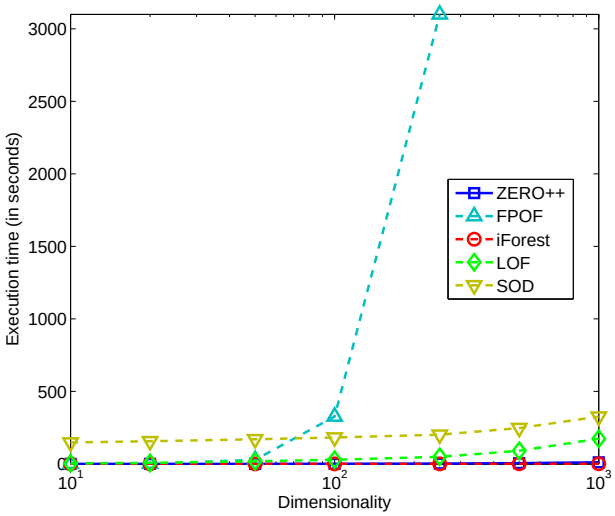

Figure 4.2: Scaleup test of ZERO++ with respect to data dimensionality using FPOF, iForest, LOF and SOD as baselines. Each data set contains 10,000 instances and its dimensionality ranges from 10 to 1,000. A logarithmic scale is used on the horizontal axis.

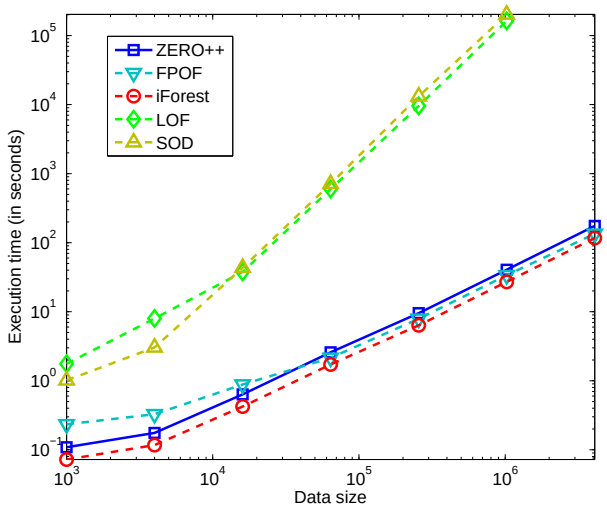

Figure 4.3: Scaleup test of ZERO++ with respect to data size using FPOF, iForest, LOF and SOD as baselines. Data size ranges from 1,000 to 4,096,000. Logarithmic scale is used on both axes.

4.5 Sensitivity examination

The subsampling size ψ and ensemble size t are the only two parameters in ZERO++. We investigated the sensitivity of ZERO++ with respect to ψ and t in all the 20 data sets. We used the default setting for ψ when conducting the sensitivity test with respect to t, and vice versa. Most data sets have similar sensitivity results for ψ and t. For better readability, we focus on discussing distinctive results here only. The results in all the 20 data sets are presented in Appendix C.

Figure 4.4 reports the AUC mean values and two standard error bars over 10 runs of ZERO++ with respect to ψ in four selected data sets. The results show that although the detection performance of ZERO++ may vary with increasing subsample size, ZERO++

normally achieves the best performance using small subsample sizes, e.g., $\leqslant 64$, in data sets of different characteristics, e.g., different data sizes, diverse dimensionality sizes and data sets with different types of attributes. ZERO++ performs stably as the two standard errors are very small, e.g., they are often smaller than 0.01. Also, it is interesting to note that the performance of ZERO++ can often converge at the very beginning with respect to ψ. Similar results can also be found in other data sets, as illustrated in Figure C.1.

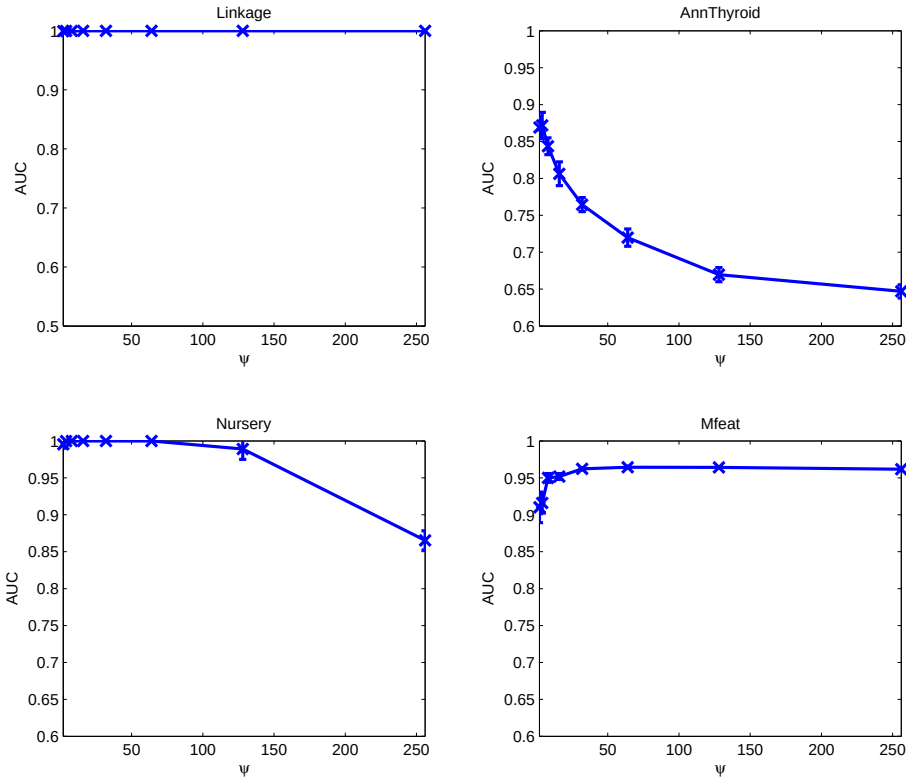

Figure 4.4: Sensitivity test with respect to ψ on the four selected data sets.

Figure 4.5 presents the AUC mean values and two standard error bars over 10 runs of ZERO++ with respect to t in the four selected data sets. The results show that the AUC performance of ZERO++ converges very quickly with respect to t. ZERO++ normally obtains the best performance and works stably using $t \geqslant 30$. Similar results can also be found in other data sets in Figure C.2.

4.6 Application on data sets without ground truth

In this section, we investigated the applications of ZERO++ on three UCI data sets without ground truth, i.e., *Mnist*, *Zoo* and *Internet Usage*. ZERO++ was used with the default setting in the following experiments.

Mnist: This data set contains 60,000 and 10,000 images of handwritten digits in the training and test sets, respectively. Each image has 784 pixel features, but most of the pixels are blank. We examined ZERO++ on the test set with reduced pixel features, i.e., to extract 96 features using the block size 14 (Maji and Malik, 2009). Since different people have different handwriting styles, we were interested in looking for handwritten exceptions for each digit. To this end, we ran ZERO++ on the test set and ranked all the instances according to their anomaly scores. The two top ranked anomalies for each digit were then picked up and visualised in Figure 4.6. These top ranked images are all poor

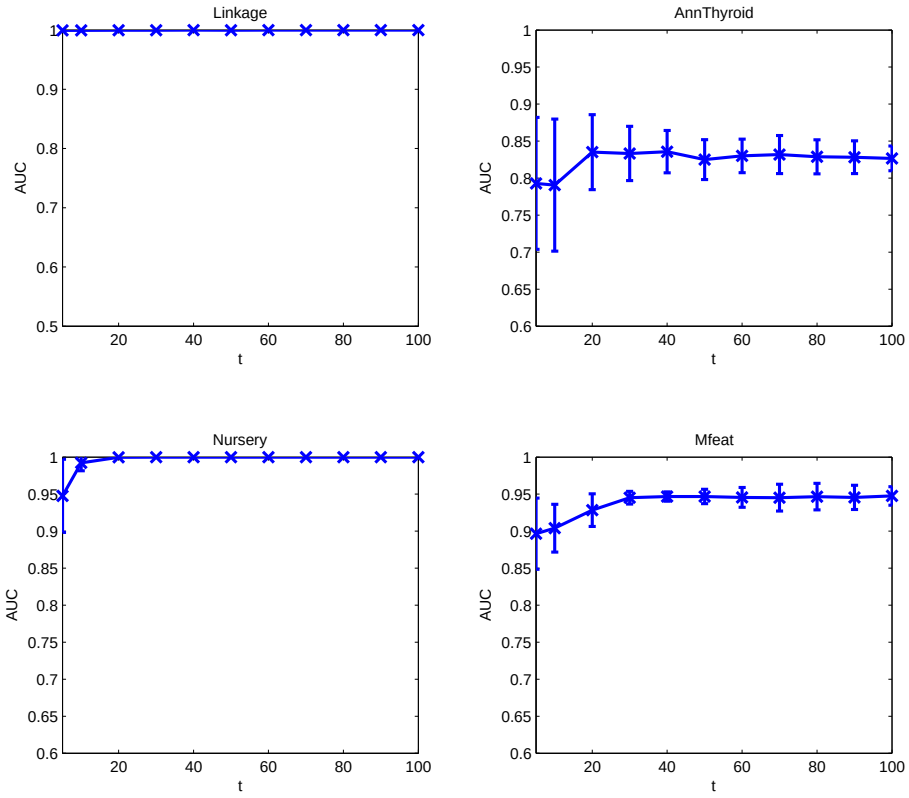

Figure 4.5: Sensitivity test with respect to t on the four selected data sets.

written digits because their pixel feature values deviate substantially from the typical digit images.

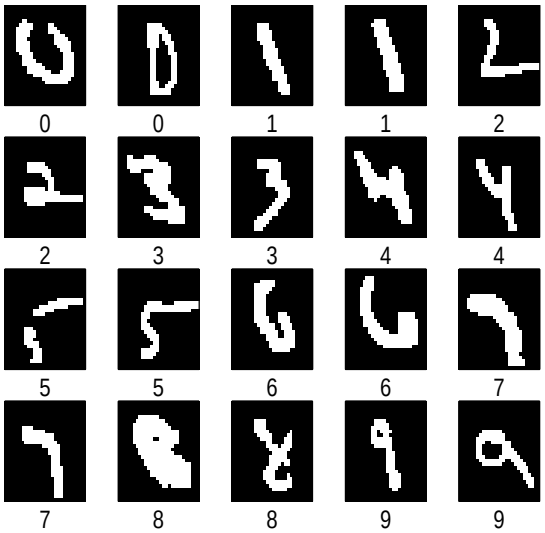

Figure 4.6: Top two anamolies for each digit in *Mnist* detected by ZERO++.

Zoo: *Zoo* consists of 101 instances from 7 species of animals, including 8 instances of insect, 10 instances of invertebrate, 20 instances of bird, 41 instances of mammal, 13 instances of fish, 4 instances of amphibian and 5 instances of reptile. There are 15 Boolean attributes and 2 numeric attributes. The two numeric attributes, including the number of legs and animal type, were regarded as nominal attributes since the difference between the attribute values are not statistically meaningful in defining an animal. We ran ZERO++

on this data and obtained the following top three anomalies: honeybee, scorpion and octopus:

- Honeybee is a top ranked anomaly because it is the only insect animal that is venomous and domestic.
- Scorpion is an unusual invertebrate animal because it has a breathing system and a tail, contrasting to all the other invertebrate animals.
- Octopus is an extreme case in the data set because none of the other animals has eight legs and is cat sized.

Similar results on this data set can also be found in (Kriegel and Zimek, 2008), where scorpion and octopus are considered as the two top ranked anomalies.

Internet usage: This data set comes from a survey about Internet usage in 1997. It consists of 10,104 instances with 71 categorical attributes plus an ID attribute. Each instance contains general demographic information on an Internet user. The number of labels in the attributes range from 2 to 129. The two top ranked anomalies detected by ZERO++ are users 99179 and 91839:

- User 99179 is a five year old nurse in Denmark with 1-3 years Internet usage.
- User 91839 is a nine year old Virginia male with a college degree and has a networking occupation but has less than six months on Internet.

4.7 Discussion

As presented in Sections 4.2 to 4.4, ZERO++ (either with the default setting $\psi = 8$ and $t = 50$ or the best parameter) is more favourable than (or competitive to) its four state-of-the-art contenders in the 20 data sets with different characteristics, i.e., diverse in data size, data type and data dimensionality. This section provides an analysis of these results from three perspectives as follows:

- **Data size**. Given an instance, its probability of having zero appearances in subspaces is dependent on the frequency of the instance. Since the nature of anomaly is rare and exceptional, anomalies have rare attribute values regardless of data size. As illustrated in Figure 3.9, such instances were very likely to have zero appearances in subspaces when using a small subsample size, e.g., if the attribute values accounted for no greater than 1%, the probability of the values having zero appearances was more than 0.9 when using $\psi = 8$.

  For LOF and SOD, their detection performance is mainly dependent on the size of its neighbourhood set, which is strongly related to the data size. In general, they required a large neighbourhood size to perform well in large data sets (e.g., range from 500 to 4,000 in *CoverType*, *Probe* and *U2R* in Table 4.9) while a small neighbourhood size was needed in small data sets (e.g., range from 10 to 80 in *AnnThyroid*, *Isolet* and *Mfeat* in Table 4.9). As reported in Tables 4.5, 4.9 and 4.13, since iForest considers a few subspaces only, it normally requires a larger subsample size ($\psi = 265$) than ZERO++ to perform well in large data sets. It is difficult to capture all the normal patterns in a data set, especially in large data sets, so a small δ is normally needed in FPOF in order to obtain a sufficient number of normal patterns.

  The runtimes of ZERO++, iForest and FPOF are linear to data size, while LOF and SOD are at least quadratic to data size. It should be noted that FPOF has

lower linear time complexity than ZERO++ with respect to data size, so ZERO++ ran slower than FPOF in large data sets with low dimensionality, such as the largest data set *Linkage*. ZERO++ also ran slower than iForest, this is because ZERO++ considers a lot more subspaces than iForest. For LOF and SOD, indexing methods can be employed to reduce time complexity from $O(n^2)$ to $O(n \log n)$, which is still much higher than ZERO++.

- **Data type**. ZERO++ and FPOF are categorical data oriented methods, and as reported in Tables 4.2 and 4.5 they performed better than the other three numeric data oriented methods (iForest, LOF and SOD) with one-of-ℓ categorical-to-numeric transformation method in categorical data. However, the reverse is not true: according to the results in Tables 4.6 and 4.8, ZERO++ and FPOF, using MS or EW discretisation methods, also performed better than or were very competitive to the three numeric data oriented methods in numeric data. This is mainly because the categorical-to-numeric transformation is a more difficult task than the discretisation, as categorical attributes do not have the notion of ordering and often contain only a few labels; and the effectiveness of the transformation is often context dependent (Boriah et al., 2008).

  It should be noted that the MS discretisation method is based on an underlying assumption that normal instances follow uni-modal distributions. Due to this assumption, ZERO++ (MS) works best in data sets where normal instances follow uni-modal distributions in relevant attributes and anomalies lie outside of the distributions, e.g., *Isolet* and *Mulcross*. It fails to work if anomalies lie at the middle of the distributions, e.g., for a two-dimensional data set, normal instances have a distribution in the shape of a doughnut and an anomaly is located in the centre of the doughnut. In data sets where normal instances follow multi-modal distributions in relevant attributes, ZERO++ (MS) can still work well if anomalies lie outside the distributions, e.g., *Satimage* and *HAR*; but it fails to work if anomalies lie inside the distributions.

  The runtime of ZERO++ and FPOF is dependent on the number of bins used in the discretisation method. Their runtime can be slightly longer if the number of bins produced is large. For example, in Tables 4.7 and 4.11, ZERO++ using the 10-bin EW discretisation method ran slower than that using the 2-bin MS method.

  It should be noted that the gap between the runtimes of ZERO++ and iForest varied in different data types: ZERO++ ran much slower than iForest in categorical data sets, but it had comparable runtime as iForest when using the 2-bin MS method in numeric data sets. This is because: in categorical data, iForest built isolation trees on binary attributes, which ran faster than building trees on attributes having a range of continuous values in numeric data; while for ZERO++, it ran slower in categorical data than that using the 2-bin MS discretisation method in numeric data because attributes in categorical data often contain multiple labels.

  For LOF and SOD, since there are too many identical attribute values in converted categorical attributes, R^*-tree or other tree indexing methods cannot work in data sets with categorical attributes only, and work less effectively in mixed data. That is why even though the R^*-tree indexing method was employed in LOF and SOD, their runtime still increased quickly with data size in the mixed data *Linkage*, as shown in Figure 4.3.

- **Data dimensionality**. In anomaly detection, one typical challenge related to data dimensionality is irrelevant attributes (Zimek et al., 2012). ZERO++, FPOF and

Table 4.14: A summary of the ability of ZERO++ to meet the four challenges stated in Section 1.2. The four challenges include the ability to handle data sets with different types of attributes (A), high detection accuracy (B), scale up to very large data size and high dimensionality (C) and tolerant to irrelevant attributes (D).

Challenges	Performance of ZERO++ (With four well-known detectors as baselines)
A & B	ZERO++ was able to identify anomalies in data sets with different types of attributes effectively. It performed consistently better than iForest in categorical, numeric and mixed data sets. It performed comparably to FPOF in categorical data, and outperformed FPOF significantly in most numeric and mixed data sets. ZERO++ performed comparably to LOF and significantly better than SOD in numeric data, and outperformed LOF and SOD significantly in all the mixed data sets.
C	ZERO++ had linear time complexity to data dimensionality and data size, so it could scale up well with very large and high dimensional data. It ran two to three orders of magnitude faster than LOF and SOD in large data sets, and is two to three orders of magnitude faster than FPOF in data set with high dimensions. ZERO++ had comparable runtime as iForest in large and low dimensional data, and ran slower than iForest in high dimensional data.
D	ZERO++ could identify anomalies in data sets with a low percentage of relevant attributes. ZERO++ worked very well in data sets with irrelevant attributes, and it outperformed the four contenders significantly in data sets with a very low percentage of relevant attributes..

iForest work on low dimensional subspaces, while LOF and SOD use the full dimensionality to define distance and thus are sensitive to irrelevant attributes. Therefore, ZERO++, FPOF and iForest performed better than LOF and SOD in data sets with a high percentage of irrelevant attributes, as shown in Figure 4.1.

iForest considers a few subspaces only, which are spanned by some randomly selected attribute subsets, and all these subspaces used in iForest are likely to be spanned by irrelevant attributes only in data sets with many irrelevant attributes; whereas the subspaces used in ZERO++ and FPOF cover all the attributes, and at least a portion of subspaces are spanned by relevant attributes. Therefore, in data sets with a very low percentage of relevant attributes, iForest is very likely to work on irrelevant subspaces only, and performs much less effectively than ZERO++ and FPOF, which work on at least a portion of relevant subspaces.

4.8 Chapter summary

A summary of the ability of ZERO++ to meet the four challenges stated in Section 1.2 is provided in Table 4.14. In general, our assessment showed that ZERO++ could identify anomalies more effectively than FPOF, iForest, LOF and SOD in data with different attribute types, and was able to identify anomalies in data sets with a very low percentage of relevant attributes. In terms of efficiency, ZERO++ scaled up well with data size and dimensionality, and ran two to three orders of magnitude faster than FPOF, LOF and SOD.

ZERO++ has two parameters, i.e., subsample size ψ and ensemble size t. Our results showed that ZERO++ often obtained the best (or close to the best) detection performance using a small subsample size, i.e., $\psi \leqslant 64$; and it converged very quickly with respect to t,

for example, it normally converged at $t = 30$. It is worth noting that ZERO++ with the default setting, i.e., $\psi = 8$ and $t = 50$, obtained favourable AUC performance in data sets with different characteristics.

Also, our results on data sets with unknown ground truth showed that ZERO++ was able to: identify unusual patterns in recognition of poor written digit images, detect unusual animal species and remove data noise in survey response data.

Chapter 5

Conclusion

This thesis proposes the anomaly detection method ZERO++ and makes the following four key contributions to the field of anomaly detection:

First, we introduce a novel anomaly detection method ZERO++ which employs the number of zero appearances in subspaces to identify anomalies. We provide a statistical justification that, given a set of subsamples, anomalies are likely to have a higher number of zero appearances in subspaces than that for normal instances. ZERO++ is unique in that it works in regions of subspaces that are not occupied by data; whereas existing methods work in regions occupied by data. It has linear time complexity with respect to data size and data dimensionality, and it has constant space complexity with a small constant.

Second, we examine two discretisation methods, i.e., the equal-width method and the $\overline{x} \pm 3s$ rule discretisation method, for enabling ZERO++ to handle numeric and mixed data. We demonstrate that although ZERO++ is based on categorical data, it can handle numeric and mixed data effectively by using a discretisation method in the preprocessing step.

Third, a series of empirical results is conducted to compare ZERO++ with four state-of-the-art anomaly detectors, including one categorical data oriented detector (FPOF) and three numeric data oriented detectors (iForest, LOF and SOD), and show that ZERO++ is superior in terms of:

- its ability to detect anomalies in data sets with different types of attributes,
- its ability to tolerate irrelevant attributes, and
- its scalability with respect to data size and data dimensionality. (Note that both iForest and ZERO++ have linear time complexity to data size and data dimensionality.)

Fourth, we have an empirical investigation on the performance of categorical (or numeric) data oriented anomaly detectors that work in numeric (or categorical) data and mixed data. There is a lack of such empirical results in the literature. Our results enable researchers to understand the detection performance of existing well-known detectors in data sets with different types of attributes. The results for categorical and mixed data sets are particularly important references because relatively few methods have been proposed for these two types of data.

In future work, we are interested in designing more advanced discretisation methods for ZERO++, such as density based variable-width discretisation methods (Kontkanen and Myllymäki, 2007), in order to handle data sets with different distributions more effectively, e.g., data sets with multi-modal distributions. We also plan to modify ZERO++ to detect all the anomalies automatically in data sets where abnormal behaviours are dependent

on different numbers of attributes, e.g., through the use of subsets of R'_m with different m values. ZERO++ might be applicable for data streams without major modifications. This is because it requires only a set of small subsamples to train detection models, so it can update models and detect anomalies quickly in data streams.

As discussed in Section 1.1.2, in categorical data, there is no formal definition to different types of anomaly (i.e., scattered point anomalies, including global anomalies and local anomalies, and clustered anomalies) proposed in numeric data, and existing research in categorical domain focuses on point anomalies only. The key challenge for defining these anomalies in categorical data lies in the definition of an effective metric based on the unordered categorical attributes. In future work, we are also interested in exploring ways to define and differentiate those different types of anomaly in categorical data, and utilise ZERO++ to detect all these anomalies.

Appendix A

Proofs of theorems

Theorem 1 The probability of $\mathcal{Z}_{\mathcal{S}}(\mathbf{y})$ is equal to its expected value $E(\mathcal{Z}_{\mathcal{S}}(\mathbf{y})) = \frac{\binom{n-r_{\mathcal{S}}(\mathbf{y})}{\psi}}{\binom{n}{\psi}}$.

Proof 1 *There are $\binom{n-r_{\mathcal{S}}(\mathbf{y})}{\psi}$ choices for sampling ψ instances from n instances while excluding instances that are identical to $\mathbf{y}$ in $\mathcal{S}$.*

On the other hand, simply sampling ψ instances from n instances has $\binom{n}{\psi}$ choices.

Therefore,

$$E(\mathcal{Z}_{\mathcal{S}}(y)) = \frac{\binom{n-r_{\mathcal{S}}(\mathbf{y})}{\psi}}{\binom{n}{\psi}}$$

Theorem 2 If $\mathcal{Z}_{\mathcal{S}}(\mathbf{y})$ are independent, then

$$0 \leq E(score(\mathbf{y}|\mathcal{D}, R)) \leq |R| \left(1 - \left(1 - \frac{\binom{n-r(\mathbf{y})}{\psi}}{\binom{n}{\psi}}\right)^{\frac{1}{|R|}}\right)$$

Moreover, if $E(\mathcal{Z}_{\mathcal{S}}(\mathbf{y}))$ are identical for every $\mathcal{S} \in R$, then

$$E(score(\mathbf{y}|\mathcal{D}, R)) = |R| \left(1 - \left(1 - \frac{\binom{n-r(\mathbf{y})}{\psi}}{\binom{n}{\psi}}\right)^{\frac{1}{|R|}}\right)$$

Proof 2 *If $\psi > n - r(\mathbf{y})$, then $\mathcal{D} \cap \{\mathbf{x} \in \mathcal{D} : \mathbf{x} = \mathbf{y}\} \neq \emptyset$*

Therefore, $\exists \mathbf{x} \in \mathcal{D}$ s.t. $\mathbf{x} = \mathbf{y}$, and so

$$P_{\mathcal{S}}(\mathbf{y}|\mathcal{D}) \neq 0, \forall \mathcal{S} \in R$$

And thus

$$score(\mathbf{y}|\mathcal{D}, R) = 0$$

If $\psi \leq n - r(\mathbf{y})$, then

$$I\left(P_{\mathcal{A}}(\mathbf{y}|\mathcal{D}) = 0\right) = 1 - \prod_{\mathcal{S} \in R} \left(1 - I\left(P_{\mathcal{S}}(\mathbf{y}|\mathcal{D}) = 0\right)\right) \quad \text{(A.1)}$$

$$= 1 - \prod_{\mathcal{S} \in R} \left(1 - \mathcal{Z}_{\mathcal{S}}(\mathbf{y})\right) \quad \text{(A.2)}$$

Since $\mathcal{Z}_{\mathcal{S}}(\mathbf{y})$ are independent, and the geometric mean is bounded above by the arithmetic mean,

$$\begin{aligned} E(I\left(P_{\mathcal{A}}(\mathbf{y}|\mathcal{D})=0\right)) &= 1-\prod_{\mathcal{S}\in R}\left(1-E(\mathcal{Z}_{\mathcal{S}}(\mathbf{y}))\right) && \text{(A.3)}\\ &\geqslant 1-\left(\frac{1}{|R|}\sum_{\mathcal{S}\in R}(1-E(\mathcal{Z}_{\mathcal{S}}(\mathbf{y})))\right)^{|R|} && \text{(A.4)}\\ &= 1-\left(1-\frac{\sum_{\mathcal{S}\in R}E(\mathcal{Z}_{\mathcal{S}}(\mathbf{y}))}{|R|}\right)^{|R|} && \text{(A.5)}\end{aligned}$$

On the other hand,

$$\begin{aligned} E(I(P_{\mathcal{A}}(\mathbf{y}|\mathcal{D})=0) &= E(\mathcal{Z}_{\mathcal{A}}(\mathbf{y})) && \text{(A.6)}\\ &= \frac{\binom{n-r(\mathbf{y})}{\psi}}{\binom{n}{\psi}} && \text{(A.7)}\end{aligned}$$

So,

$$\frac{\binom{n-r(\mathbf{y})}{\psi}}{\binom{n}{\psi}} \geqslant 1-\left(1-\frac{\sum_{\mathcal{S}\in R}E(\mathcal{Z}_{\mathcal{S}}(\mathbf{y}))}{|R|}\right)^{|R|} \quad \text{(A.8)}$$

And therefore,

$$\begin{aligned} E(score(\mathbf{y}|\mathcal{D},R)) &= \sum_{\mathcal{S}\in R}E(\mathcal{Z}_{\mathcal{S}}(\mathbf{y})) && \text{(A.9)}\\ &\leq |R|\left(1-\left(1-\frac{\binom{n-r(\mathbf{y})}{\psi}}{\binom{n}{\psi}}\right)^{\frac{1}{|R|}}\right) && \text{(A.10)}\end{aligned}$$

Moreover, if all $E(\mathcal{Z}_{\mathcal{S}}(\mathbf{y}))$ are identical, the geometric mean is the same as the arithmetic mean, so

$$E(score(\mathbf{y}|\mathcal{D},R)) = |R|\left(1-\left(1-\frac{\binom{n-r(\mathbf{y})}{\psi}}{\binom{n}{\psi}}\right)^{\frac{1}{|R|}}\right) \quad \text{(A.11)}$$

Appendix B

Details for datasets used

We used 19 real-world data sets, including *Linkage*, *Census*, *CoverType*, *Probe*, *U2R*, *AnnThyroid*, *Arrhythmia*, *Nursery*, *Chess*, *Mushroom*, *SolarFlare*, *Http*, *Smtp*, *Shuttle*, *Mammography*, *HAR*, *Satimage*, *Isolet* and *Mfeat*, and one synthetic data set *Mulcross* (Hadi, 1992), to examine the effectiveness of ZERO++ in handling data sets with different types of attributes in Section 4.2. These data sets were selected mainly because they had been used widely in previous literature.

- *Linkage* is a data set used for element-wise record linkage comparison. Its task is to decide whether underlying records match one person based on phonetic equality of first name and family name, date of birth and gender. It has two classes, 'match' and 'non-match', and contains 5,749,132 instances, of which only 20,931 instances belong to 'match'. We treat the 'match' class as the anomaly class. There are 11 attributes in its original form, but two of them have more than 99% missing values, so we remove these two attributes in our experiment. This data set was used in Ngufor and Wojtusiak (2013).

- *Census* is short for Census-Income data, which contains two classes '50K–' and '50K+', indicating whether a survey respondent has annual income over $50K. The small class '50K+' is used as anomalies. This data set was used in Ghoting et al. (2004) and Zhang and Jin (2011).

- *CoverType* is used for predicting types of forest cover from cartographic variables. It is transformed into an anomaly detection data set by keeping the smallest class (class 4 'Cottonwood/Willow') as anomalies against the largest class (class 2 'Lodgepole Pine'). This data set was used in Bay and Schwabacher (2003), Liu et al. (2012) and Ting et al. (2013).

- Following Shoemaker and Hall (2011) and Lazarevic and Kumar (2005), two data sets, *Probe* and *U2R*, are derived from KDD CUP 99 network intrusion data. *Probe* and *U2R* are with the original 41 mixed-type attributes, and they contain instances of probe and user-to-root attacks in the KDD CUP 99 data, respectively.

- *AnnThyroid* is a version of the series of Thyroid data, which is used to determine whether a patient referred to the clinic is hypothyroid. The first two classes are 'hyperfunction' and 'subnormal functioning', which are used as anomalies against the 'normal' class. This data set was previously used in (Liu et al., 2012) and Ting et al. (2013).

- *Arrhythmia* contains 16 classes. Following Liu et al. (2012) and He, Xu, Huang and Deng (2005), the eight smallest classes 3, 4, 5, 7, 8, 9, 14 and 15 are regarded as

anomalies against the rest of the classes. This data set was also used in Noto et al. (2010).

- *Nursery* is used to rank applications for nursery school. Class 'very_recom' is used as anomalies versus the class 'not_recom'. This data set was used in Noto et al. (2010).

- The data set *Chess* is converted to an anomaly detection data set by using the smallest class ('zero') as anomalies against the largest class ('fourteen'). This data set was used in Noto et al. (2010).

- *Mushroom*, which contains 23 edible or poisonous mushroom species described by 22 categorical attributes, is widely used in categorical data clustering and classification. To transform it into an anomaly detection data set, following He, Xu, Huang and Deng (2005), Koufakou and Georgiopoulos (2010) and Zimek, Gaudet, Campello and Sander (2013), we keep the large class unchanged and downsample the small class to create a rare class (consisting of 5% instances of the entire data set), evaluating the rare class as anomalies versus the large class. This data set was used in Noto et al. (2010) and Koufakou et al. (2007).

- *SolarFlare* contains 3 classes, i.e., three types of solar flare occurring in a 24 hour period. We focus on the flare class X, and use the occurrence of solar flare X as anomalies against non-occurrence normal instances. This data set was used in Tang et al. (2013).

- *Http* and *Smtp* are also taken from the KDD CUP 99 network intrusion data, but they are very different from *Probe* and *U2R*. *Http* and *Smtp* are created as follows: 4 attributes, including *service, duration, src bytes*, and *dst bytes*, out of an original 41 attributes are selected because they are regarded as the most basic attributes (Yamanishi et al., 2000), and the data is then divided into five subsets according to the five values in the *service* attribute, called *http*, *smtp*, *ftp*, *ftp_data*, and *others*. *Http* and *Smtp* are the two largest subsets. Therefore, *Http* and *Smtp* contain three numeric attributes *duration, src bytes*, and *dst bytes* only. These two data sets were used in Yamanishi et al. (2000), Liu et al. (2012) and Ting et al. (2013).

- *Shuttle* contains nine independent attributes and seven classes, of which class 1 accounts for about 80% of instances. Following Lazarevic and Kumar (2005) and Liu et al. (2012), classes 2, 3, 5, 6 and 7 are selected as anomalies against class 1.

- *Mammography* is used for detection of mammographic calcifications. Instances labelled as calcifications (2.32%) are regarded as anomalies against non-calcifications normal instances (97.68%). This data set was used in Woods et al. (1993), Lazarevic and Kumar (2005), Liu et al. (2012) and Ting et al. (2013).

- *HAR* is a Human Activity Recognition data set used in Anguita et al. (2012), which contains *walking*, *down-stair walking*, *up-stair walking*, *sitting*, *standing* and *laying* six activities. The smallest class *up-stair walking* is used as the anomaly class against three non-walking activity classes.

- *Satimage* is a landsat data set, which consists of 6,453 sub-areas of scenes. The scenes are labeled as *cotton crop* and five different soils. The *cotton crop* class is used as the anomaly class against all the soil classes. *Satimage* was used in Lazarevic and Kumar (2005).

- *Isolet* and *Mfeat* are taken from Pham and Pagh (2012). *Isolet* classifies instances based on the pronunciation of 26 letters of the alphabet while *Mfeat* (Multiple Features) consists of data of handwritten digits ('0' - '9'). For each data set, instances

from some classes having common behaviours are selected as normal instances, and 10 instances from another class as anomalies. For *Isolet*, instances of classes C, D, and E that share the 'e' sound as normal instances and 10 instances from class Y as anomalies. Instances of classes 6 and 9 in *Mfeat* are selected as normal instances because of the similarity of shapes, and 10 instances of class 0 as anomalies.

- *Mulcross* is taken from Liu et al. (2012), which is generated by the Mulcross data generator (Rocke and Woodruff, 1996). In *Mulcross*, normal instances are drawn from a multivariate normal distribution with an offset to create clustered anomalies. The data set used was generated using the following basic setting: two clustered anomalies centred a normal cluster, contamination ratio = 10% (percentage of anomalies), distance factor = 2 (distance between the centre of the normal cluster and anomaly clusters). This data contains 262,144 instances, so each anomaly cluster has more than 10,000 instances.

Appendix C

Sensitivity examination results

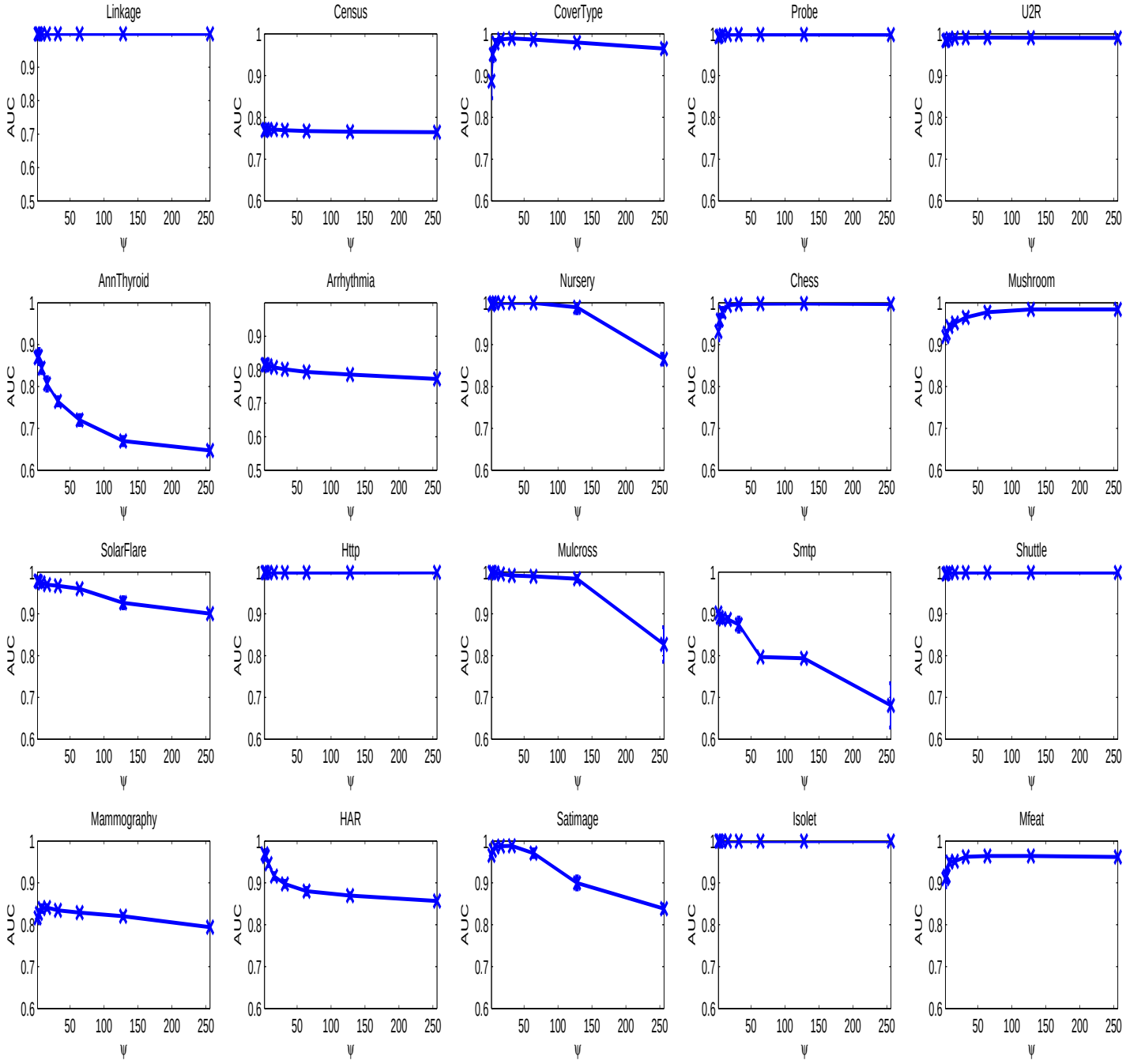

Figure C.1: Sensitivity test of ZERO++ with respect to ψ on all the 20 data sets.

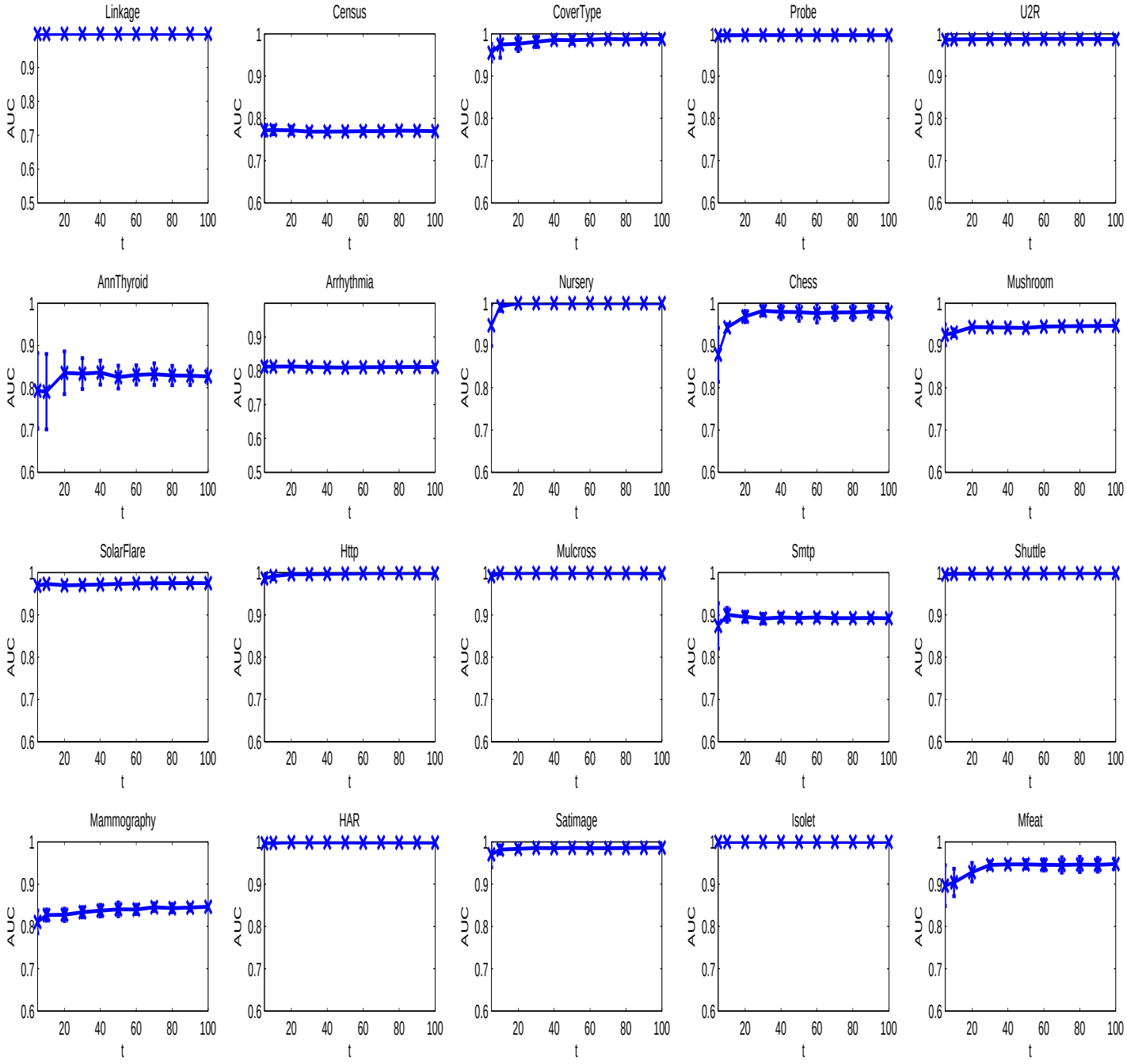

Figure C.2: Sensitivity test of ZERO++ with respect to t on all the 20 data sets.

References

Achtert, E., Kriegel, H., Schubert, E. and Zimek, A. (2013). Interactive data mining with 3d-parallel-coordinate-trees, *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2013, New York, NY, USA, June 22-27, 2013*, pp. 1009–1012.

Aggarwal, C. C. (2013a). *Outlier analysis*, Springer.

Aggarwal, C. C. (2013b). Outlier ensembles: Position paper, *ACM SIGKDD Explorations Newsletter* **14**(2): 49–58.

Agrawal, R., Imieliński, T. and Swami, A. (1993). Mining association rules between sets of items in large databases, *ACM SIGMOD Record* **22**(2): 207–216.

Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A. I. et al. (1996). Fast discovery of association rules, *Advances in knowledge discovery and data mining* **12**(1): 307–328.

Angiulli, F. and Fassetti, F. (2009). Dolphin: An efficient algorithm for mining distance-based outliers in very large datasets, *ACM Transactions on Knowledge Discovery from Data (TKDD)* **3**(1): 4.

Angiulli, F. and Pizzuti, C. (2002). Fast outlier detection in high dimensional spaces, *Principles of Data Mining and Knowledge Discovery*, Springer, pp. 15–27.

Anguita, D., Ghio, A., Oneto, L., Parra, X. and Reyes-Ortiz, J. L. (2012). Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine, *Ambient Assisted Living and Home Care*, Springer, pp. 216–223.

Asuncion, A. and Newman, D. (2007). UCI machine learning repository.

Barnett, V. and Lewis, T. (1994). *Outliers in statistical data*, Wiley New York.

Bay, S. D. and Schwabacher, M. (2003). Mining distance-based outliers in near linear time with randomization and a simple pruning rule, *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 29–38.

Beckmann, N., Kriegel, H.-P., Schneider, R. and Seeger, B. (1990). The R*-tree: An efficient and robust access method for points and rectangles, *International Conference on Management of Data*, ACM.

Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching, *Communications of the ACM* **18**(9): 509–517.

Beyer, K., Goldstein, J., Ramakrishnan, R. and Shaft, U. (1999). When is nearest neighbor meaningful?, *Database TheoryICDT99*, Springer, pp. 217–235.

Boriah, S., Chandola, V. and Kumar, V. (2008). Similarity measures for categorical data: A comparative evaluation, *Proceedings of the SIAM International Conference on Data Mining, SDM 2008*, SIAM, pp. 243–254.

Breunig, M. M., Kriegel, H.-P., Ng, R. T. and Sander, J. (2000). LOF: Identifying density-based local outliers, *ACM Sigmod Record* **29**(2): 93–104.

Chandola, V., Banerjee, A. and Kumar, V. (2009). Anomaly detection: A survey, *ACM Computing Surveys (CSUR)* **41**(3): 15.

Das, K., Schneider, J. and Neill, D. B. (2008). Anomaly pattern detection in categorical datasets, *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 169–176.

Dietterich, T. G. (2000). Ensemble methods in machine learning, *Multiple classifier systems*, Springer, pp. 1–15.

Donald, E. K. (1999). The art of computer programming, *Sorting and searching* **3**: 426–458.

Duch, W., Adamczak, R. and Grabczewski, K. (1996). Extraction of logical rules from training data using backpropagation networks, *The 1st Online Workshop on Soft Computing*, pp. 19–30.

Etemadi, N. (1981). An elementary proof of the strong law of large numbers, *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete* **55**(1): 119–122.

Ghoting, A., Otey, M. E. and Parthasarathy, S. (2004). LOADED: Link-based outlier and anomaly detection in evolving data sets, *IEEE International Conference on Data Mining*, IEEE, pp. 387–390.

Görnitz, N., Kloft, M. M., Rieck, K. and Brefeld, U. (2014). Toward supervised anomaly detection, *arXiv preprint arXiv:1401.6424* .

Guvenir, H. A., Acar, S., Demiroz, G. and Cekin, A. (1997). A supervised machine learning algorithm for arrhythmia analysis, *Computers in Cardiology 1997*, IEEE, pp. 433–436.

Hadi, A. (1992). Identifying multiple outliers in multivariate data, *Journal of the Royal Statistical Society. Series B. Methodological* **54**(3): 761–771.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I. H. (2009). The weka data mining software: An update, *ACM SIGKDD explorations newsletter* **11**(1): 10–18.

Hand, D. J. and Till, R. J. (2001). A simple generalisation of the area under the ROC curve for multiple class classification problems, *Machine Learning* **45**(2): 171–186.

Hawkins, D. M. (1980). *Identification of outliers*, Springer.

Hawkins, S., He, H., Williams, G. and Baxter, R. (2002). Outlier detection using replicator neural networks, *Data warehousing and knowledge discovery*, Springer, pp. 170–180.

He, Z., Deng, S. and Xu, X. (2005). An optimization model for outlier detection in categorical data, *Advances in Intelligent Computing*, Springer, pp. 400–409.

He, Z., Xu, X. and Deng, S. (2003). Discovering cluster-based local outliers, *Pattern Recognition Letters* **24**(9): 1641–1650.

He, Z., Xu, X., Huang, Z. J. and Deng, S. (2005). FP-outlier: Frequent pattern based outlier detection, *Computer Science and Information Systems/ComSIS* **2**(1): 103–118.

Houle, M. E., Kriegel, H.-P., Kröger, P., Schubert, E. and Zimek, A. (2010). Can shared-neighbor distances defeat the curse of dimensionality, *Scientific and Statistical Database Management*, Springer, pp. 482–500.

Huang, Z. (1997). Clustering large data sets with mixed numeric and categorical values, *Proceedings of the 1st Pacific-Asia Conference on Knowledge Discovery and Data Mining,(PAKDD)*, Singapore, pp. 21–34.

Jarvis, R. A. (1973). On the identification of the convex hull of a finite set of points in the plane, *Information Processing Letters* **2**(1): 18–21.

Jiang, M.-F., Tseng, S.-S. and Su, C.-M. (2001). Two-phase clustering process for outliers detection, *Pattern recognition letters* **22**(6): 691–700.

Jiang, S., Song, X., Wang, H., Han, J. and Li, Q. (2006). A clustering-based method for unsupervised intrusion detections, *Pattern Recognition Letters* **27**(7): 802–810.

Johnson, T., Kwok, I. and Ng, R. T. (1998). Fast computation of 2-dimensional depth contours, *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*, pp. 224–228.

Keller, F., Muller, E. and Bohm, K. (2012). HiCS: High contrast subspaces for density-based outlier ranking, *IEEE International Conference on Data Engineering (ICDE)*, IEEE, pp. 1037–1048.

Knorr, E. M. and Ng, R. T. (1997). A unified notion of outliers: Properties and computation, *Proceedings of the 3rd ACM International Conference on Knowledge Discovery and Data Mining*, pp. 219–222.

Knox, E. M. and Ng, R. T. (1998). Algorithms for mining distance based outliers in large datasets, *Proceedings of the International Conference on Very Large Data Bases*, Citeseer, pp. 392–403.

Kontkanen, P. and Myllymäki, P. (2007). MDL histogram density estimation, *International Conference on Artificial Intelligence and Statistics*, pp. 219–226.

Koufakou, A. and Georgiopoulos, M. (2010). A fast outlier detection strategy for distributed high-dimensional data sets with mixed attributes, *Data Mining and Knowledge Discovery* **20**(2): 259–289.

Koufakou, A., Ortiz, E. G., Georgiopoulos, M., Anagnostopoulos, G. C. and Reynolds, K. M. (2007). A scalable and efficient outlier detection strategy for categorical data, *19th IEEE International Conference on Tools with Artificial Intelligence*, IEEE, pp. 210–217.

Kriegel, H.-P., Kröger, P., Schubert, E. and Zimek, A. (2009). Outlier detection in axis-parallel subspaces of high dimensional data, *Advances in Knowledge Discovery and Data Mining*, Springer, pp. 831–838.

Kriegel, H.-P., Kröger, P. and Zimek, A. (2009). Outlier detection techniques, *Tutorial at the 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining*.

Kriegel, H.-P. and Zimek, A. (2008). Angle-based outlier detection in high-dimensional data, *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 444–452.

Lazarevic, A. and Kumar, V. (2005). Feature bagging for outlier detection, *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, ACM, pp. 157–166.

Lin, D. (1998). An information-theoretic definition of similarity, *In Proceedings of the 15th International Conference on Machine Learning*, pp. 296–304.

Liu, F. T., Ting, K. M. and Zhou, Z.-H. (2010). On detecting clustered anomalies using sciforest, *Machine Learning and Knowledge Discovery in Databases*, Springer, pp. 274–290.

Liu, F. T., Ting, K. M. and Zhou, Z.-H. (2012). Isolation-based anomaly detection, *ACM Transactions on Knowledge Discovery from Data (TKDD)* **6**(1): 3.

Liu, H., Hussain, F., Tan, C. L. and Dash, M. (2002). Discretization: An enabling technique, *Data mining and knowledge discovery* **6**(4): 393–423.

Ma, J. and Perkins, S. (2003). Time-series novelty detection using one-class support vector machines, *Proceedings of the International Joint Conference on Neural Networks*, IEEE, pp. 1741–1745.

Maji, S. and Malik, J. (2009). Fast and accurate digit classification, *EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-159* .

Mukkamala, S., Janoski, G. and Sung, A. (2002). Intrusion detection using neural networks and support vector machines, *Proceedings of the 2002 International Joint Conference on Neural Networks*, IEEE, pp. 1702–1707.

Muller, E., Schiffer, M. and Seidl, T. (2011). Statistical selection of relevant subspace projections for outlier ranking, *2011 IEEE 27th International Conference on Data Engineering (ICDE)*, IEEE, pp. 434–445.

Ngufor, C. and Wojtusiak, J. (2013). Learning from large-scale distributed health data: an approximate logistic regression approach, *Proc. ICML 13: Role of Machine Learning in Transforming Healthcare* .

Nguyen, H. V., Ang, H. H. and Gopalkrishnan, V. (2010). Mining outliers with ensemble of heterogeneous detectors on random subspaces, *Database Systems for Advanced Applications*, Springer, pp. 368–383.

Noto, K., Brodley, C. and Slonim, D. (2010). Anomaly detection using an ensemble of feature models, *2010 IEEE 10th International Conference on Data Mining (ICDM)*, IEEE, pp. 953–958.

Papadimitriou, S., Kitagawa, H., Gibbons, P. B. and Faloutsos, C. (2003). LOCI: Fast outlier detection using the local correlation integral, *International Conference on Data Engineering*, IEEE, pp. 315–326.

Pham, N. and Pagh, R. (2012). A near-linear time approximation algorithm for angle-based outlier detection in high-dimensional data, *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 877–885.

Preparatat, F. P. and Shamos, M. I. (1985). Computational geometry: An introduction.

Ramaswamy, S., Rastogi, R. and Shim, K. (2000). Efficient algorithms for mining outliers from large data sets, *ACM SIGMOD Record* **29**(2): 427–438.

Rocke, D. M. and Woodruff, D. L. (1996). Identification of outliers in multivariate data, *Journal of the American Statistical Association* **91**(435): 1047–1061.

Shoemaker, L. and Hall, L. O. (2011). Anomaly detection using ensembles, *Multiple Classifier Systems*, Springer, pp. 6–15.

Sugiyama, M. and Borgwardt, K. (2013). Rapid distance-based outlier detection via sampling, *Advances in Neural Information Processing Systems* pp. 467–475.

Tan, P.-N., Steinbach, M. and Kumar, V. (2006). *Introduction to data mining*, Pearson Addison Wesley Boston.

Tan, P.-N., Steinbach, M., Kumar, V. et al. (2006). *Introduction to data mining*, Pearson Addison Wesley Boston.

Tang, G., Bailey, J., Pei, J. and Dong, G. (2013). Mining multidimensional contextual outliers from categorical relational data, *Proceedings of the 25th International Conference on Scientific and Statistical Database Management*, ACM.

Tang, J., Chen, Z., Fu, A. W.-C. and Cheung, D. W. (2002). Enhancing effectiveness of outlier detections for low density patterns, *Advances in Knowledge Discovery and Data Mining*, Springer, pp. 535–548.

Ting, K. M., Zhou, G.-T., Liu, F. T. and Tan, S. C. (2013). Mass estimation, *Machine learning* **90**(1): 127–160.

Tukey, J. W. (1977). *Exploratory data analysis*, Addison-Wesley, Massachusetts. US.

Webb, G. I., Boughton, J. R. and Wang, Z. (2005). Not so naive bayes: aggregating one-dependence estimators, *Machine learning* **58**(1): 5–24.

Woods, K. S., Doss, C. C., Bowyer, K. W., Solka, J. L., Priebe, C. E. and Kegelmeyer JR, W. P. (1993). Comparative evaluation of pattern recognition techniques for detection of microcalcifications in mammography, *International Journal of Pattern Recognition and Artificial Intelligence* **7**(06): 1417–1436.

Wu, M. and Jermaine, C. (2006). Outlier detection by sampling with accuracy guarantees, *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 767–772.

Wu, S. and Wang, S. (2013). Information-theoretic outlier detection for large-scale categorical data, *IEEE Transactions on Knowledge and Data Engineering* **25**(3): 589–602.

Yamanishi, K., Takeuchi, J.-I., Williams, G. and Milne, P. (2000). On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms, *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 320–324.

Zhang, K. and Jin, H. (2011). An effective pattern based outlier detection approach for mixed attribute data, *Advances in Artificial Intelligence*, Springer, pp. 122–131.

Zimek, A., Campello, R. J. and Sander, J. (2013). Ensembles for unsupervised outlier detection: Challenges and research questions, *ACM SIGKDD Explorations Newsletter* **15**(1): 11–22.

Zimek, A., Gaudet, M., Campello, R. J. and Sander, J. (2013). Subsampling for efficient and effective unsupervised outlier detection ensembles, *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 428–436.

Zimek, A., Schubert, E. and Kriegel, H.-P. (2012). A survey on unsupervised outlier detection in high-dimensional numerical data, *Statistical Analysis and Data Mining* **5**(5): 363–387.