NEW METHODS IN OUTLIER DETECTION

by

Guanting Tang

M.Sc., Georgia Southwestern State University, 2008

B.Sc. Shanghai Ocean University, 2007

Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of

Doctor of Philosophy

in the
School of Computing Science
Faculty of Applied Sciences

SIMON FRASER UNIVERSITY
Spring 2015

APPROVAL

Name:	Guanting Tang
Degree:	Doctor of Philosophy
Title of Thesis:	NEW METHODS IN OUTLIER DETECTION
Examining Committee:	Dr. Ramesh Krishnamurti Chair
	Dr. Jian Pei, Senior Supervisor, Professor, Computing Science, SFU
	Dr. Wo-Shun Luk, Supervisor, Professor, Computing Science, SFU
	Dr. Yupin Yang, Supervisor, Assistant Professor, Beedie Business School, SFU
	Dr. Jiangchuan Liu, Internal Examiner, Professor, Computing Science, SFU
	Dr. Alex Thomo, External Examiner, Associate Professor, Computer Science, University of Victoria
Date Approved:	April 22nd, 2015

Partial Copyright Licence

Abstract

Outlier detection has been studied extensively in data mining. However, as the emergence of huge data sets in real-life applications nowadays, outlier detection faces a series of new challenges. Many traditional outlier detection techniques do not work well in such an environment. Therefore, developing up-to-date outlier detection methods becomes urgent tasks.

In this thesis, we propose several new methods for detecting different kinds of outliers in high-dimensional data sets from two different perspectives, namely, detecting the outlying aspects of a data object and detecting outlying data objects of a data set. Specifically, for detecting the outlying aspects of a data object, we propose the problems of mining outlying aspects and mining contrast subspaces; for detecting outlying data objects of a data set, we propose the problems of mining contextual outliers and mining Markov blanket based outliers. We develop efficient and scalable algorithms to tackle the computational challenges. We also conduct comprehensive empirical studies on real and synthetic data sets to verify the effectiveness of the proposed outlier detection techniques and the efficiency of our algorithms.

Keywords: outlier detection; outlying aspects detection; outlier interpretation

To my family

"Everyone is special in their own way."

— *The Perks of Being a Wallflower*, Stephen Chbosky, *1999*

Acknowledgments

I wish to express my deepest gratitude to my senior supervisor and mentor, Dr. Jian Pei. I thank him for his continuous encouragement, confidence and support during my Ph.D study. He shared with me not only valuable knowledge and research skills but also the wisdom of life. Dr. Pei's guidance enables me to hurdle all the obstacles in the completion of this research work. I believe what he has trained me in graduate school will benefit my career all the time.

My gratitude goes to my supervisors, Dr. Wo-Shun Luk and Dr. Yupin Yang, for providing insightful comments and helpful suggestions that helped me to improve the quality of my research. Their visionary thoughts and energetic working style have inuenced me greatly. Part of this work is done in collaboration with Dr. James Bailey, Dr. Guozhu Dong and Dr. Lei Duan. I thank them for the knowledge and skills they imparted through the collaboration.

I am very thankful to Dr. Jiangchuang Liu and Dr. Alex Thomo for serving in my thesis examining committee. I would also like to thank Dr. Ramesh Krishnamurti to chair my thesis defense.

I would also like to thank many people in our department, support staff and faculty, for always being helpful over the years. I thank my friends at Simon Fraser University for their accompany and help. A particular acknowledgement goes to Crystal (Zhengzheng) Xing, Carrie (Cong) Wang, Ming Hua, Bin Jiang, Bin Zhou, Qiang Jiang, Yi Cui, Hossein Maserrat, Luping Li, Zhenhua Lin, Da Huang, Yi Han, Xiao Meng, Xiangbo Mao, Guangtong Zhou, Juhua Hu, Jiaxing Liang, Chuangcong Gao, Xiaoning Xu, Yu Tao, Ruiwen Chen, Yu Yang, Lin Liu, Berer Lu, Ye Wang, Lumin Zhang and Li Xiong. They have made my graduate school enjoyable and memorable.

Last but not least, I am very grateful to my families who always support, encourage, and tolerate me without limits. Their love accompanies me wherever I go.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

An outlier is a data object that is signicantly different from the rest of the data objects in a data set. Hawkins [53] provided a formal definition as following:
"An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism."

Outlier detection is the process of identifying data objects, which are drastically different from the rest of the data set. It is an important data mining task with broad applications, such as credit card fraud detection [18, 19], insurance claim fraud detection [109, 124], medical diagnosis [28, 97], image processing [30, 94], intrusion detection [73, 103], and event detection [26, 59].

Outlier detection has been studied extensively in the data mining research community. However, as the emergence of huge data sets in real-life practice nowadays, outlier detection faces a series of new challenges. Many traditional outlier detection methods do not work well in such an environment. Therefore, developing up-to-date outlier detection methods becomes urgent tasks. In this thesis, we aim at developing new outlier detection methods, which try to tackle some of the new challenges, specifically, *dimensionality* and *understandability*.

- **Dimensionality.** How to find meaningful subspaces for outlier detection? Real-life data sets often have many dimensions, which may lead to a huge number of subspaces. Detecting outliers in every subspace is ineffective and inefficient mainly because of three reasons. First, in high-dimensional subspaces, data becomes sparse; and the similarities among data objects in full space become indistinguishable. The

true outliers are often covered up by the noises generated by the high dimensionality. Second, in order to assist business users to effectively analyze outliers, it is helpful to avoid reporting outliers from meaningless subspaces. Third, brute force methods, such as testing all data objects in all possible subspaces, are not feasible when a data set has a huge number of subspaces.

- **Understandability.** Understanding why the detected objects are outliers is important to business users. Outliers usually conveying useful information regarding different properties of the underlying data sets. Such information can help business users to propose suitable action plans for the outliers. In order to facilitate the understandability of outliers, an outlier detection method needs to provide justifications of the detection, such as the contexts of the outliers.

1.1 Motivation

Our study propose four different outlier detection methods from two different perspectives, detecting the outlying aspects of a data object and detecting outlying data objects of a data set. In some application scenarios, using the appropriate outlier detection perspective can provide useful information.

Example 1.1. (Mining outlying aspects and outliers in healthcare applications). Despite the tremendous improvements in healthcare practice and systems nowadays, having sustainable healthcare is still one of the most important and urgent issues. From the individual point of view, healthcare is important to every one, even for those who are currently in good health. No one wants to get sick, but every one will get sick at some point of his life. From the global point of view, healthcare has a notable impact on a country's economy. According to the well-known Canadian Institute for Health Information, the total spending on healthcare in Canada in 2013 was projected to exceed \$211 billion, or $\$5,988$ per person.

Healthcare has many outlier detection applications. In this example, we consider four different application scenarios from two different perspectives, identifying outlying aspects of individuals (Scenario 1 and Scenario 2) and detecting outliers under different circumstances (Scenario 3 and Scenario 4).

Scenario 1. Identifying Most Important Factors in Medical Treatment. Assume you are a medical doctor having many patients diagnosed with cardiovascular diseases. When facing a particular patient, you may want to know the most outlying (or dangerous) aspects of the patient, though the patient may not be top ranked on those aspects or on any other among all patients. After some analysis, you notice that the patient's glycemic aspect, which includes triglyceride ("TG") value: 420 mg/dL and low-density lipoprotein ("LDL") value: 185 mg/dL, is outlying most. Reducing the risk factor values in glycemic aspect is very important to this patient. Based on the previous analysis, you probably make a treatment plan focusing on improving the patient's current glycemic status, which includes medicines such as Statins and Fibrates, and suggestions of reducing the amount of "bad" fat in his diet and exercising more. ■

The scenario illustrated in the above example is different from traditional outlier detection. Specifically, instead of searching for outliers from a data set, here we are given a query object and want to find the outlying aspects whereby the object is most unusual. The query object itself may or may not be an outlier in the full space or in any specific subspaces. In this problem, we are not interested in other individual outliers or inliers. The outlying aspect finding questions cannot be answered by the existing outlier detection methods directly.

In Chapter 3, we will explore the novel and interesting problem of finding outlying aspects of a query object on multidimensional numeric data. We systematically developed a model and a heuristic method.

Scenario 2. Reduce Misdiagnosis by Providing Useful Information. The occurrence of medical errors, such as diagnostic error, is a serious problem. Tehrani et al. [100] pointed out that about 28.6% of the paid malpractice medical claims in US had been misdiagnosed.

Imagine you are a medical doctor facing a patient having symptoms of being overweight, short of breath, and some others. You want to check the patient on two specic possible diseases: coronary artery disease and adiposity. Please note that clogged arteries are among the top-5 most commonly misdiagnosed diseases [29]. You have a set of reference samples of both diseases. Then, you may naturally ask "In what aspect is this patient most similar to cases of coronary artery disease and, at the same time, dissimilar to adiposity?" ■

The above scenario cannot be addressed well using an existing data mining method, and thus suggests a novel data mining problem. In a multidimensional data set of two classes,

given a query object and a target class, we want to find the subspace where the query object is most likely to belong to the target class against the other class. We call such a subspace a *contrast subspace* since it contrasts the likelihood of the query object in the target class against the other class.

In Chapter 4, we will study the interesting problem of mining contrast subspaces to discover the aspects that a query object most similar to a class and dissimilar to the other class. A heuristic method with embedded pruning techniques is developed.

Scenario 3. Detect Fraud Suspects in Health Insurance. Fraud detection is an important task for business analysts in health insurance industry. According to the Canadian Health Care Anti-Fraud Association, about 2% to 10% of Canadian's healthcare spending is lost to fraud every year [42].

Suppose you are a business analyst in the auditing department of a health insurance company. Your job is to identify frauds. You may not only want to find out all fraud suspects, but also the insightful reasons about why these people are suspicious. Particularly, for a small group of fraud suspects, you may want to know in what aspects they share similarity with the majority of normal patients; and in what aspects they deviate dramatically from the normal patients.

A concrete example is: "Among the patients who consume narcotic drugs in the Greater Vancouver area, a small group of 10 patients purchasing narcotic drugs from more than 60 different pharmacies is an outlier group, comparing to a reference group of 3000 patients buying narcotic drugs from fewer than 5 different pharmacies." The 10 patients in the outlier group have a high probability to submit fraud claims to the insurance company. Since among the patients who take narcotic drugs in the Greater Vancouver area, this small group of people have a very different purchase pattern from the majority. ■

In Chapter 5, we will we develop a notion of contextual outliers to solve the previous scenario. We systematically develop a model, including a concise representation, for contextual outlier analysis, and devise a detection algorithm that leverages the state-of-the-art data cube computation techniques.

Scenario 4. Discover Diseases and Related Factors in Medical Diagnosis. Assume you are a medical doctor having a set of patients' medical records. Each record contains many different features of a patient, such as the patient's age, weight, skin humidity, blood type, work place, lab test results, symptoms, and diagnosis history.

In order to find out patients who have serious illness and the possible causes of the illness, you may want to identify some patients who are very different from the others in a certain feature and its causal features. For example, a leukemia patient is usually very different from the others in features related to blood, such as complete blood count, blood chemistry tests, and bleeding and clotting factors. Other features such as "age" and "skin humidity" may be irrelevant for detecting this type of outlier, but are relevant for discovering the abnormal patients with the "dehydration" status. ■

In Chapter 6, we will explore the problem of detecting Markov blanket based outliers. Particularly, we develop a model to first select meaningful subspaces using Markov blanket. And then, the model will detect, characterize, and summarize outliers in the selected subspaces.

Example 1.1 demonstrates the great needs of detecting meaningful outlying aspects and outliers in practice. Traditional outlier detection techniques do not work well here. It is not only because of infamous "curse of dimensionality", which reveals that the distances between all pairs of data objects are similar in high-dimensional spaces [15], but also because of the lack of contexts for the detected outliers. In real-life applications, a business analyst may want to see not only the outliers detected, but also the possible contexts about the outliers.

Besides the application scenarios mentioned in Example 1.1, detecting outlying aspects and outliers have many other applications, such as intrusion detection and image processing.

1.2 Challenges and Contributions

While being useful in many real-life applications, detecting meaningful outlying aspects and outliers pose special challenges.

- *How to model the outlying aspects of a data object?* Scenario 1 and Scenario 2 in Example 1.1 illustrate two different applications in mining outlying aspects of a data object. They not only show the great practical value of mining outlying aspects, but also raises a fundamental question: how can we formulate the outlying aspects of a data object that capture the most outstanding characteristics of the data object and suit application needs?

  In particular, we need to consider the following two aspects. First, how to quantify

the outlyingness of an object in a subspace? Second, how to compare the outlyingness of an object in different subspaces?

- *How to model meaningful outliers in multidimensional data sets?* As shown in Scenario 3 and Scenario 4 of Example 1.1, context is an important component in outlier detection. Outliers usually convey useful information regarding different properties of the underlying data set. Proper contextual information of the outliers can help a business analyst to better understand and investigate individual outliers and propose action plans suitable for such outliers.

  Therefore, we need to tackle two problems in developing meaningful outliers in multidimensional data sets. First, how to model outliers according to different application interests? Second, how to find contexts best manifesting the outlyingness of the detected outliers?

- *How to explore the exponential number of subspaces in a multidimensional data set efficiently?* Given a data set with n attributes, there are $2^n - 1$ non-empty subspaces. A brute force method, which examines subspaces one by one, incurs heavy computational cost when the dimensionality is high. For example, on a data set of 100 dimensions, $2^{100} - 1 = 1.27 \times 10^{30}$ subspaces have to be examined, which is unfortunately computationally prohibitive using the state-of-the-art technology.

  Thus, it is important to develop effective techniques, which can efficiently process the exponential number of subspaces of a multidimensional data set.

In this thesis, we propose several new methods for detecting different kinds of outliers in multidimensional data sets. In particular, we make the following contributions.

- We formulate the problem of detecting outlying aspects of a data object in both unsupervised data sets and supervised data sets. And we develop models to address the two application scenarios in Example 1.1, respectively.

  In particular, for unsupervised data sets, we use the rank of the probability density of an object in a subspace to measure the outlyingness of the object in the subspace. For supervised data sets, we use the ratio of the likelihood of a query object in the target class against that in the other class as the outlyingness measure, which is essentially the Bayes factor [64] on the query object.

- For detecting meaningful outliers in multidimensional data sets, we develop two outlier detection methods to tackle the application Scenario 3 and Scenario 4 of Example 1.1. Specifically, we propose the problems of mining contextual outlier and mining Markov blanket based outlier.

  We argue that the contextual information of an outlier is an important component in the outlier detection process. Thus, we integrate the task of identifying proper context information for outliers in both of the proposed methods.

- We develop a series of heuristic techniques to improve the efficiency of all methods proposed in this thesis. These techniques can be categorized into two categories: subspace pruning techniques and object pruning techniques.

  Particularly, subspace pruning techniques are developed for the purpose of processing as fewer candidate subspaces as possible; while object pruning techniques are developed for the purpose of calculating as fewer candidate data objects in a subspace as possible.

Table 1.1 summarizes the four new outlier detection problems, their challenges, and our contributions presented in this thesis.

Outlying level	Outlying aspects	Outliers
General problem	Given a data set and a query, find top-k subspaces that make the query most outlying	Given a data set, find all possible outlier data objects (or outlier data object groups) and their corresponding contexts
Data type	Numeric data	Categorical data
What to find	Outlying subspace / Contrast subspace	Contextual outlier / Markov blanket based outlier
Application scenarios	Scenario 1 / Scenario 2	Scenario 3 / Scenario 4
Major challenges	Measuring and comparing the outlyingness of an object in different subspaces / Measuring the contrast of the similarity between the query object and the target class and the difference between the query object and the other class	Modeling meaningful contexts for outliers / Selecting meaningful subspaces for outlier detection, assessing and characterizing outliers in different subspaces
Major contributions	• Using rank statistics of the probability density of an object to measure and compare the outlyingness of an object in different subspaces. • Developing efficient detection algorithm. / • Using the ratio of the likelihood of the query object in the target class against that in the other class as the measure. • Proving MAX SNP-hardness of the problem. • Developing efficient detection algorithm.	• Developing a notion of multidimensional contextual outliers to model the contexts of an outlier. • Providing analysis to reduce redundancy among outliers. • Designing a simple, yet effective algorithm. / • Using Markov blanket based method to select meaningful subspaces for outlier detection. • Designing an outlier score based on the joint probabilities of local Bayesian networks learned from the selected subspaces. • Characterizing outliers by using the subspaces that maximize the outlier score.
Chapter	Chapters 3 / Chapters 4	Chapters 5 / Chapters 6

Table 1.1: A summary of problems, challenges, and contributions.

1.3 Organization of the Thesis

The rest of the thesis is organized as follows.

- In Chapter 2, we review the related work and explain how they are related to and different from this thesis.

- In Chapter 3, we propose a novel problem of mining outlying aspects on numeric data, which finds the subspaces best manifesting the unusualness of a specified query object, using the other objects as the background in comparing different subspaces. This problem is related to, but critically different from traditional outlier detection. Specifically, instead of searching for outliers from a data set, here we are given a query object and want to find the outlying aspects whereby the object is most unusual. The query object itself may or may not be an outlier in the full space or in any specific subspaces.

- In Chapter 4, we study the problem of mining contrast subspaces on numeric data. In a multidimensional data set of two classes, given a query object and a target class, we want to find the subspace where the query object is most likely to belong to the target class against the other class. While there are many existing studies on outlier detection and contrast mining, they focus on collective patterns that are shared by many cases of the target class. The contrast subspace mining problem addressed here is different. It focuses on one query object and finds the customized contrast subspace. This problem is a supervised version of the problem we proposed in Chapter 3.

- In Chapter 5, we propose the framework of detecting contextual outliers on categorical data. A notion of contextual outliers is introduced to model the context of an outlier. Intuitively, a contextual outlier is a small group of objects that share strong similarity with a signicantly larger reference group of objects on some attributes, but deviate dramatically on some other attributes. We presented a detection algorithm leveraging the state-of-the-art data cube computation techniques.

- In Chapter 6, we propose the frame work of detecting Markov blanket based outliers on categorical data. Given a data set, we regard the Markov blanket of each dimension and the dimension itself as a Markov blanket subspace. Markov blanket based outliers are outlying data objects, which are detected from the Markov blanket subspaces. The

experimental results using synthetic and real-world data validate the effectiveness, efficiency and scalability of the proposed model.

- In Chapter 7, we conclude the thesis and discuss the future works.

Chapter 2

Related Work

Outlier analysis is a well studied subject in data mining. Several recent surveys on the topic [1, 4, 24, 57, 78, 79, 92, 125, 126] as well as dedicated chapters in classical data mining textbooks [50] provide thorough treatments.

In this chapter, we review the existing studies related to this thesis. First, we review some of the state-of-art methods on traditional outlier detection. Then, we review the recent studies on subspace outlier detection. Last, we discuss the related work on each of the proposed problems in this thesis.

2.1 Traditional Outlier Detection Method

Given a set of data objects, traditional outlier detection focuses on finding outlier objects that are significantly different from the rest of the data set in the full attribute space.

A number of outlier detection methods have been proposed. We categorize them into three categories according to how outliers are defined, namely, statistical methods, proximity-based methods, and clustering-based methods.

2.1.1 Statistical methods

Statistical methods [38, 41, 61, 104, 123] assume that data objects in a given data set are generated by a certain statistical model. Outliers are data objects that significantly deviate from the statistical model. Statistical methods usually contain two steps. First, assume (or train) a distribution (or a probabilistic) model based on the given data set. Then, test

if a data object is significantly deviating from the assumed (or trained) model. If yes, the tested data object is an outlier. Based on whether the assumption of data distribution is made or not, statistical methods can be categorized into two categories, namely, parametric method and non-parametric method.

Parametric methods [45, 104, 106, 123] assume data is generated by certain distribution (or probabilistic) models. For example, in early 1930s, Shewhart [104] proposed to apply a simple outlier detection technique to the quality control process. Particularly, he first assumed that all data objects were following the Gaussian distribution. Then, any data object that is more than 3σ distance away from the mean μ is an outlier, where σ is the standard deviation of the Gaussian distribution.

Later, mixture probabilistic models [45, 123] are used to deal with more complex forms of data distribution. For instance, Yang *et al.* [123] proposed to use the Expectation-Maximization (EM) algorithm [33] to fit a Gaussian Mixture Model (GMM) with Gaussians centered at each data object to a given data set. Each mixture proportion represents the degree of a data object being a cluster center. The outlier score of each data object is defined as a weighted sum of the mixture proportions with weights representing the similarities to other data objects. The smaller the outlier score of a data object, the more likely it is an outlier.

Non-parametric methods [56, 61, 88] do not assume any priori data distribution (or probabilistic) models. For example, Javitz and Valdes [61] proposed a histograms based method for intrusion detection. Histograms were constructed to represent the profile of normal behaviors. A testing behavior is a potential intrusion if it does not fall into any of the bins in the constructed histograms.

Besides the histograms technique, kernel function is another popular technique used under this theme. For instance, Palpanas *et al.* [88] proposed a framework to detect outliers in a sensor network. In essence, they used kernel density estimator [102] to estimate the expected distribution of normal values generated by a sensor at a certain time. Epanechnikov kernel was adopted. Given a time t, an observed value is outlying if it does not fall into a certain range of the values in the expected distribution. A sensor is an outlier if it generated more than n such outlying values within a certain time period.

The effectiveness of statistical methods heavily rely on whether the given data set fits the chosen statistical model or not.

2.1.2 Proximity-based methods

Proximity-based methods [20, 27, 66, 96, 108] assume that outliers are data objects isolated from the rest of the data set. In other words, the proximity of an outlier object to its neighbors is very different from that of a normal object. Based on the choice of proximity measurements, proximity-based methods can be further divided into two categories, namely, density-based methods and distance-based methods.

Density-based methods [20, 27, 108] assume that the density or relative density around an outlier object is substantially different from that around its neighbors. For instance, Breunig *et al.* [20] proposed the famous Local Outlier Factor (LOF) algorithm. The basic idea of LOF is to use the relative density of a data object against its neighbors as the indicator of the outlyingness degree of the data object. Specifically, given a data object p and the number of data objects in the local neighborhood of a data object $MinPts$, the reachability distance of p with respect to data object o, denoted by $reach\text{-}dist_{MinPts(p,o)}$, is the maximum value between the distance from p to its farthest neighbor in p's $MinPts$-neighborhood and the distance from p to o. The local reachability density of p is defined as

$$lrd_{MinPts}(p) = 1/\Big(\frac{\sum_{o \in MinPts(p)} reach\text{-}dist_{MinPts(p,o)}}{|N_{MinPts}(p)|}\Big),$$

where $|N_{MinPts}(p)|$ is the number of data objects falling into p's $MinPts$-neighborhood. And the LOF of p in a given data set is defined as

$$LOF_{MinPts}(p) = \frac{\sum_{o \in MinPts(p)} \frac{lrd_{MinPts(o)}}{lrd_{MinPts(p)}}}{|N_{MinPts}(p)|}.$$

A data object is likely to be an outlier if it has a LOF score that is significantly greater than 1.

Subsequently, many density-based outlier detection techniques have been developed based on the idea of LOF. For instance, Chiu and Fu [27] proposed a method named *GridLOF*. *GridLOF* uses a simple grid-based technique to prune non-outlier objects, such that the computation of LOF for all data objects can be avoided. Another interesting variant of LOF is the connectivity-based outlier factor (COF) proposed by Tang *et al.* [108], which aims to improve the effectiveness of LOF when a potential outlier has similar local neighborhood density as that of its neighbors.

Distance-based methods [9, 14, 46, 66, 96, 120] assume that data objects which are far away from their neighbors have high probabilities to be outliers. Knorr and Ng [66] firstly

introduced the notion of distance-based outlier [1]. Given a radius ε and a percentage p, a data object o is a outlier if at least $p\%$ of all other data objects have a distance to o greater than ε.

Later, this distance-based outlier notion [66] has been extended based on the distance of a data object to its kth nearest neighbor, which is known as the nearest neighbour (k-NN) method. For example, given a data set, Ramaswamy *et al.* [96] considered data objects with the top-k highest D^ks as outliers. The D^k of a data object o is the distance between o and its k^{th} nearest neighbor. The distance between two data objects can be measured by different distance metrics, such as manhattan distance, euclidean distance, and so on. In this paper, the authors use the square of the euclidean distance instead of the euclidean distance itself, for the sake of reducing computational costs. Angiulli and Pizzuti [9] provided another way of defining the k-NN distance based outliers. Specifically, the outlier score of a data object o is defined as the sum of the distances from o to all its k nearest neighbors. The outliers are data objects with top-k highest outliers scores. Many methods [14, 46, 120] have been developed to improve the efficiency of the distance-based outlier detection methods.

The effectiveness of proximity-based methods highly depend on the proximity measure adopted.

2.1.3 Clustering-based Methods

Clustering-based methods identify three different cases of outliers: (1) outliers are data objects that do not belong to any cluster; (2) outliers are data objects which are far away from their closest cluster centroid; (3) outliers are data objects that belong to very small and distant clusters.

Methods for detecting data objects that do not belong to any cluster. Clustering algorithms [39, 37, 48] that do not require every data object belonging to at least one cluster can be naturally applied here. Take the well known *DBSCAN* algorithm proposed by Ester *et al.* [39] as an example. Given a neighborhood radius ε and the minimum number of data objects contained in the neighborhood $MinPts$, *DBSCAN* check the ε-neighborhood for each data object. If a data object o has more than $MinPts$ objects in its ε-neighborhood, a new cluster is formed, and o is the core of this cluster. All data object in o's ε-neighborhood are added to this cluster, as well as their own ε-neighborhood only if the ε-neighborhood also have more than $MinPts$ objects. The cluster grows when no more core object can be found. Then, a new unvisited data object will be chosen to

repeat the previous process. The whole clustering will stop when no more new object can be added into any cluster. As a result, data objects that do not belong to any cluster can be considered as outliers.

Methods to tackle the second and third cases often have two phases, namely, the clustering phase and the testing phase. In the clustering phase, clustering algorithms such as k-means [76] or *DBSCAN* [39], are applied to identify clusters or dense regions. The testing phase computes the outlyingness scores for data objects. Assume o is an arbitrary data object in a given data set. People usually consider the following three aspects when designing the outlyingness score: (1) the fitness of o for the clusters identified in clustering phase; (2) the sizes of the clusters that o belongs to; (3) the distances between the clusters that o belongs to and the other clusters.

Methods for detecting data objects which are far away from their closest cluster centroid. Many methods [12, 71, 95] designed for intrusion detection are under this assumption. Barbará *et al.* [12] proposed a bootstrap outlier detection method by considering the similarity between data points and the clusters in a training data set. The authors rst separated normal data from abnormal data in the training data set by frequent itemsets based method. Then, normal data in the training data set were clustered into groups. Last, testing whether a data point is an outlier with respect to the clusters from the previous step.

Methods for detecting data objects that belong to very small and distant clusters. Various methods [54, 63, 87] proposed under this theme. For example, Jiang *et al.* [63] proposed a two-step clustering process to detect outliers belonging to very small and distant clusters. In the first step, a modified k-means algorithm is applied to find clusters. This algorithm uses a heuristic that a new data object is the center of a new cluster if it is far away from the centers of all existing clusters. The modified k-means algorithm produces clusters in a coarse way by allowing the number of clusters to be greater than k. Note that k is the number of cluster set by the user. In the second step, each cluster is represented by its center node; and a minimum spanning tree (MST) is constructed based on the distance between every two nodes. The longest edge of the MST is removed; and the MST is replaced by the two newly obtained subtrees. We can repeat the process of removing the longest edges of the subtrees until k subtrees have been obtained. Outliers are data objects that belong to the clusters in the smallest subtree.

The effectiveness of clustering-based methods highly depends on whether the used

clustering algorithms can capture the true clustering structure of normal objects or not.

2.2 Subspace Outlier Detection Methods

Data sets from real life often have very high dimensionalities. Due to the curse of dimensionality, measurements designed to calculate the differences between an object and the other objects, such as distance and probability density, become meaningless in the full space [16]. Thus, many conventional outlier detection techniques may not work well in such situation. Identifying outliers in various subspaces becomes a possible approach to solve practical outlier detection problems like this.

Given a set of high dimensional data objects, subspace outlier detection techniques aim to identify outlier objects, which are drastically deviating from the majority in some selected subspaces.

Most studies in subspace outlier detection have two major components, subspace selection and outlyingness measurement design. Many subspace selection methods have been proposed based on the different assumptions of meaningful subspaces. For example, Keller *et al.* [65] and Böhm *et al.* [17] selected subspaces with high contrast by statistical approaches. Another example is that Kriegel *et al.* [68] explored the outlyingness of a data object in a subspace made up by its nearest neighbors. Similar to the traditional outlier detection, various outlyingness measurements are designed for different purposes. For instance, Müller *et al.* [81] designed the outlier score of a data object based on the size and the dimensionality of the corresponding reference cluster. Some studies [17, 65, 85] directly adopted the existing outlyingness measurement from the traditional outlier detection.

For example, Kriegel *et al.* [67] introduced *SOD*, a method to detect data objects that do not fit well into their axis-parallel subspaces. Given a data object o, the axis-parallel subspace of o is a subspace spanned by o's reference points. The reference points of o are modeled as a set of data objects, each of which shares at least n of its k-nearest neighbors with that of o. Note that n and k are two user input parameters. n is the minimum number of nearest neighbors that a reference point and o have in common; and k is the number of nearest neighbors that a data object needs to consider. The outlier score of o is designed based on two elements: (1) the distance between o and the mean values of the reference points in each dimension; (2) the variance of reference points in each dimension.

Müller *et al.* [83] proposed the *OUTRES* approach. *OUTRES* aims to assess the

contribution of some selected subspaces where an object deviates from its neighborhood (modeled as the closest cluster). The selected subspaces are chosen to provide a large contrast between the object and its immediate neighborhood (modeled as a set of objects within a specified radius. The chosen radius is subspace dependent). For a chosen object, *OUTRES* computes an aggregate outlier score that incorporates only the contribution of subspaces where the object has significantly low density (more than two standard deviations from the mean). The challenge of calibrating density measures in different subspaces is tackled by using an adaptive neighborhood which grows according to the number of features in the subspace.

The effectiveness of techniques in subspace outlier detection depend on both the quality of the selected subspaces and the fitness of the adopted outlyingness measurements.

2.3 Related Work to the Proposed Problems

In this section, we review the related work for each of the proposed problems, and briefly point out their relationships to our studies and the significant differences.

2.3.1 Mining Outlying Subspaces

The mining outlying subspaces problem is mainly related to the existing work on outlying property detection and subspace outlier detection. In this section, we review the previous studies and show the signicant differences.

Outlying Property Detection

Given a data set, *outlying property detection* problem discovers the properties, which distinguish an outlier from the normal data objects in the data set. To the best of our knowledge, [7, 8, 80] are the only studies on finding explanation of outlying aspects, and are most relevant to our problem of mining outlying subspaces.

Given a multidimensional *categorical* database and an outlier, Angiulli *et al.* [7] found the top-k subsets of attributes (i.e., subspaces) from which the outlier receives the highest outlyingness scores. The outlyingness score for a given object in a subspace is calculated based on the frequency of the value that the outlier takes in the subspace. It tries to find subspaces E and S such that the outlier is frequent in one and much less frequent than

expected in the other. Searching all such rules is computationally costly. To reduce the cost within a manageable scope, the method takes two parameters, σ and θ, to constrain the frequencies of the given object in subspaces E and S, respectively. Therefore, if a query object is not outlying compared to the other objects, no outlying properties may be detected. Angiulli *et al.* [8] provide a version of *numeric* data for this problem.

Micenková *et al.* [80] proposed a method to find a subspace as possible explanations for a detected outlier. Particularly, given a data set and an outlier, this method sampled a subset of the original data based on the outlier for classification. Subset feature selection methods were used to select candidate explanation subspaces. Classification accuracy was used to measure the outlierness of a data object. A candidate subspace where the outlier has the highest outlierness score will be selected as the explanation subspace.

How is our work different? There are several essential differences between [7, 8] and this study. First, the problems in [7, 8] find contextual rule based explanations, while our method returns individual subspaces where the query object is mostly outlying comparing to the other subspaces. The meaning of the two types of explanation is fundamentally different. Second, Angiulli *et al.* [7] focuses on categorical data, and our method targets on numeric data. Although [8] considers numeric data, its mining target is substantially different from this work. Specifically, given a set of objects O in a multi-dimensional space D and a query object $q \in O$, [8] finds the pairs (E, d) satisfying $E \subseteq D$ and $d \in D \setminus E$, such that there exists a subset $O' \subseteq O$, including q, in which objects are similar on E (referred to as *explanation*), while q is essentially different from the other objects in O' on d (referred to as *property*). Besides one-dimensional attributes, our method can find outlying subspaces with arbitrary dimensionality.

Micenková *et al.* [80] is also very different from our study. First, they require a detected outlier to be the input query; while we do not. Second, their method only finds one subspace; while our method finds top-k subspaces. Third, they use the classification accuracy to measure the outlierness; while we use density rank as outlierness measure.

Subspace Outlier Detection

Subspace outlier detection aims to detect data objects, which are drastically deviating from the majority in some subspaces. Detailed literature review can be found in Section 2.2.

How is our work different? The problem settings of outlying subspaces mining and subspace outlier detection are different. In outlying subspaces mining, we focus on the

following question. Given a query object o in a multidimensional numeric data set O, in which subspace is o most outlying? Note that the query itself may or may not be an outlier.

Recently, some subspaces outlier detection studies attempt to use the selected subspaces as explanations to demonstrate the outlying properties of outliers. The explanations may be a byproduct of outlier detection. For example, Böhm *et al.* [17] and Keller *et al.* [65] proposed statistical approaches *CMI* and *HiCS* to select subspaces for a multidimensional database, where there may exist outliers with high deviations. Both *CMI* and *HiCS* are fundamentally different from our method. They choose highly contrasting subspaces for all possible outliers in a data set, while our method chooses subspaces based on the query object. Kriegel *et al.* [67] introduced *SOD*, a method to detect outliers in axis-parallel subspaces. There are two major differences between *SOD* and our work. First, *SOD* is still an outlier detection method, and the hyperplane is a byproduct of the detection process. Our method does not detect outliers at all. Second, the models to identify the outlying subspaces in the two methods are very different. When calculating the outlyingness score, *SOD* only considers the nearest neighbors as references in the full space. Our method considers all objects in the database and their relationship with the query object in subspaces.

Other Data Mining Techniques

To some extent, outlyingness is related to *uniqueness* and *uniqueness mining*. Paravastu *et al.* [90] discovered the feature-value combinations that make a particular record unique. Their task formulation is reminiscent of infrequent itemset mining, and uses a level-wise Apriori enumeration strategy [3]. It needs a discretization step. Our method is native for continuous data.

Our method uses probability density to measure outlying degree in a subspace. There are a few *density-based outlier detection methods*, such as [20, 69, 55, 2]. Our method is inherently different from those, since we do not find outlier objects at all.

2.3.2 Mining Contrast Subspaces

The problem of mining contrast subspaces is related to the existing work on contrast mining, subspace outlier detection and typicality queries. We briefly review previous works and show the difference in the following.

Contrast mining

Contrast mining discovers patterns and models that manifest drastic differences between datasets. Dong and Bailey [35] presented a comprehensive review of contrast mining, together with a range of real-life applications. Some of the best known types of contrast patterns include emerging patterns [36], contrast sets [13] and subgroups [119]. Although their definitions vary, the mining methods share many similarities [86].

How is our work different? Contrast pattern mining identifies patterns by considering all objects of all classes in the complete pattern space. Orthogonally, contrast subspace mining focuses on one object, and identifies subspaces where a query object demonstrates the strongest overall similarity to one class against the other. These two mining problems are fundamentally different. To the best of our knowledge, the contrast subspace mining problem has not been systematically explored in the data mining literature. Though Chen and Dong [25] presented a contrast-pattern length based algorithm to detection global outliers, their problem setting is different from ours.

Subspace Outlier Detection

As reviewed in Section 2.2, *subspace outlier detection* discovers objects that significantly deviate from the majority in some subspaces.

How is our work different? In contrast subspace mining, the query object may or may not be an outlier. We are trying to find the top-k subspaces, in which a query object is the most typical in the current class and is very unlikely to occur in other classes.

Some recent studies find subspaces that may contain substantial outliers. Keller *et al.* [65] and Böhm *et al.* [17] proposed statistical approaches *HiCS* and *CMI* to select subspaces for a multidimensional database, where there may exist outliers with high deviations. Both *HiCS* and *CMI* differ from our method. Technically, they choose subspaces for all outliers in a given database, while our method chooses the most contrasting subspaces for a query object. In *HiCS* and *CMI*, *contrast* refers to the differences between the assumptions on whether the subspaces are mutually independent or not. In our work, *contrast* is defined as the differences of the likelihoods that a query belongs to the given class or not.

Typicality Queries

Given a data set and a query object, *typicality queries* focus on the problem of finding the most typical data objects according to the query. Hua *et al.* [58] introduced a novel top-k *typicality query*, which ranks objects according to their typicality in a data set or a class of objects. Density estimation based methods are used to calculate the typicality of a query object with respect to a set of data objects. Cai *et al.* [21] proposed a method that adopted concepts from human cognition, to answer the top-k typicality queries. The typicality of an object with respect to a set of data objects was calculated based on the similarity and support of the object with respect to the set of data objects.

How is our work different? The problem settings are different. Works in typicality query [58, 21] aim to find the most typical data objects according to the query object; in contrast subspace mining, we find the most contrasting subspaces for a query object. Although both [58] and our work use density estimation methods to calculate the typicality/likelihood of a query object with respect to a set of data objects, typicality queries [58] do not consider subspaces.

2.3.3 Mining Contextual Outliers

In this section, we briefly review the previous works on contextual outlier detection, rule-based outlier detection and some other data mining techniques related to the proposed contextual outlier detection problem. We show the relationships and differences between the existing works and the proposed problem.

Contextual Outlier Detection

Given a set of data objects, *contextual outlier detection* focuses on finding outlier objects under some specific circumstances. A contextual outlier usually associates with a set of contextual attributes and a set of behavioral attributes. Contextual attributes define the context of the outliers; while behavioral attributes are used to determine outliers. For example, "a temperature of 30°C in winter in Vancouver" is an outlier. "winter" and "Vancouver" are the contextual attributes; and "a temperature of 30°C" is the behavioral attribute. Contextual outliers are also known as conditional outliers [107].

Song *et al.* [107] proposed the notion of conditional outliers to model the outliers manifested by a set of behavioral attributes (e.g. temperature) conditionally depending

on a set of contextual attributes (e.g. longitude and latitude). The behavioral attributes and the contextual attributes are pre-defined.

Li *et al.* [116] introduced a hypergraph-based outlier detection test (HOT) to identify outliers and their contexts. Given a dataset and a minimum support threshold, a hypergraph is built based on the frequent itemsets in the dataset. Particularly, a vertex is a data object; and a hyperedge denotes a group of objects containing frequent itemsets. A deviation score has been designed to calculate the outlyingness of a data object in a certain attribute with respect to a hyperedge. Given a deviation threshold θ, a data object o is an outlier in attribute A with respect to a hyperedge he, if the deviation score of o in A with respect to he is lower than θ.

There are a few other contextual outlier detection studies, which detection outliers and their corresponding contexts from different angles. For example, Valko *et al.* [111] detected conditional anomalies using a training set of labeled examples with possible label noise. And Wang and Davidson [114] used random walks to find context and outliers.

How is our work different? The contextual outlier mining problem proposed in this thesis is different from the existing studies in problem settings and solutions. For example, in [107], both the behavioral attributes and the contextual attributes have to be predefined; in contrast, our method can automatically identify these attributes. Take HOT [116] as another example. There are two major differences between HOT and our method. First, the detected outliers are very different. Outliers detected by HOT are single data objects (vertexes). Outliers detected by our methods are group-by tuples (hyperedges). Second, outlying attributes are different. An outlying attribute detected by HOT only contains one attribute; while an outlying subspace detected by our method can have several attributes. Thus, the problem and the framework proposed by our work are more general than the ones proposed by HOT.

Rule-based Outlier Detection

Rule-based outlier detection learns rules, which can represent the normal behaviors of a given data set. A data object is an outlier if it does not follow any of the learned rule. There are a number of methods [41, 23, 77, 84, 117] using rules in outlier detection. A contextual outlier (r, o) identified in our method can be written as a pair of rules: $cond(r, o) \Rightarrow avs(r) - cond(r, o)$ for the reference group, and $cond(r, o) \Rightarrow avs(o) - cond(r, o)$ for the outlier group.

How is our work different? Our method differs from the existing methods in several

important aspects. First, most of the existing rule-based methods focus on detecting individual outliers, and may not be able to identify outlier groups and measure the outlyingness accordingly. Second, many existing rule-based methods use rules to model only the normal objects or strong associations. Outliers are individual objects that do not follow those rules. Those methods do not model and analyze context explicitly. Lastly, many existing methods, such as [31, 74, 117], set strict constraints on the size of the rules or the aggregate groups to be considered, such as a very small number of items/attributes allowed in a rule or only the parents and their sibling groups.

Other Data Mining Techniques

Our study is also related to the previous work on *emerging pattern mining* ([36] and the consequent studies) and *contrast mining* (see [34] and the references there). Conceptually, mining contextual outliers can be regarded as mining the non-redundant set of emerging patterns in all possible subspaces and under all possible shared AVSs as the constraints. Chen and Dong [25] provided an emerging pattern length based outlier detection method, but it is limited to global outliers. To the best of our knowledge the contextual outlier detection problem has not been systematically explored in the emerging/contrast pattern mining area.

More broadly, our study uses related concepts and techniques in *data cube computation* [47, 16] and *formal concept analysis* [44]. However, to the best of our knowledge, no previous study systematically integrates the techniques to tackle the contextual outlier detection problem.

2.3.4 Mining Markov Blanket Based Outliers

The problem of mining Markov blanket based outliers is related to the existing work on subspace outlier detection and Bayesian network based outlier detection. We briefly review previous works and show the difference in the following

Subspace Outlier Detection

As mentioned before, a brief literature review of *subspace outlier detection* can be found in Section 2.2.

How is our work different? We adopt a novel subspace search approach for the subspace selection, which can significantly reduce the computational cost in subspace selection.

Most existing subspace outlier detection algorithms have prohibitive computational costs in subspace selection. For example, some state-of-the-art subspace outlier detection algorithms, such as *OUTRES* [83], *HiCS* [65] and *CMI* [17], employ an Apriori-style subspace search scheme. Given a data set with n attributes, an Apriori-style subspace search scheme has $2^n - 1$ possible candidate subspaces, and thus incurs very expensive computational cost. Recently, Nguyen *et al.* [85] proposed the $4S$ algorithm to mitigate the problem of exponential runtime. Instead of a levelwise subspace search strategy, the $4S$ algorithm computes the correlation of each pair of dimensions and transforms the top K pairs with the largest correlations to an undirected correlation graph. Then, $4S$ performs a direct mining of correlated subspaces in the graph. The $4S$ algorithm still needs to explore an exponentially large number of candidate subspaces.

We use the concept of Markov blankets in Bayesian networks for the subspace selection. Particulary, given a data set with n attributes, we find outliers from only n subspaces selected by a Markov blanket based method, instead of $2^n - 1$ arbitrary subspaces.

Bayesian Network Based Outlier Detection

Bayesian network based outlier detection usually builds a Bayesian network [93] over a full feature space first. And then, for a testing data object, it uses the learned Bayesian network to estimate the posterior probability of being an outlier for the testing data object. A number of works in outlier detection use this method, such as [10, 22, 118].

For example, Wong *et al.* [118] proposed a Bayesian network based outlier detection technique for the disease outbreak detection. Given a database with Emergency Department (or "ED" for short) records, all attributes are divided into two groups by domain experts, namely, environmental set and indicator set. Environmental set is a group of attributes, which can form the trends of disease outbreaks. The rest attributes are put into the indicator set. The Bayesian network, which is used to represent the baseline distribution, is conditioned on forming relation only between attributes belonging to environmental set to attributes in the indicator set. The distribution of the testing data is compared against the baseline distribution to identify outlying patterns that may cause disease outbreaks.

How is our work different? Our method detects subspace outlier, which is different from most of the existing works in Bayesian networks outlier detection. Methods in most existing

Bayesian networks outlier detection search outliers in a single space of given attributes, and are unable to detect outliers hidden in subspaces.

Chapter 3

Mining Outlying Aspects

When we are investigating an object in a data set, which itself may or may not be an outlier, can we identify unusual (i.e., outlying) aspects of the object?

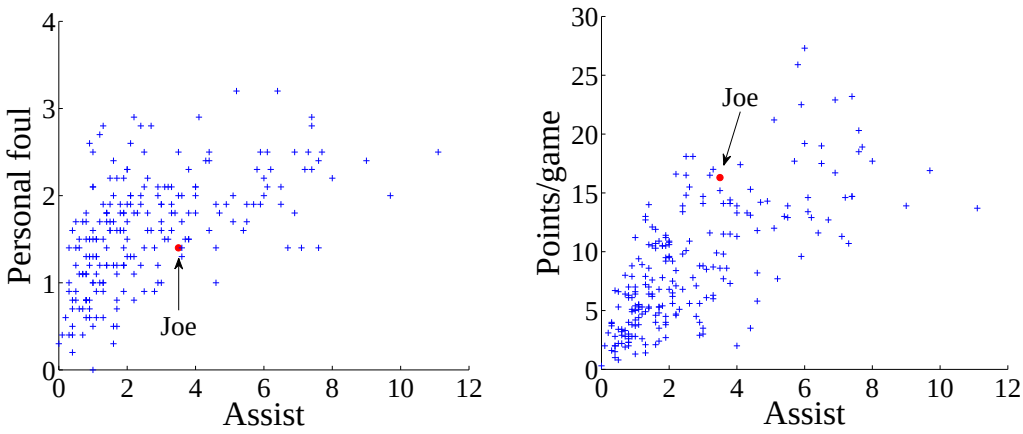
In this chapter, we identify the novel problem of *mining outlying aspects on numeric data*. Given a query object o in a multidimensional numeric data set O, in which subspace is o most outlying? Technically, we use the rank of the probability density of an object in a subspace to measure the outlyingness of the object in the subspace. A minimal subspace where the query object is ranked the best is an outlying aspect. Computing the outlying aspects of a query object is far from trivial. A naïve method has to calculate the probability densities of all objects and rank them in every subspace, which is very costly when the dimensionality is high. We systematically develop a heuristic method that is capable of searching data sets with tens of dimensions efficiently. Our empirical study using both real data and synthetic data demonstrates that our method is effective and efficient.

3.1 Motivation

In many application scenarios, a user may wish to investigate a specific object, in particular, the aspects where the object is most unusual compared to the rest of the data. In addition to the application Scenario 1 of the Example 1.1, there are quite a few interesting applications of outlying aspects mining in practice.

For example, when a commentator mentions an NBA player, the commentator may want to name the most distinguishing features of the player, though the player may not be top ranked on those aspects or on any others among all players. Take the technical statistics of

the 220 guards on assist, personal foul and points/game in the NBA Season 2012-2013 as an example (Figure 3.1), an answer for Joe Johnson may be "the most distinguishing feature of Joe Johnson is his scoring ability with respect to his performance on personal foul". Note we collect the NBA Season 2012-2013 statistics from http://sports.yahoo.com/nba/stats.



(a) Assist w.r.t personal foul

(b) Assist w.r.t points/game

(c) Personal foul w.r.t points/game

Figure 3.1: Performance of NBA guards on assist, personal foul and points/game in the 2012-2013 Season (the solid circle (•) represents Joe Johnson)

As another example, when evaluating an applicant to a university program, who herself /himself may not necessarily be outstanding among all applicants, one may want to know the strength or weakness of the applicant, such as "the student's strength is the combination of GPA and volunteer experience, ranking her/him in the top 15% using these combined

aspects". Moreover, in an insurance company, a fraud analyst may collect the information about various aspects of a claim, and wonder in which aspects the claim is most unusual. Furthermore, in commercial promotion, when designing an effective advertisement, it may be useful for marketers to know the most distinctive set of features characterizing the product. Similar examples can easily be found in other analytics applications.

The questions illustrated in the above examples are different from traditional outlier detection. Specifically, instead of searching for outliers from a data set, here we are given a query object and want to find the outlying aspects whereby the object is most unusual. The query object itself may or may not be an outlier in the full space or in any specific subspaces. In this problem, we are not interested in other individual outliers or inliers. The outlying aspect finding questions cannot be answered by the existing outlier detection methods directly.

We emphasize that investigating specific objects is a common practice in anomaly and fraud detection and analysis. Specifying query objects is an effective way to explicitly express analysts' background knowledge about data. Moreover, finding outlying aspects extends and generalizes the popular exercise of checking a suspect of anomaly or fraud. Currently, more often than not an analyst has to check the features of an outlying object one by one to find outlying features, but still cannot identify combinations of features where the object is unusual.

We address several technical challenges and make solid contributions on several fronts.

First, we identify and formulate the problem of outlying aspect mining on numeric data. Although [7, 8] recently studied detecting outlying properties of exceptional objects, their methods find contextual rule based explanations. We discuss the differences between our model and theirs in detail in Section 2.3.1 in Chapter 2. As illustrated, outlying aspect mining has immediate applications in data analytics practice.

Second, how can we compare the outlyingness of an object in different subspaces? While comparing the outlyingness of different objects in the same subspace is well studied and straightforward, comparing outlyingness of the same object in different subspaces is subtle, since different subspaces may have different scales and distribution characteristics. We propose a simple yet principled approach. In a subspace, we rank all objects in the ascending order of probability density. A smaller probability density and thus a better rank indicates that the query object is more outlying in the subspace. Then, we compare the rank statistics of the query object in different subspaces, and return the subspaces of the best rank as

the outlying aspects of the query object. To avoid redundancy, we only report minimal subspaces. That is, if a query object is ranked the same in subspaces S and S' such that S is a proper subspace of S' (i.e., $S \subset S'$), then S' is not reported since S is more general. Our model can be extended to many outlyingness measures other than probability density, which we leave for future work.

Third, how can we compute the outlying aspects fast, particularly on high dimensional data sets? A naïve method using the definition of outlying aspects directly has to calculate the probability densities of all objects and rank them in every subspace. This method incurs heavy cost when the dimensionality is high. On a data set of 100 dimensions, $2^{100} - 1 \approx 1.27 \times 10^{30}$ subspaces have to be examined, which is unfortunately computationally prohibitive using the state-of-the-art technology. To tackle the problem, we systematically develop a heuristic method that is capable of searching data sets with dozens of dimensions efficiently. Specifically, we develop pruning techniques that can avoid computing the probability densities of many objects in many subspaces. These effective pruning techniques enable our method to mine outlying aspects on data sets with tens of dimensions, as demonstrated later in our experiments.

Last, to evaluate outlying aspect mining, we conduct an extensive empirical study on both real and synthetic data sets. We illustrate the characteristics of discovered outlying aspects, and justify the value of outlying aspect mining. Moreover, we examine the effectiveness of our pruning techniques and the efficiency of our methods.

The rest of the chapter is organized as follows. We formulate the problem of outlying aspect mining Section 3.2. In Section 3.3, We recall the basics of kernel density estimation, which is used to estimate the probability density of objects, and present the framework of our method *OAMiner* (for Outlying Aspect Miner). In Section 3.4, we discuss the critical techniques in *OAMiner*. We report a systematic empirical study in Section 3.5, and conclude the chapter in Section 3.6.

3.2 Problem Definition

Let $D = \{D_1, \ldots, D_d\}$ be a d-dimensional space, where the domain of D_i is $\mathbb{R}$, the set of real numbers. A *subspace* $S \subseteq D$ ($S \neq \emptyset$) is a subset of D. We also call D the *full space*.

Consider a set O of n objects in space D. For an object $o \in O$, denote by $o.D_i$ the value of o in dimension D_i ($1 \leq i \leq d$). For a subspace $S = \{D_{i_1}, \ldots, D_{i_l}\} \subseteq D$, the *projection* of

o in S is $o^S = (o.D_{i_1}, \ldots, o.D_{i_l})$. The *dimensionality* of S, denoted by $|S|$, is the number of dimensions in S.

In a subspace $S \subseteq D$, we assume that we can define a measure of *outlyingness degree* $OutDeg(\cdot)$ such that for each object $o \in O$, $OutDeg(o)$ measures the outlyingness of o. Without loss of generality, we assume that the lower the outlyingness degree $OutDeg(o)$, the more outlying the object o.

In this study, we assume the generative model. That is, the set of objects O are generated (i.e., sampled) from an often unknown probability distribution. Thus, we can use the probability density of an object o, denoted by $f(o)$, as the outlyingness degree. The smaller the value of $f(o)$, the more outlying the object o. We discuss how to estimate the probability densities in Section 3.3.1.

How can we compare the outlyingness of an object in different subspaces? Unfortunately, we cannot compare the outlyingness degree or probability density values directly, since the outlyingness degree and the probability density values depend on the properties of specific subspaces, such as their scales. For example, it is well known that probability density tends to be low in subspaces of higher dimensionality, since such subspaces often have a larger "volume" and thus are sparser.

To tackle this issue, we propose to use rank statistics. Specifically, in a subspace S, we rank all objects in O in their outlyingness degree ascending order. For an object $o \in O$, we denote by

$$rank_S(o) = |\{o' \mid o' \in O, OutDeg(o') < OutDeg(o)\}| + 1 \tag{3.1}$$

the *outlyingness rank* of o in subspace S. The smaller the rank value, the more outlying the object is comparing to the other objects in O in subspace S. We can compare the outlyingness of an object o in two subspaces S_1 and S_2 using $rank_{S_1}(o)$ and $rank_{S_2}(o)$. Object o is more outlying in the subspace where it has the smaller rank. Apparently, in Equation 3.1, for objects with the same outlyingness degree (probability density value), their outlyingness ranks are the same.

Suppose for object o there are two subspaces S and S' such that $S \subset S'$ and $rank_S(o) = rank_{S'}(o)$. Since S is more general than S', S is more significant in manifesting the outlyingness of o at the rank of $rank_S(o)$ relative to the other objects in the data set. Therefore, S' is redundant given S in terms of outlying aspects. Note that we use rank statistics instead of the absolute outlyingness degree values to compare the outlyingness of an object in different subspaces.

Rank statistics allows us to compare outlyingness in different subspaces, which is an advantage. At the same time, in high dimensional subspaces where the probability density values of objects are very small, comparing the ranks may not be reliable, since the subtle differences in probability density values may be due to noise or sensitivity to parameter settings in the density estimation. Ranking such objects may be misleading. Moreover, more often than not, users do not want to see high dimensional subspaces as answers, since high dimensional subspaces are hard to understand. Thus, we assume a *maximum dimensionality threshold* $\ell > 0$, and consider only subspaces whose dimensionality are not greater than ℓ. Please note that the problem cannot be solved using a minimum density threshold, since the density values are space and dimensionality sensitive, as explained before.

Based on the above discussion, we formalize the problem as follows.

Definition 3.1 (Problem definition)**.** *Given a set of objects O in a multidimensional space D, a query object $q \in O$ and a maximum dimensionality threshold $0 < \ell \leq |D|$, a subspace $S \subseteq D$ $(0 < |S| \leq \ell)$ is called a* ***minimal outlying subspace*** *of q if*

1. *(Rank minimality) there does not exist another subspace $S' \subseteq D$ $(S' \neq \emptyset)$, such that $rank_{S'}(q) < rank_S(q)$; and*
2. *(Subspace minimality) there does not exist another subspace $S'' \subset S$ such that $rank_{S''}(q) = rank_S(q)$.*

The problem of ***outlying aspect mining*** *is to find the minimal outlying subspaces of q.* ■

Apparently, given a query object q, there exists at least one, and may be more than one minimal outlying subspace.

3.3 The Framework

In this section, we first review the essentials of kernel density estimation techniques. Then, we present the framework of our *OAMiner* method.

3.3.1 Kernel Density Estimation

We use kernel density estimation [102, 105] to estimate the probability density given a set of objects O. Given a random sample $\{o_1, o_2, \ldots, o_n\}$ drawn from some distribution with an unknown probability density f in space $\mathbb{R}$, the probability density f at a point $o \in \mathbb{R}$ can be estimated by

$$\hat{f}_h(o) = \frac{1}{n}\sum_{i=1}^{n} K_h(o - o_i) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{o - o_i}{h}\right)$$

where $K(\cdot)$ is a kernel, and h is a smoothing parameter called the *bandwidth*. A widely adopted approach to estimate the bandwidth is Silverman's rule of thumb [105], which suggests $h = 1.06\sigma n^{-\frac{1}{5}}$, σ being the standard deviation of the sample. To further reduce the sensitivity to outliers, in this work, we use a better rule of thumb [51] and set

$$h = 1.06 \min\{\sigma, \frac{R}{1.34}\} n^{-\frac{1}{5}} \tag{3.2}$$

where $R = X_{[0.75n]} - X_{[0.25n]}$, and $X_{[0.25n]}$ and $X_{[0.75n]}$, respectively, are the first and the third quartiles.

For the d-dimensional case ($d \geq 2$), $o = (o.D_1, \ldots, o.D_d)^T$, and $o_i = (o_i.D_1, \ldots, o_i.D_d)^T$ ($1 \leq i \leq n$). Then, the probability density of f at point $o \in \mathbb{R}^d$ can be estimated by

$$\hat{f}_H(o) = \frac{1}{n}\sum_{i=1}^{n} K_H(o - o_i)$$

where H is a bandwidth matrix.

The product kernel, which consists of the product of one-dimensional kernels, is a good choice for multivariate kernel density estimator in practice [102, 52]. We have

$$\hat{f}_H(o) = \frac{1}{n\prod_{j=1}^{d} h_{D_j}} \sum_{i=1}^{n}\left\{\prod_{j=1}^{d} K\left(\frac{o.D_j - o_i.D_j}{h_{D_j}}\right)\right\} \tag{3.3}$$

where h_{D_i} is the bandwidth of dimension D_i ($1 \leq i \leq d$).

Note that the product kernel does not assume that the dimensions are independent. Otherwise, the density estimation would be

$$\hat{f}_H(o) = \prod_{j=1}^{d}\left(\frac{1}{n \cdot h_{D_j}}\sum_{i=1}^{n} K\left(\frac{o.D_j - o_i.D_j}{h_{D_j}}\right)\right)$$

In this study, we adopt the Gaussian kernel, which has been popularly used. The distance between two objects is measured by Euclidean distance. The kernel function is

$$K\left(\frac{o-o_i}{h}\right) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(o-o_i)^2}{2h^2}} \tag{3.4}$$

Note that other kernel functions and distance functions may be used in our framework.

Plugging Equation 3.4 into Equation 3.3, the density of a query object $q \in O$ in subspace S can be estimated as

$$\hat{f}_S(q) = \hat{f}_S(q^S) = \frac{1}{n(2\pi)^{\frac{|S|}{2}} \prod_{D_i \in S} h_{D_i}} \sum_{o \in O} e^{-\sum_{D_i \in S} \frac{(q.D_i - o.D_i)^2}{2h_{D_i}^2}} \tag{3.5}$$

Since we are interested in only the rank of q, that is, $rank_S(q)$, and

$$c = \frac{1}{n(2\pi)^{\frac{|S|}{2}} \prod_{D_i \in S} h_{D_i}} \tag{3.6}$$

is a factor common to every object in subspace S and thus does not affect the ranking at all, we can rewrite Equation 3.5 as

$$\hat{f}_S(q) \sim \tilde{f}_S(q) = \sum_{o \in O} e^{-\sum_{D_i \in S} \frac{(q.D_i - o.D_i)^2}{2h_{D_i}^2}} \tag{3.7}$$

where symbol "$\sim$" means equivalence for ranking.

For the sake of clarity, we call $\tilde{f}_S(q)$ the *quasi-density* of q in S. Please note that, using $\tilde{f}_S(q)$ instead of $\hat{f}_S(q)$ not only simplifies the description, but also saves computational cost for calculating $rank_S(q)$. We will illustrate the details in Section 3.4.

We can show an interesting property – invariance of ranking under linear transformation.

Proposition 3.1 (Invariance)**.** *Given a set of objects O in space $S = \{D_1, \ldots, D_d\}$, define a linear transformation $g(o) = (a_1 o.D_1 + b_1, \ldots, a_d o.D_d + b_d)$ for any $o \in O$, where $a_1, \ldots, a_d$ and $b_1, \ldots, b_d$ are real numbers. Let $O' = \{g(o) | o \in O\}$ be the transformed data set. For any objects $o_1, o_2 \in O$ such that $\tilde{f}_S(o_1) > \tilde{f}_S(o_2)$ in O, $\tilde{f}_S(g(o_1)) > \tilde{f}_S(g(o_2))$ if the product kernel is used and the bandwidths are set using Härdle's rule of thumb (Equation 3.2).*

Proof. For any dimension $D_i \in S$ $(1 \leq i \leq d)$, the mean value of $\{o.D_i \mid o \in O\}$, denoted by μ_i, is $\frac{1}{|O|} \sum_{o \in O} o.D_i$, the standard deviation of $\{o.D_i \mid o \in O\}$, denoted by σ_i,

is $\sqrt{\frac{1}{|O|}\sum_{o\in O}(o.D_i-\mu_i)^2}$, and the bandwidth of D_i (h_i) is $1.06\min\{\sigma_i, \frac{R}{1.34}\}|O|^{-\frac{1}{5}}$, where R is the difference between the first and the third quartiles of O in D_i.

We perform the linear transformation $g(o).D_i = a_i o.D_i + b_i$ for any $o \in O$. Then, the mean value of $\{g(o).D_i \mid o \in O\}$ is $\frac{1}{|O|}\sum_{o\in O}(a_i o.D_i+b_i) = a_i\mu_i+b_i$, and the standard deviation of $\{g(o).D_i \mid o \in O\}$ is $\sqrt{\frac{1}{|O|}\sum_{o\in O}(a_i o.D_i + b_i - a_i\mu_i - b_i)^2} = a_i\sqrt{\frac{1}{|O|}\sum_{o\in O}(o.D_i-\mu_i)^2} = a_i\sigma_i$.

Correspondingly, the bandwidth of D_i is $1.06\min\{a_i\sigma_i, \frac{a_iR}{1.34}\}|O|^{-\frac{1}{5}}$ after the linear transformation. As the distance between two objects in D_i is also enlarged by a_i, the quasi-density calculated by Equation 3.7 keeps unchanged. Thus, the ranking is invariant under linear transformation. ∎

Using quasi-density estimation (Equation 3.7), we can have a baseline algorithm for computing the outlyingness rank in a subspace S, as shown in Algorithm 3.1. The baseline method estimates the quasi-density of each object in a data set, and ranks them. Let the total number of objects be n. In order to compute the quasi-density of every data object, the baseline method essentially has to compute the distance between every pair of objects in every dimension of S. Therefore, the time complexity is $O(n^2|S|)$ in each subspace S.

Algorithm 3.1 $rank_S(q)$ – baseline

Input: a set of objects O, query object $q \in O$, and subspace S
Output: $rank_S(q)$

1: **for** each object $o \in O$ **do**
2: compute $\tilde{f}_S(o)$ using Equation 3.7
3: **end for**
4: **return** $rank_S(q) = |\{o \mid o \in O, \tilde{f}_S(o) < \tilde{f}_S(q)\}| + 1$

3.3.2 The Framework of *OAMiner*

To reduce the computational cost, we present Algorithm 3.2, the framework of our method *OAMiner* (for Outlying Aspect Miner).

First of all, *OAMiner* removes the dimensions where all values of objects are identical, since no object is outlying in such dimensions. As a result, the standard deviations of all dimensions involved for outlying aspect mining are greater than 0.

In order to ensure that *OAMiner* can find the most outlying subspaces, we have to enumerate all possible subspaces in a systematic way. Here, we adopt the set enumeration

Algorithm 3.2 The framework of *OAMiner*

Input: a set of objects O and query object $q \in O$
Output: the set of minimal outlying subspaces for q

```
1: initialize r_best ← |O| and Ans ← ∅;
2: remove D_i from D if the values of all objects in D_i are identical;
3: compute rank_{D_i}(q) in each dimension D_i ∈ D;
4: sort all dimensions in rank_{D_i}(q) ascending order;
5: for each subspace S searched by traversing the set enumeration tree in a depth-first
   manner do
6:     compute rank_S(q);
7:     if rank_S(q) < r_best then
8:         r_best ← rank_S(q), Ans ← {S};
9:     end if
10:    if rank_S(q) = r_best and S is minimal then
11:        Ans ← Ans ∪ {S};
12:    end if
13:    if a subspace pruning condition is true then
14:        prune all super-spaces of S
15:    end if
16: end for
17: return Ans
```

tree approach [99], which has been popularly used in many data mining methods. Conceptually, a set enumeration tree takes a total order on the set, the dimensions in the context of our problem, and then enumerates all possible combinations systematically. For example, Figure 3.2 shows a set enumeration tree that enumerates all subspaces of space $D = \{D_1, D_2, D_3, D_4\}$.

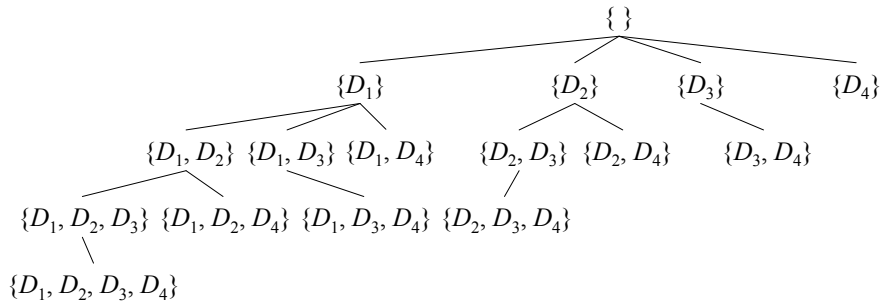


Figure 3.2: A set enumeration tree.

OAMiner searches subspaces by traversing the subspace enumeration tree in a depth-first manner. Given a set of objects O, a query object $q \in O$, and a subspace S, if $rank_S(q) = 1$, then every super-space of S cannot be a minimal outlying subspace and thus can be pruned.

Pruning Rule 3.1. *If $rank_S(q) = 1$, according to the dimensionality minimality condition in the problem definition (Definition 3.1), all super-spaces of S can be pruned.* ■

In the case of $rank_S(q) > 1$, *OAMiner* prunes subspaces according to the current best rank of q in the search process. More details will be discussed in Section 3.4.3.

Heuristically, we want to find subspaces early where the query object q has a low rank, so that the pruning techniques take better effect. Motivated by this observation, we compute the outlyingness rank of q in each dimension D_i, and order all dimensions in the ascending order of $rank_{D_i}(q)$.

In general, the outlyingness rank does not have any monotonicity with respect to subspaces. That is, for subspaces $S_1 \subset S_2$, neither $rank_{S_1}(q) \leq rank_{S_2}(q)$ nor $rank_{S_1}(q) \geq rank_{S_2}(q)$ holds in general. Example 3.1 illustrates this situation with a toy data set.

Example 3.1. Given a set of objects $O = \{o_1, o_2, o_3, o_4\}$ with 2 numeric attributes D_1 and D_2. The values of each object in O are listed in Table 3.1. Using Equation 3.7, we estimate the quasi-density values of each object on different subspaces (Table 3.2). We can see that $\tilde{f}_{\{D_1\}}(o_2) > \tilde{f}_{\{D_1\}}(o_4)$ and $\tilde{f}_{\{D_2\}}(o_2) > \tilde{f}_{\{D_2\}}(o_4)$, which indicate $rank_{\{D_1\}}(o_2) > rank_{\{D_1\}}(o_4)$ and $rank_{\{D_2\}}(o_2) > rank_{\{D_2\}}(o_4)$. However, for subspace $\{D_1, D_2\}$, since $\tilde{f}_{\{D_1,D_2\}}(o_2) < \tilde{f}_{\{D_1,D_2\}}(o_4)$, $rank_{\{D_1,D_2\}}(o_2) < rank_{\{D_1,D_2\}}(o_4)$.

object	$o_i.D_1$	$o_i.D_2$
o_1	14.23	1.5
o_2	13.2	1.78
o_3	13.16	2.31
o_4	14.37	1.97

Table 3.1: A numeric data set example

■

object	$\tilde{f}_{\{D_1\}}(o_i)$	$\tilde{f}_{\{D_2\}}(o_i)$	$\tilde{f}_{\{D_1,D_2\}}(o_i)$
o_1	2.229	1.832	1.305
o_2	2.220	2.529	1.300
o_3	2.187	1.626	1.185
o_4	2.113	2.474	1.314

Table 3.2: Quasi-density values of objects in Table 3.1

To make the situation even more challenging, probability density itself does not have any monotonicity with respect to subspaces. Given a query object q, and subspaces $S_1 \subset S_2$. According to Equation 3.5, we have

$$\begin{aligned}\frac{\hat{f}_{S_1}(q)}{\hat{f}_{S_2}(q)} &= \frac{\sum_{o\in O} e^{-\sum_{D_i\in S_1}\frac{(q.D_i-o.D_i)^2}{2h_{D_i}^2}}}{n(2\pi)^{\frac{|S_1|}{2}}\prod_{D_i\in S_1} h_{D_i}} \Big/ \frac{\sum_{o\in O} e^{-\sum_{D_i\in S_2}\frac{(q.D_i-o.D_i)^2}{2h_{D_i}^2}}}{n(2\pi)^{\frac{|S_2|}{2}}\prod_{D_i\in S_2} h_{D_i}} \\ &= (2\pi)^{\frac{|S_2|-|S_1|}{2}}\prod_{D_i\in S_2\setminus S_1} h_{D_i}\frac{\sum_{o\in O} e^{-\sum_{D_i\in S_1}\frac{(q.D_i-o.D_i)^2}{2h_{D_i}^2}}}{\sum_{o\in O} e^{-\sum_{D_i\in S_2}\frac{(q.D_i-o.D_i)^2}{2h_{D_i}^2}}}\end{aligned}$$

Since $S_1 \subset S_2$, $\sum_{o\in O} e^{-\sum_{D_i\in S_1}\frac{(q.D_i-o.D_i)^2}{2h_{D_i}^2}} \Big/ \sum_{o\in O} e^{-\sum_{D_i\in S_2}\frac{(q.D_i-o.D_i)^2}{2h_{D_i}^2}} \geq 1$ and $(2\pi)^{\frac{|S_2|-|S_1|}{2}} > 1$. However, in the case $\prod_{D_i\in S_2\setminus S_1} h_{D_i} < 1$, there is no guarantee that $\frac{\hat{f}_{S_1}(q)}{\hat{f}_{S_2}(q)} > 1$ always holds. Thus, neither $\hat{f}_{S_1}(q) \leq \hat{f}_{S_2}(q)$ nor $\hat{f}_{S_1}(q) \geq \hat{f}_{S_2}(q)$ holds in general.

3.4 Critical Techniques in *OAMiner*

In this section, we present a bounding-pruning-refining algorithm to efficiently compute the outlyingness rank of an object in a subspace, and discuss the critical techniques to prune subspaces.

Table 3.3 lists the frequently used notations in this Chapter.

Notation	Description
D	a d-dimensional space
O	a set of objects in space D
h_{D_i}	the bandwidth of dimension D_i
$rank_S(o)$	outlyingness rank of object o in subspace S
$\tilde{f}_S(o)$	quasi-density of object o in subspace S estimated by Equation 4.3
$\tilde{f}_S^{O'}(o)$	the sum of quasi-density contributions of objects in set O' to object o in subspace S
$TN_S^{\epsilon,o}$	ϵ-tight neighborhood of object o in subspace S
$LN_S^{\epsilon,o}$	ϵ-loose neighborhood of object o in subspace S
$dc_S(o,o')$	the quasi-density contribution of object o' to object o in subspace S
$Comp_S(o)$	a set of objects where *OAMiner* can determine that their densities are less than the density of o in subspace S and its super-spaces

Table 3.3: Summary of frequently used notations in Chapter 3.

3.4.1 Bounding Probability Density

In order to obtain the rank statistics about outlyingness, *OAMiner* has to compare the density of the query object with the densities of other objects. To speed up density estimation of objects, we observe that the contributions from remote objects to the density of an object are very small, and the density of an object can be bounded. Technically, we can derive upper and lower bounds of the probability density of an object using a neighborhood. Again, we denote by $\tilde{f}_S(o)$ the quasi-density of object o in subspace S.

For the sake of clarity, we introduce two notations at first. Given objects $o, o' \in O$, a subspace S, and a subset $O' \subseteq O$, we denote by $dc_S(o,o')$ the quasi-density contribution of o' to o in S, and $\tilde{f}_S^{O'}(o)$ the sum of quasi-density contributions of objects in O' to o. That is,

$$dc_S(o,o') = e^{-\sum\limits_{D_i \in S} \frac{(o.D_i - o'.D_i)^2}{2h_{D_i}^2}}$$

$$\tilde{f}_S^{O'}(o) = \sum_{o' \in O'} e^{-\sum\limits_{D_i \in S} \frac{(o.D_i - o'.D_i)^2}{2h_{D_i}^2}}$$

To efficiently estimate the bounds of $\tilde{f}_S(o)$, we define two kinds of neighborhoods. For an object $o \in O$, a subspace S, and $\{\epsilon_{D_i} \mid \epsilon_{D_i} > 0, D_i \in S\}$, the *$\epsilon$-tight neighborhood* of o in S, denoted by $TN_S^{\epsilon,o}$, is $\{o' \in O \mid \forall D_i \in S, |o.D_i - o'.D_i| \leq \epsilon_{D_i}\}$, the *$\epsilon$-loose neighborhood* of o in S, denoted by $LN_S^{\epsilon,o}$, is $\{o' \in O \mid \exists D_i \in S, |o.D_i - o'.D_i| \leq \epsilon_{D_i}\}$. An object is called

as an *ϵ-tight (loose) neighbor* if it is in the ϵ-tight (loose) neighborhood. We will illustrate how to efficiently compute $TN_S^{\epsilon,o}$ and $LN_S^{\epsilon,o}$ in Section 3.4.2.

According to the definitions of $TN_S^{\epsilon,o}$ and $LN_S^{\epsilon,o}$, we obtain the following properties.

Property 3.1. $TN_S^{\epsilon,o} \subseteq LN_S^{\epsilon,o}$. ∎

Property 3.2. $TN_S^{\epsilon,o} = LN_S^{\epsilon,o}$ *if* $|S| = 1$. ∎

Based on $TN_S^{\epsilon,o}$ and $LN_S^{\epsilon,o}$, O can be divided into three disjoint subsets: $TN_S^{\epsilon,o}$, $LN_S^{\epsilon,o} \setminus TN_S^{\epsilon,o}$ and $O \setminus LN_S^{\epsilon,o}$. For any object $o' \in O$, we obtain a lower bound and an upper bound of $dc_S(o, o')$ as follows.

Theorem 3.2 (Single quasi-density contribution bounds)**.** *Given an object $o \in O$, a subspace S, and a set $\{\epsilon_{D_i} \mid \epsilon_{D_i} > 0, D_i \in S\}$. Then, for any object $o' \in TN_S^{\epsilon,o}$,*

$$dc_S^{\epsilon} \leq dc_S(o, o') \leq dc_S^{max}(o)$$

for any object $o' \in LN_S^{\epsilon,o} \setminus TN_S^{\epsilon,o}$,

$$dc_S^{min}(o) \leq dc_S(o, o') \leq dc_S^{max}(o)$$

for any object $o' \in O \setminus LN_S^{\epsilon,o}$,

$$dc_S^{min}(o) \leq dc_S(o, o') < dc_S^{\epsilon}$$

where

$$\begin{aligned} dc_S^{\epsilon} &= e^{-\sum_{D_i \in S} \frac{\epsilon_{D_i}^2}{2h_{D_i}^2}} \\ dc_S^{max}(o) &= e^{-\sum_{D_i \in S} \frac{\min_{o' \in O} \{|o.D_i - o'.D_i|\}^2}{2h_{D_i}^2}} \\ dc_S^{min}(o) &= e^{-\sum_{D_i \in S} \frac{\max_{o' \in O} \{|o.D_i - o'.D_i|\}^2}{2h_{D_i}^2}} \end{aligned}$$

Proof. (i) Given an object $o' \in TN_S^{\epsilon,o}$, for any dimension $D_i \in S$, $\min_{o'' \in O}\{|o.D_i - o''.D_i|\} \leq |o.D_i - o'.D_i| \leq \epsilon_{D_i}$. Thus,

$$e^{-\sum_{D_i \in S} \frac{\epsilon_{D_i}^2}{2h_{D_i}^2}} \leq e^{-\sum_{D_i \in S} \frac{|o.D_i - o'.D_i|^2}{2h_{D_i}^2}} \leq e^{-\sum_{D_i \in S} \frac{\min_{o'' \in O} \{|o.D_i - o''.D_i|\}^2}{2h_{D_i}^2}}.$$

That is, $dc_S^{\epsilon} \leq dc_S(o, o') \leq dc_S^{max}(o)$.

(ii) Given an object $o' \in LN_S^{\epsilon,o} \backslash TN_S^{\epsilon,o}$, for any dimension $D_i \in S$, $\min_{o'' \in O}\{|o.D_i - o''.D_i|\} \leq |o.D_i - o'.D_i| \leq \max_{o'' \in O}\{|o.D_i - o''.D_i|\}$. Thus,

$$e^{-\sum_{D_i \in S} \frac{\max_{o'' \in O}\{|o.D_i - o''.D_i|\}^2}{2h_{D_i}^2}} \leq e^{-\sum_{D_i \in S} \frac{|o.D_i - o'.D_i|^2}{2h_{D_i}^2}} \leq e^{-\sum_{D_i \in S} \frac{\min_{o'' \in O}\{|o.D_i - o''.D_i|\}^2}{2h_{D_i}^2}}.$$

That is, $dc_S^{min}(o) \leq dc_S(o, o') \leq dc_S^{max}(o)$.

(iii) Given an object $o' \in O \setminus LN_S^{\epsilon,o}$, for any dimension $D_i \in S$, $\epsilon_{D_i} < |o.D_i - o'.D_i| \leq \max_{o'' \in O}\{|o.D_i - o''.D_i|\}$. Thus,

$$e^{-\sum_{D_i \in S} \frac{\max_{o'' \in O}\{|o.D_i - o''.D_i|\}^2}{2h_{D_i}^2}} \leq e^{-\sum_{D_i \in S} \frac{|o.D_i - o'.D_i|^2}{2h_{D_i}^2}} < e^{-\sum_{D_i \in S} \frac{\epsilon_{D_i}^2}{2h_{D_i}^2}}.$$

That is, $dc_S^{min}(o) \leq dc_S(o, o') < dc_S^{\epsilon}$. ∎

Using the size of $TN_S^{\epsilon,o}$ and $LN_S^{\epsilon,o}$, we obtain a lower bound and an upper bound of $\tilde{f}_S(o)$ as follows.

Corollary 3.1 (Bounds by neighborhood size)**.** *For any object $o \in O$,*

$$|TN_S^{\epsilon,o}|\ dc_S^{\epsilon} + (|O| - |TN_S^{\epsilon,o}|)\ dc_S^{min}(o) \leq \tilde{f}_S(o)$$

$$\tilde{f}_S(o) \leq |LN_S^{\epsilon,o}|\ dc_S^{max}(o) + (|O| - |LN_S^{\epsilon,o}|)\ dc_S^{\epsilon}$$

Proof. We divide O into three disjoint subsets $TN_S^{\epsilon,o}$, $LN_S^{\epsilon,o} \setminus TN_S^{\epsilon,o}$ and $O \setminus LN_S^{\epsilon,o}$. By Theorem 3.2, for objects belonging to $TN_S^{\epsilon,o}$, we have

$$|TN_S^{\epsilon,o}|\ dc_S^{\epsilon} \leq \sum_{o' \in TN_S^{\epsilon,o}} dc_S(o, o') \leq |TN_S^{\epsilon,o}|\ dc_S^{max}(o)$$

For objects belonging to $LN_S^{\epsilon,o} \setminus TN_S^{\epsilon,o}$, we have

$$(|LN_S^{\epsilon,o}| - |TN_S^{\epsilon,o}|)\ dc_S^{min}(o) \leq \sum_{o' \in LN_S^{\epsilon,o} \backslash TN_S^{\epsilon,o}} dc_S(o, o') \leq (|LN_S^{\epsilon,o}| - |TN_S^{\epsilon,o}|)\ dc_S^{max}(o)$$

For objects belonging to $O \setminus LN_S^{\epsilon,o}$, we have

$$(|O| - |LN_S^{\epsilon,o}|)\ dc_S^{min}(o) \leq \sum_{o' \in O \backslash LN_S^{\epsilon,o}} dc_S(o, o') < (|O| - |LN_S^{\epsilon,o}|)\ dc_S^{\epsilon}$$

As

$$\tilde{f}_S(o) = \sum_{o' \in O} dc_S(o, o') = \sum_{o' \in TN_S^{\epsilon,o}} dc_S(o, o') + \sum_{o' \in LN_S^{\epsilon,o} \setminus TN_S^{\epsilon,o}} dc_S(o, o') + \sum_{o' \in O \setminus LN_S^{\epsilon,o}} dc_S(o, o'),$$

Thus,

$$\begin{aligned} \tilde{f}_S(o) &\geq |TN_S^{\epsilon,o}|\ dc_S^{\epsilon} + (|LN_S^{\epsilon,o}| - |TN_S^{\epsilon,o}|)\ dc_S^{min}(o) + (|O| - |LN_S^{\epsilon,o}|)\ dc_S^{min}(o) \\ &= |TN_S^{\epsilon,o}|\ dc_S^{\epsilon} + (|O| - |TN_S^{\epsilon,o}|)\ dc_S^{min}(o) \end{aligned}$$

$$\begin{aligned} \tilde{f}_S(o) &\leq |TN_S^{\epsilon,o}|\ dc_S^{max}(o) + (|LN_S^{\epsilon,o}| - |TN_S^{\epsilon,o}|)\ dc_S^{max}(o) + (|O| - |LN_S^{\epsilon,o}|)\ dc_S^{\epsilon} \\ &= |LN_S^{\epsilon,o}|\ dc_S^{max}(o) + (|O| - |LN_S^{\epsilon,o}|)\ dc_S^{\epsilon} \end{aligned}$$

Moreover, if $LN_S^{\epsilon,o} \subset O$, i.e. $O \setminus LN_S^{\epsilon,o} \neq \emptyset$, then

$$\tilde{f}_S(o) < |LN_S^{\epsilon,o}|\ dc_S^{max}(o) + (|O| - |LN_S^{\epsilon,o}|)\ dc_S^{\epsilon}$$

■

Corollary 3.1 allows us to compute the quasi-density bounds of an object without computing the quasi-density contributions of other objects to it.

Moreover, by Theorem 3.2, we can obtain following corollaries.

Corollary 3.2 (Bounds by ϵ-tight neighbors)**.** *For any object $o \in O$ and $O' \subseteq TN_S^{\epsilon,o}$,*

$$\tilde{f}_S^{O'}(o) + (|TN_S^{\epsilon,o}| - |O'|)\ dc_S^{\epsilon} + (|O| - |TN_S^{\epsilon,o}|)\ dc_S^{min}(o) \leq \tilde{f}_S(o)$$

$$\tilde{f}_S(o) \leq \tilde{f}_S^{O'}(o) + (|LN_S^{\epsilon,o}| - |O'|)\ dc_S^{max}(o) + (|O| - |LN_S^{\epsilon,o}|)\ dc_S^{\epsilon}$$

Proof. Since $O' \subseteq TN_S^{\epsilon,o}$, for objects belonging to $O \setminus O'$, we divide them into $TN_S^{\epsilon,o} \setminus O'$, $LN_S^{\epsilon,o} \setminus TN_S^{\epsilon,o}$ and $O \setminus LN_S^{\epsilon,o}$. Then

$$\tilde{f}_S(o) = \tilde{f}_S^{O'}(o) + \sum_{o' \in TN_S^{\epsilon,o} \setminus O'} dc_S(o, o') + \sum_{o' \in LN_S^{\epsilon,o} \setminus TN_S^{\epsilon,o}} dc_S(o, o') + \sum_{o' \in O \setminus LN_S^{\epsilon,o}} dc_S(o, o'),$$

By Theorem 3.2, for objects belonging to $TN_S^{\epsilon,o} \setminus O'$, we have

$$(|TN_S^{\epsilon,o}| - |O'|)\ dc_S^{\epsilon} \leq \sum_{o' \in TN_S^{\epsilon,o} \setminus O'} dc_S(o, o') \leq (|TN_S^{\epsilon,o}| - |O'|)\ dc_S^{max}(o)$$

For objects belonging to $LN_S^{\epsilon,o} \setminus TN_S^{\epsilon,o}$, we have

$$(|LN_S^{\epsilon,o}| - |TN_S^{\epsilon,o}|)\ dc_S^{min}(o) \leq \sum_{o' \in LN_S^{\epsilon,o} \setminus TN_S^{\epsilon,o}} dc_S(o,o') \leq (|LN_S^{\epsilon,o}| - |TN_S^{\epsilon,o}|)\ dc_S^{max}(o)$$

For objects belonging to $O \setminus LN_S^{\epsilon,o}$, we have

$$(|O| - |LN_S^{\epsilon,o}|)\ dc_S^{min}(o) \leq \sum_{o' \in O \setminus LN_S^{\epsilon,o}} dc_S(o,o') < (|O| - |LN_S^{\epsilon,o}|)\ dc_S^{\epsilon}$$

Thus,

$$\begin{aligned} \tilde{f}_S(o) &\geq \tilde{f}_S^{O'}(o) + (|TN_S^{\epsilon,o}| - |O'|)\ dc_S^{\epsilon} + (|LN_S^{\epsilon,o}| - |TN_S^{\epsilon,o}|)\ dc_S^{min}(o) + (|O| - |LN_S^{\epsilon,o}|)\ dc_S^{min}(o) \\ &= \tilde{f}_S^{O'}(o) + (|TN_S^{\epsilon,o}| - |O'|)\ dc_S^{\epsilon} + (|O| - |TN_S^{\epsilon,o}|)\ dc_S^{min}(o) \end{aligned}$$

$$\begin{aligned} \tilde{f}_S(o) &\leq \tilde{f}_S^{O'}(o) + (|TN_S^{\epsilon,o}| - |O'|)\ dc_S^{max}(o) + (|LN_S^{\epsilon,o}| - |TN_S^{\epsilon,o}|)\ dc_S^{max}(o) + (|O| - |LN_S^{\epsilon,o}|)\ dc_S^{\epsilon} \\ &= \tilde{f}_S^{O'}(o) + (|LN_S^{\epsilon,o}| - |O'|)\ dc_S^{max}(o) + (|O| - |LN_S^{\epsilon,o}|)\ dc_S^{\epsilon} \end{aligned}$$

Moreover, if $LN_S^{\epsilon,o} \subset O$, i.e. $O \setminus LN_S^{\epsilon,o} \neq \emptyset$, then

$$\tilde{f}_S(o) < \tilde{f}_S^{O'}(o) + (|LN_S^{\epsilon,o}| - |O'|)\ dc_S^{max}(o) + (|O| - |LN_S^{\epsilon,o}|)\ dc_S^{\epsilon}$$

■

Corollary 3.3 (Bounds by ϵ-loose neighbors)**.** *For any object $o \in O$ and $TN_S^{\epsilon,o} \subset O' \subseteq LN_S^{\epsilon,o}$,*

$$\tilde{f}_S^{O'}(o) + (|O| - |O'|)\ dc_S^{min}(o) \leq \tilde{f}_S(o)$$

$$\tilde{f}_S(o) < \tilde{f}_S^{O'}(o) + (|LN_S^{\epsilon,o}| - |O'|)\ dc_S^{max}(o) + (|O| - |LN_S^{\epsilon,o}|)\ dc_S^{\epsilon}$$

Proof. Since $TN_S^{\epsilon,o} \subset O' \subseteq LN_S^{\epsilon,o}$, for objects belonging to $O \setminus O'$, we divide them into $LN_S^{\epsilon,o} \setminus O'$ and $O \setminus LN_S^{\epsilon,o}$. Then

$$\tilde{f}_S(o) = \tilde{f}_S^{O'}(o) + \sum_{o' \in LN_S^{\epsilon,o} \setminus O'} dc_S(o,o') + \sum_{o' \in O \setminus LN_S^{\epsilon,o}} dc_S(o,o'),$$

By Theorem 3.2, for objects belonging to $LN_S^{\epsilon,o} \setminus O'$, we have

$$(|LN_S^{\epsilon,o}| - |O'|)\ dc_S^{min}(o) \leq \sum_{o' \in LN_S^{\epsilon,o} \setminus TN_S^{\epsilon,o}} dc_S(o,o') \leq (|LN_S^{\epsilon,o}| - |O'|)\ dc_S^{max}(o)$$

For objects belonging to $O \setminus LN_S^{\epsilon,o}$, we have

$$(|O| - |LN_S^{\epsilon,o}|)\ dc_S^{min}(o) \leq \sum_{o' \in O \setminus LN_S^{\epsilon,o}} dc_S(o,o') < (|O| - |LN_S^{\epsilon,o}|)\ dc_S^{\epsilon}$$

Thus,

$$\begin{aligned} \tilde{f}_S(o) &\geq \tilde{f}_S^{O'}(o) + (|LN_S^{\epsilon,o}| - |O'|)\ dc_S^{min}(o) + (|O| - |LN_S^{\epsilon,o}|)\ dc_S^{min}(o) \\ &= \tilde{f}_S^{O'}(o) + (|O| - |O'|)\ dc_S^{min}(o) \\ \tilde{f}_S(o) &\leq \tilde{f}_S^{O'}(o) + (|LN_S^{\epsilon,o}| - |O'|)\ dc_S^{max}(o) + (|O| - |LN_S^{\epsilon,o}|)\ dc_S^{\epsilon} \end{aligned}$$

Moreover, if $LN_S^{\epsilon,o} \subset O$, i.e. $O \setminus LN_S^{\epsilon,o} \neq \emptyset$, then

$$\tilde{f}_S(o) < \tilde{f}_S^{O'}(o) + (|LN_S^{\epsilon,o}| - |O'|)\ dc_S^{max}(o) + (|O| - |LN_S^{\epsilon,o}|)\ dc_S^{\epsilon}$$

■

Corollary 3.4 (Bounds by a supper set of ϵ-loose neighbors)**.** *For any object $o \in O$ and $LN_S^{\epsilon,o} \subset O' \subseteq O$,*

$$\tilde{f}_S^{O'}(o) + (|O| - |O'|)\ dc_S^{min}(o) \leq \tilde{f}_S(o)$$

$$\tilde{f}_S(o) \leq \tilde{f}_S^{O'}(o) + (|O| - |O'|)\ dc_S^{\epsilon}$$

Proof. Since $LN_S^{\epsilon,o} \subset O' \subseteq O$, Then

$$\tilde{f}_S(o) = \tilde{f}_S^{O'}(o) + \sum_{o' \in O \setminus O'} dc_S(o,o'),$$

By Theorem 3.2, for objects belonging to $O \setminus O'$, we have

$$(|LN_S^{\epsilon,o}| - |O'|)\ dc_S^{min}(o) \leq \sum_{o' \in O \setminus O'} dc_S(o,o') \leq (|O| - |O'|)\ dc_S^{\epsilon}$$

Thus,

$$\tilde{f}_S(o) \geq \tilde{f}_S^{O'}(o) + (|O| - |O'|)\ dc_S^{min}(o)$$

$$\tilde{f}_S(o) \leq \tilde{f}_S^{O'}(o) + (|O| - |O'|)\ dc_S^{\epsilon}$$

■

Since the density of o is the sum of the density contributions of all objects in O, and the density contribution decreases with the distance, *OAMiner* first computes the quasi-density contributions from the objects in $TN_S^{\epsilon,o}$, then from the objects in $LN_S^{\epsilon,o} \setminus TN_S^{\epsilon,o}$, and last from the objects in $O \setminus LN_S^{\epsilon,o}$.

By computing the bounds of $\tilde{f}_S(o)$, *OAMiner* takes a bounding-pruning-refining method, shown in Algorithm 3.3, to efficiently perform density comparison in subspace S. Initially, *OAMiner* estimates the quasi-density of query object q, which is denoted by $\tilde{f}_S(q)$. Then, for an object o, *OAMiner* firstly computes the bounds of $\tilde{f}_S(o)$ by the sizes of $TN_S^{\epsilon,o}$ and $LN_S^{\epsilon,o}$ (Corollary 3.1), and compares the bounds with $\tilde{f}_S(q)$ (Steps 1-8). If $\tilde{f}_S(q)$ is smaller than the lower bounds or greater than the upper bound, then we have $\tilde{f}_S(q) < \tilde{f}_S(o)$ or $\tilde{f}_S(q) > \tilde{f}_S(o)$. That is, the relationship between $\tilde{f}_S(q)$ and $\tilde{f}_S(o)$ is determined. Thus, Algorithm 3.3 stops. Otherwise, *OAMiner* updates the lower and upper bounds of $\tilde{f}_S(o)$ by involving the quasi-density contributions of objects in $TN_S^{\epsilon,o}$ (Steps 10-20), in $LN_S^{\epsilon,o} \setminus TN_S^{\epsilon,o}$ (Steps 21-31), and in $O \setminus LN_S^{\epsilon,o}$ (Steps 32-42) one by one, and repeatedly compares the updated bounds with $\tilde{f}_S(q)$, until the relationship between $\tilde{f}_S(q)$ and $\tilde{f}_S(o)$ is fully determined.

Algorithm 3.3 Density comparison

Input: quasi-density of the query object $\tilde{f}_S(q)$, object $o \in O$, subspace S, the ϵ-tight neighborhood of o $TN_S^{\epsilon,o}$, and the ϵ-loose neighborhood of o $LN_S^{\epsilon,o}$.

Output: a boolean value indicating $\tilde{f}_S(o) < \tilde{f}_S(q)$ is true or not.

$L \leftarrow$ the lower bound of $\tilde{f}_S(o)$ computed by Corollary 3.1; // bounding
if $L > \tilde{f}_S(q)$ **then**
    return false; // pruning
end if
$U \leftarrow$ the upper bound of $\tilde{f}_S(o)$ computed by Corollary 3.1; // bounding
if $U < \tilde{f}_S(q)$ **then**
    return true; // pruning
end if
$O' \leftarrow \emptyset$; $\tilde{f}_S^{O'}(o) \leftarrow 0$;
for each $o' \in TN_S^{\epsilon,o}$ **do**
    $\tilde{f}_S^{O'}(o) \leftarrow \tilde{f}_S^{O'}(o) + dc_S(o, o')$; $O' \leftarrow O' \cup \{o'\}$; // refining
    $L \leftarrow$ the lower bound of $\tilde{f}_S(o)$ computed by Corollary 3.2; // bounding
    if $L > \tilde{f}_S(q)$ **then**
        return false; // pruning
    end if
    $U \leftarrow$ the upper bound of $\tilde{f}_S(o)$ computed by Corollary 3.2; // bounding
    if $U < \tilde{f}_S(q)$ **then**
        return true; // pruning
    end if
end for
for each $o' \in LN_S^{\epsilon,o} \setminus TN_S^{\epsilon,o}$ **do**
    $\tilde{f}_S^{O'}(o) \leftarrow \tilde{f}_S^{O'}(o) + dc_S(o, o')$; $O' \leftarrow O' \cup \{o'\}$; // refining
    $L \leftarrow$ the lower bound of $\tilde{f}_S(o)$ computed by Corollary 3.3; // bounding
    if $L > \tilde{f}_S(q)$ **then**
        return false; // pruning
    end if
    $U \leftarrow$ the upper bound of $\tilde{f}_S(o)$ computed by Corollary 3.3; // bounding
    if $U < \tilde{f}_S(q)$ **then**
        return true; // pruning
    end if
end for
for each $o' \in O \setminus LN_S^{\epsilon,o}$ **do**
    $\tilde{f}_S^{O'}(o) \leftarrow \tilde{f}_S^{O'}(o) + dc_S(o, o')$; $O' \leftarrow O' \cup \{o'\}$; // refining
    $L \leftarrow$ the lower bound of $\tilde{f}_S(o)$ computed by Corollary 3.4; // bounding
    if $L > \tilde{f}_S(q)$ **then**
        return false; // pruning
    end if
    $U \leftarrow$ the upper bound of $\tilde{f}_S(o)$ computed by Corollary 3.4; // bounding
    if $U < \tilde{f}_S(q)$ **then**
        return true; // pruning
    end if
end for
return false;

In *OAMiner*, the neighborhood distance in dimension D_i, denoted by ϵ_{D_i}, is defined as $\alpha\sigma_{D_i}$, where σ_{D_i} is the standard deviation in dimension D_i, and α is a parameter. Our experiments show that α is not sensitive, and can be set in the range of $0.8 \sim 1.2$, by which *OAMiner* runs efficiently. It is still an open question about how to set the best neighborhood distance for bounding — this is a future research problem. Theorem 3.3 guarantees that no matter how to set the neighborhood distance, the ranking results keep unchanged.

Theorem 3.3. *Given an object $o \in O$, and a subspace S, for any neighborhood distances ϵ_1 and ϵ_2, $rank_S^{\epsilon_1}(o) = rank_S^{\epsilon_2}(o)$, where $rank_S^{\epsilon_1}(o)$ ($rank_S^{\epsilon_2}(o)$) is the outlyingness rank of o in S computed using ϵ_1 (ϵ_2).*

Proof. We prove by contradiction.

Given a set of objects O, a subspace S, two neighborhood distances ϵ_1 and ϵ_2. Let $q \in O$ be the query object. For an object $o \in O$, denote by L_{ϵ_1} the lower bound of $\tilde{f}_S(o)$ estimated by ϵ_1, U_{ϵ_2} the upper bound of $\tilde{f}_S(o)$ estimated by ϵ_2.

Assume that $\tilde{f}_S(q) < L_{\epsilon_1}$ and $\tilde{f}_S(q) > U_{\epsilon_2}$.

As L_{ϵ_1} is a lower bound of $\tilde{f}_S(o)$, and U_{ϵ_2} is an upper bound of $\tilde{f}_S(o)$, so that $L_{\epsilon_1} < \tilde{f}_S(o) < U_{\epsilon_2}$. Then, we have $\tilde{f}_S(q) < L_{\epsilon_1} < \tilde{f}_S(o)$ and $\tilde{f}_S(o) < U_{\epsilon_2} < \tilde{f}_S(q)$. Consequently, $\tilde{f}_S(o) < \tilde{f}_S(q) < \tilde{f}_S(o)$. A contradiction.

Thus, $rank_S^{\epsilon_1}(q) = |\{o \in O \mid \tilde{f}_S(o) < \tilde{f}_S(q)\}| + 1 = rank_S^{\epsilon_2}(q)$. ∎

3.4.2 Efficiently Estimating Density Bounds

In this subsection, we present strategies in Algorithm 3.3 that efficiently estimate the lower and upper bounds of quasi-density.

Consider a candidate subspace $S \subseteq D$, and an object $o \in O$. To estimate lower and upper bounds of $\tilde{f}_S(o)$, *OAMiner* has to compute $TN_S^{\epsilon,o}$, $LN_S^{\epsilon,o}$, dc_S^{ϵ}, $dc_S^{min}(o)$, $dc_S^{max}(o)$ and $dc_S(o, o')$, where $o' \in O$.

In the case $|S| = 1$, we compute $TN_S^{\epsilon,o}$, dc_S^{ϵ}, $dc_S^{min}(o)$, $dc_S^{max}(o)$ and $dc_S(o, o')$ based on their definitions directly. As pointed out in Section 3.4.1, $TN_S^{\epsilon,o} = LN_S^{\epsilon,o}$ in this case. Moreover, the density contribution is symmetrical, so that the computational cost for $dc_S(o', o)$ can be saved if $dc_S(o, o')$ is available.

Please recall that *OAMiner* searches subspaces by traversing the subspace enumeration tree in a depth-first manner. For a subspace S satisfying $|S| \geq 2$, denote by $par(S)$ the

parent subspace of S. Suppose $S \setminus par(S) = D'$ ($|D'| = 1$). Then, we have

$$TN_S^{\epsilon,o} = TN_{par(S)}^{\epsilon,o} \cap TN_{D'}^{\epsilon,o} \tag{3.8}$$

$$LN_S^{\epsilon,o} = LN_{par(S)}^{\epsilon,o} \cup LN_{D'}^{\epsilon,o} \tag{3.9}$$

$$dc_S^{\epsilon} = e^{-\sum\limits_{D_i \in S} \frac{\epsilon_{D_i}^2}{2h_{D_i}^2}} = e^{-(\sum\limits_{D_i \in par(S)} \frac{\epsilon_{D_i}^2}{2h_{D_i}^2} + \frac{\epsilon_{D'}^2}{2h_{D'}^2})} = dc_{par(S)}^{\epsilon} \cdot dc_{D'}^{\epsilon} \tag{3.10}$$

$$\begin{aligned} dc_S^{min}(o) &= e^{-(\sum\limits_{D_i \in par(S)} \frac{\max\limits_{o' \in O} \{|o.D_i - o'.D_i|\}^2}{2h_{D_i}^2} + \frac{\max\limits_{o' \in O} \{|o.D' - o'.D'|\}^2}{2h_{D'}^2})} \\ &= dc_{S \setminus par(S)}^{min}(o) \cdot dc_{D'}^{min}(o) \end{aligned} \tag{3.11}$$

$$\begin{aligned} dc_S^{max}(o) &= e^{-(\sum\limits_{D_i \in par(S)} \frac{\min\limits_{o' \in O} \{|o.D_i - o'.D_i|\}^2}{2h_{D_i}^2} + \frac{\min\limits_{o' \in O} \{|o.D' - o'.D'|\}^2}{2h_{D'}^2})} \\ &= dc_{S \setminus par(S)}^{max}(o) \cdot dc_{D'}^{max}(o) \end{aligned} \tag{3.12}$$

$$\begin{aligned} dc_S(o, o') &= e^{-\sum\limits_{D_i \in S} \frac{(o.D_i - o'.D_i)^2}{2h_{D_i}^2}} = e^{-(\sum\limits_{D_i \in par(S)} \frac{(o.D_i - o'.D_i)^2}{2h_{D_i}^2} + \frac{(o.D' - o'.D')^2}{2h_{D'}^2})} \\ &= dc_{par(S)}(o, o') \cdot dc_{D'}(o, o') \end{aligned} \tag{3.13}$$

Thus, it is efficient for *OAMiner* to estimate the bounds of $\tilde{f}_S(o)$ using $par(S)$ and $S \setminus par(s)$.

3.4.3 Subspace Pruning

Recall that *OAMiner* searches subspaces by traversing the subspace enumeration tree in a depth-first manner. During the search process, let $\mathcal{S}_1$ be the set of subspaces that *OAMiner* has searched, and $\mathcal{S}_2$ the set of subspaces that *OAMiner* has not searched yet. Clearly, $|\mathcal{S}_1 \cup \mathcal{S}_2| = 2^{|D|} - 1$. Given a query object q, let $r_{best} = \min\limits_{S \in \mathcal{S}_1} \{rank_S(q)\}$ be the best rank that q has achieved so far. We can use r_{best} to prune some subspaces not searched yet. Specifically, for a subspace $S \in \mathcal{S}_2$, once we can determine that $rank_S(q) > r_{best}$, then S cannot be an outlying aspect, and thus can be pruned.

Observation 3.1. *When subspace S is met in a depth-first search of the subspace set enumeration tree, let r_{best} be the best rank of q in all the subspaces searched so far. Given object q with $rank_S(q) \geq 1$, if for every proper super-space $S' \supset S$, $rank_{S'}(q) > r_{best}$, then all proper super-spaces of S can be pruned.* ∎

For the case that $rank_S(q) = 1$, all super-spaces of S can be pruned directly due to the

dimensionality minimality condition in the problem definition (Pruning Rule 3.1). Thus, we only consider the case $rank_S(q) > 1$ here.

To implement Observation 3.1, in a subspace S where $rank_S(q) > 1$, we check whether there are at least r_{best} objects that are ranked better than q in every super-space of S. If so, all the super-spaces of S can be pruned. Please note that the condition *OAMiner* checks is sufficient, but not necessary.

Recall that the common factor c (Equation 3.6) does not affect the outlyingness rank. For simplicity, *OAMiner* computes the quasi-density $\tilde{f}_S(o)$ (Equation 3.7) instead of probability density $\hat{f}_S(o)$ (Equation 3.5) for ranking. Then, we have the following monotonicity of $\tilde{f}_S(o)$ with respect to subspaces.

Lemma 3.1. *Consider a set of objects O, and two subspaces S and S' satisfying $S' \supset S$. Let $D_i \in S' \setminus S$. If the standard deviation of O in D_i is greater than 0, then for any object $o \in O$, $\tilde{f}_S(o) > \tilde{f}_{S'}(o)$.*

Proof. Consider $D_i \in S' \setminus S$, for any object $o' \in O$, we have $\frac{(o.D_i - o'.D_i)^2}{2h_{D_i}^2} \geq 0$. Since the standard deviation of O in D_i is greater than 0, there exists at least one object $o'' \in O$, such that $\frac{(o.D_i - o''.D_i)^2}{2h_{D_i}^2} > 0$, that is, $e^{-\frac{(o.D_i - o''.D_i)^2}{2h_{D_i}^2}} < 1$. Thus,

$$\begin{aligned}
\tilde{f}_S(o) &= \sum_{o' \in O} e^{-\sum\limits_{D_i \in S} \frac{(o.D_i - o'.D_i)^2}{2h_{D_i}^2}} \\
&> \sum_{o' \in O} e^{-\left(\sum\limits_{D_i \in S} \frac{(o.D_i - o'.D_i)^2}{2h_{D_i}^2} + \sum\limits_{D_i \in S' \setminus S} \frac{(o.D_i - o'.D_i)^2}{2h_{D_i}^2}\right)} = \tilde{f}_{S'}(o)
\end{aligned}$$

■

Recall that *OAMiner* removes the dimensions with standard deviation 0 in the pre-processing step (Step 2 in Algorithm 3.2). Thus, the standard deviation of any dimension $D_i \in S' \setminus S$ is greater than 0.

OAMiner sorts all dimensions in D in the ascending order of $rank_{D_i}(q)$ ($D_i \in D$), and traverses the subspace set enumeration tree in the depth-first manner. Denote by R the ascending order of $rank_{D_i}(q)$. For a subspace $S = \{D_{i_1}, ..., D_{i_m}\}$, listing in R, let $R(S) = \{D_j \mid D_j \text{ is behind } D_{i_m} \text{ in } R\}$. By Lemma 3.1, for any subspace S' such that $S \subset S' \subseteq S \cup R(S)$, the minimum quasi-density of q, denoted by $\tilde{f}^{min}_{sup(S)}(q)$, is $\tilde{f}_{S \cup R(S)}(q)$.

An object $o \in O$ is called a *competitor* of q in S if $\tilde{f}_S(o) < \tilde{f}^{min}_{sup(S)}(q)$. The set of competitors of q in S is denoted by $Comp_S(q)$. Clearly, for any $o \in Comp_S(q)$, by Lemma 3.1 we have $\tilde{f}_{S'}(o) < \tilde{f}_S(o) < \tilde{f}^{min}_{sup(S)}(q) \leq \tilde{f}_{S'}(q)$. Thus, $rank_{S'}(o) < rank_{S'}(q)$. Moreover, we have the following property of $Comp_S(q)$.

Property 3.3. *Given a query object q and a subspace S, for any subspace S' such that $S \subset S'$, $Comp_S(q) \subseteq Comp_{S'}(q)$.*

Proof. Since $S \subset S'$, by Lemma 3.1, for any $o \in Comp_S(q)$,

$$\tilde{f}_{S'}(o) < \tilde{f}_S(o) < \tilde{f}^{min}_{sup(S)}(q).$$

Since $\tilde{f}^{min}_{sup(S)}(q) \leq \tilde{f}^{min}_{sup(S')}(q)$, we have

$$\tilde{f}_{S'}(o) < \tilde{f}_S(o) < \tilde{f}^{min}_{sup(S)}(q) \leq \tilde{f}^{min}_{sup(S')}(q).$$

Thus, $o \in Comp_{S'}(q)$. That is, $Comp_S(q) \subseteq Comp_{S'}(q)$. ∎

Correspondingly, *OAMiner* performs subspace pruning based on the number of competitors.

Pruning Rule 3.2. *When S is met in a depth-first search of the subspace set enumeration tree, let r_{best} be the best rank of q in all the subspaces searched so far. If there are at least r_{best} competitors of q in S, i.e., $|Comp_S(q)| \geq r_{best}$, then all proper super-spaces of S can be pruned.* ∎

Next, we discuss how to compute $\tilde{f}^{min}_{sup(S)}(q)$ when the maximum dimensionality threshold of an outlying aspect, ℓ, is less than $|S| + |R(S)|$. In this situation, $|S| < |S'| \leq \ell < |S| + |R(S)|$. Clearly, it is unsuitable to use $\tilde{f}_{S \cup R(S)}(q)$ as $\tilde{f}^{min}_{sup(S)}(q)$. Intuitively, we can set $\tilde{f}^{min}_{sup(S)}(q)$ to $\min\{\tilde{f}_{S'}(q) \mid |S'| = \ell, S \subset S' \subset S \cup R(S)\}$. However, the computational cost may be high, since the number of candidates is $\binom{|R(S)|}{\ell - |S|}$. Alternatively, we suggest a method to efficiently compute $\tilde{f}^{min}_{sup(S)}(q)$, which uses a lower bound of $\tilde{f}_{S'}(q)$.

For object o', the quasi-density contribution of o' to q in S, denoted by $\tilde{f}_S(q, o')$, is $e^{-\sum\limits_{D_i \in S} \frac{(q.D_i - o'.D_i)^2}{2h^2_{D_i}}}$. Let $R(S, o')$ be the set of $(\ell - |S|)$ dimensions in $R(S)$ with the largest values of $\frac{|q.D_j - o'.D_j|}{h_{D_j}}$ $(D_j \in R(S))$. Then, the minimum quasi-density contribution of o' to q in S' $(S \subset S')$ is $\tilde{f}_{S \cup R(S,o')}(q, o')$. Since $\tilde{f}_{S'}(q) = \sum\limits_{o' \in O} \tilde{f}_{S'}(q, o')$, we have $\tilde{f}^{min}_{sup(S)}(q) = \sum\limits_{o' \in O} \tilde{f}_{S \cup R(S,o')}(q, o') \leq \tilde{f}_{S'}(q)$.

Please note that if we compare $\tilde{f}^{min}_{sup(S)}(q)$ with the quasi-density values of all objects in O, the computational cost for density estimation is considerably high. Especially, when the size of O is large, for the sake of efficiency, we make a tradeoff between subspace pruning and object pruning. Specifically, when we are searching a subspace S, we care more about the relationship between $rank_S(q)$ and r_{best} than the completeness of $Comp_S(q)$. We terminate the search of S as long as we can determine $rank_S(q) > r_{best}$, no matter $Comp_S(q)$ is complete or not.

Algorithm 3.4 gives the pseudo-code of computing outlyingness rank and pruning subspaces in *OAMiner*. Theorem 3.4 guarantees that Algorithm 3.4 can find all minimal outlying subspaces.

Algorithm 3.4 $rank_S(q)$ – *OAMiner*

Input: query object $q \in O$, subspace S, the set of competitors of q discovered in the parent-subspace of S $Comp$ ($Comp$ is empty if $|S| = 1$), and the best rank of q in the subspaces searched so far r_{best}

Output: $rank_S(q)$

```
1: compute f̃_S(q) using Equation 3.7;
2: rank_S(q) ← |Comp| + 1;
3: for each object o ∈ O \ Comp do
4:     if f̃_S(o) < f̃_S(q) then
5:         rank_S(q) ← rank_S(q) + 1;
6:         if f̃_S(o) < f̃^min_sup(S)(q) then
7:             Comp ← Comp ∪ {o};
8:             if |Comp| = r_best then
9:                 prune super-spaces of S and return; // pruning rule 3.2
10:            end if
11:        end if
12:        if rank_S(q) > r_best then
13:            return;
14:        end if
15:    end if
16: end for
17: return rank_S(q);
```

Theorem 3.4 (Completeness of *OAMiner*)**.** *Given a set of objects O in a multi-dimensional space D, a query object $q \in O$ and a maximum dimensionality threshold $0 < \ell \leq |D|$, OAMiner finds all minimal outlying subspaces of q.*

Proof. We prove by contradiction.

Let Ans be the set of minimal outlying subspaces of q found by *OAMiner*, r_{best} the best rank. Assume that subspace $S \notin Ans$ satisfying $S \subseteq D$ and $0 < |S| \leq \ell$ is a minimal outlying subspace of q.

Recall that *OAMiner* searches subspaces by traversing the subspace enumeration tree in a depth-first manner. As $S \notin Ans$, S is pruned by Pruning Rule 3.1 or Pruning Rule 3.2.

In the case that S is pruned by Pruning Rule 3.1, S is not minimal. A contradiction;

In the case that S is pruned by Pruning Rule 3.2, then there exist a subspace S', such that S' is a parent of S in the subspace enumeration tree and $Comp_{S'}(q) \geq r_{best}$. By the property of competitors, we have $Comp_{S'}(q) \subseteq Comp_S(q)$. Correspondingly, $rank_S(q) \geq |Comp_S(q)| \geq |Comp_{S'}(q)| \geq r_{best}$. A contradiction. ∎

3.5 Empirical Evaluation

In this section, we report a systematic empirical study using several real data sets and synthetic data sets to verify the effectiveness and efficiency of our method. All experiments were conducted on a PC with an Intel Core i7-3770 3.40 GHz CPU and 8 GB main memory, running the Windows 7 operating system. The algorithms were implemented in Java and compiled by JDK 7. Since it may likely be too hard for the user to understand the meaning of subspaces with dimensionality more than 5, we set $\ell = 5$ and $\alpha = 1.0$ as default in *OAMiner*.

3.5.1 Mining Outlying Aspects on Real Data Sets

NBA coaches, sport agents, and commentators may want to know in which aspects a player is most unusual. Using this application scenario as a case study, we first investigate the outlying aspects of all NBA guards, forwards and centers in the 2012-2013 Season. We collect the technical statistics on 20 numerical attributes from http://sports.yahoo.com/nba/stats. Table 3.4 shows the names of dimensions. The statistics for centers on 3-points (items 6, 7 and 8) are removed since the statistics for most centers are 0. Besides, we apply *OAMiner* to several real world data sets from the UCI repository [11]. In our experiments, we remove non-numerical attributes and all instances containing missing values. Table 3.5 shows the data characteristics.

For each data set, we take each record as a query object q, and apply *OAMiner* to

1: Game played	6: 3-Points (M)	11: Free throw (Pct)	16: Turnover
2: Minutes	7: 3-Points (A)	12: Rebounds (Off)	17: Steal
3: Field goal (M)	8: 3-Points (Pct)	13: Rebounds (Def)	18: Block
4: Field goal (A)	9: Free throw (M)	14: Rebounds (Tot)	19: Personal foul
5: Field goal (Pct)	10: Free throw (A)	15: Assist	20: Points/game

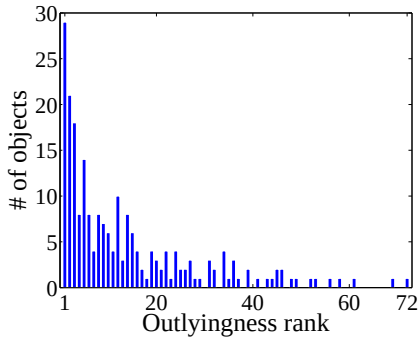
Table 3.4: The 20 technical statistics

Data set	# objects	# attributes
Guards	220	20
Forwards	160	20
Centers	46	17
Breast cancer	194	33
Climate model	540	18
Concrete slump	103	10
Parkinsons	195	22
Wine	178	13

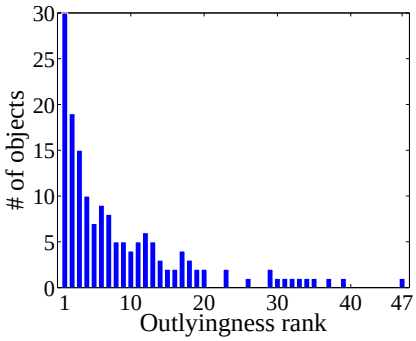
Table 3.5: Data set characteristics

discover the outlying aspects of q. Figure 3.3 shows the distributions of the best outlyingness ranks of objects on the data sets. Surprisingly, the best outlyingness ranks of most objects are small, that is, most objects are ranked very good in outlyingness in some subspaces. For example, 90 guards (40.9%), 81 forwards (50.6%) and 32 centers (69.6%) have an outlyingness rank of 5 or better. Most players have some subspaces where they are substantially different from the others. The observation justifies the need for outlying aspect mining.

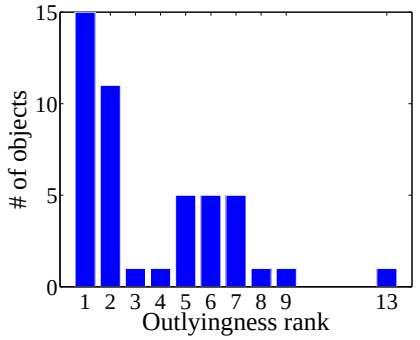
Figure 3.4 shows the distributions of the number of the minimal outlying subspaces where the objects achieve the best outlyingness rank on the data sets. For most objects, the number of outlying aspects is small, which is also surprising. As shown in Figure 3.4(a), 150 (68.2%) objects in Guards have only 1 outlying aspect. This indicates that most objects can be distinguished from the others using a small number of factors.



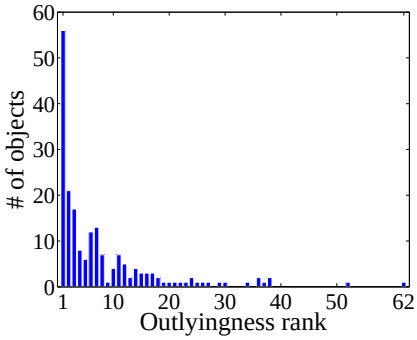
(a) Guards



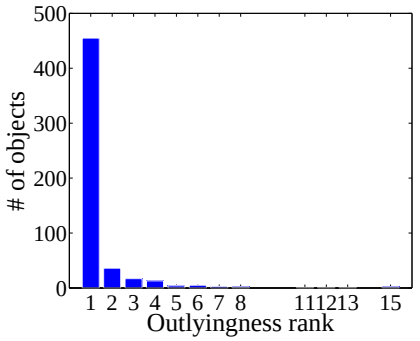
(b) Forwards



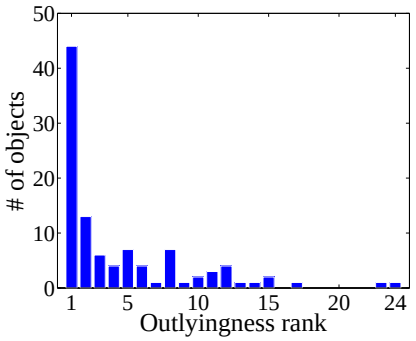
(c) Centers



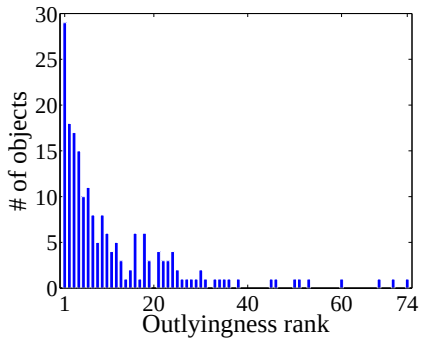
(d) Breast cancer



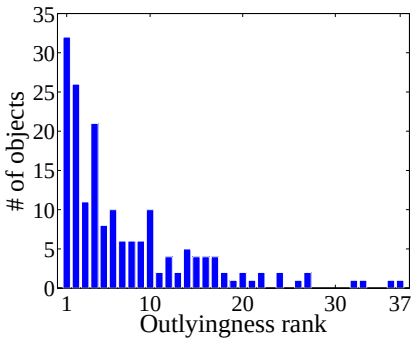
(e) Climate model



(f) Concrete slump

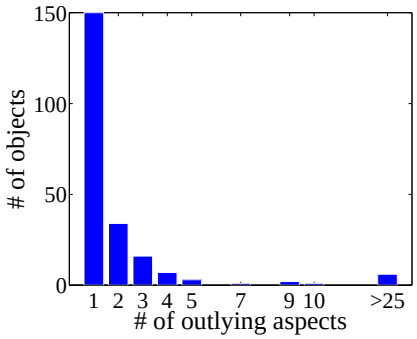


(g) Parkinsons

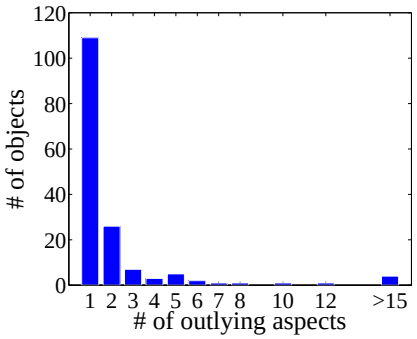


(h) Wine

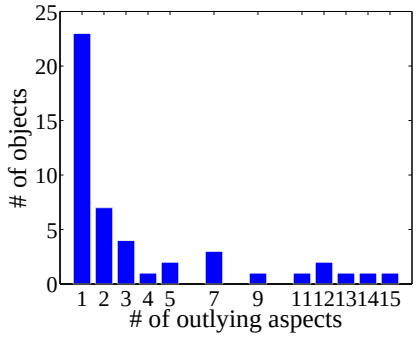
Figure 3.3: Distributions of outlyingness ranks ($\ell = 5$)



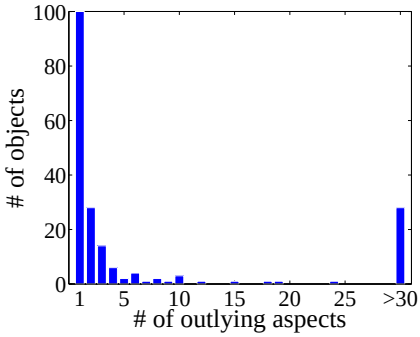
(a) Guards



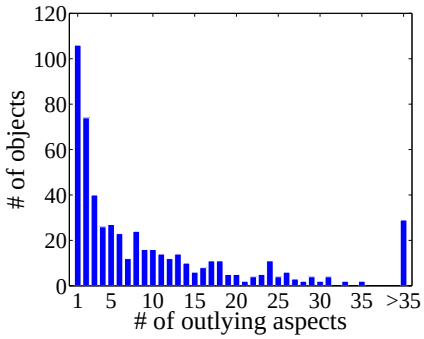
(b) Forwards



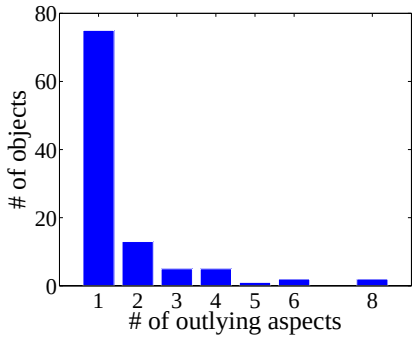
(c) Centers



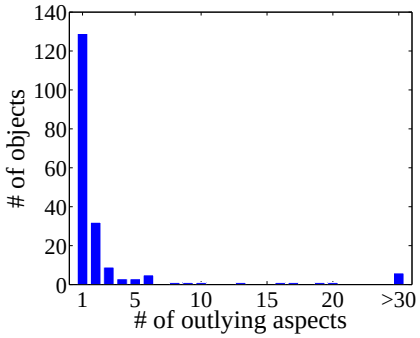
(d) Breast cancer



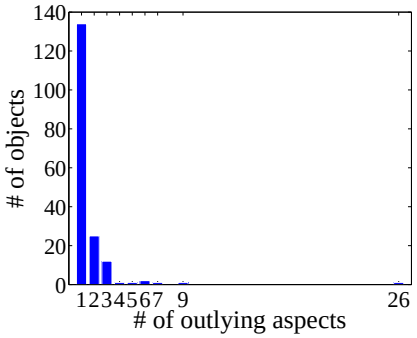
(e) Climate model



(f) Concrete slump



(g) Parkinsons



(h) Wine

Figure 3.4: Distributions of total number of outlying aspects ($\ell = 5$)

Table 3.6 summarizes the mining results of *OAMiner* on real data sets when $\ell = 4, 5, 6$, respectively. Not surprisingly, the smallest values of outlyingness rank, number of outlying aspects, dimensionality are 1. With larger value of ℓ, the average outlyingness rank decreases, while the average number of outlying aspects and the average dimensionality increase. In addition, we can see that more outlying aspects with a higher dimensionality can be found on data sets with more attributes and more instances. For example, the average number of outlying aspects discovered from Breast cancer is the largest.

Data set	ℓ	Outlyingness rank			# of outlying aspects			Dimensionality		
		Min.	Max.	Avg.	Min.	Max.	Avg.	Min.	Max.	Avg.
Guards	4	1	72	13.94	1	49	2.02	1	4	2.79
	5	1	72	13.70	1	111	3.05	1	5	3.68
	6	1	72	13.50	1	359	5.67	1	6	4.83
Forwards	4	1	48	8.79	1	40	2.24	1	4	2.77
	5	1	47	8.54	1	41	2.37	1	5	3.13
	6	1	46	8.43	1	71	2.93	1	6	3.77
Centers	4	1	13	3.70	1	15	3.28	1	4	2.74
	5	1	13	3.57	1	15	3.65	1	5	3.08
	6	1	13	3.54	1	18	3.61	1	6	3.23
Breast cancer	4	1	70	8.04	1	232	9.57	1	4	3.47
	5	1	62	7.74	1	2478	43.37	1	5	4.67
	6	1	56	7.57	1	11681	243.10	1	6	5.77
Climate model	4	1	33	1.97	1	30	4.57	1	4	3.65
	5	1	15	1.45	1	78	10.18	1	5	4.43
	6	1	15	1.28	1	149	16.97	1	6	5.07
Concrete slump	4	1	27	4.67	1	8	1.56	1	4	2.38
	5	1	24	4.44	1	8	1.64	1	5	2.59
	6	1	24	4.41	1	8	1.65	1	6	2.66
Parkinsons	4	1	74	12.13	1	156	4.20	1	4	3.25
	5	1	74	11.51	1	400	7.63	1	5	4.09
	6	1	74	11.33	1	889	14.30	1	6	5.01
Wine	4	1	37	7.65	1	26	1.49	1	4	2.66
	5	1	37	7.47	1	26	1.59	1	5	2.96
	6	1	37	7.46	1	26	1.66	1	6	3.09

Table 3.6: Sensitivity of *OAMiner*'s effectiveness w.r.t. parameter ℓ

3.5.2 Outlying Aspects Discovery on Synthetic Data Sets

[65] provided a collection of synthetic data sets, each consisting 1000 data objects. Each data set contains some subspace outliers, which deviate from all clusters in at least one 2-5 dimensional subspace. As stated in [65], an object can be an outlier in multiple subspaces independently. We perform test on the data sets of 10, 20, 30, 40, 50 dimensions, and denote the data sets by $Synth\_10D$, $Synth\_20D$, $Synth\_30D$, $Synth\_40D$, $Synth\_50D$, respectively.

For an outlier q in a data set, let S be the ground truth about outlying subspace of q. Please note that S may not be an outlying aspect of q if there exists another outlier more outlying than q in S, since *OAMiner* finds the subspaces whereby the query object is most outlying. To verify the effectiveness of *OAMiner* using the known ground truth about outlying subspaces, in the case of multiple implanted outliers in S, we keep q and remove the other outliers, and take q as the query object. Since q is the only implanted strong outlier in subspace S, *OAMiner* is expected to find the ground truth outlying subspace S where q takes rank 1 in outlyingness, that is, $rank_S(q) = 1$.

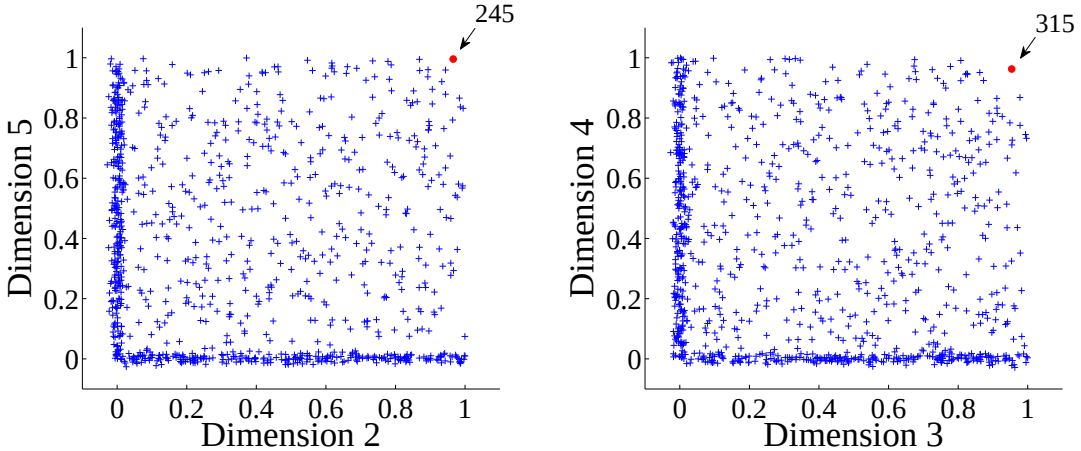
We divide the mining results of *OAMiner* into the following 3 cases:

- Case 1: only the ground truth outlying subspace is discovered by *OAMiner* with outlyingness rank 1.
- Case 2: besides the ground truth outlying subspace, *OAMiner* finds other outlying aspects with outlyingness rank 1.
- Case 3: instead of the ground truth outlying subspace, *OAMiner* finds a subset of the ground truth as an outlying aspect with outlyingness rank 1.

Table 3.7 lists the mining results on $Synth\_10D$. (Note that the object ids and dimension ids in Tables 3.7 and 3.8 are consistent with the original data sets in [65].) For all outliers (query objects), outlying aspects with outlyingness rank 1 are discovered. Moreover, we can see that for objects 183, 315, 577, 704, 754, 765 and 975, *OAMiner* finds not only the ground truth outlying subspace, but also some other outlying subspaces (Case 2). For object 245, the outlying aspect discovered by *OAMiner* is a subset of the ground truth outlying subspace (Case 3). For the other 11 objects, the outlying aspects discovered by *OAMiner* are identical with the ground truth outlying subspaces (Case 1).

Query object	Ground truth outlying subspace	Outlying aspect with outlyingness rank 1	Description
172	$\{8,9\}$	$\{8,9\}$	Case 1
183	$\{0,1\}$	$\{0,1\},\{0,6,8\}$	Case 2
184	$\{6,7\}$	$\{6,7\}$	Case 1
207	$\{0,1\}$	$\{0,1\}$	Case 1
220	$\{2,3,4,5\}$	$\{2,3,4,5\}$	Case 1
245	$\{2,3,4,5\}$	$\{2,5\}$	Case 3
315	$\{0,1\},\{6,7\}$	$\{0,1\},\{6,7\},\{3,4\},\{3,5,9\},\{4,6,9\}$	Case 2
323	$\{8,9\}$	$\{8,9\}$	Case 1
477	$\{0,1\}$	$\{0,1\}$	Case 1
510	$\{0,1\}$	$\{0,1\}$	Case 1
577	$\{2,3,4,5\}$	$\{2,3,4,5\},\{0,3,7\}$	Case 2
654	$\{2,3,4,5\}$	$\{2,3,4,5\}$	Case 1
704	$\{8,9\}$	$\{8,9\},\{0,2,3,4\}$	Case 2
723	$\{2,3,4,5\}$	$\{2,3,4,5\}$	Case 1
754	$\{6,7\}$	$\{6,7\},\{2,4,8\},\{2,6,8\},\{4,6,8\}$	Case 2
765	$\{6,7\}$	$\{6,7\},\{1,4,6\},\{3,4,5,6\}$	Case 2
781	$\{6,7\}$	$\{6,7\}$	Case 1
824	$\{8,9\}$	$\{8,9\}$	Case 1
975	$\{8,9\}$	$\{8,9\},\{2,5,9\},\{5,6,8\},\{2,3,5,8\}$	Case 2

Table 3.7: Outlying aspects on *Synth_10D*



(a) Object 245 (the solid circle •)

(b) Object 315 (the solid circle •)

Figure 3.5: Outlying aspects for objects 245 and 315 in *Synth_10D*

To further demonstrate the effectiveness of *OAMiner*, for object 245 in Case 2, we illustrate the outlying aspect $\{2, 5\}$ in Figure 3.5(a), and for object 315 in Case 3, we illustrate the outlying aspect $\{3, 4\}$ in Figure 3.5(b). Visually, the objects show outlying characteristics in the corresponding outlying aspects.

Table 3.8 summarizes the mining results of *OAMiner* on the synthetic data sets of 10, 20, 30, 40, 50 dimensions. As *OAMiner* finds all subspaces in which the outlyingness rank of the query object are the minimum, we can see that the number of Case 2 increases with higher dimensionality. In other words, more outlying aspects can be found on data sets with more attributes. Please note that this observation is consistent with the experimental observations in real data sets (Section 3.5.1). In addition, the number of Case 3 increases a bit, since *OAMiner* applies the dimensionality minimality condition to outlying aspect mining.

Data set	# of outliers	# of Case 1	# of Case 2	# of Case 3
$Synth\_10D$	19	11	7	1
$Synth\_20D$	25	1	23	1
$Synth\_30D$	44	0	40	4
$Synth\_40D$	53	0	52	1
$Synth\_50D$	68	0	65	3

Table 3.8: Statistics on the mining results of *OAMiner* on synthetic data sets

3.5.3 Outlying Aspects Discovery on NBA Data Sets

As a real case study, we verified the usefulness of outlying aspect mining by analyzing the outlying aspects of some NBA players.

Please note that "outlying" is different from "outstanding". A player receives a good outlyingness rank in a subspace if very few other players are close to him in the subspace, regardless of whether the performance is "good" or not. Table 3.9 lists 10 guards who have the largest number of rank-1 outlying aspects, where the dimensions are represented by their serial numbers in Table 3.4. Note that Table 3.9 only lists the outlying aspects whose dimensionality are not greater than 3.

In Table 3.9, the first several players are not well-known. Their low outlyingness ranks arise due to no other players having similar statistics. For example, Quentin Richardson, who has 18 outlying aspects, just played one game in which he played very well at rebounds,

Name	Outlying aspects ($\ell = 3$)
Quentin Richardson	$\{1\}$, $\{12\}$, $\{14\}$, $\{2,17\}$, $\{3,4\}$, $\{3,13\}$, $\{4,17\}$, $\{5,8\}$, $\{5,11\}$, $\{5,13\}$, $\{13,17\}$, $\{13,20\}$, $\{2,3,16\}$, $\{2,4,5\}$, $\{2,5,6\}$, $\{2,5,7\}$, $\{2,5,9\}$, $\{4,5,7\}$
Will Conroy	$\{2,5\}$, $\{5,8\}$, $\{5,11\}$, $\{5,12\}$, $\{5,13\}$, $\{5,14\}$, $\{5,16\}$, $\{4,5,6\}$, $\{4,5,9\}$, $\{4,5,10\}$, $\{4,5,7\}$, $\{4,5,19\}$, $\{5,6,7\}$, $\{5,7,9\}$
Brandon Rush	$\{5\}$, $\{1,19\}$, $\{2,19\}$, $\{17,19\}$
Ricky Rubio	$\{3,17\}$, $\{7,17\}$, $\{16,17\}$, $\{17,20\}$
Rajon Rondo	$\{15\}$, $\{16\}$, $\{1,17\}$, $\{1,2,20\}$
Scott Machado	$\{19\}$, $\{2,16\}$, $\{5,8,18\}$
Kobe Bryant	$\{3\}$, $\{4\}$, $\{20\}$
Jamal Crawford	$\{19,20\}$, $\{4,19\}$, $\{2,3,19\}$
James Harden	$\{9\}$, $\{10\}$
Stephen Curry	$\{6\}$, $\{7\}$

Table 3.9: The guards having the most rank-1 outlying aspects

but poor at field goal. Will Conroy played four games and his performance on shooting is poor. Brandon Rush played two games, and his number of personal fouls is large. Ricky Rubio performs well at stealing. Rajon Rondo's ability to assist is impressive, but his statistics for turnover is large. Scott Machado did not make any personal foul in the six games he played. The last four players in Table 3.9 are famous. Their overall performance on every aspect is much better than most of the other guards. For example, Kobe Bryant is a great scorer, Jamal Crawford's personal fouls are very low, James Harden is excellent at the free throw, and Stephen Curry leads in 3-points scoring.

Please note that different objects may share some outlying aspects with the same outlyingness rank. For example, both Quentin Richardson and Will Conroy are ranked number 1 in $\{5,8\}$. There are two reasons for this situation. First, the values of objects are identical in these subspaces. Second, the difference between the outlyingness degrees is so tiny that it is beyond the precision of the program.

Table 3.10 lists the guards who have poor outlyingness ranks overall (i.e. there are not any subspaces where they are ranked particularly well). Their performance statistics is in the middle of the road, and do not have any obvious shortcomings. They may be important to be included in a team as "the sixth man", even though they are not star performers.

As mentioned in Section 2.3.1 of Chapter 2, subspace outlier detection is fundamentally different from outlying aspect mining, since subspace outlier detection finds contrast

Outlyingness rank	Name	Outlying aspects
72	Terrence Ross	{11}
70	E'Twaun Moore	{18}
69	C.J. Watson	{8, 12, 13, 14, 18}
61	Jerryd Bayless	{2, 3, 4, 19, 20}
58	Nando De Colo	{1, 2}, {3, 4, 5, 11, 20}
56	Alec Burks	{2, 9, 10, 11}
55	Rodrigue Beaubois	{1, 2, 8, 11, 15}
52	Marco Belinelli	{9, 10, 12}
49	Aaron Brooks	{2, 3, 5, 7, 16}
48	Nick Young	{1, 3, 16, 18, 20}

Table 3.10: The guards having poor ranks in outlying aspects

Position	Name	$rank_{HL}$	$rank_{SOD}$	$rank_S$ (# of outlying aspects)
guard	Quentin Richardson	1	1	1 (54)
	Kobe Bryant	1	9	1 (3)
	Brandon Roy	32	1	1 (4)
forward	Carmelo Anthony	1	5	1 (26)
	Kevin Love	3	1	1 (41)
center	Dwight Howard	1	2	1 (15)
	Andrew Bogut	10	1	1 (9)

Table 3.11: The outlyingness ranks of players ranked top in *HiCS* or *SOD*

subspaces for all possible outliers. However, we can make use of the results of subspace outlier ranking to verify to some extent our discovered outlying aspects. Specifically, we look at the objects that are ranked the best by either *HiCS* [65] or *SOD* [67], and check their outlyingness ranks. As *HiCS* randomly selects subspace slices, we run it 3 times independently on each data set with the default parameters. The parameter for the number of nearest neighbors in both *LOF* and *SOD* was varied across 5, 10 and 20, and the best ranks were reported. In *SOD* [67], the parameter l specifying the size of the reference sets cannot be larger than the number of nearest neighbors. We set it to the number of nearest neighbors. For a given object, we denote by $rank_{HL}$ and $rank_{SOD}$ the ranks computed by *HiCS* and by *SOD*, respectively. We denote by $rank_S$ the outlyingness rank computed by *OAMiner*. Table 3.11 shows the results. The results clearly show that every player ranked top in either *HiCS* or *SOD* has some outlying subspaces where he is ranked number 1. The results of outlying aspect mining are consistent with those of subspace outlier ranking. At the same time, we notice that the rankings of *HiCS* and *SOD* are not always consistent with each other, such as for Kobe Bryant, Brandon Roy and Andrew Bogut.

3.5.4 Efficiency

To the best of our knowledge, there is no other method tackling the exact same problem as *OAMiner*. Therefore, we only evaluate the efficiency of *OAMiner* and its variations. Specifically, we implemented the baseline method (Algorithm 3.1 with Pruning Rule 3.1). Recall that *OAMiner* uses both upper and lower bounds of quasi-density to speed up the computation of outlyingness ranks. To evaluate the efficiency of our techniques for quasi-density comparison, we implemented a version *OAMiner*-part that does not use bounds in quasi-density estimation and strategies presented in Section 3.4.2. Moreover, we implement the full version *OAMiner*-full that uses all techniques.

Once again, we used a synthetic data set from [65]. The dimensionality of the data set is 50, and the data set consists of 1000 data points. We randomly chose 10 data points (non-outliers) from the data set as query objects, and reported the average runtime. Again, we set $\ell = 5$ for all three methods and $\alpha = 1.0$ for *OAMiner*-full by default.

Figure 3.6 shows the runtime with respect to data set size. The runtime is plotted using the logarithmic scale. The baseline method is time consuming, which is consistent with our analysis. Our pruning techniques can achieve a roughly linear runtime in practice. Both versions of *OAMiner* are substantially faster than the baseline method. Moreover, *OAMiner*-full is more efficient than *OAMiner*-part.

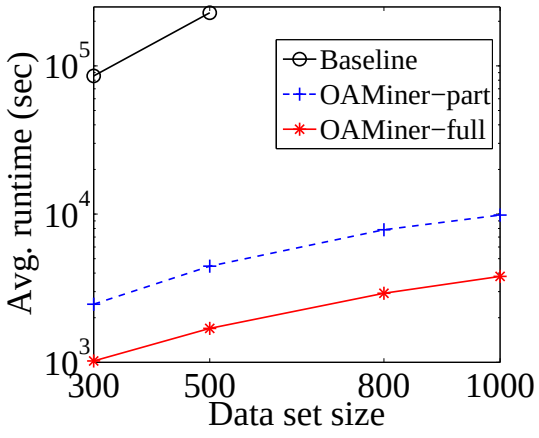


Figure 3.6: Runtime w.r.t. data set size.

Figure 3.7 shows the runtime with respect to dimensionality. The runtime is also plotted using the logarithmic scale. As dimensionality increases, the runtime increases

exponentially. However, our heuristic pruning techniques speed up the search in practice. Again, *OAMiner*-full is more efficient than *OAMiner*-part.

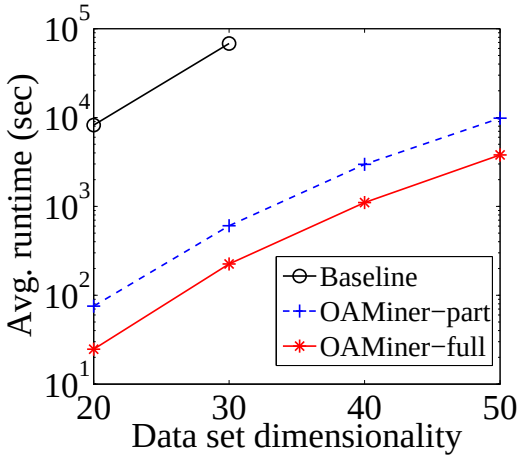


Figure 3.7: Runtime w.r.t. data set dimensionality.

Figure 3.8 shows the runtime with respect to maximum dimensionality threshold (ℓ). The runtime is plotted using the logarithmic scale, too. As ℓ increases, more subspaces will be enumerated. Correspondingly, the runtime increases. Once more, both versions of *OAMiner* are considerably faster than the baseline method, and *OAMiner*-full is more efficient than *OAMiner*-part.

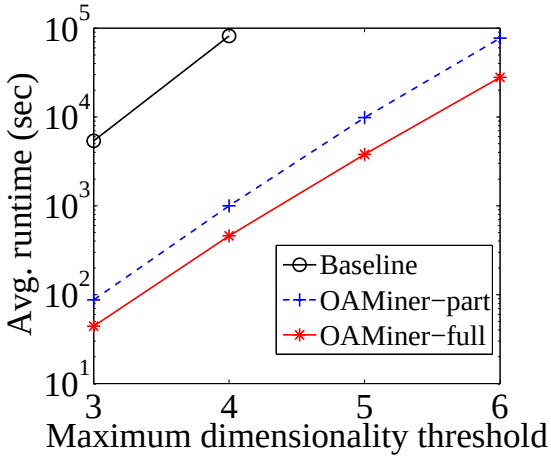
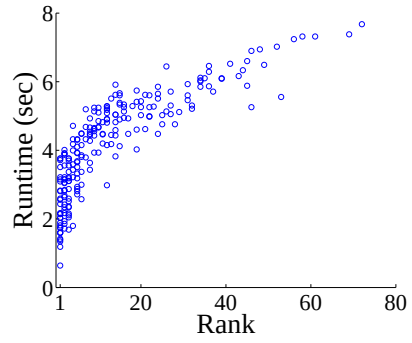


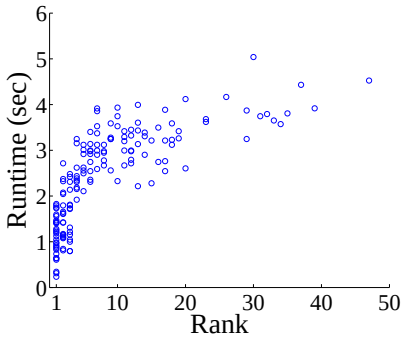
Figure 3.8: Runtime w.r.t. ℓ.

Figure 3.6, Figure 3.7 and Figure 3.8 show the necessity of our pruning techniques. Even when ℓ is a relatively small constant, exhaustive search (the baseline method) is extremely time consuming due to the expensive computational costs of the quasi-density estimation for data objects.

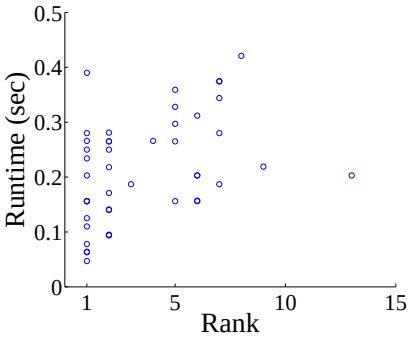
We also notice that the runtime of *OAMiner* is related with the outlyingness rank of the query object. Figure 3.9 shows the runtime with respect to outlyingness rank on each real data set. Not surprisingly, the objects with large outlyingness rank cost more runtime, since *OAMiner* prunes subspaces based on the rank of the query object by either Pruning Rule 3.1 or Pruning Rule 3.2.



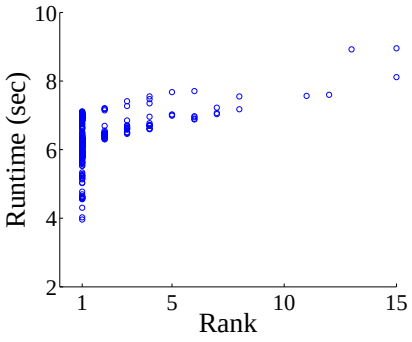
(a) Guards



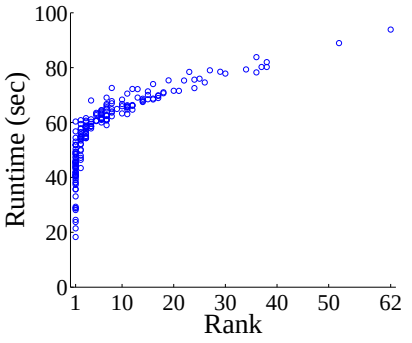
(b) Forwards



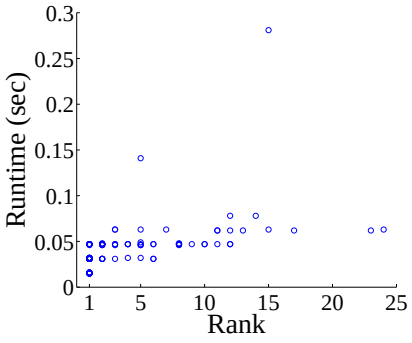
(c) Centers



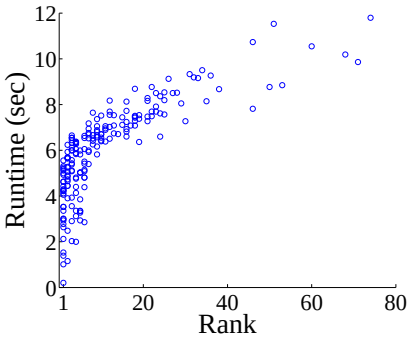
(d) Climate model



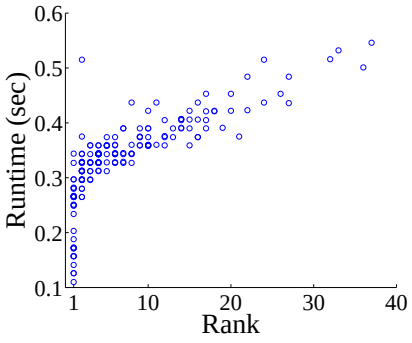
(e) Breast cancer



(f) Concrete slump



(g) Parkinsons



(h) Wine

Figure 3.9: Runtime w.r.t. outlyingness rank

Last, we test the sensitivity of the parameter α for bounding quasi-density. We vary the parameter α, which sets the ϵ-neighborhood distance. Table 3.12 lists the average runtime of *OAMiner* for a query object on each real data set. The runtime of *OAMiner* is not sensitive to α in general. Experimentally, the shortest runtime of *OAMiner* (bold values in Table 3.12) happens when α is in $[0.8, 1.2]$.

Data set	Average runtime (sec)				
	$\alpha = 0.6$	$\alpha = 0.8$	$\alpha = 1.0$	$\alpha = 1.2$	$\alpha = 1.4$
Guards	4.459	4.234	**4.213**	4.303	4.315
Forwards	2.810	2.519	2.424	2.418	**2.413**
Centers	0.260	0.234	0.216	**0.212**	0.220
Breast cancer	58.476	58.228	57.927	**57.613**	57.982
Climate model	6.334	6.268	6.339	**6.253**	6.410
Concrete slump	0.047	**0.044**	0.044	0.045	0.045
Parkinsons	6.164	6.154	**6.083**	6.218	6.243
Wine	0.351	0.341	**0.339**	0.344	0.350

Table 3.12: Average runtime of *OAMiner* w.r.t parameter α

3.6 Conclusions

In this chapter, we studied the novel and interesting problem of finding outlying aspects of a query object on multidimensional numeric data. We systematically developed a model and a heuristic method. Using both real and synthetic data sets, we verified that mining outlying aspects is interesting and useful. Moreover, our experiments show that our outlying aspect mining method is effective and efficient.

Chapter 4

Mining Contrast Subspaces

In this chapter, we tackle the novel problem of mining contrast subspaces.

Given a set of multidimensional objects in two classes C_+ and C_- and a query object o, we want to find the top-k subspaces that maximize the ratio of likelihood of o in C_+ against that in C_-. Such subspaces are very useful for characterizing an object and explaining how it differs between two classes. We demonstrate that this problem has important applications, and, at the same time, is very challenging, being MAX SNP-hard. We present *CSMiner*, a mining method that uses kernel density estimation in conjunction with various pruning techniques. We experimentally investigate the performance of *CSMiner* on a range of datasets, evaluating its efficiency, effectiveness and stability and demonstrating it is substantially faster than a baseline method.

4.1 Motivation

Mining contrast subspaces is an interesting problem with important applications.

Besides the application Scenario 2 of the Example 1.1, another example is, when an analyst in an insurance company is investigating a suspicious claim, she may want to compare this suspicious case against samples of frauds and normal claims. A useful question to ask is "In what aspects is this suspicious case most similar to fraudulent cases and different from normal claims?". In other words, finding the contrast subspaces for the suspicious claim is informative for the analyst and serves as a useful input for deeper exploration.

While there are many existing studies on outlier detection and contrast mining, they focus on collective patterns that are shared by many cases of the target class. The contrast

subspace mining problem addressed here is different. It focuses on one query object and finds the customized contrast subspaces. This critical difference makes the problem formulation, the suitable applications, and the mining methods rather different.

To tackle the problem of mining contrast subspaces, we need to address several technical challenges.

First, we need to have a simple yet informative contrast measure to quantify the similarity between the query object and the target class and the difference between the query object and the other class. In this chapter, we use the ratio of the likelihood of the query object in the target class against that in the other class as the contrast measure. This is essentially the Bayes factor on the query object, and comes with a well recognized explanation [62].

Second, the problem of mining contrast subspaces is computationally challenging. Exhaustive search, which enumerates every non-empty subspace and computes the contrast measure, is very costly on data sets with a non-trivial dimensionality. We show that the problem of contrast subspace mining is MAX SNP-hard, and thus does not allow polynomial time approximation methods unless P=NP. Therefore, the only hope is to develop heuristics that may work well in practice.

Third, one might attempt a brute-force method to tackle the contrast mining problem. One major obstacle preventing effective pruning is that the contrast measure does not have any monotonicity with respect to the subspace-superspace relationship. To tackle the problem, we develop pruning techniques based on bounds of likelihood and contrast ratio. Our experimental results on real data sets clearly verify the effectiveness, stability and efficiency of our method.

The rest of this chapter is organized as follows. In Section 4.2, we formalize the problem, and analyze it theoretically. We present a heuristic method in Section 4.3, and evaluate our method empirically using real data sets in Section 4.4. We summarize the chapter in Section 4.5.

4.2 Problem Formulation and Analysis

In this section, we first formulate the problem. Then, we recall the basics of kernel density estimation for estimating the probability density of objects. Last, we investigate the complexity of the problem.

4.2.1 Problem Definition

Let $D = \{D_1, \ldots, D_d\}$ be a d-dimensional space, where the domain of D_i is $\mathbb{R}$, the set of real numbers. A *subspace* $S \subseteq D$ ($S \neq \emptyset$) is a subset of D. We also call D the *full space*.

Consider an object o in space D. We denote by $o.D_i$ the value of o in dimension D_i ($1 \leq i \leq d$). For a subspace $S = \{D_{i_1}, \ldots, D_{i_l}\} \subseteq D$, the *projection* of o in S is $o^S = (o.D_{i_1}, \ldots, o.D_{i_l})$. For a set of objects $O = \{o_j \mid 1 \leq j \leq n\}$, the *projection* of O in S is $O^S = \{o_j^S \mid o_j \in O, 1 \leq j \leq n\}$.

Given a set of objects O, we assume a latent distribution $\mathcal{Z}$ that generates the objects in O. For a query object q, denote by $L_D(q \mid \mathcal{Z})$ the likelihood of q being generated by $\mathcal{Z}$ in full space D. The posterior probability of q given O, denoted by $L_D(q \mid O)$, can be estimated by $L_D(q \mid \mathcal{Z})$. For a non-empty subspace S ($S \subseteq D$, $S \neq \emptyset$), denote by $\mathcal{Z}^S$ the projection of $\mathcal{Z}$ in S. The *subspace likelihood* of object q with respect to $\mathcal{Z}$ in S, denoted by $L_S(q \mid \mathcal{Z})$, can be used to estimate the posterior probability of object q given O in S, denoted by $L_S(q \mid O)$.

In this study, we assume that the objects in O belong to two classes, C_+ and C_-, exclusively in full space D. Thus, $O = O_+ \cup O_-$ and $O_+ \cap O_- = \emptyset$, where O_+ and O_- are the subsets of objects of O belonging to C_+ and C_-, respectively. Given a query object q, we are interested in how likely q belongs to C_+ and does not belong to C_-. To measure these two factors comprehensively, we define the *likelihood contrast* as $LC(q) = \frac{L_D(q|O_+)}{L_D(q|O_-)}$.

Generally, given a set of observations Q, the plausibility of two models M_1 and M_2 can be assessed by the Bayes factor $K = \frac{Pr(Q|M_1)}{Pr(Q|M_2)}$. Likelihood contrast is essentially the Bayes factor of object q as the observation. In other words, we can regard O_+ and O_- as representing two models, and we need to choose one of them based on query object q. Consequently, the ratio of likelihoods indicates the plausibility of model represented by O_+ against that by O_-. Jeffreys [62] gave a scale for interpretation of Bayes factor. When $LC(q)$ is in the ranges of < 1, 1 to 3, 3 to 10, 10 to 30, 30 to 100, and over 100, respectively, the strength of the evidence is negative, barely worth mentioning, substantial, strong, very strong, and decisive.

We can extend likelihood contrast to subspaces. For a non-empty subspace $S \subseteq D$, we define the likelihood contrast in the subspace as $LC_S(q) = \frac{L_S(q|O_+)}{L_S(q|O_-)}$. To avoid triviality in subspaces where $L_S(q \mid O_+)$ is very small, we introduce a minimum likelihood threshold $\delta > 0$, and consider only the subspaces S where $L_S(q \mid O_+) \geq \delta$. The number of likelihood

contrast subspaces will be reduced with larger δ.

Now, we formally define the problem. Given a multidimensional data set O in full space D, a query object q, a minimum likelihood threshold $\delta > 0$, and a parameter $k > 0$, the *problem of mining contrast subspaces* is to find the top-k subspaces S ordered by the subspace likelihood contrast $LC_S(q)$ subject to $L_S(q \mid O_+) \geq \delta$.

Table 4.1 lists the frequently used notations in this chapter.

Notation	Description
$D = \{D_1, \ldots, D_d\}$	a d-dimensional space
S	a subspace, $S \subseteq D$ $(S \neq \emptyset)$
O	a set of objects in space D
$o.D_i$	the value of data object o in dimension $D_i (1 \leq i \leq d)$
q	a query object
$L_S(q \mid O)$	the subspace likelihood of object q with respect to O in S
O_+ (or O_-)	objects that belong to class C_+ (or C_-)
$LC_S(q)$	the subspace likelihood contrast of object q with respect to O in subspace S, $LC_S(q) = \frac{L_S(q\|O_+)}{L_S(q\|O_-)}$
$\hat{f}_S(q, O)$	the density of a query object q in subspace S
h_S	the bandwidth of subspace S
$dist_S(q, o)$	the distance between objects q and o in subspace S

Table 4.1: Summary of frequently used notations in Chapter 4

4.2.2 Kernel Density Estimation

We can use kernel density estimation to estimate the likelihood $L_S(q \mid O)$. Given a set of objects O, we denote by $\hat{f}_S(q, O)$ the density of a query object q in subspace S. Following [105], the general formula for multivariate kernel density estimation with kernel K and bandwidth parameter h_S in subspace S is defined as follows

$$\hat{f}_S(q, O) = \hat{f}_S(q^S, O) = \frac{1}{|O| h_S^{|S|}} \sum_{o \in O} K\{\frac{1}{h_S}(q - o)\} \tag{4.1}$$

Choosing K to be a radially symmetric unimodal probability density function, in this

study, we adopt the Gaussian kernel

$$K(x) = \frac{1}{(2\pi)^{|S|/2}} e^{-\frac{1}{2}x^T x} \tag{4.2}$$

which is natural and widely used in density estimation. Note that if it was not unimodal, then there could be multiple peaks at different distances from the query, which is counter to intuition. Similarly, we have no basis for preferring any direction over another, so symmetry is natural. This is a common requirement for kernel density functions [102, 105].

This then leads to

$$\hat{f}_S(q, O) = \hat{f}_S(q^S, O) = \frac{1}{|O|(\sqrt{2\pi}h_S)^{|S|}} \sum_{o \in O} e^{\frac{-dist_S(q,o)^2}{2{h_S}^2}}$$

where $dist_S(q, o)^2 = \sum_{D_i \in S} (q.D_i - o.D_i)^2$.

Silverman [105] suggested that the optimal bandwidth value for smoothing normally distributed data with unit variance is $h_{S\_opt} = A(K)|O|^{-1/(|S|+4)}$, where $A(K) = \{4/(|S| + 2)\}^{1/(|S|+4)}$ for the Gaussian kernel.

As the kernel is radially symmetric and the data is not normalized in subspaces, we can use a single scale parameter σ_S in subspace S and set $h_S = \sigma_S \cdot h_{S\_opt}$. As Silverman [105] suggested, a reasonable choice for σ_S is the root of the average marginal variance in S.

Using kernel density estimation, we can estimate $L_S(q \mid O)$ as

$$L_S(q \mid O) = \hat{f}_S(q, O) = \frac{1}{|O|(\sqrt{2\pi}h_S)^{|S|}} \sum_{o \in O} e^{\frac{-dist_S(q,o)^2}{2{h_S}^2}} \tag{4.3}$$

Correspondingly, the likelihood contrast of object q in subspace S is given by

$$LC_S(q, O_+, O_-) = \frac{\hat{f}_S(q, O_+)}{\hat{f}_S(q, O_-)} = \frac{|O_-|}{|O_+|} \cdot \left(\frac{h_{S_-}}{h_{S_+}}\right)^{|S|} \cdot \frac{\sum_{o \in O_+} e^{\frac{-dist_S(q,o)^2}{2{h_{S_+}}^2}}}{\sum_{o \in O_-} e^{\frac{-dist_S(q,o)^2}{2{h_{S_-}}^2}}} \tag{4.4}$$

We often omit O_+ and O_- and write $LC_S(q)$ if O_+ and O_- are clear from context.

4.2.3 Complexity Analysis

Before developing any algorithms to tackle the contrast subspace mining problem, let us first investigate its complexity. We will show that the contrast subspace mining problem is MAX

SNP-hard by constructing a linear-reduction (L-reduction for short) from the emerging pattern mining problem [36], which was been shown to be MAX SNP-hard [113]. The L-reduction linearly preserves approximability features of the original problem after the transformation, thus the name "linear reduction".

To make the discussion self-contained, a brief description of the emerging pattern mining problem is given as follows. Let $D' = \{D'_1, D'_2, \ldots, D'_d\}$ denote a set of d items. A transaction o'_i is represented by a binary vector of length d whose element $o'_{ij} = 1$ if item D'_j is present, and 0 otherwise. A pattern S' is a subset of items in D'. A transaction o'_i satisfies S' if $o'_{ij} = 1,\ \forall D'_j \in S'$. A transaction database O' is a set of transactions. Let $\mathrm{Sat}_{O'}(S')$ denote the set of transactions in O' satisfying S'.

Definition 4.1 (Emerging Pattern Mining (EP))**.** *Given two transaction databases O'_+ and O'_-, find the pattern S' such that the cost function $c_{\boldsymbol{EP}}(S') = |Sat_{O'_+}(S')|$ is maximized subject to the feasibility condition $|Sat_{O'_-}(S')| = 0$.* ■

We consider the following simplified version of the contrast subspace mining problem, where the bandwidth parameters h_{S_+} and h_{S_-} for all subspaces are set to the same value h.

Definition 4.2 (Contrast Subspace Mining (CS))**.** *Given $\{q, O_+, O_-\}$ where q is the query and O_+ and O_- are the two classes, find the subspace S maximizing the cost function*

$$c_{\boldsymbol{CS}}(S, q) = \sum_{o \in O_+} \exp\left(\frac{-dist_S(q,o)^2}{2h^2}\right) / \sum_{o \in O_-} \exp\left(\frac{-dist_S(q,o)^2}{2h^2}\right)$$

(which is equivalent to the likelihood contrast, up to a constant multiplicative factor $\frac{|O_-|}{|O_+|}$). ■

In addition, we define the *complete contrast subspace mining problem* as follows:

Definition 4.3 (Complete Contrast Subspace Mining (Complete-CS))**.** *Given $\{O_+, O_-\}$ find the subspace S such that the cost function*

$$c(S) = \max_{o_i \in O_+} c_{\boldsymbol{CS}}(S, q = o_i)$$

is maximized. ■

It can be seen that Complete-**CS** can be solved by solving at most $|O_+|$ **CS** sub-problems corresponding to unique data points in O_+. We will now prove that Complete-**CS** is MAX SNP-hard, via the following reduction from the emerging pattern mining problem.

Reduction 1. *The **EP** $\rightarrow$ Complete-**CS** reduction:*

- *For each item D'_i, set up a corresponding dimension D_i.*
- *For each transaction $o'_i \in O'_+$, insert 2 copies of o'_i into O_+.*
- *For each transaction $o'_i \in O'_-$, insert $2|O'_+|$ identical data points o'_i into O_-.*
- *Insert 1 item (a numeric vector) with all 1's into O_-.*
- *Let h be an arbitrary user-specified bandwidth parameter, replace each occurrence of the 0 value in $O = O_+ \cup O_-$ with a unique value in the set $\{2\gamma h, 3\gamma h, 4\gamma h \ldots\}$ where γ is some fixed large constant.*
- *Replace each occurrence of the value 1 in O with γh where γ is the same as the one used above.*

■

This transformation can be done in $\mathcal{O}(|O_+||O_-|)$ time. An example illustrating the transformation is given in Table 4.2.

Database	Transactions	O_+	O_-
O'_+	$[0,1,1,0]$	$[2\gamma h, 1\gamma h, 1\gamma h, 3\gamma h]$ $[4\gamma h, 1\gamma h, 1\gamma h, 5\gamma h]$	
	$[0,1,0,0]$	$[6\gamma h, 1\gamma h, 7\gamma h, 8\gamma h]$ $[9\gamma h, 1\gamma h, 10\gamma h, 11\gamma h]$	
O'_-	$[1,1,0,0]$		$[1\gamma h, 1\gamma h, 12\gamma h, 13\gamma h]$ $[1\gamma h, 1\gamma h, 14\gamma h, 15\gamma h]$ $[1\gamma h, 1\gamma h, 16\gamma h, 17\gamma h]$ $[1\gamma h, 1\gamma h, 18\gamma h, 19\gamma h]$
	$[0,0,0,1]$		$[20\gamma h, 21\gamma h, 22\gamma h, 1\gamma h]$ $[23\gamma h, 24\gamma h, 25\gamma h, 1\gamma h]$ $[26\gamma h, 27\gamma h, 28\gamma h, 1\gamma h]$ $[29\gamma h, 30\gamma h, 31\gamma h, 1\gamma h]$
			$[1\gamma h, 1\gamma h, 1\gamma h, 1\gamma h]$

Table 4.2: An example transformation from a transaction database to a numeric dataset according to the **EP** $\rightarrow$ Complete-**CS** reduction

Theorem 4.1. *The reduction **EP** $\rightarrow$ Complete-**CS** defined above is an L-reduction, denoted by **EP** $\rightarrow_L$ Complete-**CS**.*

For completeness, the formal definition of the L-reduction [89] is given as follows:

Definition 4.4 (L-Reduction)**.** *Let $\mathbf{\Pi}_1$ and $\mathbf{\Pi}_2$ be two optimization problems. We say that $\mathbf{\Pi}_1$ L-reduces to $\mathbf{\Pi}_2$ if there are two polynomial time algorithms f, g and constants $\alpha, \beta > 0$ such that, for any instance I of $\mathbf{\Pi}_1$, $f(I)$ forms an instance of $\mathbf{\Pi}_2$ and*

- ***c.1*** *$OPT(f(I)) \leq \alpha OPT(I)$ where $OPT(.)$ denotes the optimal value of the respective optimization problem.*
- ***c.2*** *Given any solution s of $f(I)$, algorithm g produces a solution $g(s)$ of I satisfying $|c_{\mathbf{\Pi}_1}(g(s)) - OPT(I)| \leq \beta |c_{\mathbf{\Pi}_2}(s) - OPT(f(I))|$, where $c_{\mathbf{\Pi}_i}(.)$ denotes the cost function of the corresponding optimization problem.*

Proof. First, we note that for any bandwidth value h, we can set γ to a large value such that $\exp\left(\frac{-\text{dist}_S(q,o)^2}{2h^2}\right)$ can be arbitrarily close to 0 for all $q \in O$ such that $q^S \neq o^S$. The cost function for **CS** can be computed as

$$c_{\mathbf{CS}}(S, q) = \frac{\sum_{o \in O_+} \exp\left(\frac{-\text{dist}_S(q,o)^2}{2h^2}\right)}{\sum_{o \in O_-} \exp\left(\frac{-\text{dist}_S(q,o)^2}{2h^2}\right)} = \frac{|O_+^{S,q}| + \epsilon_+(S, q)}{|O_-^{S,q}| + \epsilon_-(S, q)} \tag{4.5}$$

where $O^{S,q}$ denotes the set of data points in O having values identical to q in the subspace S, and

$$\epsilon_+(S, q) = \sum_{o \in O_+ \setminus O_+^{S,q}} \exp\left(\frac{-\text{dist}_S(q, o)^2}{2h^2}\right),$$

$$\epsilon_-(S, q) = \sum_{o \in O_- \setminus O_-^{S,q}} \exp\left(\frac{-\text{dist}_S(q, o)^2}{2h^2}\right).$$

Let $M > 1$ be the maximum integer value such that $M\gamma h$ is a value occurring in O (e.g. $M = 31$ in the example in Table 4.2). Then $|S|\gamma^2 h^2 < \text{dist}_S(q, o)^2 < M^2 |S| \gamma^2 h^2$ for all $o \in O_+ \cup O_-$. Thus

$$(|O_+| - |O_+^{S,q}|) \exp\left(-|S|\gamma^2 M^2\right) < \epsilon_+(S, q) < (|O_+| - |O_+^{S,q}|) \exp\left(-|S|\gamma^2\right) \ll 1$$

and similarly

$$(|O_-| - |O_-^{S,q}|) \exp\left(-|S|\gamma^2 M^2\right) < \epsilon_-(S, q) < (|O_-| - |O_-^{S,q}|) \exp\left(-|S|\gamma^2\right) \ll 1$$

Note that $\lim_{\gamma \to \infty} \epsilon_+(S, q) = 0$ and $\lim_{\gamma \to \infty} \epsilon_-(S, q) = 0$. Now, it can be seen that:

- If a pattern S' is an emerging pattern, then by construction *at least one* object $q \in O_+$ must have $|O_+^{S,q}| \geq 2$ and $|O_-^{S,q}| = 1$. This is because S' only appears in O'_+, and for each transaction $o'_i \in O'_+$, we have inserted 2 copies of o'_i into O_+. On the other hand, S' does not appear in O'_- and the only object having values identical to q in the subspace S is the object containing all γh's. Therefore,

$$c_{\mathbf{CS}}(S,q) = \frac{|O_+^{S,q}| + \epsilon_+(S,q)}{|O_-^{S,q}| + \epsilon_-(S,q)} \geq \frac{2 + \epsilon_+(S,q)}{1 + \epsilon_-(S,q)} > 1 \tag{4.6}$$

- If a pattern S' is *not* an emerging pattern, then by construction *all* objects $q \in O_+$ must have $|O_-^{S,q}| \geq |O_+^{S,q}| + 1 > |O_+^{S,q}|$. Therefore,

$$c_{\mathbf{CS}}(S,q) = \frac{|O_+^{S,q}| + \epsilon_+(S,q)}{|O_-^{S,q}| + \epsilon_-(S,q)} < 1 \tag{4.7}$$

With these observations, we are ready to prove the main complexity result. We need to verify that the reduction **EP** $\rightarrow$ Complete-**CS** satisfies the two conditions (c1) and (c2) of the L-reduction.

c.1 For any instance I of **EP**, if S' is the most frequent emerging pattern with $c_{\mathbf{EP}}(S') = |\mathrm{Sat}_{O'_+}(S')|$ and $|\mathrm{Sat}_{O'_-}(S')| = 0$, then the corresponding optimal S solution for Complete-**CS** must have a cost value of

$$c(S) = \frac{2|\mathrm{Sat}_{O'_+}(S')| + \epsilon_+(S,q)}{1 + \epsilon_-(S,q)} \simeq 2|\mathrm{Sat}_{O'_+}(S')| = 2c_{\mathbf{EP}}(S') \tag{4.8}$$

where q is any data point in O_+ corresponding to the transaction containing pattern S'. This is because for each transaction o'_i containing S' in O'_+, we have inserted 2 copies of o'_i into O_+. The '1' in the denominator is due to the object containing all γh in O_-. Thus condition 1 is satisfied with $\alpha = 2$ when γ is sufficiently large.

c.2 For any solution S of Complete-**CS**, if $c(S) = \lambda \geq 2$ then the corresponding pattern S' constructed from S will be an emerging pattern. Further, let $[\lambda]$ be the nearest integer to λ. Then $[\lambda]$ must be even, and $[\lambda]/2$ will be the cost of the corresponding **EP** problem. Let λ^* denote the optimal cost of Complete-**CS**, then

$$\left| \frac{[\lambda]}{2} - \frac{[\lambda^*]}{2} \right| = \frac{1}{2}|[\lambda] - [\lambda^*]| \simeq \frac{1}{2}|\lambda - \lambda^*| \leq |\lambda - \lambda^*| \tag{4.9}$$

Thus condition 2 is satisfied with $\beta = 1$.

■

*Since **EP** $\rightarrow_L$ Complete-**CS**, if there exists a polynomial time approximation algorithm for Complete-**CS** with performance guarantee $1-\epsilon$, then there exists a polynomial time approximation algorithm for **EP** with performance guarantee $1-\alpha\beta\epsilon$. Since **EP** is MAX SNP-hard, it follows that Complete-**CS** must also be MAX SNP-hard.*

*Last, we draw the connection between Complete-**CS** and **CS**.*

Theorem 4.2. *If there exists a polynomial time approximation scheme (PTAS) for **CS** then there must also be a PTAS for Complete-**CS**.*

Proof. This is straightforward, as Complete-**CS** can be solved by a series of $|O_+|$ **CS** problems. ■

Unless P=NP, there exists no PTAS for Complete-**CS**, implying no PTAS for **CS**.

The above theoretical result indicates that the problem of mining contrast subspaces is even hard to approximate – it is impossible (unless P=NP) to design a good approximation algorithm. In the rest of the paper, we turn to practical heuristic methods.

4.3 Mining Methods

In this section, we first describe a baseline method that examines every possible non-empty subspace. Then, we present the design of our method *CSMiner* (for Contrast Subspace Miner) which employs smarter strategies for search.

4.3.1 A Baseline Method

A baseline naive method enumerates all possible non-empty spaces S and calculates the exact values of both $L_S(q \mid O_+)$ and $L_S(q \mid O_-)$, since both $L_S(q \mid O_+)$ and $L_S(q \mid O_-)$ are not monotonic with respect to the subspace-superspace relationship. Then, it returns the top-k subspaces S with the largest $LC_S(q)$ values. To ensure the completeness and efficiency of subspace enumeration, the baseline method traverses the set enumeration tree [99] of subspaces in a depth-first manner. A set enumeration tree takes a total order on a set, the set of dimensions in our problem, and enumerates all possible subsets in the lexicographical order. Figure 4.1 shows a set enumeration tree that enumerates all subspaces of $D = \{D_1, D_2, D_3, D_4\}$.

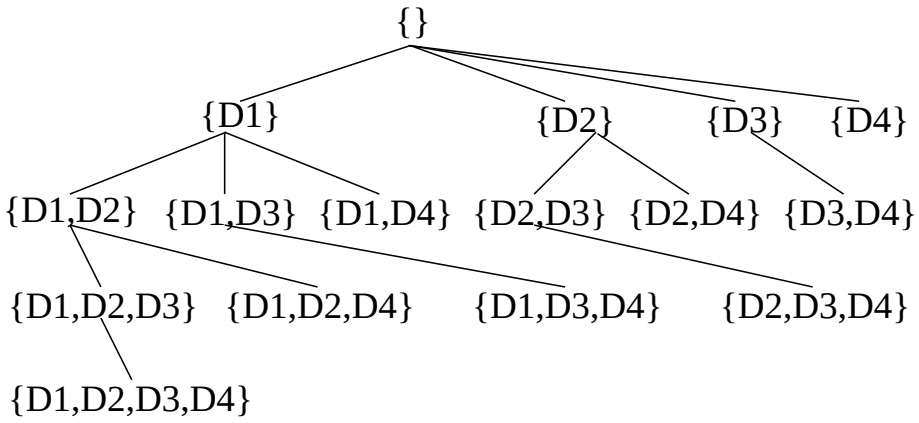


Figure 4.1: A set enumeration tree.

Using Equations 4.3 and 4.4, the baseline algorithm, shown in Algorithm 4.1, computes the likelihood contrast for every subspace where $L_S(q \mid O_+) \geq \delta$, and returns the top-$k$ subspaces. The time complexity is $O(2^{|D|} \cdot (|O_+| + |O_-|))$.

Algorithm 4.1 The baseline algorithm

Input: q: query object, O_+: objects belonging to C_+, O_-: objects belonging to C_-, δ: likelihood threshold, k: positive integer

Output: k subspaces with the highest likelihood contrast

1: let Ans be the current top-k list of subspaces, initialize Ans as k null subspaces associated with likelihood contrast 0
2: traverse the subspace set enumeration tree in a depth-first search manner
3: **for** each subspace S **do**
4: compute σ_{S+}, σ_{S-}, h_{opt};
5: compute $L_S(q \mid O_+)$ and $L_S(q \mid O_-)$ using Equation 4.3;
6: **if** $L_S(q \mid O_+) \geq \delta$ and $\exists S' \in Ans$ s.t. $\frac{L_S(q|O_+)}{L_S(q|O_-)} > LC_{S'}(q)$ **then**
7: insert S into Ans and remove S' from Ans;
8: **end if**
9: **end for**
10: **return** Ans;

4.3.2 The Framework of *CSMiner*

$L_S(q \mid O_+)$ is not monotonic in subspaces. To prune subspaces using the minimum likelihood threshold δ, we develop an upper bound of $L_S(q \mid O_+)$. We sort all the dimensions in their standard deviation descending order. Let $\mathcal{S}$ be the set of descendants of S in the subspace

set enumeration tree using the standard deviation descending order. Define

$$L_S^*(q \mid O_+) = \frac{1}{|O_+|(\sqrt{2\pi}\sigma'_{min}h'_{opt\_min})^\tau} \sum_{o \in O_+} e^{\frac{-dist_S(q,o)^2}{2(\sigma_S h'_{opt\_max})^2}} \tag{4.10}$$

where $\sigma'_{min} = \min\{\sigma_{S'} \mid S' \in \mathcal{S}\}$, $h'_{opt\_min} = \min\{h_{S'\_opt} \mid S' \in \mathcal{S}\}$, $h'_{opt\_max} = \max\{h_{S'\_opt} \mid S' \in \mathcal{S}\}$, and

$$\tau = \begin{cases} |S| & \text{if } \sqrt{2\pi}\sigma'_{min}h'_{opt\_min} \geq 1 \\ \max\{|S'| \mid S' \in \mathcal{S}\} & \text{if } \sqrt{2\pi}\sigma'_{min}h'_{opt\_min} < 1 \end{cases}$$

We have the following result.

Theorem 4.3 (Monotonic Density Bound)**.** *For a query object q, a set of objects O, and subspaces S_1, S_2 such that S_1 is an ancestor of S_2 in the subspace set enumeration tree in which dimensions in full space D are sorted by their standard deviation descending order, it is true that $L^*_{S_1}(q \mid O) \geq L_{S_2}(q \mid O)$.*

Proof. Let $\mathcal{S}$ be the set of descendants of S_1 in the subspace set enumeration tree using the standard deviation descending order in O. We define $\sigma'_{min} = \min\{\sigma_{S'} \mid S' \in \mathcal{S}\}$, $h'_{opt\_min} = \min\{h_{S'\_opt} \mid S' \in \mathcal{S}\}$, $h'_{opt\_max} = \max\{h_{S'\_opt} \mid S' \in \mathcal{S}\}$, and

$$\tau = \begin{cases} |S_1| & \text{if } \sqrt{2\pi}\sigma'_{min}h'_{opt\_min} \geq 1 \\ \max\{|S'| \mid S' \in \mathcal{S}\} & \text{if } \sqrt{2\pi}\sigma'_{min}h'_{opt\_min} < 1 \end{cases}$$

(Note that the computing of σ'_{min}, h'_{opt_min}, and h'_{opt_max} has linear complexity. As introduced in Section 4.2.2, $\sigma_{S'}$ is the root of the average marginal variance in S' and $h_{S'\_opt}$ depends on the values of $|O|$ and $|S'|$. Let $S'' \in \mathcal{S}$ such that for any subspace $S' \in \mathcal{S}$, $S' \subseteq S''$. Recall that the dimensions in the set enumeration tree are sorted by the standard deviation descending order, then, σ'_{min} can be obtained by checking dimensions in $S'' \setminus S_1$ one by one in the standard deviation ascending order. Moreover, h'_{opt_min} (h'_{opt_max}) can be obtained by comparing $h_{S'\_opt}$ with different values of $|S'| \in [|S_1|+1, |S''|]$.) As $S_2 \in \mathcal{S}$, we have $1 \leq |S_1| < |S_2| \leq \max\{|S'| \mid S' \in \mathcal{S}\}$, and $\sigma_{S_1} \geq \sigma_{S_2} \geq \sigma'_{min}$. Then, $\sigma_{S_2}h_{S_2\_opt} \geq \sigma'_{min}h'_{opt\_min}$. Thus,

$$(\sqrt{2\pi}\sigma_{S_2}h_{S_2\_opt})^{|S_2|} > (\sqrt{2\pi}\sigma'_{min}h'_{opt\_min})^\tau$$

Moreover, for $o \in O$, $dist_{S_1}(q,o) \leq dist_{S_2}(q,o)$. Correspondingly,

$$\frac{-dist_{S_2}(q,o)^2}{2(\sigma_{S_2}h_{S_2\_opt})^2} \leq \frac{-dist_{S_1}(q,o)^2}{2(\sigma_{S_1}h'_{opt\_max})^2}$$

By Equation 4.3,

$$\begin{aligned} L_{S_2}(q \mid O) &= \frac{1}{|O|(\sqrt{2\pi}\sigma_{S_2} h_{S_2\_opt})^{|S_2|}} \sum_{o \in O} e^{\frac{-dist_{S_2}(q,o)^2}{2(\sigma_{S_2} h_{S_2\_opt})^2}} \\ &\leq \frac{1}{|O|(\sqrt{2\pi}\sigma'_{min} h'_{opt\_min})^{\tau}} \sum_{o \in O} e^{\frac{-dist_{S_1}(q,o)^2}{2(\sigma_{S_1} h'_{opt\_max})^2}} \\ &= L^*_{S_1}(q \mid O) \end{aligned}$$

■

Using Theorem 4.3, in addition to $L_S(q \mid O_+)$ and $L_S(q \mid O_-)$, we also compute $L^*_S(q \mid O_+)$ for each subspace S. We are now in a position to state a pruning rule based on this theorem.

Pruning Rule 4.1. *Given a minimum likelihood threshold δ, if $L^*_S(q \mid O_+) < \delta$ in a subspace S, all superspaces of S can be pruned.* ■

Note that by using depth-first search, the distance between two objects in a super-space can be computed incrementally from the distance among the objects in a subspace. Given two objects q and o, let subspace $S' = S \cup \{D_i\}$. We have $dist_{S'}(q,o)^2 = dist_S(q,o)^2 + (q.D_i - o.D_i)^2$.

Algorithm 4.2 shows the pseudo code of the framework of *CSMiner*. Similar to the baseline method (Algorithm 4.1), *CSMiner* conducts a depth-first search on the subspace set enumeration tree. For a candidate subspace S, *CSMiner* calculates $L^*_S(q \mid O_+)$ using Equation 4.10. If $L^*_S(q \mid O_+)$ is less than the minimum likelihood threshold, all superspaces of S can be pruned by Theorem 4.3. Due to the hardness of the problem shown in Section 4.2.3 and the heuristic nature of this method, the time complexity of *CSMiner* is $O(2^{|D|} \cdot (|O_+| + |O_-|))$, the same as the exhaustive baseline method. However, as shown by our empirical study, *CSMiner* is substantially faster than the baseline method.

As stated in Algorithm 4.2, *CSMiner* starts with reading q, O_+ and O_-. For a candidate subspace S, *CSMiner* stores σ_{S+}, σ_{S-}, σ'_{min}, h_{opt}, h'_{opt_min}, and h'_{opt_max} to compute $L^*_S(q \mid O_+)$, and $LC_S(q)$. As *CSMiner* traverses the subspace set enumeration tree in a depth-first manner and finds top-k subspaces with the highest likelihood contrast, *CSMiner* only stores the likelihood contrast information of k candidate subspaces. The space complexity of *CSMiner* is $\mathcal{O}(|O_+| + |O_-| + k)$. Observe that $k \leq 2^{|D|}$ (D representing the full space).

Algorithm 4.2 *CSMiner*(q, O_+, O_-, δ, k)

Input: q: query object, O_+: objects belonging to C_+, O_-: objects belonging to C_-, δ: likelihood threshold, k: positive integer

Output: k subspaces with the highest likelihood contrast

1: let Ans be the current top-k list of subspaces, initialize Ans as k null subspaces associated with likelihood contrast 0
2: traverse the subspace set enumeration tree in a depth-first search manner
3: **for** each subspace S **do**
4: compute σ_{S+}, σ_{S-}, σ'_{min}, h_{opt}, h'_{opt_min}, and h'_{opt_max};
5: compute $L^*_S(q \mid O_+)$ using Equation 4.10;
6: **if** $L^*_S(q \mid O_+) < \delta$ **then**
7: prune all descendants of S and go to Step 2; // Pruning Rule 4.1
8: **else**
9: compute $L_S(q \mid O_+)$ and $L_S(q \mid O_-)$ using Equation 4.3;
10: **if** $L_S(q \mid O_+) \geq \delta$ and $\exists S' \in Ans$ s.t. $\frac{L_S(q|O_+)}{L_S(q|O_-)} > LC_{S'}(q)$ **then**
11: insert S into Ans and remove S' from Ans;
12: **end if**
13: **end if**
14: **end for**
15: **return** Ans;

4.3.3 A Bounding-Pruning-Refining Method

For a query object q and a set of objects O, the likelihood $L_S(q \mid O)$, computed by Equation 4.3, is the sum of density contributions of objects in O to q in subspace S. In Gaussian kernel estimation, given object $o \in O$, the contribution from o to $L_S(q \mid O)$ is $\frac{1}{|O|(\sqrt{2\pi}h_S)^{|S|}} e^{\frac{-dist_S(q,o)^2}{2{h_S}^2}}$. We observe that the contribution of o decays exponentially as the distance between q and o increases, and $L_S(q \mid O)$ can be bounded.

For a query object q and a set of objects O, the *ϵ-neighborhood* ($\epsilon > 0$) of q in subspace S is $N^\epsilon_S(q) = \{o \in O \mid dist_S(q, o) \leq \epsilon\}$. We can divide $L_S(q \mid O)$ into two parts, that is, $L_S(q \mid O) = L_{N^\epsilon_S}(q \mid O) + L^{rest}_S(q \mid O)$. The first part is contributed by the objects in the ϵ-neighborhood, that is, $L_{N^\epsilon_S}(q \mid O) = \frac{1}{|O|(\sqrt{2\pi}h_S)^{|S|}} \sum_{o \in N^\epsilon_S(q)} e^{\frac{-dist_S(q,o)^2}{2{h_S}^2}}$, and the second part is by the objects outside the ϵ-neighborhood, that is, $L^{rest}_S(q \mid O) = \frac{1}{|O|(\sqrt{2\pi}h_S)^{|S|}} \sum_{o \in O \setminus N^\epsilon_S(q)} e^{\frac{-dist_S(q,o)^2}{2{h_S}^2}}$.

Let $\overline{dist}_S(q \mid O)$ be the maximum distance between q and all objects in O in subspace

S. We have,

$$\frac{|O| - |N_S^\epsilon(q)|}{|O|(\sqrt{2\pi}h_S)^{|S|}} \cdot e^{-\frac{\overline{dist}_S(q,O)^2}{2{h_S}^2}} \leq L_S^{rest}(q \mid O) \leq \frac{|O| - |N_S^\epsilon(q)|}{|O|(\sqrt{2\pi}h_S)^{|S|}} \cdot e^{-\frac{\epsilon^2}{2{h_S}^2}}$$

Example 4.1. Figure 4.2 illustrates an example of a ϵ-neighborhood of object q with respect to object set O in a 2-dimensional subspace S. From Figure 4.2, we can see that $N_S^\epsilon(q) = \{o_1, o_2, o_3, o_4, o_5\}$, and $\overline{dist}_S(q \mid O) = dist_S(q, o_{10})$. ∎

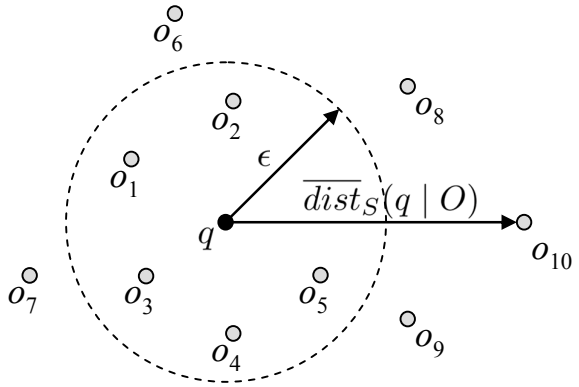


Figure 4.2: An example of an ϵ-neighborhood in a 2-dimensional subspace (within the dashed circle)

Using the above, an upper bound of $L_S^*(q \mid O_+)$ using ϵ-neighborhood, denoted by $L_S^{*\epsilon}(q \mid O_+)$, is

$$L_S^{*\epsilon}(q \mid O_+) = \frac{\sum\limits_{o \in N_S^\epsilon(q)} e^{\frac{-dist_S(q,o)^2}{2(\sigma_S h'_{opt\_max})^2}} + (|O_+| - |N_S^\epsilon(q)|)e^{-\frac{\epsilon^2}{2(\sigma_S h'_{opt\_max})^2}}}{|O_+|(\sqrt{2\pi}\sigma'_{min}h'_{opt\_min})^\tau} \tag{4.11}$$

where, the meanings of σ'_{min}, h'_{opt_min}, h'_{opt_max}, and τ are the same as those in Equation 4.10.

Pruning Rule 4.2. *Given a minimum likelihood threshold δ, if $L_S^{*\epsilon}(q \mid O_+) < \delta$ in a subspace S, all superspaces of S can be pruned.* ∎

Moreover, using the ϵ-neighborhood, we have the following upper and lower bounds of $L_S(q \mid O)$.

Theorem 4.4 (Bounds)**.** *For a query object q, a set of objects O and $\epsilon \geq 0$,*

$$LL_S^\epsilon(q \mid O) \leq L_S(q \mid O) \leq UL_S^\epsilon(q \mid O)$$

where

$$LL_S^\epsilon(q \mid O) = \frac{1}{|O|(\sqrt{2\pi}h_S)^{|S|}} \left(\sum_{o \in N_S^\epsilon(q)} e^{\frac{-dist_S(q,o)^2}{2{h_S}^2}} + (|O| - |N_S^\epsilon(q)|) e^{-\frac{\overline{dist}_S(q,O)^2}{2{h_S}^2}} \right)$$

and

$$UL_S^\epsilon(q \mid O) = \frac{1}{|O|(\sqrt{2\pi}h_S)^{|S|}} \left(\sum_{o \in N_S^\epsilon(q)} e^{\frac{-dist_S(q,o)^2}{2{h_S}^2}} + (|O| - |N_S^\epsilon(q)|) e^{-\frac{\epsilon^2}{2{h_S}^2}} \right)$$

Proof. For any object $o \in O \setminus N_S^\epsilon(q)$, $\epsilon^2 \leq dist_S(q,o)^2 \leq \overline{dist}_S(q,O)^2$. Then,

$$e^{-\frac{\epsilon^2}{2{h_S}^2}} \geq e^{\frac{-dist_S(q,o)^2}{2{h_S}^2}} \geq e^{-\frac{\overline{dist}_S(q,O)^2}{2{h_S}^2}}$$

Thus,

$$(|O| - |N_S^\epsilon(q)|) e^{-\frac{\epsilon^2}{2{h_S}^2}} \geq (|O| - |N_S^\epsilon(q)|) e^{\frac{-dist_S(q,o)^2}{2{h_S}^2}} \geq (|O| - |N_S^\epsilon(q)|) e^{-\frac{\overline{dist}_S(q,O)^2}{2{h_S}^2}}$$

Correspondingly,

$$LL_S^\epsilon(q \mid O) \leq L_S(q \mid O) \leq UL_S^\epsilon(q \mid O)$$

■

We obtain an upper bound of $LC_S(q)$ based on Theorem 4.4 and Equation 4.4.

Corollary 4.1 (Likelihood Contrast Upper Bound)**.** *For a query object q, a set of objects O_+, a set of objects O_-, and $\epsilon \geq 0$, $LC_S(q) \leq \frac{UL_S^\epsilon(q|O_+)}{LL_S^\epsilon(q|O_-)}$.*

Proof. By Theorem 4.4, we have $L_S(q \mid O_+) \leq UL_S^\epsilon(q \mid O_+)$ and $L_S(q \mid O_-) \geq LL_S^\epsilon(q \mid O_-)$. Then,

$$LC_S(q) = \frac{L_S(q \mid O_+)}{L_S(q \mid O_-)} \leq \frac{UL_S^\epsilon(q \mid O_+)}{L_S(q \mid O_-)} \leq \frac{UL_S^\epsilon(q \mid O_+)}{LL_S^\epsilon(q \mid O_-)}$$

■

Using Corollary 4.1, we have the following.

Pruning Rule 4.3. *For a subspace S, if there are at least k subspaces whose likelihood contrast are greater than $\frac{UL_\epsilon^S(q|O_+)}{LL_\epsilon^S(q|O_-)}$, then S cannot be a top-k subspace of the largest likelihood contrast.* ■

We implement the bounding-pruning-refining method in *CSMiner* to compute bounds of likelihood and contrast ratio. We call this version *CSMiner*-BPR. For a candidate subspace S, *CSMiner*-BPR calculates $UL_S^\epsilon(q \mid O_+)$, $LL_S^\epsilon(q \mid O_-)$ and $L_S^{*\epsilon}(q \mid O_+)$ using the ϵ-neighborhood. If $UL_S^\epsilon(q \mid O_+)$ is less than the minimum likelihood threshold (δ), *CSMiner*-BPR checks whether it is true that $L_S^{*\epsilon}(q \mid O_+) < \delta$ (Pruning Rule 4.2) or $L_S^*(q \mid O_+) < \delta$ (Pruning Rule 4.1). Otherwise, *CSMiner*-BPR checks whether the likelihood contrasts of the current top-k subspaces are larger than the upper bound of $LC_S(q)$ ($= \frac{UL_\epsilon^S(q|O_+)}{LL_\epsilon^S(q|O_-)}$). If not, *CSMiner*-BPR refines $L_S^*(q \mid O_+)$, $L_S(q \mid O_+)$ and $L_S(q \mid O_-)$ by involving objects that are out of the ϵ-neighborhood. S will be added into the current top-k list if $L_S^*(q \mid O_+) \geq \delta$ and the ratio of $L_S(q \mid O_+)$ to $L_S(q \mid O_-)$ is larger than one of the current top-k ones. Note that the computational cost for $L_S^*(q \mid O_+)$ can be high, especially, when the size of O_+ is large. Thus for efficiency, we employ a tradeoff between Pruning Rule 4.1 and Pruning Rule 4.3. Specifically, when we are searching a subspace S, once we can determine that S cannot be a top-k contrast subspace, then we terminate the search of S immediately. Therefore, *CSMiner*-BPR accelerates *CSMiner* by avoiding the cost for computing the likelihood contributions of objects outside the ϵ-neighborhood to q when $L_S^*(q \mid O_+) < \delta$ (Pruning Rule 4.2) or $\frac{UL_\epsilon^S(q|O_+)}{LL_\epsilon^S(q|O_-)} < \delta$ (Pruning Rule 4.3).

Computing ϵ-neighborhood is critical in *CSMiner*-BPR. The distance between objects increases when dimensionality increases. Thus, the value of ϵ should not be fixed. The standard deviation expresses the variability of a set of data. For subspace S, we set $\epsilon = \sqrt{\alpha \cdot \sum_{D_i \in S} (\sigma_{D_i^+}^2 + \sigma_{D_i^-}^2)}$ ($\alpha \geq 0$), where $\sigma_{D_i^+}^2$ and $\sigma_{D_i^-}^2$ are the marginal variances of O_+ and O_-, respectively, on dimension D_i ($D_i \in S$), and α is a system defined parameter. Our experiments show that α can be set in the range of $0.8 \sim 1.2$, and is not sensitive. Algorithm 4.3 provides the pseudo-code of *CSMiner*-BPR. Theorem 4.5 guarantees that no matter how the neighborhood distance (ϵ) is varied, the mining result of *CSMiner*-BPR is unchanged.

Theorem 4.5. *Given data set O, query object q, minimum likelihood threshold δ, and parameter k, for any neighborhood distances ϵ_1 and ϵ_2, $CS^{\epsilon_1}(q \mid O) = CS^{\epsilon_2}(q \mid O)$, where $CS^{\epsilon_1}(q \mid O)$ ($CS^{\epsilon_2}(q \mid O)$) is the set of contrast subspaces discovered by CSMiner-BPR using ϵ_1 (ϵ_2).*

Proof. We prove by contradiction.

Assume that subspace $S \in CS^{\epsilon_1}(q \mid O)$ but $S \notin CS^{\epsilon_2}(q \mid O)$. As $S \in CS^{\epsilon_1}(q \mid O)$, we

Algorithm 4.3 *CSMiner*-BPR$(q, O_+, O_-, \delta, k, \alpha)$

Input: q: a query object, O_+: the set of objects belonging to C_+, O_-: the set of objects belonging to C_-, δ: a likelihood threshold, k: positive integer, α: neighborhood parameter

Output: k subspaces with the highest likelihood contrast

1: let Ans be the current top-k list of subspaces, initialize Ans as k null subspaces associated with likelihood contrast 0
2: **for** each subspace S in the subspace set enumeration tree, searched in the depth-first manner **do**
3: compute ϵ, σ_{S+}, σ_{S-}, σ'_{min}, h_{opt}, h'_{opt_min}, and h'_{opt_max};
4: $N_S^\epsilon(q)_+ \leftarrow \emptyset$; $N_S^\epsilon(q)_- \leftarrow \emptyset$; $\overline{dist}_S(q \mid O_-) \leftarrow 0$;
5: **for** each object $o \in O_+ \cup O_-$ **do**
6: $dist_S(q,o)^2 \leftarrow dist_{S^p}(q,o)^2 + (q.D' - o.D')^2$; // $S^p (= S \cup \{D'\})$ is the parent of S.
7: **if** $o \in O_+$ and $dist_S(q,o) < \epsilon$ **then**
8: $N_S^\epsilon(q)_+ \leftarrow N_S^\epsilon(q)_+ \cup \{o\}$;
9: **end if**
10: **if** $o \in O_-$ **then**
11: **if** $dist_S(q,o) < \epsilon$ **then**
12: $N_S^\epsilon(q)_+ \leftarrow N_S^\epsilon(q)_+ \cup \{o\}$;
13: **end if**
14: **if** $\overline{dist}_S(q \mid O_-) < dist_S(q,o)$ **then**
15: $\overline{dist}_S(q \mid O_-) \leftarrow dist_S(q,o)$;
16: **end if**
17: **end if**
18: **end for**
19: compute $UL_S^\epsilon(q \mid O_+)$, $LL_S^\epsilon(q \mid O_-)$ and $L_S^{*\epsilon}(q \mid O_+)$; // bounding
20: **if** $UL_S^\epsilon(q \mid O_+) < \delta$ **then**
21: **if** $L_S^{*\epsilon}(q \mid O_+) < \delta$ **then**
22: prune all descendants of S and go to Step 2; // Pruning Rule 4.2
23: **end if**
24: compute $L_S^*(q \mid O_+)$;
25: **if** $L_S^*(q \mid O_+) < \delta$ **then**
26: prune all descendants of S and go to Step 2; // Pruning Rule 4.1
27: **end if**
28: **else**
29: **if** $\exists S' \in Ans$ s.t. $\frac{UL_S^\epsilon(q|O_+)}{LL_S^\epsilon(q|O_-)} \geq LC_{S'}(q)$ **then**
30: compute $L_S^*(q \mid O_+)$ using Equation 4.10; // refining
31: **if** $L_S^*(q \mid O_+) < \delta$ **then**
32: prune all descendants of S and go to Step 2; // Pruning Rule 4.1
33: **else**
34: compute $L_S(q \mid O_+)$ and $L_S(q \mid O_-)$ using Equation 4.3; // refining
35: **if** $L_S(q \mid O_+) \geq \delta$ and $\exists S' \in Ans$ s.t. $\frac{L_S(q|O_+)}{L_S(q|O_-)} > LC_{S'}(q)$ **then**
36: insert S into Ans and remove S' from Ans;
37: **end if**
38: **end if**
39: **end if**
40: **end if**
41: **end for**
42: **return** Ans;

have ($\star$) $L_S(q \mid O_+) \geq \delta$. On the other hand, $S' \notin CS^{\epsilon_2}(q \mid O)$ means that (i) $L_S^{*\epsilon_2}(q \mid O_+) < \delta$, or (ii) $\exists S' \in CS^{\epsilon_2}(q \mid O)$ such that $S' \notin CS^{\epsilon_1}(q \mid O)$ and $\frac{UL_S^{\epsilon_1}(q|O_+)}{LL_S^{\epsilon_1}(q|O_-)} < LC_{S'}(q)$. For case (i), as $L_S(q \mid O_+) \leq L_S^*(q \mid O_+) \leq L_S^{*\epsilon_2}(q \mid O_+)$, we have $L_S(q \mid O_+) < \delta$, contradicting ($\star$). For case (ii), as $LC_S(q) \leq \frac{UL_S^{\epsilon_1}(q|O_+)}{LL_S^{\epsilon_1}(q|O_-)}$, we have $LC_S(q) < LC_{S'}(q)$, contradicting $S' \notin CS^{\epsilon_1}(q \mid O)$. ∎

Corollary 4.2. *Given data set O, query object q, minimum likelihood threshold δ, and parameter k, the mining result of CSMiner-BPR, no matter what the value of parameter α is, the output is the same as that of CSMiner.*

Proof. For subspace S, suppose ϵ, computed by parameter α, is not less than $\overline{dist}_S(q \mid O)$. We have $N_S^\epsilon(q) = \emptyset$. Correspondingly, $UL_S^\epsilon(q \mid O_+) = L_S(q \mid O_+)$, $LL_S^\epsilon(q \mid O_-) = L_S(q \mid O_-)$, and $L_S^{*\epsilon}(q \mid O_+) = L_S^*(q \mid O_+)$. Then the execution flow of *CSMiner*-BPR (Algorithm 4.3) is the same as that of *CSMiner* (Algorithm 4.2). Furthermore, by Theorem 4.5, the mining result of *CSMiner*-BPR is unchanged no matter what the value of neighborhood distance is. ∎

4.4 Empirical Evaluation

In this section, we report a systematic empirical study using real data sets to verify the effectiveness and efficiency of *CSMiner* (*CSMiner*-BPR). In general, we study how sensitive are our methods to the running parameters, such as δ, k, and α, in terms of discovered contrast subspaces and running time; and how sensitive are our methods to different bandwidth values and kernel function, in terms of the similarity of mining results. All experiments were conducted on a PC computer with an Intel Core i7-3770 3.40 GHz CPU, and 8 GB main memory, running Windows 7 operating system. All algorithms were implemented in Java and compiled by JDK 7. We set $\delta = 0.001, k = 10$ and, $\alpha = 0.8$ as defaults in our experiments.

4.4.1 Effectiveness

We use 6 real data sets from the UCI machine learning repository [11]. We remove non-numerical attributes and all instances containing missing values. Table 4.3 shows the data characteristics.

Data set	# objects	# attributes	# classes
Breast Cancer Wisconsin (BCW)	683	9	2
Climate Model Simulation Crashes (CMSC)	540	18	2
Glass Identification (Glass)	214	9	2
Pima Indians Diabetes (PID)	768	8	2
Waveform	5000	21	3
Wine	178	13	3

Table 4.3: Data set characteristics

As shown in Table 4.3, BCW, Glass, PID and Wine are typical small data sets which contain hundreds of objects with around 10 numerical attributes. The objects in BCW, Glass and PID are divided into 2 classes, respectively, while the objects in Wine are divided into 3 classes. Compared with BCW, Glass, PID and Wine, CMSC and Waveform contain more numerical attributes. We note that CMSC is an unbalanced data set, in which the number of objects in the two classes are 46 and 494, respectively. Among all selected data sets, Waveform containing 5000 objects is the largest one with the highest dimensionality.

For each data set, we take each record as a query object q, and all records except q belonging to the same class as q forming the set O_1, and records belonging to the other classes forming the set O_2. Using *CSMiner*, we compute for each record (1) the *inlying contrast subspace* taking O_1 as O_+ and O_2 as O_-, and (2) the *outlying contrast subspace* taking O_2 as O_+ and O_1 as O_-. In this experiment, we only compute the top-1 subspace. For clarity, we denote the likelihood contrasts of inlying contrast subspace by $LC_S^{in}(q)$ and those of outlying contrast subspace by $LC_S^{out}(q)$. The minimum likelihood threshold (δ) is set to 0.001.

Tables 4.4 $\sim$ 4.9 list the joint distributions of $LC_S^{in}(q)$ and $LC_S^{out}(q)$ in each data set. Consider that the query object has the same class label as objects in O_1 in the original data set. Thus, it is expected that, for most objects, $LC_S^{in}(q)$ are larger than $LC_S^{out}(q)$. However, interestingly a good portion of objects have strong outlying contrast subspaces. For example, in CMSC, more than 40% of the objects have outlying contrast subspaces satisfying $LC_S^{out}(q) \geq 10^3$. Moreover, we can see that, except PID, a non-trivial number of objects in each data set have both strong inlying and outlying contrast subspaces (e.g., $LC_S^{in}(q) \geq 10^4$ and $LC_S^{out}(q) \geq 10^2$).

		$LC_S^{out}(q)$					
		< 1	[1,3)	[3,10)	$[10, 10^2)$	$\geq 10^2$	Total
$LC_S^{in}(q)$	$< 10^4$	0	3	0	7	23	33
	$[10^4, 10^5)$	7	4	2	4	7	24
	$[10^5, 10^6)$	21	21	5	8	9	64
	$[10^6, 10^7)$	184	33	5	4	9	235
	$\geq 10^7$	121	31	74	66	35	327
Total		333	92	86	89	83	683

Table 4.4: Distribution of $LC_S(q)$ in BCW ($\delta = 0.001, k = 1$)

		$LC_S^{out}(q)$					
		$[10, 10^2)$	$[10^2, 10^3)$	$[10^3, 10^4)$	$[10^4, 10^5)$	$\geq 10^5$	Total
$LC_S^{in}(q)$	$< 10^3$	1	11	12	2	0	26
	$[10^3, 10^4)$	6	35	47	6	6	100
	$[10^4, 10^5)$	10	46	44	8	2	110
	$[10^5, 10^6)$	11	40	32	8	2	93
	$\geq 10^6$	39	110	50	11	1	211
Total		67	242	185	35	11	540

Table 4.5: Distribution of $LC_S(q)$ in CMSC ($\delta = 0.001, k = 1$)

		$LC_S^{out}(q)$					
		< 1	[1,3)	[3,10)	$[10, 10^2)$	$\geq 10^2$	Total
$LC_S^{in}(q)$	$< 10^2$	0	0	0	1	7	8
	$[10^2, 10^3)$	2	8	4	4	7	25
	$[10^3, 10^4)$	28	91	6	4	5	134
	$[10^4, 10^5)$	1	4	0	0	3	8
	$\geq 10^5$	0	1	0	30	8	39
Total		31	104	10	39	30	214

Table 4.6: Distribution of $LC_S(q)$ in Glass ($\delta = 0.001, k = 1$)

		$LC_S^{out}(q)$					
		< 1	[1,3)	[3,10)	$[10, 30)$	≥ 30	Total
$LC_S^{in}(q)$	< 1	0	0	1	0	0	1
	$[1, 3)$	2	241	62	8	2	315
	$[3, 10)$	36	328	31	3	0	398
	$[10, 30)$	23	23	2	0	0	48
	≥ 30	3	3	0	0	0	6
Total		64	595	96	11	2	768

Table 4.7: Distribution of $LC_S(q)$ in PID ($\delta = 0.001, k = 1$)

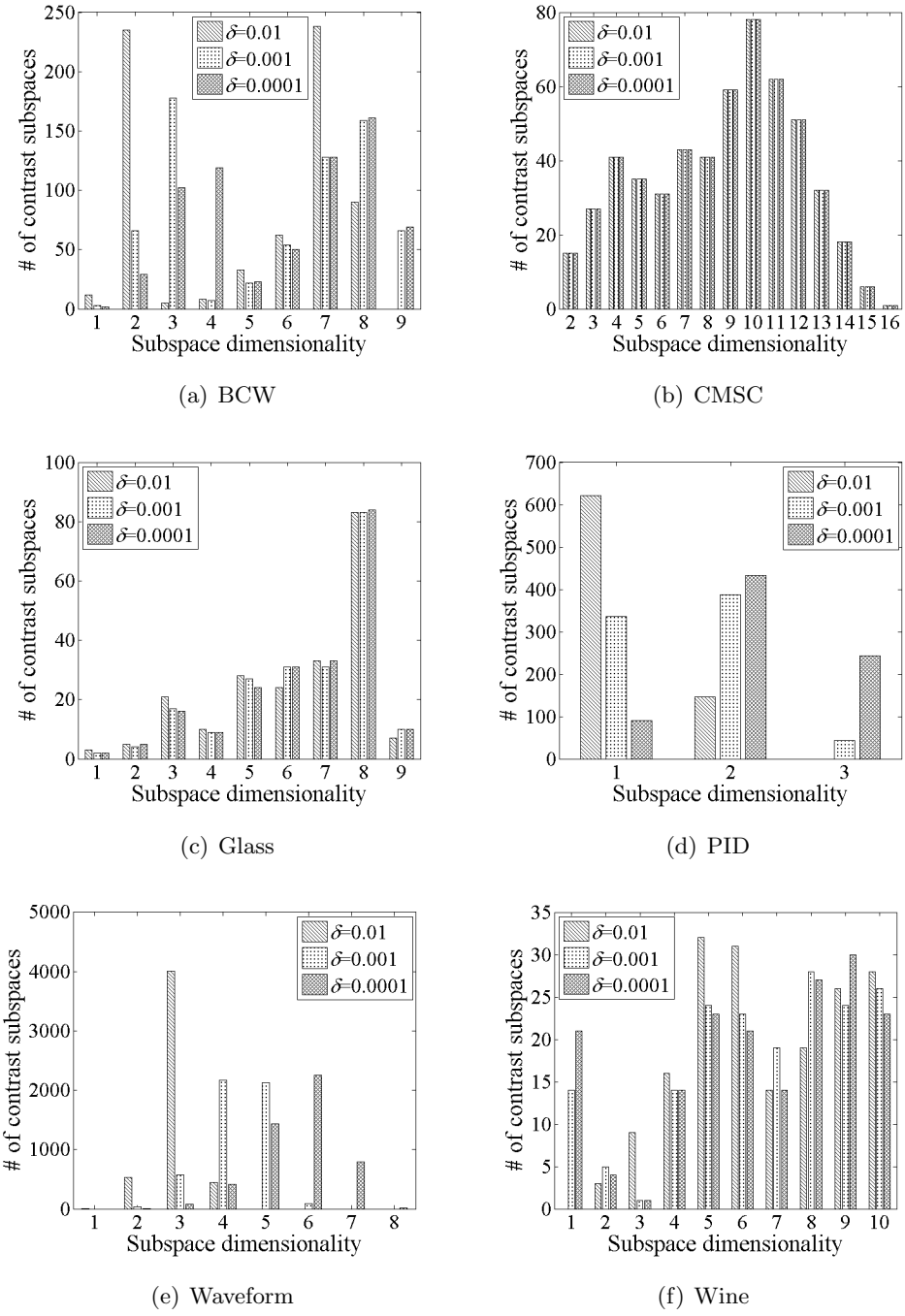
		$LC_S^{out}(q)$					
		$[1, 3)$	[3,10)	$[10, 10^2)$	$[10^2, 10^3)$	$\geq 10^3$	Total
$LC_S^{in}(q)$	< 10	0	24	34	8	2	68
	$[10, 10^2)$	204	676	772	190	71	1913
	$[10^2, 10^3)$	471	1049	981	228	56	2785
	$[10^3, 10^4)$	53	103	67	4	4	231
	$\geq 10^4$	0	2	1	0	0	3
Total		728	1854	1855	430	133	5000

Table 4.8: Distribution of $LC_S(q)$ in Waveform ($\delta = 0.001, k = 1$)

		$LC_S^{out}(q)$					
		< 1	[1,3)	[3,10)	$[10, 10^2)$	$\geq 10^2$	Total
$LC_S^{in}(q)$	$< 10^3$	0	13	8	7	5	33
	$[10^3, 10^4)$	1	18	11	4	0	34
	$[10^4, 10^5)$	2	23	12	5	2	44
	$[10^5, 10^6)$	3	7	5	1	0	16
	$\geq 10^6$	7	20	16	4	4	51
Total		13	81	52	21	11	178

Table 4.9: Distribution of $LC_S(q)$ in Wine ($\delta = 0.001, k = 1$)

Figures 4.3 and Figure 4.4 show the distributions of dimensionality of top-1 inlying and outlying contrast subspaces with different minimum likelihood thresholds (δ), respectively. The dimensionality distribution is an interesting feature characterizing a data set. For example, in most cases the contrast subspaces tend to have low dimensionality. However, in CMSC and Wine, the inlying contrast subspaces tend to have high dimensionality. Moreover, we can see that with the decrease of δ, the number of subspaces with higher dimensionality is typically increased.



(a) BCW

(b) CMSC

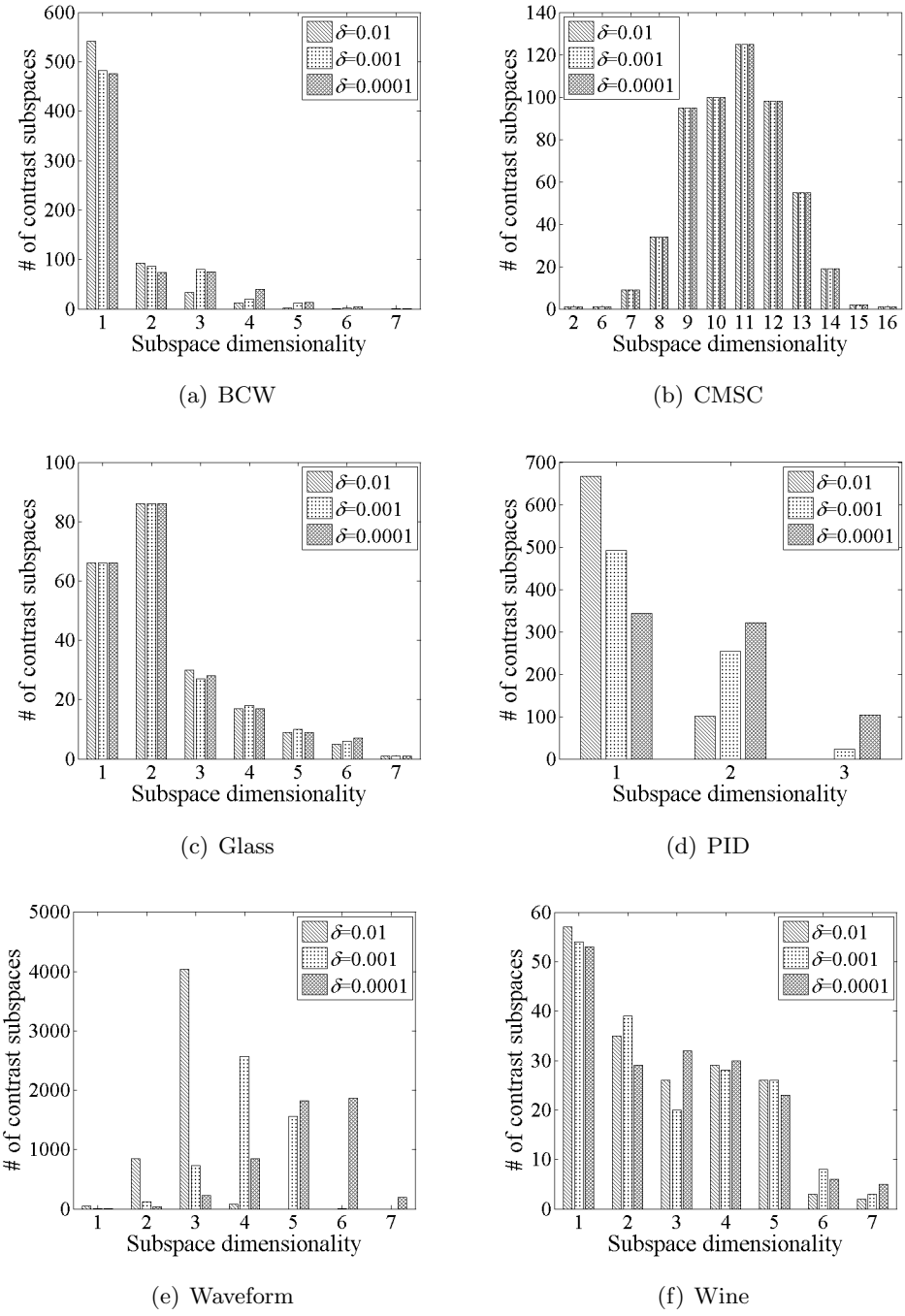
(c) Glass

(d) PID

(e) Waveform

(f) Wine

Figure 4.3: Dimensionality distributions of top inlying contrast subspaces ($k = 1$)



(a) BCW

(b) CMSC

(c) Glass

(d) PID

(e) Waveform

(f) Wine

Figure 4.4: Dimensionality distributions of top outlying contrast subspaces ($k = 1$)

4.4.2 Efficiency

To the best of our knowledge, there is no previous method tackling the exact same mining problem. Therefore, we evaluate the efficiency of *CSMiner* and its variations. Specifically, we implemented the baseline method (Algorithm 4.1). To evaluate the efficiency of our pruning techniques for contrast subspace mining, we also implemented *CSMiner* (Algorithm 4.2), and *CSMiner*-BPR (Algorithm 4.3) using the bounding-pruning-refining method.

We report the results on the Waveform data set only, since it is the largest one with the highest dimensionality. We randomly select 100 records from Waveform as query objects, and report the average runtime. The results on the other data sets follow similar trends.

Figure 4.5 shows the runtime with respect to the minimum likelihood threshold δ.A logarithmic scale has been used for the runtime to better demonstrate the difference in the behavior between *CSMiner* and the baseline. The baseline performs exhaustive subspace search and thus its runtime is unchanged across different δ values. For *CSMiner* and *CSMiner*-BPR, as δ decreases, their runtime increase exponentially. However, the heuristic pruning techniques implemented in *CSMiner* and *CSMiner*-BPR accelerate the search substantially in practice. Moreover, *CSMiner*-BPR is slightly more efficient than *CSMiner*.

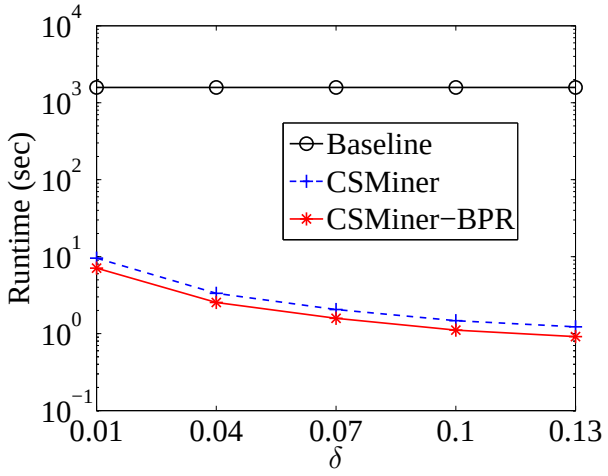


Figure 4.5: Scalability test w.r.t $\delta(k = 10, \alpha = 0.8)$

Figure 4.6 shows the runtime with respect to the data set size, which is measured by the number of objects. Again, the runtime is plotted using the logarithmic scale. We can see

that our pruning techniques can achieve a roughly linear runtime in practice. Both *CSMiner* and *CSMiner*-BPR are considerably faster than the baseline method, and *CSMiner*-BPR is slightly more efficient than *CSMiner*.

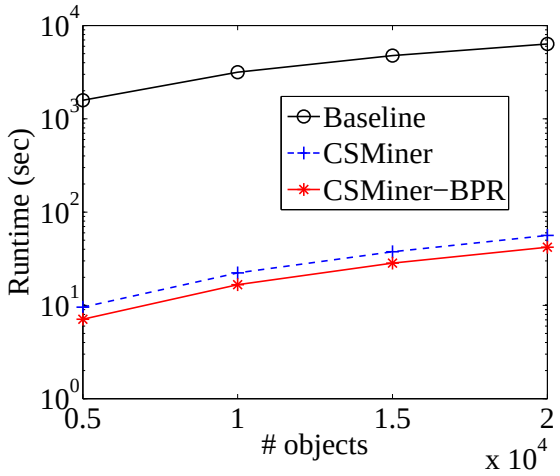


Figure 4.6: Scalability test w.r.t data set size ($k = 10, \delta = 0.01, \alpha = 0.8$)

Figure 4.7 shows the runtime with respect to the dimensionality of the data set. The runtime is also plotted using the logarithmic scale. As dimensionality increases, more candidate subspaces are generated. Correspondingly, the runtime increases exponentially. However, our heuristic pruning techniques implemented in *CSMiner* and *CSMiner*-BPR speed up the search in practice. Moreover, *CSMiner*-BPR is faster than *CSMiner*.

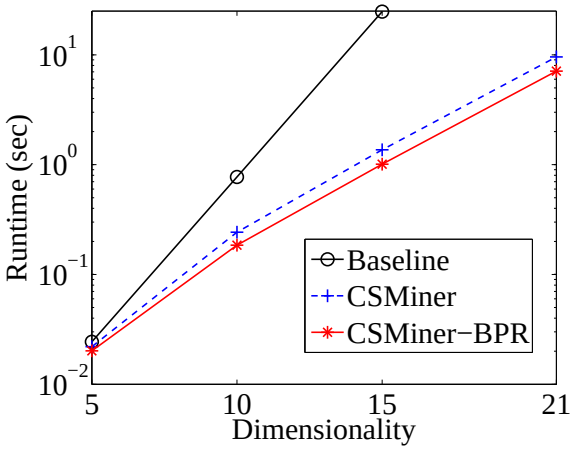


Figure 4.7: Scalability test w.r.t dimensionality ($k = 10, \delta = 0.01, \alpha = 0.8$)

CSMiner-BPR employs a user defined parameter α to define the ϵ-neighborhood. Table 4.10 lists the average runtime of *CSMiner*-BPR for a query object with respect to α on each real data set. The runtime of *CSMiner*-BPR is not sensitive to α in general. Experimentally, the shortest runtime of *CSMiner*-BPR happens when α is in $[0.6, 1.0]$.

Data set	Average runtime (millisecond)				
	$\alpha = 0.6$	$\alpha = 0.8$	$\alpha = 1.0$	$\alpha = 1.2$	$\alpha = 1.4$
BCW	20.97	20.14	17.76	16.32	**15.59**
CMSC	**11446.2**	11643.5	12915.1	14125.0	15210.2
Glass	16.13	15.83	**15.62**	15.69	15.76
PID	4.21	**4.17**	4.23	4.25	4.29
Waveform	**6807.1**	7102.3	7506.7	7874.7	8183.7
Wine	18.51	**18.16**	18.42	18.69	19.12

Table 4.10: Average runtime of *CSMiner*-BPR w.r.t α ($k = 10, \delta = 0.01$)

Figure 4.8 illustrates the relative runtime of *CSMiner*-BPR with respect to k on each real data set, showing that *CSMiner*-BPR is linearly scalable with respect to k. Note that we show relative performance in Figure 4.8 so that the scalability of *CSMiner*-BPR with respect to k on different data sets can be compared in one figure. The absolute performance of *CSMiner*-BPR with $k = 10$, $\delta = 0.01$ and $\alpha = 0.8$ can be found in Table 4.10.

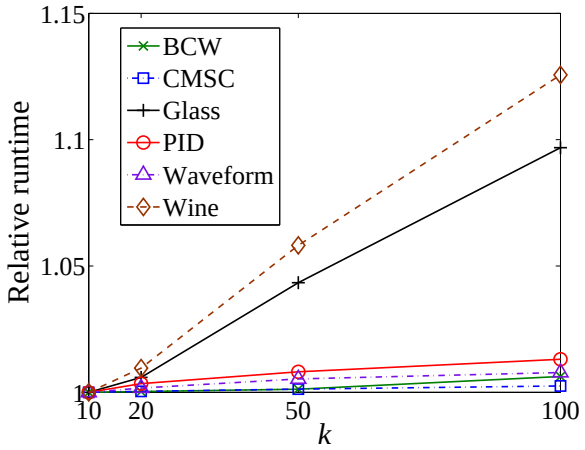


Figure 4.8: Relative runtime of *CSMiner*-BPR w.r.t k ($\delta = 0.01, \alpha = 0.8$)

4.4.3 Sensitivity to the Bandwidth

To test the sensitivity of the top-k contrast subspaces with respect to the bandwidth value, we begin by defining the similarity measure for two lists of top-k contrast subspaces.

For any two subspaces S_1 and S_2, we measure the similarity between S_1 and S_2 by the Jaccard similarity coefficient, denoted by $Jaccard(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}$.

Given a positive integer r, let $\mathbb{P}^r$ be the set of all permutations of the set $\{i \mid 1 \leq i \leq r\}$. Correspondingly, $|\mathbb{P}^r| = r!$. For permutation $P \in \mathbb{P}^r$, we denote the j-th ($1 \leq j \leq r$) element in P by $P[j]$. For example, by writing each permutation as a tuple, we have $\mathbb{P}^3 = \{(1,2,3),(1,3,2),(2,1,3),(2,3,1),(3,1,2),(3,2,1)\}$. Suppose $P = (2,3,1)$, then $P[2] = 3$.

To the best of our knowledge, there is no previous work on measuring the similarity between two ranked lists of subspaces. Given two ranked lists of top-k contrast subspaces ℓ_1 and ℓ_2, without loss of generality, we follow the definition of *average overlap* [115] (also named as *average accuracy* [121], or *intersection metric* [40]), which derives the similarity measure by averaging the overlaps of two ranked lists at each rank, to measure the similarity between ℓ_1 and ℓ_2. In addition, in consideration of the fact that each subspace in a list is a set of dimensions, we introduce the Jaccard similarity coefficient into the overlap calculation. Specifically, let $\ell_1[i]$ be the element (subspace) at rank i ($1 \leq i \leq k$) in list ℓ_1. The *agreement* of lists ℓ_1 and ℓ_2 at rank r ($1 \leq r \leq k$), $Agr(\ell_1, \ell_2, r)$, is

$$Agr(\ell_1, \ell_2, r) = \frac{1}{r} \max\{\sum_{i=1}^{r} Jaccard(\ell_1[P_1[i]], \ell_2[P_2[i]]) \mid P_1, P_2 \in \mathbb{P}^r\}$$

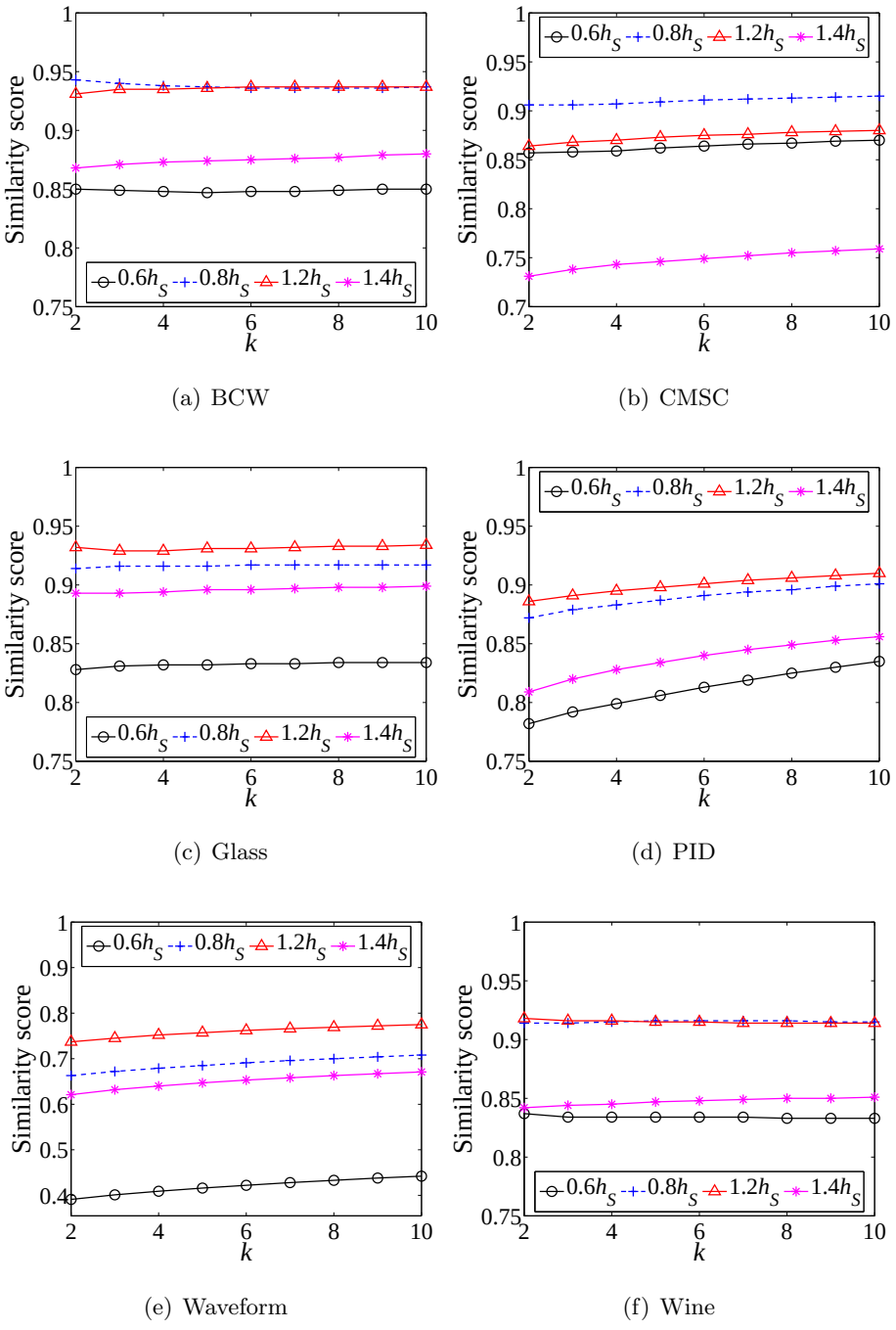
Then, the similarity between ℓ_1 and ℓ_2, denoted by $Sim(\ell_1, \ell_2)$, is

$$Sim(\ell_1, \ell_2) = \frac{1}{k} \sum_{r=1}^{k} Agr(\ell_1, \ell_2, r) \tag{4.12}$$

Clearly, $0 \leq Sim(\ell_1, \ell_2) \leq 1$. The larger the value of $Sim(\ell_1, \ell_2)$, the more similar ℓ_1 and ℓ_2 are.

Given a set of objects O, and a query object q, to find top-k contrast subspaces for q with respect to O by *CSMiner* (Algorithm 4.2), as discussed in Section 4.2.2, we first fix the bandwidth value $h_S = \sigma_S \cdot h_{S\_opt}$, and use the Gaussian kernel function to estimate the subspace likelihood of q with respect to O in subspace S. We then vary the bandwidth value from $0.6h_S$ to $1.4h_S$ for density estimation in S. Let ℓ_{h_S} be the top-k contrast subspaces computed using the default bandwidth value h_S, and $\ell_{\widetilde{h_S}}$ be the top-k contrast

subspaces computed using other bandwidth values. For each object $q \in O$, we discover top inlying contrast subspaces and top outlying contrast subspaces of q by *CSMiner* using different bandwidth values. Figure 4.9 illustrates the average value of $Sim(\ell_{h_S}, \ell_{\widetilde{h_S}})$ of inlying contrast subspaces with respect to k, and Figure 4.10 illustrates the average value of $Sim(\ell_{h_S}, \ell_{\widetilde{h_S}})$ of outlying contrast subspaces with respect to k. From the results, we can see that the contrast subspaces computed using different bandwidth values are similar in most data sets. As expected, using a bandwidth whose value is closer to h causes less difference. Moreover, we observe that with increasing k, the value of $Sim(\ell_{h_S}, \ell_{\widetilde{h_S}})$ slightly increases.

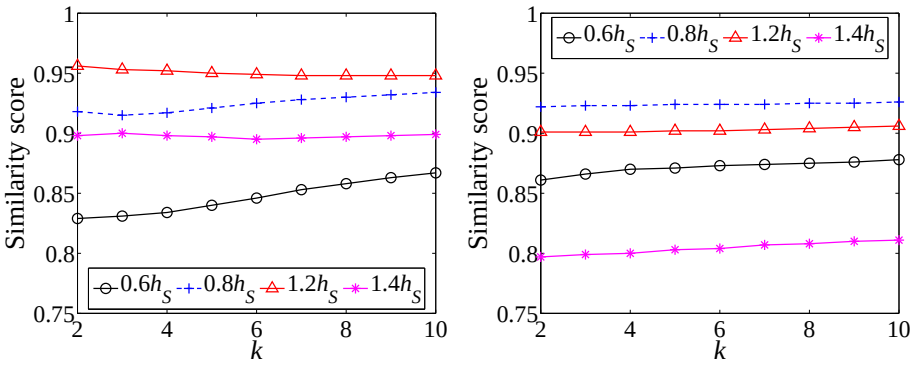


(a) BCW (b) CMSC

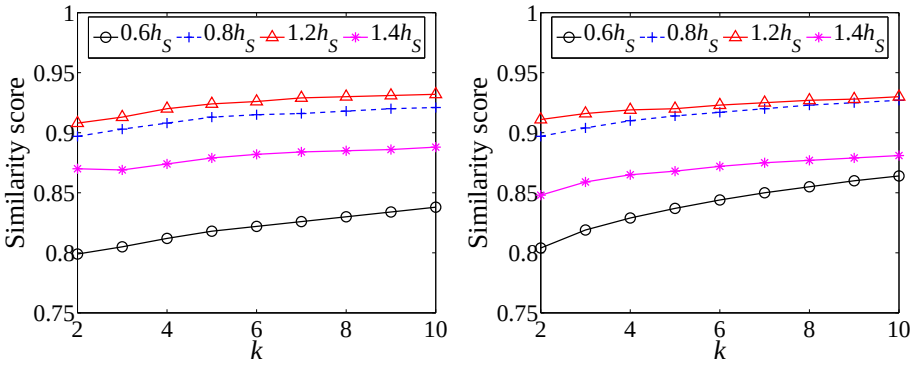
(c) Glass (d) PID

(e) Waveform (f) Wine

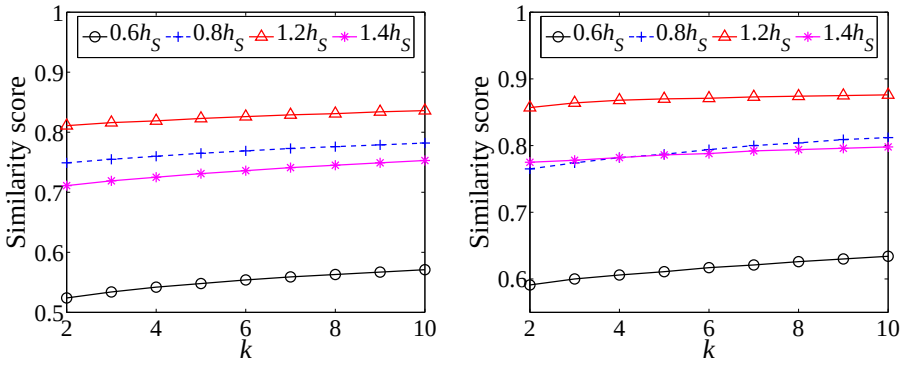
Figure 4.9: The similarity scores of inlying contrast subspaces using different bandwidth values with respect to k ($\delta = 0.001$)



(a) BCW (b) CMSC



(c) Glass (d) PID



(e) Waveform (f) Wine

Figure 4.10: The similarity scores of outlying contrast subspaces using different bandwidth values with respect to k ($\delta = 0.001$)

4.4.4 Comparison with Epanechnikov Kernel

Besides Gaussian kernel (Equation 4.2), another possible kernel for multivariate kernel density estimation is the multivariate Epanechnikov kernel

$$K_e(x) = \begin{cases} \frac{1}{2}{c_d}^{-1}(d+2)(1-x^Tx) & \text{if } x^Tx < 1 \\ 0 & \text{otherwise} \end{cases}$$

where c_d is the volume of the unit d-dimensional sphere, and can be expressed by making use of the Gamma function. It is,

$$c_d = \frac{\pi^{d/2}}{\Gamma(1+d/2)} = \begin{cases} \pi^{d/2}/(d/2)! & \text{if } d \geq 0 \text{ is even} \\ \pi^{\lfloor d/2 \rfloor} 2^{\lceil d/2 \rceil}/d!! & \text{if } d \geq 0 \text{ is odd} \end{cases}$$

where $d!!$ is the double factorial.

Plugging $K_e(x)$ into Equation 4.1, the density of a query object q with respect to a set of objects O in subspace S can be estimated as

$$\hat{f}_S(q,O) = \frac{1}{|O|h_S^{|S|}} \sum_{o \in O \wedge \frac{dist_S(q,o)^2}{h_S^2} < 1} \left(\frac{1}{2}{c_{|S|}}^{-1}(|S|+2)(1 - \frac{dist_S(q,o)^2}{h_S^2}) \right) \tag{4.13}$$

where h_S is the bandwidth for subspace S.

Similar to calculating the bandwidth using Gaussian kernel in Section 4.2.2, we calculate h_S as follows.

$$h_S = \sigma_S \cdot h_{S\_opt}$$

As Silverman suggested [105], σ_S is a single scale parameter that equals to the root of the average marginal variance in S, and h_{S_opt} is the optimal bandwidth value which equals to $A(K)|O|^{-1/(|S|+4)}$, where $A(K) = \{8{c_{|S|}}^{-1}(|S|+4)(2\sqrt{\pi})^{|S|}\}^{1/(|S|+4)}$ for the Epanechnikov kernel.

We implemented *CSMiner* (Algorithm 4.2) using the Epanechnikov kernel for contrast subspace mining as follows. Given a subspace S, let $\mathcal{S}$ be the set of descendants of S in the subspace set enumeration tree using the standard deviation descending order. Then, $L_S(q \mid O_+)$ and $L_S(q \mid O_-)$ can be computed by Equation 4.13, and $L_S^*(q \mid O_+) =$

$$\frac{1}{|O|(\sigma'_{min}h'_{opt\_min})^\tau} \sum_{o \in O \wedge \frac{dist_S(q,o)^2}{(\sigma_S h'_{opt\_max})^2} < 1} \left(\frac{1}{2}{c_{\mathcal{S}}^{min}}^{-1}(d_{\mathcal{S}}^{max}+2)(1 - \frac{dist_S(q,o)^2}{(\sigma_S h'_{opt\_max})^2}) \right)$$

where $d_{\mathcal{S}}^{max} = \max\{|S'| \mid S' \in \mathcal{S}\}$, $c_{\mathcal{S}}^{min} = \min\{c_d \mid |S| < d \leq d_{\mathcal{S}}^{max}\}$, and the meaning of σ'_{min}, h'_{opt_min}, h'_{opt_max}, τ are the same as those in Equation 4.10.

Technically, the Epanechnikov kernel could also be implemented using the *CSMiner*-BPR approach (Algorithm 4.3). However, the performance improvement by the bounding-pruning-refining method would be less significant. The reason lies in the fact that on the one hand, different from using the Gaussian kernel that each object o has a non-zero likelihood contribution to the query object q, the contribution of o satisfying $\frac{dist_S(q,o)^2}{h_S^2} \geq 1$ is 0 (by the definition) to q when uses the Epanechnikov kernel. On the other hand, computing the neighborhood requires additional computational overhead.

Note that when using the Epanechnikov kernel, $\hat{f}_S(q, O_-) = 0$ if for any object $o \in O_-$, $\frac{dist_S(q,o)^2}{h_S^2} < 1$. Correspondingly, $LC_S(q) = \frac{\hat{f}_S(q,O_+)}{\hat{f}_S(q,O_-)} = +\infty$. Given data set O (composed by O_+ and O_-), we denote by $O_E^{+\infty}$ the set of objects whose maximum likelihood contrast, computed using the Epanechnikov kernel, is infinity. That is, $O_E^{+\infty} = \{o \in O \mid \exists S \text{ s.t. } LC_S(o) = +\infty\}$.

Let ℓ_G be the top-k contrast subspaces computed using the Gaussian kernel, and ℓ_E be the top-k contrast subspaces computed using the Epanechnikov kernel. For each object $q \in O$, we discover the top-10 inlying contrast subspaces and the top-10 outlying contrast subspaces of q using the Gaussian kernel and the Epanechnikov kernel, respectively, and compute $Sim(\ell_G, \ell_E)$ in each data set. For subspaces whose likelihood contrast values are infinity ($LC_S(q) = +\infty$), we sort them by $\hat{f}_S(q, O_+)$ in descending order. Table 4.11 and Table 4.12 list the minimum, maximum and average values of $Sim(\ell_G, \ell_E)$, as well as the ratio of $|O_E^{+\infty}|$ to $|O|$.

Data set O	$Sim(\ell_G, \ell_E)$			$\frac{\lvert O_E^{+\infty}\rvert}{\lvert O\rvert}$
	Min	Max	Avg	
BCW	0.168	0.980	0.539	590/683 = 0.864
CMSC	0.066	0.826	0.391	540/540 = 1.0
Glass	0.242	0.984	0.814	76/214 = 0.355
PID	0.620	1.0	0.924	1/768 = 0.001
Waveform	0.189	0.981	0.690	2532/5000 = 0.506
Wine	0.159	0.993	0.670	145/178 = 0.815

Table 4.11: Similarity between top-10 inlying contrast subspaces using different kernel functions in data set O ($\delta = 0.001$)

Data set O	$Sim(\ell_G, \ell_E)$			$\frac{\lvert O_E^{+\infty}\rvert}{\lvert O\rvert}$
	Min	Max	Avg	
BCW	0.239	1.0	0.916	67/683 = 0.098
CMSC	0.174	0.926	0.614	540/540 = 1.0
Glass	0.358	1.0	0.906	16/214 = 0.075
PID	0.655	1.0	0.938	1/768 = 0.001
Waveform	0.364	0.998	0.820	894/5000 = 0.179
Wine	0.209	1.0	0.804	40/178 = 0.225

Table 4.12: Similarity between top-10 outlying contrast subspaces using different kernel functions in data set O ($\delta = 0.001$)

From the results, shown in Tables 4.11 and 4.12, we can see that the value of $Sim(\ell_G, \ell_E)$ is related to $\frac{|O_E^{+\infty}|}{|O|}$. Specifically, the smaller the value of $\frac{|O_E^{+\infty}|}{|O|}$ the more similar ℓ_G and ℓ_E are. For example, when mining inlying contrast subspaces (Table 4.11), the values of $\frac{|O_E^{+\infty}|}{|O|}$ in BCW, CMSC, Waveform and Wine are larger than 0.5, which is larger than the values of $\frac{|O_E^{+\infty}|}{|O|}$ in PID and Glass, while the values of $Sim(\ell_G, \ell_E)$ are lower in BCW, CMSC, Waveform and Wine than those values in PID and Glass. When mining outlying contrast subspaces (Table 4.12), we note that the values of $\frac{|O_E^{+\infty}|}{|O|}$ are less than 0.1 in BCW, Glass and PID, while the values of $Sim(\ell_G, \ell_E)$ in these data sets are over 0.9.

Furthermore, we compute $Sim(\ell_G, \ell_E)$ in $O \setminus O_E^{+\infty}$ for each data set except CMSC, because for CMSC, $O \setminus O_E^{+\infty} = \emptyset$. From the results shown in Table 4.13 (inlying contrast subspace mining) and Table 4.14 (outlying contrast subspace mining), we can see that ℓ_G is more similar to ℓ_E without considering the objects whose maximum likelihood contrast is infinity.

Data set $O \setminus O_E^{+\infty}$	$Sim(\ell_G, \ell_E)$			$\lvert O \setminus O_E^{+\infty}\rvert$
	Min	Max	Avg	
BCW	0.643	0.980	0.922	93
Glass	0.720	0.984	0.929	138
PID	0.620	1.0	0.924	767
Waveform	0.324	0.981	0.754	2468
Wine	0.527	0.988	0.904	33

Table 4.13: Similarity between top-10 inlying contrast subspaces using different kernel functions in data set $O \setminus O_E^{+\infty}$ ($\delta = 0.001$)

Data set $O \setminus O_E^{+\infty}$	$Sim(\ell_G, \ell_E)$			$\lvert O \setminus O_E^{+\infty} \rvert$
	Min	Max	Avg	
BCW	0.561	1.0	0.934	616
Glass	0.629	1.0	0.925	198
PID	0.655	1.0	0.938	767
Waveform	0.437	0.998	0.836	4106
Wine	0.482	1.0	0.863	138

Table 4.14: Similarity between top-10 outlying contrast subspaces using different kernel functions in data set $O \setminus O_E^{+\infty}$ ($\delta = 0.001$)

4.5 Conclusions

In this chapter, we studied the novel and interesting problem of mining contrast subspaces to discover the aspects in which a query object is most similar to a class and dissimilar to another class. We demonstrated theoretically that the problem is very challenging and is MAX-SNP hard. We presented a heuristic method based on pruning rules and upper and lower bounds of likelihood and likelihood contrast. Our experiments on real data sets clearly show that our method improves contrast subspace mining substantially compared to the baseline method.

Chapter 5

Mining Contextual Outliers

The interpretability of outliers, that is, explaining in what ways and to what extent an object is an outlier, is a critical issue in outlier detection.

In this chapter, we develop a notion of *contextual outliers* on categorical data. Intuitively, a contextual outlier is a small group of objects that share strong similarity with a significantly larger reference group of objects on some attributes, but deviate dramatically on some other attributes. An example is: "Among computer science senior undergraduate students at university X, a small group of 3 students not enrolled in the data structure course is an outlier against the reference group of 128 students enrolled in the course." We systematically develop a model, including a concise representation, for contextual outlier analysis, and devise a detection algorithm that leverages the state-of-the-art data cube computation techniques. We conduct extensive experiments to evaluate our approach.

5.1 Motivation

More often than not, an analyst may want to see not only the outliers detected, but also insightful explanations about the outliers. Particularly, an analyst may want to know, for an outlier, a reference group of objects which the outlier deviates from in some aspects and shares similarity with in some other aspects, and a set of features manifesting the outlier's unusual/deviating behavior, the outlier degree, and the other similar outliers sharing the same context. Such contextual information can help an analyst to better understand and investigate individual outliers and propose action plans suitable for such outliers.

In addition to the application Scenario 3 of the Example 1.1, we provide one more

example in the following. Figure 5.1 shows a contextual outlier found from a data set containing all the 958 possible board configurations at the end of tic-tac-toe games. (For details on this data set, see Section 5.5.) Here, "x" and "o" mark positions occupied by the x and o players respectively, "b" marks positions not occupied by any player, and "∗" is a wildcard matching any value among "x", "o", and "b".

(a) An outlier group
cover: 2

b	∗	b
∗	x	∗
b	∗	o

(b) A reference group
cover: 35

x	∗	o
∗	∗	∗
o	∗	o

(c) Outlier instance c_1

b	x	b
o	x	x
b	o	o

(d) Outlier instance c_2

b	o	b
x	x	b
b	x	o

Figure 5.1: A contextual outlier in the tic-tac-toe data set.

Figure 5.1(c) and Figure 5.1(d) show two rare situations, where 3 of the 4 corners are not occupied by any players at the end of the games. These two rare situations can be summarized into an outlier group using wildcard symbol "∗", as shown in Figure 5.1(a). To manifest the outlier, Figure 5.1(b) shows a reference group where "x" occupies the left-top corner and "o" the other three corners. The reference group and the outlier group share the common feature that the right-bottom corner is taken by "o". The reference group, which is matched by 35 configurations, is dramatically more popular than the outlier group, which is matched by only two configurations.

The patterns of the outlier and reference groups suggest that it may be a good strategy to occupy as many corners in the game, since 33 out of the 35 configurations matching Figure 5.1(b) are won by "o".

The outlier group and the reference group cannot be found by the existing outlier

detection methods. To the best of our knowledge, even though an existing outlier detection method can detect the outlier group, it cannot find the reference group that clearly manifests the outliers.

We argue that the contextual information about outliers should be an integral component in the outlier detection process. Unfortunately, most of the existing outlier detection methods do not provide rich and detailed contextual information for outlier analysis.

In this chapter, we tackle the problem of contextual outlier detection on categorical data, and we do so by making three main contributions. First, we develop a notion of multidimensional contextual outliers to model the context of an outlier. Intuitively, a contextual outlier is a small group of objects that share similarity, on some attributes, with a significantly larger reference group of objects, but deviate dramatically on some other attributes. An example is: "Among the computer science senior undergraduate students at University X, a small group of 3 students not enrolled in the data structure course is an outlier against the reference group of 128 students enrolled in the course."

In contextual outlier detection, we identify not only the outliers, but also their associated contextual information including (1) comparing to what reference group of objects the detected object(s) is/are an outlier; (2) the attributes defining the unusual behavior of the outlier(s) compared against the reference group; (3) the population of similar outliers sharing the same context; and (4) the outlier degree, which measures the population ratio between the reference normal group and the outlier group.

Second, there may exist many contextual outliers in a data set, and some of them can be very similar or even "equivalent" to each other. It is clearly important to identify and reduce redundancy among outliers in order to assist users to effectively analyze the outliers. We develop an approach to systematically identify redundant contextual outliers and propose a concise representation of contextual outliers.

Third, we design a simple, yet effective algorithm that leverages the state-of-the-art data cube computation techniques. The focus of our method is to find outliers together with their contextual information. We conduct extensive experiments to evaluate the feasibility and usefulness of our approach.

The rest of the chapter is organized as follows. We propose the notion of contextual outliers in Section 5.2, and give a general analysis of the collection of contextual outliers in Section 5.3. We develop a contextual outlier detection algorithm in Section 5.4. We

evaluate our approach in Section 5.5. We conclude the chapter in Section 5.6.

5.2 Contextual Outliers

In this study, we consider outlier detection on multidimensional categorical data. Specifically, we consider a **base table** $T(D_1, \ldots, D_d)$, where $D_1, \ldots, D_d$ are categorical attributes with finite domains. We assume that each object is represented by a tuple in the base table and is associated with an identifier *tid*, which is used as a reference to the object only, and does not carry any other meaning. For an object $t \in T$, let $t.D_i$ and $t.tid$ denote the value of t on attribute D_i $(1 \leq i \leq d)$ and the identifier of t respectively.

A **subspace** is a subset of attributes. In order to summarize a group of objects, we add a wildcard meta-symbol $*$ to the domain of every attribute D_i $(1 \leq i \leq d)$. Symbol $*$ matches any possible values in the domain. A **group-by tuple** (or **group** for short) is a tuple $g = (g.D_1, \ldots, g.D_d)$ such that $g.D_i$ takes either a value in the domain of D_i or meta-symbol $*$. The **cover** of g is the set of objects in T matching g, that is, $cov(g) = \{t \in T \mid t.D_i = g.D_i \text{ for all } i \text{ such that } (1 \leq i \leq d \& g.D_i \neq *)\}$. The set $space(g) = \{D_i \mid 1 \leq i \leq d, g.D_i \neq *\}$ is called the **subspace** of g, and the set $avs(g) = \{D_i = g.D_i \mid 1 \leq i \leq d, g.D_i \neq *\}$ is called the **non-$*$ attribute-value set** (**AVS** for short) of group g. For an AVS V, we overload the operator $space(\cdot)$ by defining $space(V) = \{D_i \mid D_i \text{ occurs in } V\}$. Thus, $space(avs(g)) = space(g)$ always holds.

For two distinct groups g_1 and g_2, g_1 is an **ancestor** of g_2, and g_2 a **descendant** of g_1, denoted by $g_1 \succ g_2$, if $avs(g_1) \subset avs(g_2)$, that is, for every attribute D_i $(1 \leq i \leq n)$ such that $g_1.D_i \neq *$, we have $g_1.D_i = g_2.D_i$. We write $g_1 \succeq g_2$ if $g_1 \succ g_2$ or $g_1 = g_2$.

c-id	*City*	*Type*	*Branch*	*Package*
C1	L_1	T_1	B_1	Gold
C2	L_1	T_1	B_1	Silver
C3	L_1	T_1	B_1	Gold
C4	L_1	T_1	B_2	None
C5	L_2	T_2	B_1	Silver
C6	L_1	T_2	B_1	Gold

Table 5.1: A table T of a set of customers.

Property 5.1 (Monotonicity)**.** *For two groups g_1 and g_2 such that $g_1 \succ g_2$, $cov(g_1) \supseteq cov(g_2)$.* ∎

Example 5.1. Consider the base table T in Table 5.1, which contains the cities, customer-types, home-branches, and service packages of some investment service customers. For group $g = (L_1, T_1, *, *)$, $cov(g) = \{C1, C2, C3, C4\}$, $space(g) = \{\text{city}, \text{type}\}$, and $avs(g) = \{\text{city} = L_1, \text{type} = T_1\}$.

Let $g' = (L_1, T_1, B_1, *)$. Then, g is an ancestor of g' and g' a descendant of g, that is, $g \succ g'$. Moreover, $cov(g') = \{\text{C1, C2, C3}\} \subset cov(g)$. ∎

We are now ready to define contextual outliers. Intuitively, for a group of outlier objects, the contextual information consists of a group of reference objects that manifest the outlier group in a subspace. The comparison of the two groups in population size is also included.

Definition 5.1 (Contextual outlier)**.** *Let T be a base table, and g_r, g_o be two groups such that $space(g_r) = space(g_o) \neq \emptyset$. Given an **outlier degree threshold** $\Delta > 1$, the pair (g_r, g_o) is a **contextual outlier** if the **outlier degree** $deg(g_r, g_o) = \frac{|cov(g_r)|}{|cov(g_o)|} \geq \Delta$. We call g_r the **reference group**, g_o the **outlier group**, $out(g_r, g_o) = space(g_r) - space(cond(g_r, g_o))$ the **outlier subspace**, and $cond(g_r, g_o) = avs(g_r) \cap avs(g_o)$ the **shared AVS**. It is possible that $cond(g_r, g_o)$ is empty.* ∎

The shared AVS $cond(g_r, g_o)$ provides a context subspace for the outlier analysis about g_o. The objects in groups g_o and g_r belong to the same context subspace, that is, they take the same values on those attributes that occur in $cond(g_r, g_o)$. If $cond(g_r, g_o) = \emptyset$, g_r and g_o do not share any common features. In such a special case, g_o is a global outlier that is small in population and different from a large reference group g_r in space $space(g_o) = space(g_r)$.

The reference group g_r indicates the normal or dominating objects to which g_o is compared. The outlier group g_o and the outlier subspace $out(g_r, g_o)$ indicate the outlier objects $cov(g_o)$ and the attributes that manifest the deviation of g_o from g_r. The outlier degree measures how exceptional the group g_o is when compared to g_r. The larger the outlier degree is, the more outlying g_o is.

Example 5.2. In T (Table 5.1), let the outlier degree threshold be $\Delta = 2$. Then, $(g_r, g_o) = ((L_1, T_1, B_1, *), (L_1, T_1, B_2, *))$ is a contextual outlier, where $(L_1, T_1, B_1, *)$ is the reference group, $\{\text{city} = L_1, \text{type} = T_1\}$ is the shared AVS, $(L_1, T_1, B_2, *)$ is the outlier group, $\{\text{branch}\}$ is the outlier subspace, and the outlier degree is $deg(g_r, g_o) = 3$.

One object may be an outlier in more than one context. For example, customer C4 is an outlier in the above context and $((*, T_1, *, \text{Gold}), (*, T_1, *, \text{None}))$. ∎

Using only an outlier degree threshold may lead to many contextual outliers, since many groups of very small cover size, such as 1, may be identified as outliers. We will address this issue in Section 5.3.3.

The definitions and frequently used notations are summarized in Table 5.2.

Notation	***Description***
$T(D_1, \ldots, D_d)$	A base table, where $D_1, \ldots, D_d$ are categorical attributes with finite domains.
$*$	A widecard meta-symbol, matches any possible values in a domain.
$g(= (g.D_1, \ldots, g.D_d))$	A group-by tuple (or group for short), $g.D_i$ takes either a value in the domain of D_i or meta-symbol $*$.
$cov(g)$	The cover of g, the set of objects in T matching g, that is, $cov(g) = \{t \in T \mid t.D_i = g.D_i$ for all i such that $(1 \leq i \leq d \& g.D_i \neq *)\}$.
$space(g)$	The subspace of g, $space(g) = \{D_i \mid 1 \leq i \leq d, g.D_i \neq *\}$.
$avs(g)$	The non-$*$ attribute-value set (AVS for short) of g, $avs(g) = \{D_i = g.D_i \mid 1 \leq i \leq d, g.D_i \neq *\}$.
$g_1 \succ g_2$	g_1 is an ancestor of g_2, and g_2 a descendant of g_1, if $avs(g_1) \subset avs(g_2)$.
(g_r, g_o)	A contextual outlier, where g_r is the reference group and g_o is the outlier group.
$deg(g_r, g_o)$	The outlier degree of contextual outlier (g_r, g_o), $deg(g_r, g_o) = \frac{\lvert cov(g_r)\rvert}{\lvert cov(g_o)\rvert}$.
Δ	The outlier degree threshold.
$cond(g_r, g_o)(= avs(g_r) \cap avs(g_o))$	The shared AVS of contextual outlier (g_r, g_o).
$out(g_r, g_o)(= space(g_r) - space(cond(g_r, g_o)))$	The outlier subspace of contextual outlier (g_r, g_o).
$\alpha(g_r, g_o)$	The significance of a contextual outlier (g_r, g_o) (Definition 5.6).
$\Phi(g_1, g_2)$	The assembly of an ordered pair (g_1, g_2) (Definition 5.7).

Table 5.2: Summary of definitions and frequently used notations in Chapter 5.

5.3 Contextual Outlier Analysis

Enumerating all possible contextual outliers in a base table is ineffective due to three reasons. First, some contextual outliers are highly similar and even equivalent to each other. Outlier detection is often followed by business actions, which are expensive to perform. Including redundant outliers may lead to unnecessary extra cost in the "analysis and actions" process and may also overwhelm users. Second, it is informative and important to analyze the relationships among outliers. The contextual outliers in a data set may not be independent of each other. A systematic analysis of the relationships among outliers may, for example, support business decisions based on the relationships among a collection of outliers, instead of on individual outliers only. Third, as mentioned at the end of Section 5.2, using an outlier degree threshold alone may lead to many insignificant contextual outliers, which may overwhelm users in practice. In this section, we conduct contextual outlier analysis to address the above three aspects.

5.3.1 Redundancy Removal Using Closures

We immediately observe the following:

Lemma 5.1 (Non-closure attributes)**.** *For two contextual outliers* (g_{r_1}, g_{o_1}) *and* (g_{r_2}, g_{o_2}) *in a base table* T*, if* $g_{r_1} \succ g_{r_2}$*,* $g_{o_1} \succ g_{o_2}$*,* $cov(g_{r_1}) = cov(g_{r_2})$*, and* $cov(g_{o_1}) = cov(g_{o_2})$*, then* $deg(g_{r_1}, g_{o_1}) = deg(g_{r_2}, g_{o_2})$*.* ∎

In the two contextual outliers (g_{r_1}, g_{o_1}) and (g_{r_2}, g_{o_2}) in Lemma 5.1, the two groups g_{r_1} and g_{r_2} capture the same set of objects. Hence (g_{r_1}, g_{o_1}) is redundant given (g_{r_2}, g_{o_2}) or vice versa. Since $g_{r_1} \succ g_{r_2}$, g_{r_2} contains some extra attributes in addition to those in g_{r_1}. Hence g_{r_2} is more informative and descriptive than g_{r_1} as a reference group. It is better to include (g_{r_2}, g_{o_2}) for outlier analysis.

Definition 5.2 (Closure group/outlier)**.** *Given a base table* T*, a group* g *is a* ***closure group*** *if for any descendant group* $g' \prec g$*,* $cov(g') \subset cov(g)$*.* (g_r, g_o) *is called a* ***closure outlier*** *if there does not exist another contextual outlier* $(g_{r'}, g_{o'})$ *such that* $g_{r'} \prec g_r$*,* $g_{o'} \prec g_o$*,* $cov(g_r) = cov(g_{r'})$*, and* $cov(g_o) = cov(g_{o'})$*.* ∎

Example 5.3 (Closure group/outlier)**.** In table T (Table 5.1), for contextual outliers $g_{u_1} = ((*, T_1, B_1, *), (*, T_1, B_2, *))$ and $g_{u_2} = ((L_1, T_1, B_1, *), (L_1, T_1, B_2, *))$, since

$cov((*, T_1, B_1, *)) = cov((L_1, T_1, B_1, *))$ and $cov((*, T_1, B_2, *)) = cov((L_1, T_1, B_2, *))$, g_{u_1} is redundant given g_{u_2}. The reference group in g_{u_2} is a closure one. It can be verified that g_{u_2} is a closure outlier. ■

We now establish a relationship between closure groups and closure outliers.

Theorem 5.1 (Closure group/outlier)**.** *Contextual outlier (g_r, g_o) is a closure outlier if and only if either g_r or g_o is a closure group.*

Proof. (If) There are two cases. In the first case, g_r is a closure group. Then, there does not exist $(g_{r'}, g_{o'})$ such that $g_{r'} \prec g_r$ and $cov(g_r) = cov(g_{r'})$. In the second case, g_o is a closure group. Then, there does not exist $(g_{r'}, g_{o'})$ such that $g_{o'} \prec g_o$ and $cov(g_o) = cov(g_{o'})$. In both cases, (g_r, g_o) satisfies the definition of closure outliers.

(Only-if) We prove by contradiction. Assume that (g_r, g_o) is a closure outlier, but neither g_r nor g_o is a closure group. Then, there exist $g_{r'}$ and $g_{o'}$ such that $g_{r'} \prec g_r$, $g_{o'} \prec g_o$, $cov(g_r) = cov(g_{r'})$ and $cov(g_o) = cov(g_{o'})$. This contradicts the assumption that (g_r, g_o) is a closure outlier. ■

We can generalize the idea of closure outliers to reduce redundancy further.

Example 5.4. Consider T (Table 5.1) and the outlier degree threshold $\Delta = 3$. C4 is in two contextual outliers, $g_{u_1} = ((L_1, *, B_1, *), (L_1, *, B_2, *))$ with $deg(g_{u_1}) = 4$, and $g_{u_2} = ((L_1,$ $T_1,$ $B_1, *)$, $(L_1,$ $T_1,$ $B_2, *))$ with $deg(g_{u_2}) = 3$. Comparing to g_{u_1}, the reference group in g_{u_2} is more specific and thus is closer to the outlier group and more informative. ■

We generalize the observation in Example 5.4 and introduce the notion of tight outlier, which is essentially an outlier and its most specific reference group.

Definition 5.3 (Tight outlier)**.** *Let T be a base table and (g_r, g_o) a contextual outlier with respect to outlier degree threshold Δ. We call (g_r, g_o) a **tight outlier** if there does not exist another outlier $(g_{r'}, g_{o'})$ with respect to threshold Δ such that $g_r \succ g_{r'}$ and $cov(g_o) = cov(g_{o'})$.* ■

Corollary 5.1. *A tight outlier is a closure outlier.* ■

As shown in Example 5.4, not every closure outlier is tight.

5.3.2 Relationships among Outliers

An outlier group may have more than one reference group in the same outlier space. Consider two contextual outliers (g_{r_1}, g_o) and (g_{r_2}, g_o) such that $space(g_{r_1}) = space(g_{r_2})$, $cond(g_{r_1}, g_o) = cond(g_{r_2}, g_o)$. The two outliers have the same outlier subspace, since $out(g_{r_1}, g_o) = space(g_{r_1}) - space(cond(g_{r_1}, g_o)) = space(g_{r_2}) - space(cond(g_{r_2}, g_o)) = out(g_{r_2}, g_o)$. If $|cov(g_{r_1})| > |cov(g_{r_2})|$, g_{r_1} is a stronger reference group than g_{r_2} and g_o deviates more from g_{r_1} than from g_{r_2}. Thus, (g_{r_2}, g_o) is redundant given (g_{r_1}, g_o). In words, we only need to report the reference group that an outlier deviates most. This is modeled in the following notion of strong outliers.

Definition 5.4 (Strong outlier)**.** *A contextual outlier (g_r, g_o) is a **strong outlier** if there does not exist another outlier $(g_{r'}, g_{o'})$ such that $space(g_r) = space(g_{r'})$, $cond(g_r, g_o) = cond(g_{r'}, g_{o'})$, $cov(g_o) = cov(g_{o'})$, and $|cov(g_{r'})| > |cov(g_r)|$.* ■

Example 5.5 (Strong outlier)**.** For T (Table 5.1) and the outlier degree threshold $\Delta = 2$, consider contextual outliers $g_{u_1} = ((*, *, *, \text{Gold}), (*, *, *, \text{None}))$ with $deg(u_1) = 3$, and $g_{u_2} = ((*, *, *, \text{Silver}), (*, *, *, \text{None}))$ with $deg(g_{u_2}) = 2$. u_1 is stronger than g_{u_2} in outlier degree, and they have the same shared AVS. g_{u_2} is redundant given g_{u_1}. In fact, g_{u_1} is a strong outlier. ■

A group of objects may appear in multiple strong and tight contextual outliers, such as $C4$ in T (Table 5.1). To comprehensively understand an outlier group, we can group all contextual outliers together with respect to the same set of outlier objects.

Definition 5.5 (Contextual outlier group)**.** *Given an outlier degree threshold $\Delta > 1$, a set of contextual outliers $\{(g_{r_1}, g_{o_1}), \ldots, (g_{r_l}, g_{o_l})\}$ is called a **contextual outlier group** if (1) $(g_{r_1}, g_{o_1}), \ldots, (g_{r_l}, g_{o_l})$ are all strong and tight contextual outliers; (2) $cov(g_{o_1}) = \cdots = cov(g_{o_l})$; and (3) there does not exist a proper superset of contextual outliers satisfying the above two requirements. We denote by $Context_\Delta(g_o)$ the contextual outlier group $\{(g_{r_1}, g_{o_1}), \ldots, (g_{r_l}, g_{o_l})\}$ such that $cov(g_{o_1}) = \cdots = cov(g_{o_l}) = g_o$.*

*The **(geometric) average degree** of a contextual outlier group $Context_\Delta(g_o) = \{(g_{r_1}, g_{o_1}), \ldots, (g_{r_l}, g_{o_l})\}$ is $deg(g_o, \Delta) = \sqrt[l]{\prod_{i=1}^{l} deg(g_{r_i}, g_{o_i})}$. In particular, when $l = 0$, we have $deg(g_o, \Delta) = 1$.* ■

We use the geometric average degree instead of the arithmetic average to aggregate the outlierness of an outlier group in multiple contexts because the contexts in general may

not be completely independent. The geometric average prefers outlier groups whose outlier degrees in multiple contexts are more balanced against those that are extremely outlying in only a small number of contexts, but not the others. Of course other aggregate functions might also be adopted.

5.3.3 Finding Significant Outliers

As mentioned earlier, using only an outlier degree threshold may lead to many contextual outliers, since many groups of very small cover size and with a small difference from large reference groups, such as on one attribute, may be identified as outliers. To tackle the problem and avoid overwhelming users by many insignificant outliers, we need to test the statistical significance of contextual outliers. In other words, we use statistical test to avoid reporting groups that are caused by random noise.

Intuitively, we use the global distribution in the base table as the background distribution by ignoring the tuples taking the same values with the reference group, and measure the statistical significance of the outlier group. For global outliers, we assume the uniform distribution as the background.

Definition 5.6 (Outlier significance)**.** *Let (g_r, g_o) be a contextual outlier in a base table T. The **background distribution** of (g_r, g_o) in subspace $out(g_r, g_o)$ is the distribution of tuples in $T - \{t \in T \mid t.D = g_r.D, \forall D \in out(g_r, g_o)\}$, if $cond(g_r, g_o) \neq \emptyset$, and the uniform distribution, otherwise.*

*The **significance** of a contextual outlier (g_r, g_o), denoted by $\alpha(g_r, g_o)$, is the p-value of the null hypothesis H_0: g_o has been generated from the tuples in $cov(cond(g_r, g_o)) - cov(g_r)$ according to the background distribution in space $out(g_r, g_o)$.* ∎

In the above definition, $cov(cond(g_r, g_o)) - cov(g_r)$ is the set of tuples matching reference group g_r in subspace $cond(g_r, g_o)$ but not in subspace $out(g_r, g_o)$. The smaller the p-value $\alpha(g_r, g_o)$ is, the more significant the outlier is. The following shows the details of significance calculation.

Consider a contextual outlier (g_r, g_o). Let p_{g_o} be the probability of AVS $avs(g_o) - avs(g_r)$ in the background distribution. If the background distribution is uniform, then

$$p_{g_o} = \frac{1}{\prod_{D \in space(out(g_r, g_o))}(|D| - 1)}.$$

Let $m = |T - \{t \in T \mid t.D = g_r.D, \forall D \in out(g_r, g_o)\}|$ be the number of tuples matching r in subspace $cond(g_r, g_o)$, but not in subspace $out(g_r, g_o)$. Then, we have

$$\alpha(g_r, g_o) = \sum_{i=|cov(g_o)|}^{m} p_{g_o}^i (1 - p_{g_o})^{m-i}$$

Example 5.6 (Outlier significance)**.** Given T (Table 5.1), an outlier degree threshold $\Delta = 3$, and a significance threshold $s = 0.01$. We perform the outlier significance test on a strong and tight contextual outlier, $g_{u_1} = ((L_1, *, B_1, Gold), (L_1, *, B_2, None))$ with $deg(g_{u_1}) = 3$ and $out(g_r, g_o) = \{Branch, Package\}$.

According to the previous discussion, we have $p_{g_o} = \frac{1}{(|Branch|-1)(|Package|-1)} = \frac{1}{(2-1)(3-1)} = \frac{1}{2}$, $m = 6 - 3 = 3$, $i = 1$, and thus $\alpha(g_r, g_o) = (\frac{1}{2})^1(\frac{1}{2})^2 + (\frac{1}{2})^2(\frac{1}{2})^1 + (\frac{1}{2})^3(\frac{1}{2})^0 = \frac{3}{8} = 0.375$. Since $\alpha(g_r, g_o) > s$, g_{u_1} is not a significant contextual outlier here. ■

Based on the above discussion, we can define the problem of contextual outlier detection as, given a base table T, an outlier degree threshold $\Delta > 1$, and a significance threshold $s > 0$, find all strong and tight context outliers (g_r, g_o) such that $\alpha(g_r, g_o) < s$.

5.4 Detection Algorithms

In this section, we develop an algorithm for contextual outlier detection. We observe that group-bys are essential units in both data cube computation and contextual outlier analysis, so we can exploit state-of-the-art data cube techniques in detecting contextual outliers.

Our method is inspired by Theorem 5.1. Since every closure contextual outlier must have either the reference group or the outlier group as a closure group, we can find all closure groups in the base table first, and then use the closure groups to assemble contextual outliers.

Finding closure groups and closure patterns has been well studied in frequent pattern mining [91, 112] and data cube computation [72]. Given a base table T, we can adopt a state-of-the-art algorithm, such as the DFS algorithm in [72], to find all closure groups. Therefore, hereafter, we focus on how to use closure groups to assemble contextual outliers.

We define the following assembly operation to extract the common attributes in two closure groups.

Definition 5.7 (Assembly)**.** *Given two closure groups g_1 and g_2 on a base table T, the* ***assembly*** *of the ordered pair (g_1, g_2), denoted by $\Phi(g_1, g_2) = (g_1', g_2')$, is the ordered pair of*

groups such that for every attribute $D \in space(g_1) \cap space(g_2)$, $g_1'.D = g_1.D$ *and* $g_2'.D = g_2.D$*; for any other attribute* $B \in (T - space(g_1) \cap space(g_2))$, $g_1'.B = g_2'.B = *$. *We call* g_1' *and* g_2' *the* ***reference group*** *and the* ***outlier group*** *of the assembly, respectively.* ■

Example 5.7 (Assembly)**.** In T in Table 5.1, for closure groups $g_1 = (L_1, T_1, B_1, *)$ and $g_2 = (*, T_2, B_1, *)$, the assembly of the ordered pair $\Phi(g_1, g_2) = ((*, T_1, B_1, *), (*, T_2, B_1, *))$. ■

It is easy to verify the following, which shows that contextual outliers are fixpoints for Φ.

Corollary 5.2. *For any contextual outlier* (g_r, g_o), $\Phi(g_r, g_o) = (g_r, g_o)$. ■

Since the assembly operator takes an ordered pair of closure groups as the input and produces an ordered pair as the output, in general, $\Phi(g_1, g_2) \neq \Phi(g_2, g_1)$. Instead, the reference (outlier) group of the assembly $\Phi(g_1, g_2)$ is the outlier (reference) group of $\Phi(g_2, g_1)$. The assembly operation has the following nice property.

Corollary 5.3. *For every tight and strong contextual outlier* (g_r, g_o)*, there exists a unique ordered pair of closure groups* $(g_{r'}, g_{o'})$ *such that* $\Phi(g_{r'}, g_{o'}) = (g_r, g_o)$, $cov(g_r) = cov(g_{r'})$, *and* $cov(g_o) = cov(g_{o'})$.

Proof. According to Theorem 5.1, either g_r or g_o must be a closure group. Without loss of generality, let us assume g_r is a closure group, and thus let $g_{r'} = g_r$. If g_o is not a closure group, there exists a unique closure group $g_{o'}$ such that $cov(g_o) = cov(g_{o'})$ and $space(g_o) \subseteq space(g_{o'})$. Therefore, $\Phi(g_{r'}, g_{o'}) = \Phi(g_r, g_{o'}) = \Phi(g_r, g_o)$. Using Corollary 5.2, we have $\Phi(g_{r'}, g_{o'}) = \Phi(g_r, g_o) = (g_r, g_o)$. ■

Corollary 5.3 enables us to assemble closure groups into contextual outliers. Algorithm 5.1 presents the pseudocode of our detection method, *COD* (for Contextual Outlier Detection), which is explained in detail as follows.

For each closure group g_o, we consider all the other closure groups g_r such that $\Phi(g_r, g_o)$ is a contextual outlier. Obviously, $|cov(g_o)|$ cannot be larger than $\frac{l}{\Delta}$, where l is the largest cover size among all closure groups (calculated in Line 1). For each of such closure groups g_o, we iterate over all the other closure groups g_r such that $|cov(g_r)| \geq \Delta |cov(g_o)|$ and $\alpha(g_r, g_o)$ passes the significance threshold (the inner loop, Lines 5-11). To facilitate the

Algorithm 5.1 *COD*: the contextual outlier detection algorithm.

Require: $\mathcal{G}$: the complete set of closure groups; Δ: the outlier degree threshold; and s: the significance threshold

Ensure: the set of tight and strong contextual outliers

1: let $l = \max_{g \in \mathcal{G}}\{|cov(g)|\}$;
2: let CO be the set of contextual outliers; set $CO = \emptyset$;
3: **for** each closure group g_o such that $|cov(g_o)| \leq \frac{l}{\Delta}$ **do**
4: create a set L of tight and strong contextual outliers, set $L = \emptyset$;
5: **for** each closure group g_r such that (1) $|cov(g_r)| \geq \Delta |cov(g_o)|$; (2) $space(g_r) \subseteq space(g_o)$ or $space(g_r) \supseteq space(g_o)$; and (3) $\alpha(g_r, g_o) \leq s$ **do**
6: let $(g_{r'}, g_{o'}) = \Phi(g_r, g_o)$;
7: **if** $cov(g_r) = cov(g_{r'})$ **and** $cov(g_o) = cov(g_{o'})$ **and** there is no outlier in L that is stronger or tighter than $\Phi(g_r, g_o)$ **and** $\alpha(\Phi(g_r, g_o)) \leq s$ **then**
8: insert $\Phi(g_r, g_o)$ into L;
9: remove from L any outliers that $\Phi(g_r, g_o)$ is stronger or tighter than;
10: **end if**
11: **end for**
12: $CO = CO \cup L$;
13: **end for**
14: **return** CO;

access of closure groups according to their cover size, we sort all the closure groups in cover size ascending order.

For a pair of closure groups (g_r, g_o), we compute the assembly $\Phi(g_r, g_o) = (g_{r'}, g_{o'})$. We only consider the contextual outliers $\Phi(g_r, g_o) = (g_{r'}, g_{o'})$ such that $cov(g_r) = cov(g_{r'})$ and $cov(g_o) = cov(g_{o'})$. Otherwise, the outlier will be considered by some other closure groups according to Corollary 5.3 (the first two conditions in Line 7). If $\Phi(g_r, g_o)$ is strong and tight given the other contextual outliers using g_o as the outlier group (the second last condition in Line 7), and is statistically significant (that is, the significance is less than or equal to the significance threshold) (the last condition in Line 7), then we keep $\Phi(g_r, g_o)$ (Line 8), and use it to remove those contextual outliers found before that are not as strong or tight as $\Phi(g_r, g_o)$ (Line 9).

After the inner loop, all contextual outliers using g_o as the outlier group are computed, and the tight and strong ones are kept in L. Those tight and strong outliers are moved to CO for outputting later. The iteration continues until all closure groups that may be outlier groups are examined.

COD checks every pair of closure groups (g_r, g_o) such that $\frac{|cov(g_r)|}{|cov(g_o)|} \geq \Delta$. The correctness

follows with Corollary 5.3. Moreover, the algorithm is cubic in time with respect to the number of closure groups in T, that is, $|\mathcal{G}|$ in the algorithm. The problem of computing closure groups in a table is #P-complete by a polynomial reduction from the #P-complete problem of frequent maximal pattern mining [122]. *COD* is overall pseudo-polynomial.

5.5 Experimental Results

In this section, we report our empirical evaluation of *COD* using both real-world and synthetic data sets. All experiments were conducted on a PC computer with an Intel Core Duo E8400 3.0 GHz CPU and 4 GB main memory, running Microsoft Windows 7 operating system. The algorithms were implemented in C++ using Microsoft Visual Studio 2010.

We cannot identify any existing method that solves the exact same problem. The focus of our method is to find outliers with contextual information. Consequently, this paper does not intend to compete with the existing methods with respect to outlier detection accuracy or recall. We do compare our method with *LOF* [20] in Section 5.5.2.

5.5.1 Results on Real Data Sets

We use categorical data sets from the UCI repository [43]. We report the results on six data sets: adult, mushroom-sc, solar-flare, tic-tac-toe, credit-approval, and hayes-roth. Some statistics of the data sets are summarized in Table 5.3. Note that data set mushroom-sc is made from the data set mushroom. We select all attributes related to mushrooms' shape and color in our experiments. For the credit-approval and adult data sets, we keep the categorical attributes, and remove the numerical attributes. We also remove the records that having missing values in the selected data sets.

Data Set	*# of objects*	*# of attributes*	*# of closure groups*	*QC time (s)*
Adult	30,162	8	73,282	28.678
Mushroom-sc	8,124	8	6,265	1.879
Solar-flare	1,389	10	7,770	2.136
Tic-tac-toe	958	9	42,711	12.903
Credit-approval	690	8	5,707	1.446
Hayes-roth	160	4	277	0.047

Note: QC time refers to the time used to find all closure groups, that is, $|\mathcal{G}|$ in algorithm 5.1.

Table 5.3: The statistics of the data sets.

COD takes two parameters, the outlier degree threshold Δ and the significance threshold s. We report the results with respect to different combinations of Δ and s values for each data set. We evaluate *COD* in four aspects: outlier case studies, outlier analysis effectiveness (redundancy reduction and significance filtering), efficiency, and scalability on dimensionality.

Case Studies

We demonstrate the effectiveness of context outlier detection using case studies on the data sets tic-tac-toe, hayes-roth and mushroom-sc.

An Example on Tic-tac-toe: The tic-tac-toe data set is composed of all the 958 possible board configurations at the end of tic-tac-toe games. It is assumed that "x" plays first and then "o" plays. There are 9 attributes, each corresponding to one tic-tac-toe square. An attribute takes "x" if the corresponding square is occupied by x, "o" if occupied by o, and "b" if blank.

Figures 5.1(a) and 5.1(b) show a contextual outlier group and a reference group, respectively. The outlier degree of this contextual outlier is 17.5 and the significance is 2.76×10^{-12}. This outlier corresponds to an end-of-game board where most of the corners (3 out of 4) are not occupied by any players. This is a rare occurrence, since occupying corner squares is a common winning strategy in tic-tac-toe games. The only two such configurations are c_1 and c_2, shown in Figures 5.1(c) and 5.1(d), respectively.

An Example on Hayes-roth: The hayes-roth data set records the information about 160 people on four attributes [6, 43]. The first attribute, hobby, takes values uniformly at random [43] and we thus ignore it in our analysis, that is, all groups take value $*$ on the attribute. The description for the other three attributes are adopted from [6]: the second attribute, age, takes values in $\{30, 40, 50, g_r > 0\}$ (the meaning of value "$g_r > 0$" is unspecified in [6]); attribute education takes values in {junior-high, high-school, trade-school, college}; and the last attribute, marital-status, takes values in {single, married, divorced, widowed}. Table 5.4 shows some interesting contextual outliers with respect to $\Delta = 5$ and $s = 10^{-8}$. For the sake of simplicity, in Table 5.4, we categorize the significance score of a contextual outlier s' into three significance levels, low($1 \leq \frac{s}{s'} < 10$), medium ($10 \leq \frac{s}{s'} < 10^2$) and high ($\frac{s}{s'} \geq 10^2$).

Outlier-id	*Reference group* g_r	*Outlier group* g_o	$deg(g_r, g_o) = \frac{\lvert cov(g_r)\rvert}{\lvert cov(g_o)\rvert}$	*Significance level*
c_1	(∗, ∗, high-school, single)	(∗, ∗, high-school, divorced)	5.7 = 34/6	*high*
c_2	(∗, ∗, high-school, single)	(∗, ∗, trade-school, single)	5.7 = 34/6	*high*
c_3	(∗, ∗, high-school, single)	(∗, ∗, high-school, widowed)	8.5 = 34/4	*medium*
c_4	(∗, ∗, trade-school, married)	(∗, ∗, trade-school, widowed)	8.0 = 16/2	*low*
c_5	(∗, ∗, junior-high, divorced)	(∗, ∗, college, divorced)	8.0 = 16/2	*low*
c_6	(∗, 40, junior-high, ∗)	(∗, 40, college, ∗)	8.5 = 34/4	*medium*
c_7	(∗, 40, junior-high, ∗)	(∗, 40, trade-school, ∗)	5.7 = 34/6	*high*
c_8	(∗, 40, junior-high, ∗)	(∗, 50, junior-high, ∗)	5.7 = 34/6	*high*
c_9	(∗, 50, high-school, ∗)	(∗, 50, college, ∗)	8.0 = 16/2	*low*
c_{10}	(∗, 30, ∗, married)	(∗, 30, ∗, widowed)	8.5 = 34/4	*medium*

Table 5.4: Some contextual outliers on data set hayes-roth ($\Delta = 5$, $s = 10^{-8}$). The underlined attributes indicate the shared AVSs.

In Table 5.4, outliers c_1 and c_2 share the same reference group. The reference group consists of 34 people whose marital-status is "single" and who have high-school degrees. Outlier group c_1 is a collection of 6 "divorced" college graduates and outlier group c_2 is a collection of 6 "single" trade school graduates. Outlier c_5 is interesting: among people who are divorced, those who are college graduates are outliers compared to those with high school degrees. c_9 shows that, among people who are 50-years old, the 2 with college degrees are outliers compared to the 16 with high school degrees. Another interesting outlier is c_{10}: in the age group of 30, the 4 people widowed are outliers compared to the 34 people married. Please note that, in the whole data set, there are 59 of high-school, 59 of junior-school, 29 of trade-school and 13 of college graduates. Given $\Delta = 5$, those of trade-school and college graduates are not outliers comparing to those of high-school and junior-school. The outliers can only be explained well using the contextual information.

An Example on Mushroom-sc: The mushroom-sc data set is made from data set mushroom [43]. We select all attributes that related to mushrooms' shape and color in our experiment. The mushroom-sc data set contains 8,124 individual mushroom records on 8 attributes. The attributes that we selected are (*cap-shape, stalk-shape, cap-color, gill-color, stalk-color-above-ring, stalk-color-below-ring, veil-color, spore-print-color*). Please refer to [43] for the detailed value description of each attribute. Table 5.5 shows some interesting contextual outliers with respect to $\Delta = 50$ and $s = 10^{-3}$. Similar as the example on Hayes-roth data set, for the sake of simplicity, in Table 5.5, we

Outlier-id	Contextual Outlier	$deg(g_r, g_o) = \frac{\lvert cov(g_r)\rvert}{\lvert cov(g_o)\rvert}$	*Significance level*
c_1	g_r: $(*, *, *, *, \underline{White}, White, \underline{White}, *)$ g_o: $(*, Enlarging, *, White, \underline{White}, Brown, \underline{White}, White)$	$55.0 = \frac{3520}{64}$	high
c_2	g_r: $(\underline{Convex}, *, *, *, *, White, \underline{White}, *)$ g_o: $(\underline{Convex}, Enlarging, *, *, *, Red, \underline{White}, White)$	$63.3 = \frac{2024}{32}$	high
c_3	g_r: $(Convex, *, *, *, *, *, \underline{White}, \underline{Brown})$ g_o: $(Sunken, \underline{Enlarging}, *, *, White, White, \underline{White}, \underline{Brown})$	$62.5 = \frac{1000}{16}$	high
c_4	g_r: $(*, \underline{Tapering}, *, Buff, *, \underline{White}, \underline{White}, White)$ g_o: $(*, \underline{Tapering}, *, Brown, White, \underline{White}, \underline{White}, Purple)$	$54.0 = \frac{864}{16}$	high
c_5	g_r: $(\underline{Convex}, Tapering, *, *, White, \underline{White}, \underline{White}, *)$ g_o: $(\underline{Convex}, Enlarging, *, *, red, \underline{White}, \underline{White}, White)$	$54.0 = \frac{816}{16}$	high

Table 5.5: Some contextual outliers on data set mushroom-sc ($\Delta = 50, s = 10^{-3}$). The underlined attributes indicate the shared AVSs.

also categorize the significance score of a contextual outlier s' into three significance levels, low($1 \leq \frac{s}{s'} < 10$), medium ($10 \leq \frac{s}{s'} < 10^2$) and high ($\frac{s}{s'} \geq 10^2$).

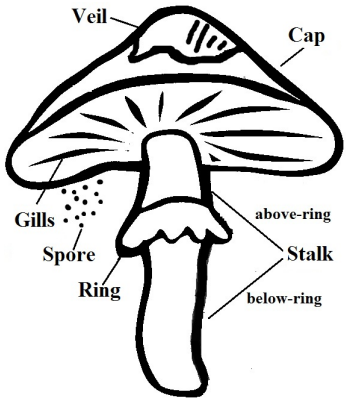


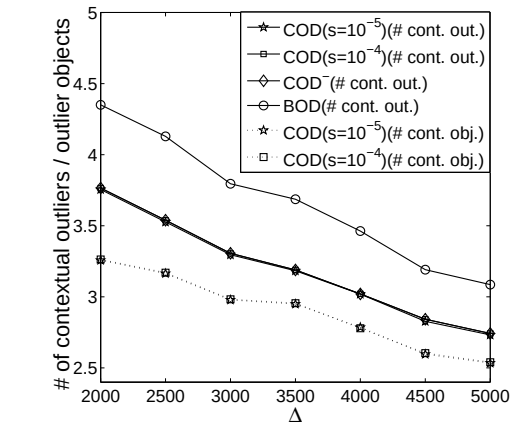
Figure 5.2: Names for the Parts of a Mushroom

Figure 5.2 shows the popular names used for different parts of a mushroom. In Table 5.5, outlier c_1 shows that, among the mushrooms that are white in the stalk above ring and the veil, the 64 mushrooms that are brown in the stalk below ring are outliers, compared to the 3,520 mushrooms that are white in the stalk below ring. Outlier c_2 tells us that, among the mushrooms that are white in the veil, brown in the spore print, and have convex caps, a small group of 32 mushrooms that are red in the stalk below ring are outliers, compared to a large group of 2,204 mushroom that are white in the stalk below ring. Outlier c_4 is interesting: among the mushrooms that are white in both the stalk below ring and the veil, and have tapering stalks, the 16 mushrooms that are brown in the gill and purple in the spore print are outliers, compared to the 864 mushrooms that are buff in the gill and white in the spore print.

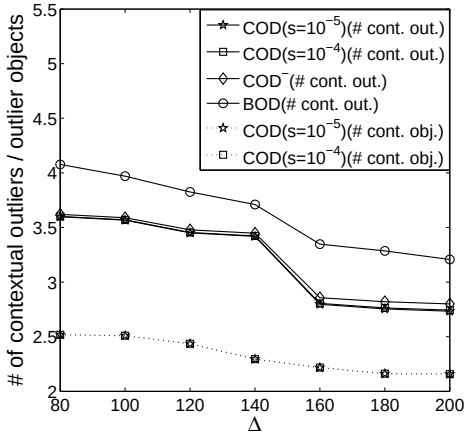
Effectiveness of Redundancy Reduction and Significance Filtering

To evaluate the effectiveness of the outlier analysis techniques developed in Section 5.3, in addition to *COD*, we consider two simplified versions, COD^- and *BOD*. Both of them work the same as *COD* except for the following changes. COD^- does not apply the significance test for contextual outliers. *BOD* does not apply either the redundancy removal techniques or the significance test.

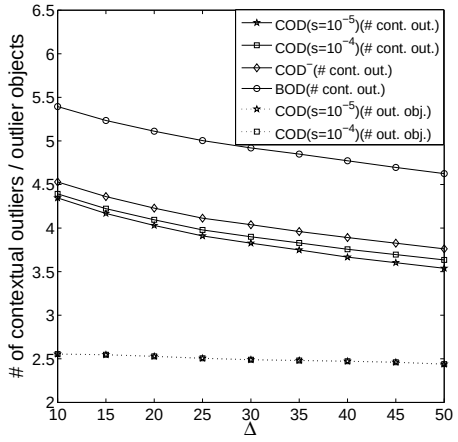
Figure 5.3 plots the number of contextual outliers and the number of outlier objects with respect to different Δ values. An object is called an outlier if it is contained in the outlier group of a context outlier. We set $s = 10^{-4}$ and $s = 10^{-5}$ respectively in *COD*. No contextual outliers exist when $\Delta \geq 50$ in the tic-tac-toe data set and when $\Delta \geq 20$ in the hayes-roth data set. *COD* outputs much less contextual outliers than COD^- and *BOD* in all cases. The number of outlier objects is small, and decreases roughly linearly as Δ increases. One object may be contained in multiple contextual outliers. Multiple contextual outliers containing the same outlier object identify how the outlier object deviates from the majority in different subspaces.



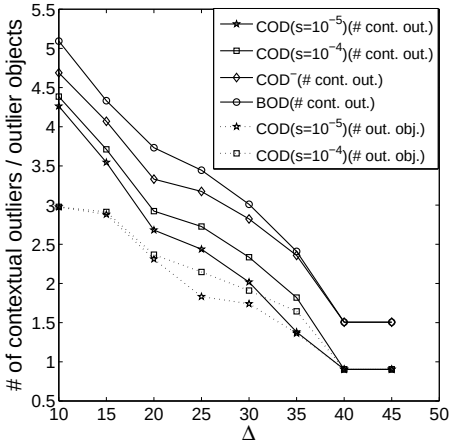
(a) Adult



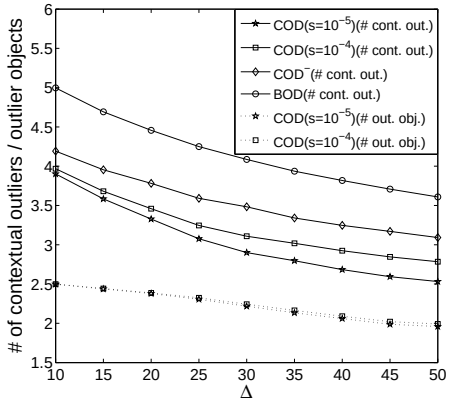
(b) Mushroom-sc



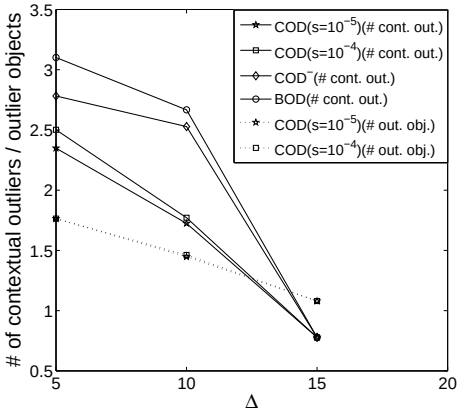
(c) Solar-flare



(d) Tic-tac-toe



(e) Credit-approval



(f) Hayes-roth

Figure 5.3: The number of contextual outliers / outlier objects w.r.t different Δ. The y-axis (# of contextual outliers / outlier objects) is in logarithmic scale.

Table 5.6 shows the number of outlier objects, outlier groups and contextual outliers with respect to different significance threshold s in *COD*. As expected, the lower the significance threshold, the less outliers are reported. Moreover, the number of outlier groups is much smaller than that of contextual outliers. This shows that outlier groups are a concise summarization of contextual outliers.

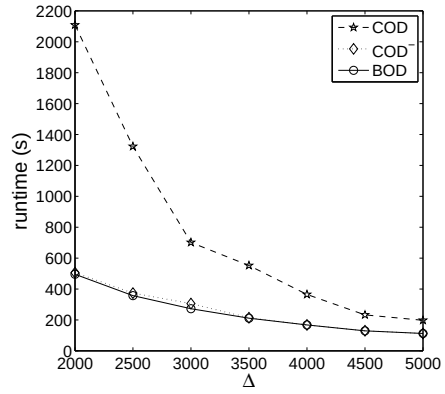
s	*Item name*	*Adult* $\Delta = 2000$	*Mushroom-sc* $\Delta = 60$	*Solar-flare* $\Delta = 150$	*Tic-tac-toe* $\Delta = 30$	*Credit-approval* $\Delta = 70$	*Hayes-roth* $\Delta = 15$
10^{-3}	# of out. obj.	1,831	346	161	222	72	12
	# of out. grp.	2,429	676	212	222	98	6
	# of cont. out.	5,823	6,659	896	664	488	6
10^{-5}	# of out. obj.	1,807	346	124	55	34	12
	# of out. grp.	2,399	676	170	55	48	6
	# of cont. out.	5,686	6,658	487	104	167	6
10^{-7}	# of out. obj.	1,388	346	113	46	27	12
	# of out. grp.	1,745	676	134	46	27	6
	# of cont. out.	2,882	6,571	304	84	62	6
10^{-9}	# of out. obj.	1,314	346	107	42	22	12
	# of out. grp.	1,506	672	118	42	14	6
	# of cont. out.	2,213	5,872	255	76	24	6

Table 5.6: The number of contextual outliers w.r.t. significance threshold.

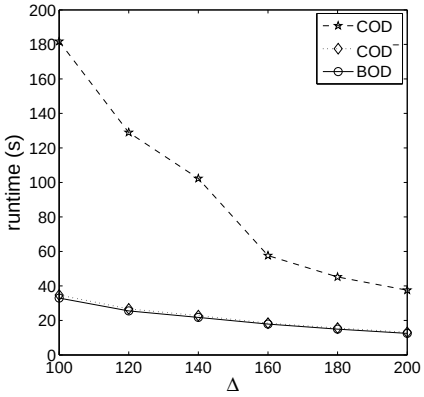
Efficiency

Figure 5.4 compares the runtime of *COD*, *COD*$^-$ and *BOD* on the four real data sets with respect to various Δ thresholds. The closure group computation time is reported in Table 5.3, and is not included in Figure 5.4.

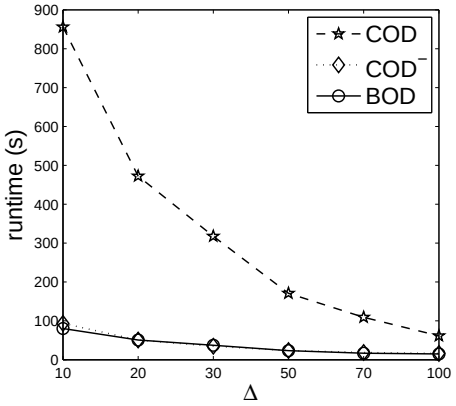
The runtime of *BOD* is the least among the three methods. *COD*$^-$ takes a very small amount of extra time on top of *BOD* to identify tight and strong outliers. *COD* uses extra time on top of *COD*$^-$ to test the statistical significance. When Δ is small, the number of contextual outliers is large, and thus *COD* and *COD*$^-$ need more extra runtime. When Δ increases, the runtime difference between these three methods decreases quickly. In practice, Δ should not be set to a small value, since one often likes to find outliers that deviate from significantly larger trends. We notice that different setting of significance threshold s does not affect the runtime of *COD* in a noticeable way.



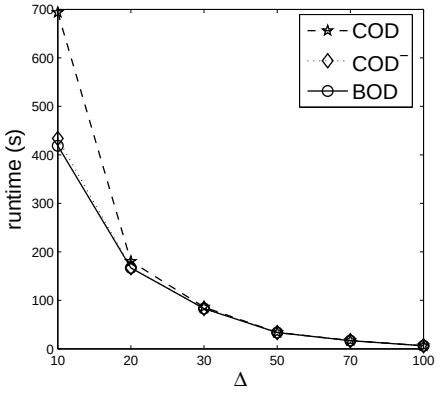
(a) Adult



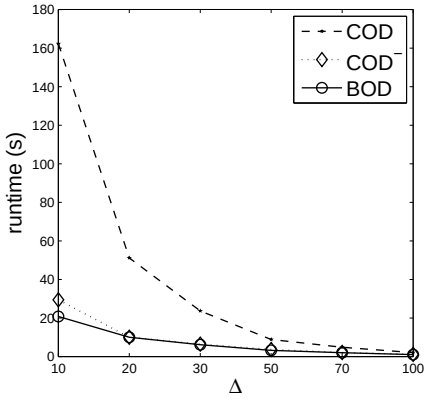
(b) Mushroom-sc



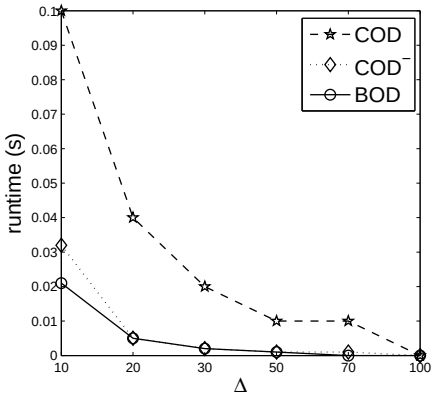
(c) Solar-flare



(d) Tic-tac-toe



(e) Credit-approval

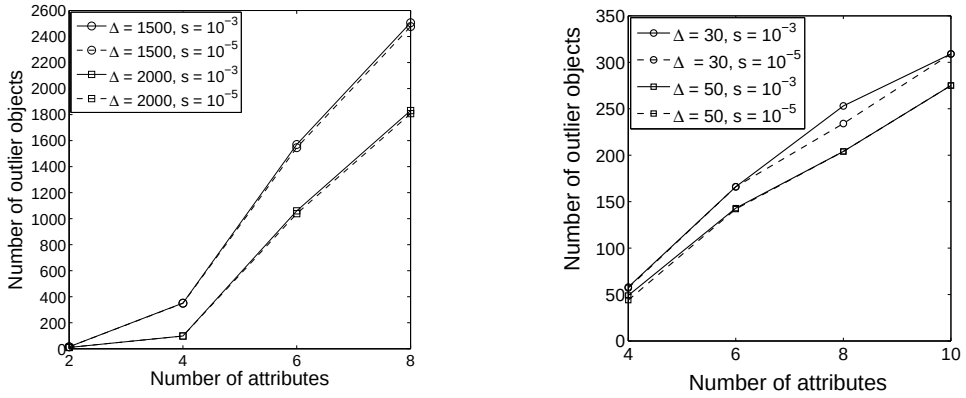


(f) Hayes-roth

Figure 5.4: The runtime of *COD*, *COD*$^-$ and *BOD* on the six real data sets ($s = 10^{-3}$ in *COD*).

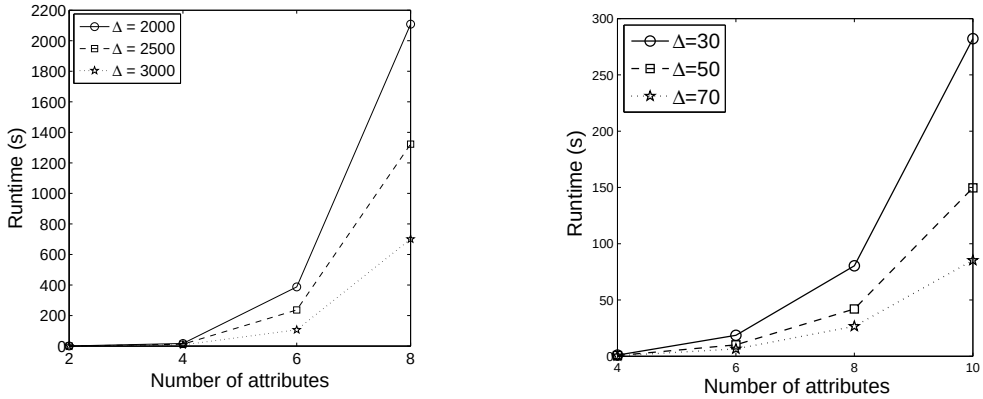
Scalability on Dimensionality

We test the scalability of *COD* with respect to dimensionality on the real data sets. We keep the first k attributes and vary k from 2 to the dimensionality of the data sets. We report the results on two data sets, adult and solar-flare, which have the largest number of tuples and the highest dimensionality, respectively, among the six real data sets. Figure 5.5 shows the number of outlier objects with respect to dimensionality; and Figure 5.6 shows the runtime with respect to dimensionality. Both the number of outlier objects and runtime increase when the dimensionality increases, since spaces of higher dimensionality can accommodate more outliers.



(a) Adult (b) Solar-flare

Figure 5.5: The number of outliers of *COD* with respect to dimensionality.

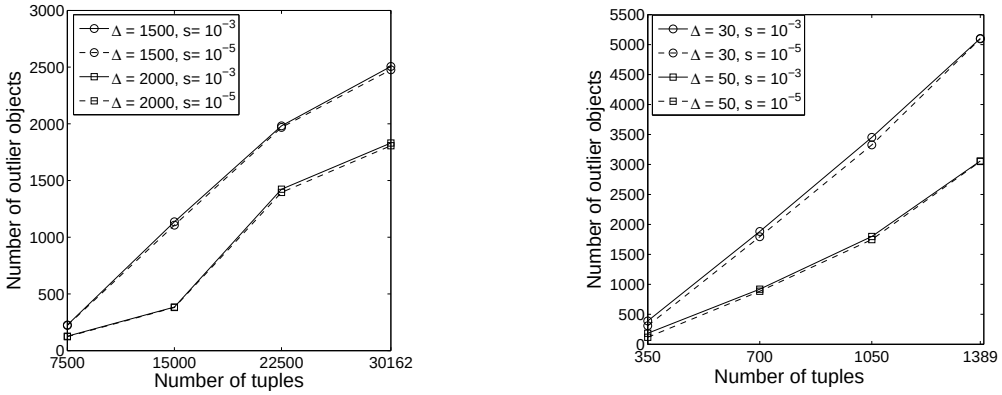


(a) Adult (b) Solar-flare

Figure 5.6: The runtime of *COD* with respect to dimensionality ($s = 10^{-3}$).

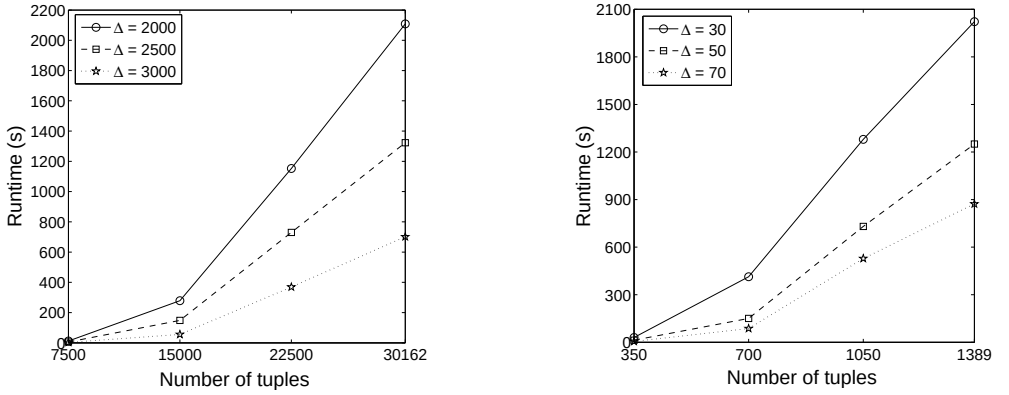
Scalability on Number of Tuples

We also test the scalability of *COD* with respect to number of tuples on the real data sets. Similar to Section 5.5.1, we report the results on two real data sets, adult and solar-flare. In adult data set, we keep the first k tuples and vary k from 7500 to the number of tuples of the data set. Similarly, in solar-flare data set, we keep the first k tuples as well and vary k from 350 to the number of tuples of the data set. Figure 5.7 shows the number of outlier objects with respect to number of tuples; and Figure 5.8 shows the runtime with respect to number of tuples. Both the number of outlier objects and runtime increase when the number of tuples increases, since more tuples can accommodate more outliers.



(a) Adult (b) Solar-flare

Figure 5.7: The number of outliers of *COD* with respect to number of tuples.



(a) Adult (b) Solar-flare

Figure 5.8: The runtime of *COD* with respect to number of tuples ($s = 10^{-3}$).

5.5.2 Results on Synthetic Data Sets

In order to test the accuracy of *COD*, we use synthetic data sets, since most real data sets do not have the complete ground-truth information. We also use synthetic data sets to test the scalability of *COD*.

Given the outlier degree threshold $\Delta > 0$, the number of subspace outliers m_s, the number of global outlier m_g, the dimensionality d, the cardinality c, and the number of tuples in the data set, we generate a synthetic data set in three steps. First, we generate m_s subspace outliers (including the reference groups and the outlier groups) satisfying the outlier degree threshold requirement. For an outlier, the shared AVS and the outlier subspace are chosen randomly. Second, we generate m_g global outliers (including the reference groups and the outlier groups) in the same manner. Again, the outlier subspaces are chosen randomly. Last, we inject independent and uniformly distributed data to fulfill the requirement on number of tuples. A synthetic data set generated as such carries the ground truth on contextual outliers. In our experiments, we fix $\Delta = 50$, $m_s = 850$, $m_g = 150$, $d = 10$, $c = 100$. By default, the number of tuples is set to 1 million.

Methods & *threshold setting*	*Precision*		*Recall*	
	Avg.	*Std.*	*Avg.*	*Std.*
$COD(\Delta = 50, s = 10^{-7})$	78.63%	10.77%	100%	0
$LOF(MinPtsLB = 10, MinPtsUB = 100)$	68.59%	11.59%	73%	13.04%

Table 5.7: The precision & recall of *COD*, and comparison with *LOF*.

Table 5.7 shows the precision of *COD*. We repeat the experiments 10 times on 10 synthetic data sets generated independently using the same parameters, and report the average and the standard deviation of the precision and recall. The results show that our method always detects all outliers in the ground-truth (100% recall). At the same time, *COD* has a good precision. Please note that, although we implant the seed outliers in the synthetic data set as the ground-truth, the noise injected in the synthetic data set may lead to some outliers that are not included in the ground-truth.

In Table 5.7, we also compare our method *COD* with *LOF* [20] using the implementation in Weka [49] and Hamming distance as the distance measure. The parameters $MinPtsLB$ and $MinPtsUB$ are set according to the suggestions in [20]. *COD* outperforms *LOF* on the synthetic data sets in both accuracy and recall. Please note that *LOF* cannot provide

contextual information for outliers, and is not designed specifically for contextual outlier detection. In fact, we cannot identify any existing method that solves the exact same problem.

Figure 5.9 tests the scalability of *COD* with respect to the number of tuples in the database. We generate data sets of different sizes, from 100 thousand to 1 million tuples. We keep the other parameters the same. Again, for each configuration, we repeat the experiment 10 times, and report results in the figure. The results clearly shows that *COD* is scalable with respect to database size.

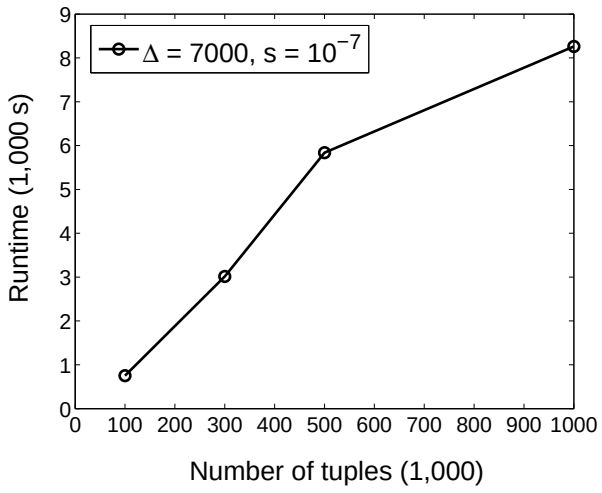


Figure 5.9: The scalability of *COD* on synthetic data sets w.r.t. number of tuples

5.6 Conclusions

In this chapter, we proposed a framework for contextual outlier detection. Our focus was to improve the interpretability of outliers. In particular, we argued that the context of an outlier should include a shared AVS, a reference group, an outlier group, and an outlier degree measure. Moreover, we developed a concise representation for contextual outliers and presented a detection algorithm leveraging the state-of-the-art data cube computation techniques.

The proposed contextual outlier concept can be very useful in real world applications. First, outlier detection tools can be developed based on this concept. The results returned by such tools can provide the analysts answers to three questions at one time. (1) What

are the outliers? Answered by the outlier groups. (2) How outlying the identified outliers are? Answered by the outlier degree. (3) What makes the identified outliers outlying? Answered by the reference group, the shared AVS and the outlier subspace. Second, some existing outlier detection tools can improve the interpretability of their results by adopting the ideas, such as the shared AVS, the outlier subspace and the reference group, in the proposed contextual outlier concept.

Chapter 6

Mining Markov Blanket Based Outliers

Mining outliers in subspaces is an important task. It is challenging to select meaningful subspaces for outlier detection and assess outliers in different subspaces.

In this chapter, to address the above challenge, we propose a Markov blanket based method. Built on the concept of Markov blankets in Bayesian networks, we can find outliers from only d Markov blanket subspaces (d is the dimensionality) instead of $2^d - 1$ non-empty subspaces. By learning a local Bayesian network for a Markov blanket subspace, we derive an outlier score measure based on joint probabilities of Bayesian networks, and then propose the *MiCOM* algorithm to mine subspaces and outliers simultaneously. Our experimental results using synthetic and real-world data validate the effectiveness, efficiency and scalability of our *MiCOM* algorithm in comparison with the state-of-the-art subspace outlier detection methods.

6.1 Motivation

In many scenarios, a data object is described by a large number of attributes. Traditional outlier detection models attempt to detect the deviation of outliers with respect to the full attribute space, and thus fail to deal with the situation where only a subset of relevant attributes (a subspace) provides the meaningful information for each object while the remaining attributes are irrelevant for this object [20, 65, 82]. For instance, the attributes like "smoking" and "coughing" are relevant for detecting the abnormal patients with a lung

disease. Other attributes such as "age" and "skin humidity" may be irrelevant for the detection of this type of outlier, but are relevant for discovering the abnormal patients with the "dehydration" status.

To tackle this problem, mining outliers in subspaces has been explored [20, 82]. The key challenge on subspace outlier detection is how to select subsets of relevant attributes (subspaces). In general, given d attributes, there are $2^d - 1$ non-empty subspaces. Most of the state-of-the-art subspace outlier detection algorithms, such as *OUTRES* [82], *HiCS* [65] and *CMI* [17], employ an Apriori-style subspace search scheme, and thus incur very expensive or even prohibitive cost on high-dimensional data. And then, the challenge is that, given two outliers in two different subspaces, how we can assess and compare their outlyingness.

To tackle the challenges, in this chapter, we propose a Markov blanket based method to select meaningful subspaces for outlier detection. The concept of Markov blankets was first introduced by Pearl [93] in Bayesian networks. In Bayesian networks, for any attribute X, its Markov blanket is composed of those attributes that have the highest correlation with X, and thus makes X independent of the remaining attributes. Therefore, an attribute and its Markov blanket form a natural subspace for outlier detection. For instance, in health surveillance data, only the attributes in the Markov blanket of "dehydration", such as "age" and "skin humidity" that are highly correlated with "dehydration", provide the meaningful information for detecting the abnormal "dehydration" status. The remaining attributes not in the Markov blanket of "dehydration" are irrelevant to"dehydration".

Our main idea of Markov blanket based outlier detection is as follows. Assuming a data set O of d attributes, we regard the Markov blanket of an attribute and the attribute itself as the Markov blanket subspace and identify d Markov blanket subspaces. Then, by learning d local Bayesian networks independently over those Markov blanket subspaces, we derive an outlyingness measure based on the joint probabilities of the Bayesian networks to assess outliers in multiple subspaces. We make the following contributions.

1. We use Markov blanket to dramatically reduce the number of candidate subspaces from $2^d - 1$ possible subspaces to d Markov blanket subspaces.

2. To detect outliers in Markov blanket subspaces, we propose three greedy strategies to learn d local Bayesian networks on Markov blanket subspaces, and then derive an outlyingness measure over multiple subspaces based on the joint probabilities of local

Bayesian networks.

3. We propose the *MiCOM* algorithm to simultaneously mine subspaces and outliers.

4. Our experimental results using synthetic and real-world data validate the effectiveness, efficiency and scalability of our *MiCOM* algorithm in comparison with the state-of-the-art subspace outlier detection methods.

The rest of the chapter is organized as follows. Section 6.2 proposes the Markov blanket subspaces. Section 6.3 develops our *MiCOM* algorithm. Section 6.4 reports our experimental results on synthetic and real-world data. Finally, Section 6.5 concludes the paper.

6.2 Markov Blankets for Subspace Discovery

6.2.1 Preliminaries

The concept of Markov blankets was first introduced by Pearl [93] in Bayesian networks. We first review Bayesian networks and then Markov blankets in detail. Then, we justify why Markov blankets can be used in subspace outlier detection. Hereafter, the terms "variable" and "attribute" will be used interchangeably.

Given a training data set $O = \{o_1, o_2, \cdots, o_n\}$ containing n training instances, D is defined on the set of dimensions $D = \{D_1, D_2, \cdots, D_d\}$ where d is the dimensionality. Let P be a joint probability distribution of a set of random variables D via a directed acyclic graph G. We call the triplet $\langle D, G, P\rangle$ a Bayesian network if $\langle D, G, P\rangle$ satisfies the Markov condition: every variable is independent of any subset of its non-descendant variables conditioned on its parents in G [93]. A simple Bayesian network of Lung Cancer as an example is shown in Figure 6.1.

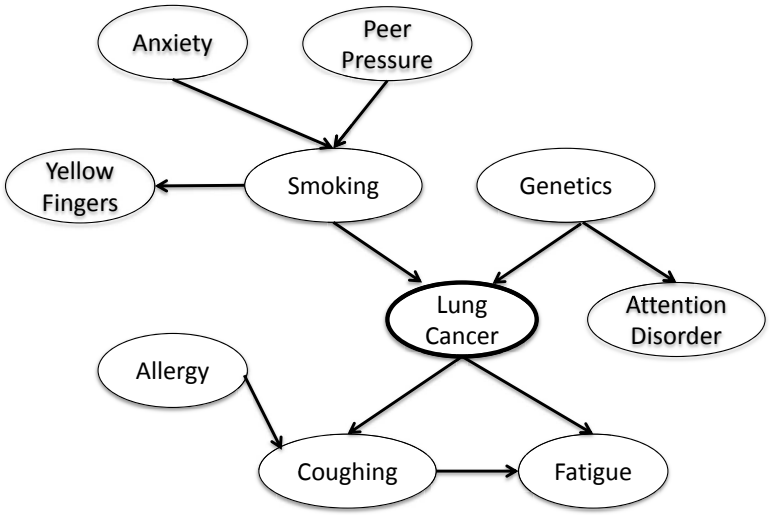


Figure 6.1: Example of a Bayesian network [75]

With the Markov condition, a Bayesian network encodes the joint probability P over a set of variables D and decomposes it into a product of the conditional probability distributions over each variable given its parents in G. Assuming $Pa(D_i)$ is the set of parents of $D_i (1 \leq i \leq d)$ in G, the joint probability P is

$$P(D_1, D_2, \cdots, D_d) = \prod_{i=1}^{d} P(D_i | Pa(D_i))$$

.

Definition 6.1 (Faithfulness)**.** *[93] A Bayesian network $\langle D, G, P \rangle$ is said to satisfy the faithfulness condition if and only if every conditional independence entailed by G is also present in P.* ∎

Theorem 6.1. *[93] If a Bayesian network satisfies the faithfulness condition, then the Markov blanket of a variable X in the Bayesian network is the set of children, parents, and spouses of X.* ∎

6.2.2 Markov Blanket Subspaces

In subspace ouliter detection, the existing studies showed that outliers only appear in correlated subspaces [82]. The challenge is that the number of candidate subspaces for

possible correlated subspaces is exponentially large. To tackle this challenge, we propose the concept of Markov blanket subspaces for efficient selection of correlated subspaces.

In a Bayesian network, for any attribute X, its Markov blanket satisfies the following property.

Property 6.1. *[5] In a Bayesian network $\langle D, G, P\rangle$, if a subset $MB(X) \subseteq D - \{X\}$ is a Markov blanket of X, then the following holds*

$$DV_{KL}(P(X|MB(X))| \mid P(X|D)) = 0$$

, where DV_{KL} is the Kullback-Leibler (KL) divergence [70] between the estimated distribution $P(X|MB(X))$ and the true distribution $P(X|D)$, given by

$$DV_{KL}(p||q) = \sum_{f \in D} p(f) log \frac{p(f)}{q(f)}$$

. ■

From Property 6.1, we can see that, given an attribute X, its Markov blanket renders X to be statistically independent from all the remaining attributes in a Bayesian network. Therefore, only the attributes in the Markov blanket of X are highly correlated with X, and provide the meaningful information to X. The remaining attributes are irrelevant to X.

For instance, in Figure 6.1, only the attributes in the Markov blanket of "Lung cancer" provide the meaningful information for detecting the abnormal patients with a lung cancer disease, while the remaining attributes not in this Markov blanket are irrelevant to"Lung cancer". Furthermore, according to [5], for any attribute, its Markov blanket is unique in a faithful Bayesian network.

Property 6.2. *[5] If a Bayesian network satisfies the faithfulness condition, then the Markov blanket of each attribute is unique.* ■

Accordingly, in Bayesian networks, an attribute and its Markov blanket form a natural subspace that makes this attribute independent of the remaining attributes.

Definition 6.2 (Markov blanket subspaces)**.** *A Markov blankt subspace consists of an attribute and its Markov blanket.* ■

Given a data set O with d attributes, it is natural for us to consider d Markov blanket subspaces as candidate subspaces for outlier detection. Interestingly, to efficiently select meaningful subspaces from $2^d - 1$ possible subspaces, Müller et al. [82, 65, 17] proposed the concept of high contrast subspaces. In the following, we analyze the connections between Markov blanket subspaces and high contrast subspaces.

Given a subspace $s \subset D$ and $\forall D_i \in s$, a high contrast subspace is measured by comparing conditional probability density $P(D_i|s - \{D_i\})$ to the corresponding marginal probability density $P(D_i)$ [65]. In contrast, if a subspace s is an uncorrelated one, it satisfies the following equation

$$P(D_1, D_2, \cdots, D_{|s|}) = \prod_{i=1}^{|s|} P(D_i).$$

For an uncorrelated subspace s, the joint probability density $P(D_1, D_2, ..., D_{|s|})$ is equal to the product of the marginal probability of each attribute in s, and thus the contrast between the marginal density $P(D_i)$ and its corresponding conditional probability density $P(D_i|s - \{D_i\})$ is equal to 1, that is,

$$\forall D_i \in s, P(D_i)/P(D_i|s - \{D_i\}) = 1.$$

Accordingly, an uncorrelated subspace is not a high contrast subspace, and does not contain any meaningful outliers [82, 65]. A high contrast subspace is only the subspace that shows high dependencies between attributes in this subspace. Using the correlation of dimensions in a subspace as an objective function for computing subspace contrasts, if a subspace s is a candidate of high contrast subspace, it should satisfy the following equation

$$\exists D_i \in s, P(D_i)/P(D_i|s - \{D_i\}) > 1.$$

The number of candidates of high contrast subspaces is exponentially large. Furthermore, to find high contrast subspaces, in any subspace s, we need to search for condition sets from $2^{|s|}$ attribute sets for each attribute $D_i \in s$ to compute the contrast between its conditional probability densities and its marginal densities. It is very computationally costly or even prohibitive.

With the discussion above, we can get the observation that for any attribute $D_i \in D(1 \leq i \leq d)$, its Markov blanket $MB(D_i)$ is the minimum condition set that makes the Markov blanket subspace, $\{D_i\} \cup MB(D_i)$, be a high contrast subspace. The explanation is that attributes in $MB(D_i)$ are highly correlated with D_i, then $P(D_i)/P(D_i|MB(D_i)) > 1$

holds. So the Markov blanket subspace, $\{D_i\} \cup MB(D_i)$, is a high contrast subspace. Assuming $\forall s \subset D$, $MB(D_i) \subset s$, according to Property 6.1, we get $P(D_i)/P(D_i|s) = P(D_i)/P(D_i|MB(D_i))$. Thus, the $MB(D_i)$ is the minimum condition set that makes $P(D_i)/P(D_i|MB(D_i)) > 1$ hold.

The observation illustrates that for any attribute $D_i \in D(1 \leq i \leq d)$, its Markov blanket not only provides the meaningful information to D_i, but also forms a minimum size of condition set for computing the marginal density $P(D_i)$ and the conditional probability density $P(D_i|MB(D_i))$. Accordingly, we can exactly use the d Markov blanket subspaces as the candidates of high contrast subspaces for outlier detection instead of enumerating $2^d - 1$ possible subspaces.

Table 6.1 lists the frequently used notations in this chapter.

Notation	Description
$D = \{D_1, \ldots, D_d\}$	a d-dimensional space
$O = \{o_1, o_2, \cdots, o_n\}$	a training data set with n training instances
X	an arbitrary attribute, $X = D_i(1 \leq i \leq d) \in D$
P	a joint probability distribution
G	a directed acyclic graph
s	a subspace, $s \subset D$ and $\forall D_i \in s$
$Pa(D_i)$	the set of parents of $D_i(1 \leq i \leq d)$ in G
$MB(X)$	a Markov blanket subspace with respect to attribute X

Table 6.1: Summary of frequently used notations in Chapter 6

6.3 Mining Outliers in Markov Blanket Subspaces

Figure 6.2 gives an overview of our proposed framework for outlier mining in Markov blanket subspaces with the key steps as follows.

1. Learn Markov blanket subspaces and construct a local Bayesian network in a Markov blanket subspace;

2. Derive a comparable outlierness measure over multiple subspaces;

3. Mine and characterize outliers in Markov blanket subspaces.

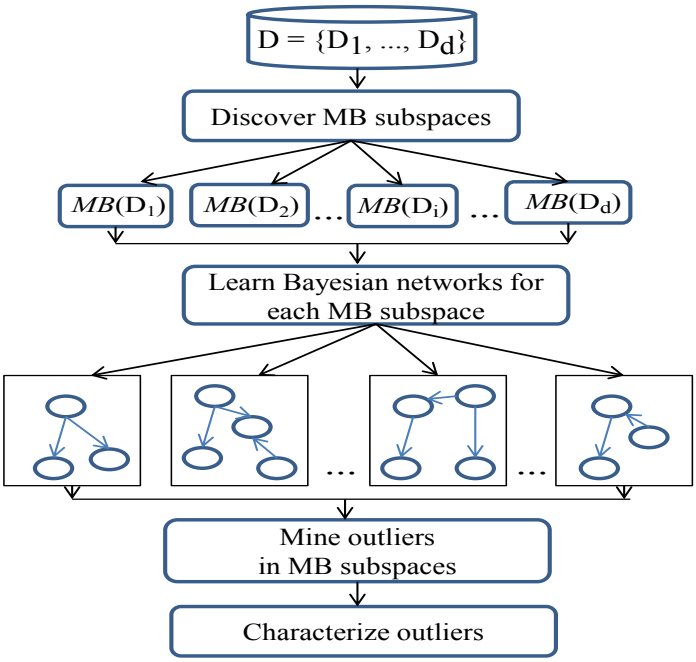


Figure 6.2: Framework of mining outliers in Markov blanket subspaces

6.3.1 Learning Bayesian Networks in Subspaces

We present three greedy strategies to learn a local Bayesian network in a Markov blanket subspace for outlier detection. Firstly, we employ the state-of-the-art MMMB methods to discover Markov blankets for each attribute (details in [5]). The MMMB algorithm identifies parents and children of a target attribute as the first step, and then discovers the spouses of the target attribute. Secondly, with the discovered Markov blankets, we learn a local Bayesian network for each Markov blanket subspace through the phases of structure learning and parameter learning.

Structure learning. We present a greedy method to learn a local Bayesian network structure in a Markov blanket subspace with the following strategies.

Strategy 1: if $A \in MB(X)$, $X \in PC(A)$ and $PC(A) - \{X\}$ is not empty, then add an edge in $G : X \rightarrow A$, or $A \rightarrow B, B \in \{PC(A) - \{X\}\}$.

Since the directions of spouses of X, the children of both the spouses and X, and X can be determined in the Markov blanket discovery phase, due to a well-known V-structure [93]: $A \rightarrow C \leftarrow B$. In this V-structure, A and B is initially independent, but will become

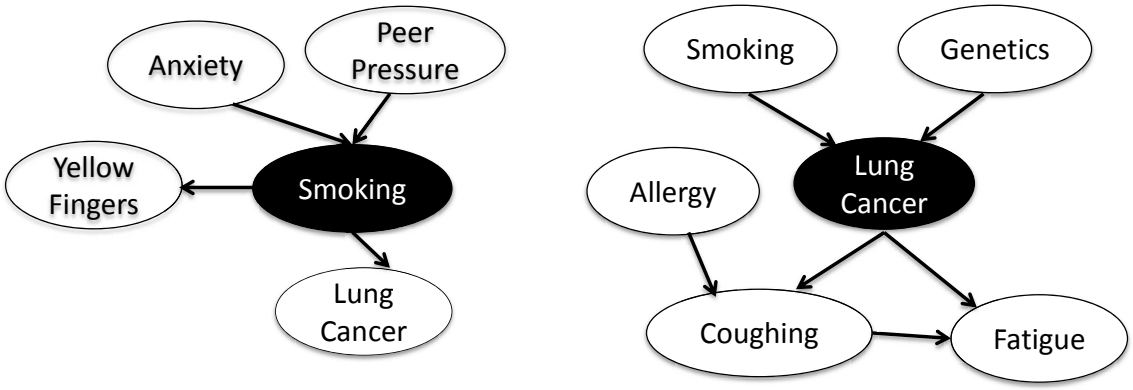


Figure 6.3: Examples of two Bayesian networks in Markov blanket subspaces

dependent when conditioned on C.

Strategy 2: if $PC(A) - \{X\}$ is empty, our method only considers adding an edge $A \rightarrow X, or\ X \rightarrow A, A \in MB(X)$.

Strategy 3: Assume $DAG(X)$ is the directed acyclic graph learned from the Markov blanket subspace of X, and $DAG(A)$ is to the Markov blanket subspace of A. If $A \in PC(X)$, $X \in PC(A)$, and edge $A \rightarrow X$ exists in $DAG(X)$, then the edge $A \rightarrow X$ should be in $DAG(A)$.

Strategy 3 keeps the directions between attributes be consistent. This is the key step for us to use Bayesian network inference to mine outliers in Markov blanket subspaces. If the directions between attributes are consistent in each local Bayesian network, then the joint probability for each attribute can be kept consistent. For example, in Figure 6.3, we can get two local Bayesian networks for "Lung-cancer" and "Smoking". In the two Bayesian networks, the directions between "Lung-cancer" and "Smoking" should be consistent.

Based on the score and our greedy search strategies, the structure learning problem can be formally expressed as follows: given a complete training data set of instances O, find a $DAG\ G^*$ such that

$$G^* = \arg\max_{G \in G_n} g(G : O)$$

where $g(G : O)$ is the scoring function measuring the degree of fitness of any candidate G to the data set, and G_n is the family of all the DAGs defined on O. To find a Bayesian network to maximize the scoring function, we employ the Bayesian Dirichlet scoring function

proposed in [32].

$$g_{BD}(G:O) = log(p(G)) + \sum_{i=1}^{n}\left[\sum_{j=1}^{q_i}\left[log(\frac{\Gamma(\eta_{ij})}{\Gamma(N_{ij}+\eta_{ij})}) + \sum_{k=1}^{r_i} log(\frac{\Gamma(N_{ijk}+\eta_{ijk})}{\Gamma(\eta_{ijk})})\right]\right]$$

where $log(p(G))$ is the log-likelihood function, the values η_{ijk} are the hyperparameters for the Dirichlet prior distributions of the parameters given the network structure, q_i is the number of states of the Cartesian product of D_i's parents, r_i is the number of states of D_i, $\eta_{ij} = \sum_{k=1}^{r_i} \eta_{ijk} \cdot \Gamma(.)$ is the Gamma function, $\Gamma(c) = \int_0^\infty e^{-u}u^{c-1}du$.

The likelihood is a function of the parameters which is proportional to the probability of the observed data, and $log(p(O|G))$ is defined as follows while $D_i^{(m)}$ is the m^{th} instance of attribute D_i.

$$\begin{aligned} logp(O|G) &= \sum_{m=1}^{n} logp(D^{(m)}|\theta) \\ &= \sum_{m=1}^{n}\sum_{i=1}^{d} logp(D_i^{(m)}|D_{\pi_i}^{(m)}, \theta_i) \end{aligned}$$

Parameter learning. We estimate parameters for Bayesian networks using maximum likelihood estimation. Given a data set O and the structure of a Bayesian network G, the maximum likelihood estimation aims to choose parameters θ that satisfy

$$L(\theta^* : O|G) = Max_{\theta\in\Theta} L(\theta : O|G)$$

The parameter Θ is defined as a hypothesis space, a set of all parameters $\Theta \in [0, 1]$. With the Markov property of Bayesian networks, $L(\Theta : O)$ can be decomposed as follows.

$$L(\theta : O|G) = \prod_i L_i(\theta_{D_i|Pa_{D_i}} : O|G)$$

where the local likelihood function for D_i is:

$$L(\theta_{D_i|Pa_{D_i}} : O) = \prod_j P(D_i^j|Pa_{D_i}^j : \theta_{D_i|Pa_{D_i}})$$

With the structure of Bayesian network G and the data set O, $L(\theta : O|G)$ is reduced to estimating $\theta_{ijk} = P(D_i = j|Pa(D_i) = k)$, that is, the maximum likelihood estimates are simply the observed frequency estimates $\hat{\theta}_{ijk} = n_{ijk}/n_{ij}$,where n_{ijk} is the number of occurrences in the training set of the k^{th} state of D_i with the j^{th} state of its parents, and n_{ij} is the sum of n_{ijk} over all k.

To deal with the situation $n_{ijk} = 0$, we use the Dirichlet prior. Then, $\hat{\theta}_{ijk}$ can be written as follows.

$$\hat{\theta}_{ijk} = \frac{n_{ijk} + \alpha_{D_i,Pa_{D_i}}}{n_{ij} + \alpha_{Pa_{D_i}}}$$

where $\alpha_{D_i,Pa_{D_i}} = \alpha \cdot P(D_i, Pa_{D_i})$, $P(D_i, Pa_{D_i}) = \frac{1}{|D_i| \cdot |Pa_{D_i}|}$, and $\alpha = \sum_{D_i,Pa_{D_i}} \alpha_{D_i,Pa_{D_i}}$. $|D_i|$ is the number of values that D_i takes, and $|Pa_{D_i}|$ is the number of (joint) values of the parents of D_i.

6.3.2 Mining and Characterizing Outliers

Score function: Using Bayesian networks to model Markov blanket subspaces, it is natural to use joint probabilities of those Bayesian networks as outlyingness scores to measure the abnormality of data objects. The low scored instances are treated as potential outliers. With d Bayesian networks over Markov blanket subspaces, given a data object, we get d scores for this data object. How can we summarize a score for the data object from the d scores?

To tackle this problem, we first define a score measure for a data object on each Markov blanket subspace. With d Markov blanket subspaces, their corresponding Bayesian networks are defined as $BN = (BN_1, BN_2, ..., BN_d)$. We define the outlyingness score for a data object o_i on $BN_i (1 \leq i \leq d)$ as follows.

$$Score(o_i : BN_i) = \prod_{X_i \in BN_i} P(X_i | Pa(X_i : BN_i))$$

With d scores, a straightforward idea is to select the minimum score for a data object. However, it is not effective for identifying interesting outliers. There are two situations accounting for low scored instances.

1. Low prior probability of an attribute leads to low conditional probability. The outliers in this situation can be detected in one-dimensional subspace by the exisitng methods [98, 66, 20] without difficulty. Users may not be insterested in this type of oulites since they might already know the existence of those outliers.

2. An attribute has a high prior probability, but it has a low posterior probability. The outliers in the condition may hidden in subspaces with more than one dimensions. The detection of this type of outliers will provide much meaningful information to users.

We focus on discovering outliers in the second condition above. According to $Score(o_i : BN_i)$, if $P(D_j)$ has high prior probability and $P(D_j|Pa(D_j))$ may get a low conditional probability, and then $P(D_j)$ and $P(D_j|Pa(D_j))$ have a strong contrast. A large contrast

indicates that the corresponding data object may be a potential outlier. Based on this idea, we define the contrast between $P(D_j)$ and $P(D_j|Pa(D_j))$ as a score for a Markov blanket subspace as follows.

$$SR(BN_i) = \arg\max_{D_j \in BN_i} P(D_j)/P(D_j|Pa(D_j))$$

With $SR(BN_i)$, we define the outlyingness score for a data object o_j as follows.

$$Outlierscore(o_j) = \max_{BN_i \in BN} SR(BN_i)$$

Outlier Detecting and Characterizing: With $Outlierscore(o_j)$, we sort the top k data objects as potential outliers. Meanwhile, for a given outlier, we characterize this outlier using the Markov blanket subspaces that maximize $Outlierscore(o_j)$. Based on the discussion above, we give the *MiCOM* algorithm for mining and characterizing outliers in Markov blanket subspaces as follows.

Algorithm 6.1 *The* MiCOM *Algorithm*

Input $O = \{D_1, D_2, \cdots, D_d\}$
// Identify Markov blanket (MB) subspaces
for $i = 1$ to d **do**
    $MB(i) = MMMB(D_i)$
end for
// Learn a local Bayesian network on a MB subspace
for $i = 1$ to d **do**
    Learn the structure of BN_i^* on $MB(i)$ with the proposed three strategies such that

$$BN_i^* = \arg\max_{BN_i \in G_n} g_{BD}(BN_i : O)$$

    Learn parameters for BN_i
end for
// Mine outliers over multiple Bayesian networks
// Assume n test instances
for $i = 1$ to n **do**
    for $p = 1$ to d **do**

$$SR(BN_p) = \arg\max_{D_j \in BN_p} P(D_j)/P(D_j|Pa(D_j))$$

        characterizing_attribute=$\{D_j, Pa(D_j)\}$
    end for
    $Outlierscore(O_i) = \max_{BN_p \in BN} SR(BN_p)$
end for
Output top-k data objects with highest scores as potential outliers;
Characterize outliers with the sets of characterizing_attribute

6.3.3 Pruning Markov blanket subspaces

With d attributes, clearly, according to Property 6.2, we get d Markov blanket subspaces. Assuming ξ is the number of attributes without parents and β is the number of attributes that have no child but have only one parent, and $\delta = d - \xi - \beta$, the upper bound of the number of Markov blanket subspaces is given as follows.

Theorem 6.2. *With the dimensionality $D = \{D_1, D_2, \cdots, D_d\}$, the number of Markov blanket subspaces γ satisfies*

$$\gamma \leqslant \delta$$

Proof. 1. Assuming a Bayesian network BN_i is built from the Markov blanket of D_i, $D_j \in BN_i$, and F is a child of D_i, if D_i has no parent, the posterior probability of D_i is $P(D_i)$. Assuming BN_F is built from the Markov blanket of F, by the following score function, we can get $SR(BN_F) \geq SR(BN_i)$.

$$SR(BN_i) = \arg\max_{D_j \in BN_i} P(D_j)/P(D_j|Pa(D_j))$$

Thus, the Markov blanket subspace of D_i can be pruned.

2. Assuming D_i has no child and has only one parent F, then the posterior probability of $P(D_i)$ computed in BN_i is the same as this posterior probability computed in BN_F. Assuming F has parents, if $SR(BN_F) > SR(BN_i)$, clearly, the Markov blanket subspace of D_i can be pruned; if $SR(BN_F) \leq SR(BN_i)$, the Markov blanket subspace of D_i can also be pruned, since the Markov blanket subspace of D_i only includes D_i and F.

■

For example, in Fig. 6.1, The Markov blanket subspaces corresponding to "anxiety", "peer pressure", and "yellow fingers" can be pruned due to the existence of the Markov blanket subspace of "smoking".

6.4 Experiment results

To evaluate the quality of our approach, we compare *MiCOM* with the state-of-the-art subspace outlier mining methods: the *HiCS* [65] and $4S\_S$ [85] algorithms, and a full-space

Bayesian network outlier mining algorithm (*BNOM* for short) [10] using synthetic and real-world data. For the $4S\_S$ algorithm, k is set to $d \log n$ where d is the dimensionality and n is the number of data objects. For *HiCS*, the number of statistical tests m chooses the default value 50. The test statistic size α and the candidate cutoff parameter are set to 0.1 and 400, respectively. For the *BNOM* method, we use the *MMHC* algorithm for structure learning of Bayesian networks (due to its superperformance in learning Bayesian networks over thousands of attributes.) [110]. For our *MiCOM* algorithm, we use the Bayesian network toolbox, which can be downloaded at https://code.google.com/p/bnt/, to learn structures and parameters of Bayesian networks.

6.4.1 Experiments on synthetic data

In order to evaluate the proposed method, we design synthetic data sets of different sizes and dimensionality. Each data set contains subspace clusters with dimensionality varying from 2 to 6 and we generate 15-30 outliers deviating from these clusters.

To evaluate the detection quality of *MiCOM*, we compare it with *HiCS*, $4S\_S$, and *BNOM* using the precision metric and recall metric. If a synthetic data has k true outliers, then for each algorithm, we select top-k data objects with the highest outlyingness scores as potential outliers. The precision metric is defined as the ratio of the number of true outliers in those top-k data objects against the number of true outliers of this data set. For the recall metric, we select the top-5 and top-15 data objects with the highest outlyingness scores from a synthetic data set, respectively. Then, the recall metric is the ratio of the number of true outliers in those top-k ($k = 5$ or 15) data objects against the value k.

We design data sets of a fixed size of 1000 objects in dimensionality ranging from 20 to 200. For each dimensionality setting, we generate 10 data sets and calculate average precision and average recall for each algorithm.

Figure 6.4 shows the average precision of each algorithm, and Figure 6.5 and Figure 6.6 give the average recall. *MiCOM* outperforms *HiCS*, $4S\_S$ and *BNOM*. *MiCOM* is able to deal with high dimensional data and shows high precision and high recall. We can observe that *MiCOM* and the other three algorithms almost achieve the same precision on data sets of low dimensionality. As dimensionality increases, *MiCOM* achieves higher precision and higher recall than the other three methods. As for the recall metric on the top-5 objects based on outlyingness scores, *MiCOM* is able to assign higher outlyness scores to potential outliers than the other three algorithms. This can make *MiCOM* rank outliers in the top

list of data objects based on outlyingness scores.

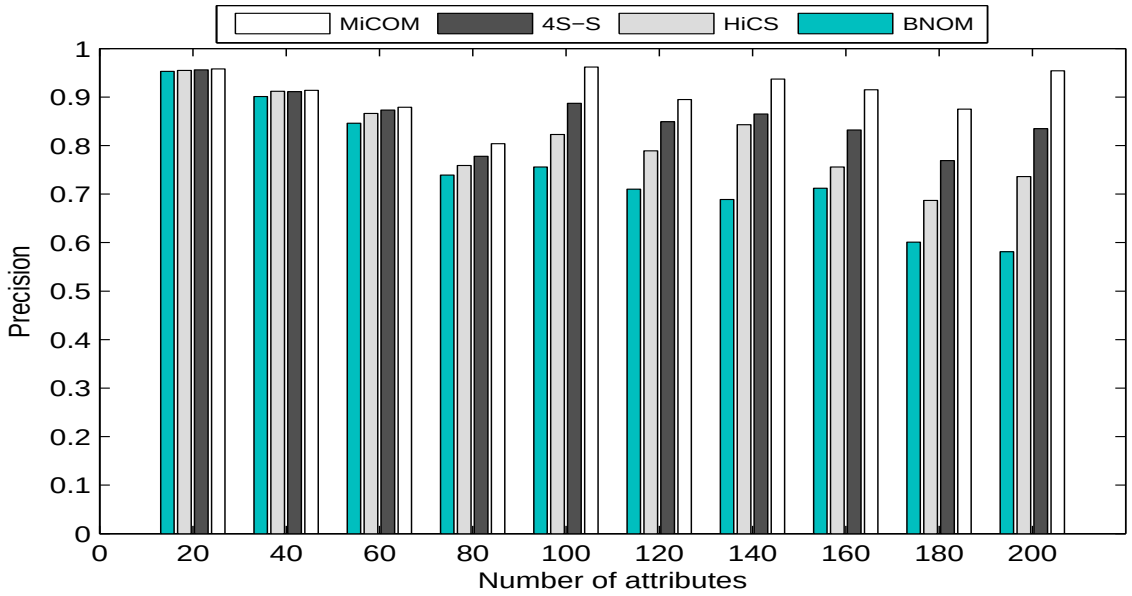


Figure 6.4: Precision of *MiCOM* against its rivals

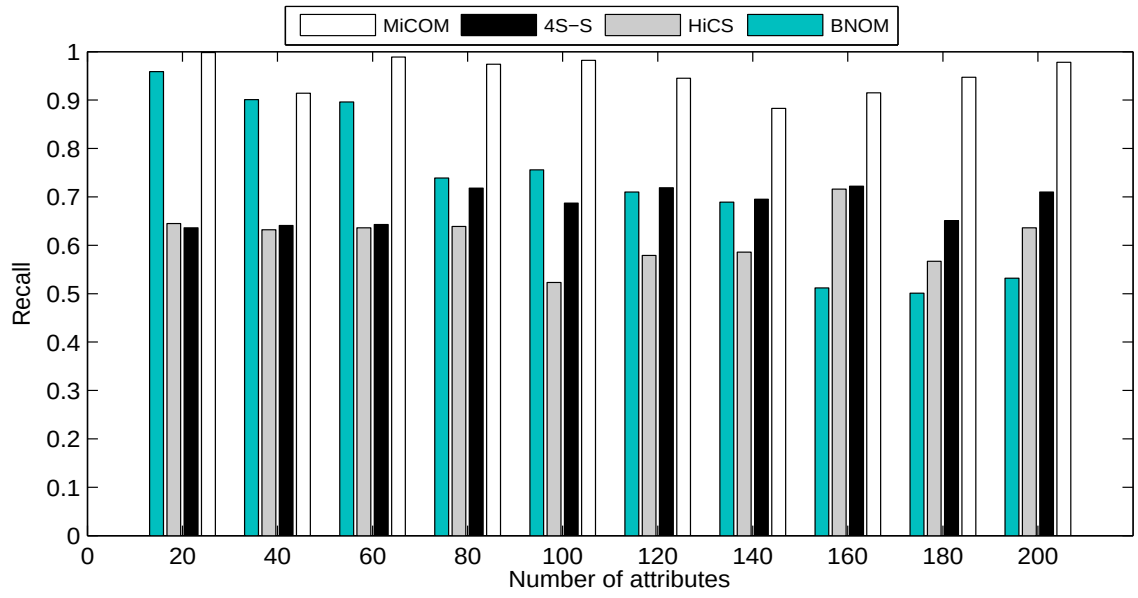


Figure 6.5: Recall of *MiCOM* against its rivals using Top 5 data objects based on outlierness scores

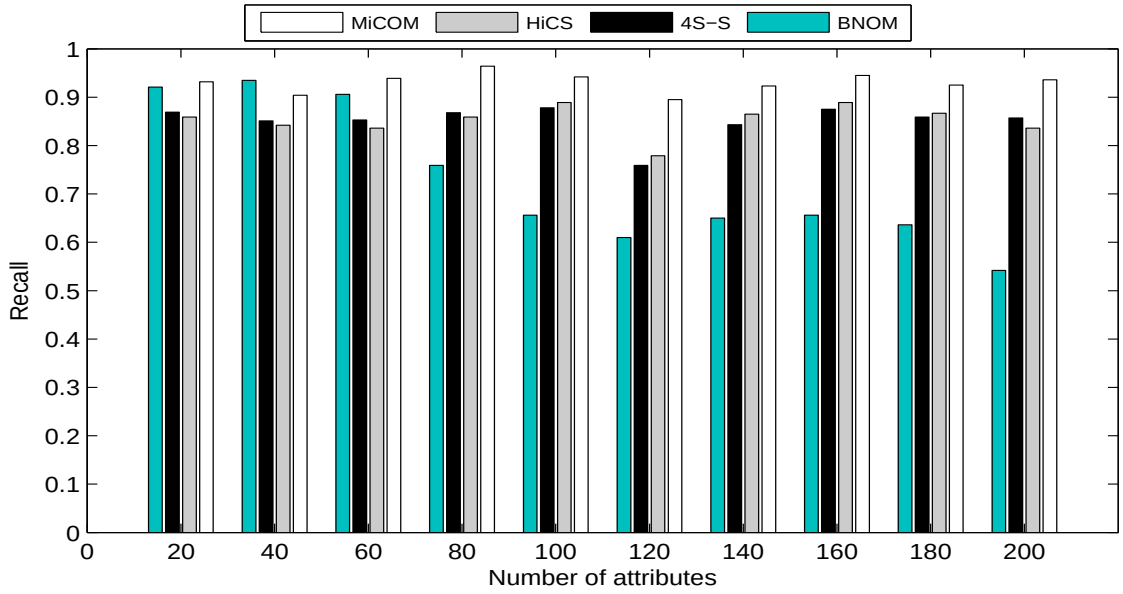


Figure 6.6: Recall of *MiCOM* against its rivals using Top 15 data objects based on outlierness scores

The explanation is that, for *HiCS*, its parameter of candidate cutoff limits the number of candidates in the bottom-up subspace processing, and then affects the quality of *HiCS*. We set the parameter to 400 in our experiments. In data sets of low dimensionality, by selecting the top 400 highest contrast subspaces, *HiCS* can achieve high precision. But with the increasing dimensionality, it is hard to select a descent candidate cutoff parameter to guarantee the quality of *HiCS*.

For $4S\_S$, it uses the parameter k (k pairs of attributes with the largest correlations) to balance its precision and running time. So with increasing dimensionality, it is also hard to set a proper value of k for $4S\_S$. With respect to *BNOM*, as the dimensionality increases, the joint probabilities between data objects will become more and more similar. Therefore, *BNOM* shows performance degradation under high dimensionality.

In addition to the precision measure, Figure 6.7 shows the running time of *MiCOM*, *HiCS*, $4S\_S$, and *BNOM* with increasing dimensionality (the running time for both subspace search and outlier detection). *MiCOM* is the fastest algorithm while *HiCS* is the slowest one among all the algorithm evaluated.

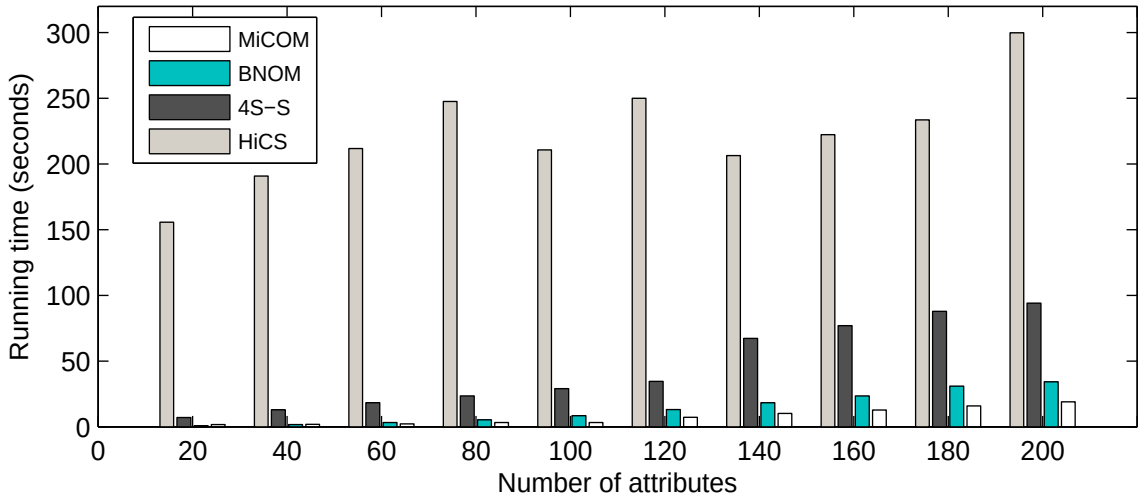


Figure 6.7: Running time of *MiCOM* against its rivals

In order to testing the scalability in high dimensionality, we compare the scalability of *MiCOM* against the other three algorithms using data sets in different sizes. In Figure 6.8, the dimensionality is fixed at 40 and the size of data sets varies from 1,000 to 10,000. We observe that the *MiCOM* algorithm is still the fastest algorithm. As the number of data objects increases, the running time of *HiCS* becomes very high. The BOMB algorithm is also very efficient since it uses *MMHC* [110], which is an efficient Bayesian network structure learning algorithm scalable to thousands of attributes.

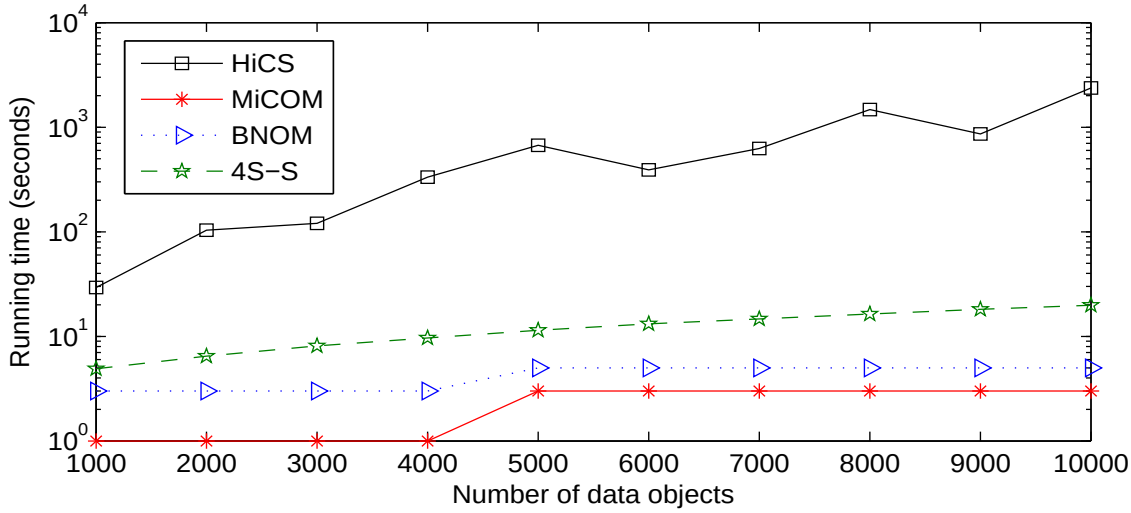
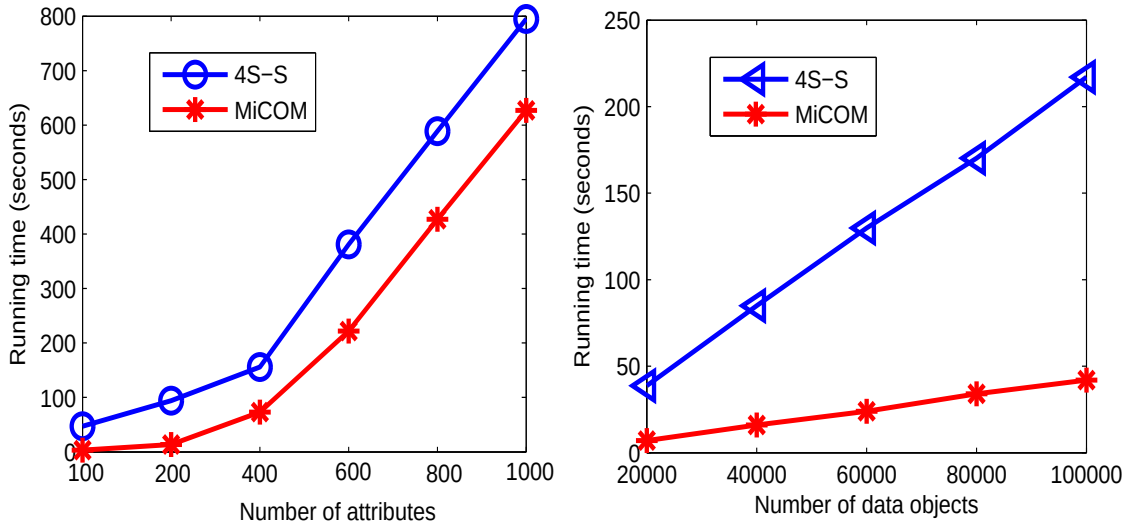


Figure 6.8: Running time of *MiCOM* against *HiCS* and *BNOM* using various data sizes

Figure 6.9 illustrates the running time of *MiCOM* and 4*S*_*S* on high-dimensional and large-scale data. In Figure 6.9 (a), the number of data object is fixed at 1,000; and in Figure 6.9 (b), the dimensionality is set to 40. We observe that both *MiCOM* and 4*S*_*S* are scalable to high dimensionality or large-scale data sizes, while *MiCOM* is more efficient than 4*S*_*S*.



(a) Runtime w.r.t. number of attributes.

(b) Runtime w.r.t. number of data objects.

Figure 6.9: The scalability of *MiCOM* against 4S-S

In summary, with Markov blanket subspaces, *MiCOM* achieves higher precision and lower running time than *HiCS*, 4*S*_*S*, and *BNOM*. Moreover, different from *HiCS* and 4*S*_*S*, *MiCOM* does not require additional parameters to balance its quality and running time.

6.4.2 Experiments on real-world data

We choose the real-world zoo data set from the UCI machine learning repository [11]. The zoo data set includes 16 predictive attributes. In Table 6.2, *MiCOM* finds 9 outliers from the zoo data set and charaterizes those outliers with the attributes in the corresponding Markov blanket subspaces. The results are ranked by their outlierness scores as shown in the third column. For example, for the mammal animal of platypus, in Markov blanket

subspace of attribute "eggs", "eggs=yes, milk=yes" is an outlying aspect. However, *HiCS*, $4S\_S$, and *BNOM* do not clearly address which subspaces characterize discovered outliers.

Outlier	**Outlier Characterizing**	**Outlier Score**
seal	hair = *yes*, feathers = *no*, milk = *yes*, legs = 0	402.6
platypus	egg = *yes*, milk = *yes*	93.96
crab	breathes = *no*, legs = 4	89.2
scorpion	egg = *no*, milk = *no*, toothed = *no*	64.74
clam	aquatic = *no*, breathes = *no*	51.6
sealion	fins = *yes*, legs = 2	50.2
frog	venomous = *yes*, legs = 4	31.2
octopus	backbone = *no*, catsize = *yes*	30.1
seasnake	egg = *no*, milk = *no*, toothed = *yes*	19.5

Table 6.2: Nine outliers discovered in the zoo data set

To further evaluate *MiCOM*, $4S\_S$, *HiCS*, and *BNOM*, Figure 6.10 gives the discovered true positive outliers using top-5, top-10, top-15, and top-20 data objects based on outlierness scores. We can see that *MiCOM* outperforms *HiCS* and *BNOM*.

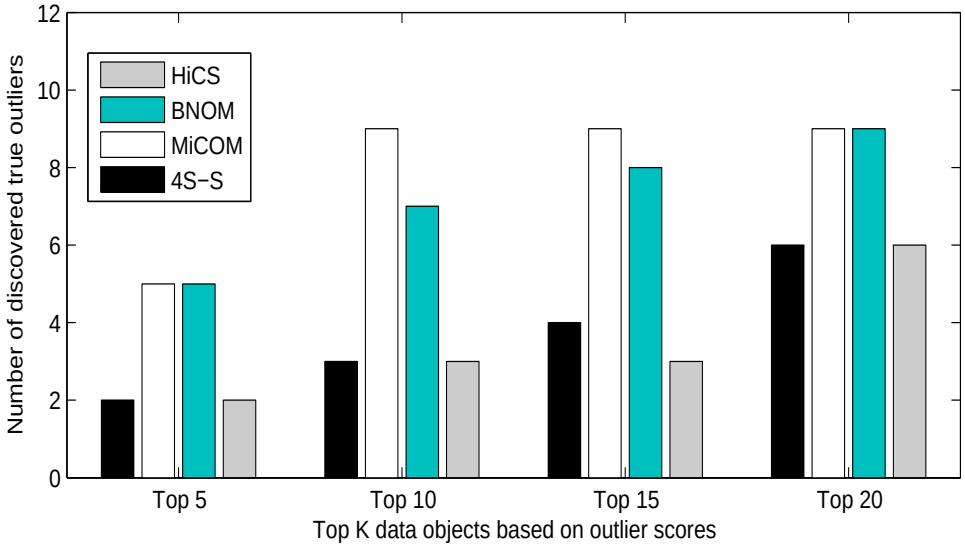


Figure 6.10: The number of the identified true positive outliers using top-k data objects

6.5 Conclusion

In this chapter, to tackle the challenge of subspace outlier detection, we propose a Markov blanket based method to dramatically reduce the number of subspaces from $2^d - 1$ possible subspaces to d Markov blanket subspaces (d is the dimensionality). To detect outliers in Markov blanket subspaces, we propose the *MiCOM* algorithm to mine subspaces and outliers simultaneously. Using synthetic and real-world data, our experimental results show that our *MiCOM* algorithm outperforms the state-of-the-art subspace outlier detection methods in terms of effectiveness, efficiency, and scalability.

Chapter 7

Conclusions

Outlier detection is an important data mining task with many critical applications, such as medical diagnosis, fraud detection, and intrusion detection. As the emergence of huge data sets in real-life applications nowadays, outlier detection faces a series of new challenges. In this thesis, we tackle the new challenges in dimensionality and understandability. In particular, we develop four new outlier detection methods from two perspectives, detecting the outlying aspects of a data object and detecting outlying data objects of a data set. We show that the proposed techniques are useful in different outlier detection application scenarios.

In this chapter, we first summarize the thesis, and then discuss a few interesting future research directions.

7.1 Summary of the Thesis

In this thesis, we study the problem of outlier detection to tackle the challenges in dimensionality and understandability. Specifically, we propose four novel outlier detection methods for different application purposes; and we make the following contributions.

- We identify two different levels of outlyingness for data objects in multidimensional data sets, namely, outlying aspects and outliers.
  - *Outlying aspects:* the most outlying subspaces of a given data object.
  - *Outliers:* the most outlying data objects of a given data set.

- For detecting outlying aspects, we formulate this problem in both unsupervised and supervised numeric data sets. And we develop two models to address two different kinds of applications, namely, outlying aspects and contrast subspaces.

  - *Outlying aspects:* given a data set and a query object, the problem of mining outlying aspects finds the subspaces best manifesting the unusualness of the specified query object, using the other objects as the background in comparing different subspaces. The query object itself may or may not be an outlier in the full space or in any specific subspaces.

    We use the rank of the probability density of an object in a subspace to measure the outlyingness of the object in the subspace. A minimal subspace where the query object is ranked the best is an outlying aspect. We systematically develop a heuristic method that is capable of searching data sets with tens of dimensions efficiently.

  - *Contrast subspaces:* in a multidimensional data set of two classes, given a query object and a target class, the problem of mining contrast subspaces finds the subspace where the query object is most likely to belong to the target class against the other class. This problem is a supervised version of the mining outlying aspects problem.

    We demonstrate theoretically that the problem is very challenging and is MAX-SNP hard. We present *CSMiner*, a mining method that uses kernel density estimation in conjunction with various pruning techniques, such as upper and lower bounds of likelihood and likelihood contrast.

- For detecting outliers, we develop two novel methods to detecting and characterizing two different kinds of outliers in multidimensional data sets, namely, contextual outliers and Markov blanket based outlier.

  - *Contextual outliers:* intuitively, a contextual outlier is a small group of objects that share strong similarity with a signicantly larger reference group of objects on some attributes, but deviate dramatically on some other attributes.

    We systematically develop a model, including a concise representation, for contextual outlier analysis, and devise a detection algorithm that leverages the state-of-the-art data cube computation techniques.

– *Markov blanket based outlier:* given a data set, we regard the Markov blanket of each dimension and the dimension itself as a Markov blanket subspace. Markov blanket based outliers are outlying data objects, which are detected from the Markov blanket subspaces.

We propose a Markov blanket based method. Built on the concept of Markov blankets in Bayesian networks, we can find outliers from only n Markov blanket subspaces (n is the dimensionality) instead of $2^n - 1$ non-empty subspaces. By learning a local Bayesian network for a Markov blanket subspace, we derive an outlier score measure based on joint probabilities of Bayesian networks, and then propose the *MiCOM* algorithm to mine subspaces and outliers simultaneously.

7.2 Future Research

It is interesting to extend the outlier detection models and techniques developed in this thesis to other related outlier detection problems. Some of them are listed below.

- *Outlying aspects:* in Chapter 3, we study the problem of mining outlying aspects on numeric data. There are several interesting issues that deserve research effort in the future. First, to further examine the quality of outlying aspects, we can compute a statistical confidence interval on the rank via bootstrap sampling, and select the subspace with tighter confidence interval on the rank. Second, since *OAMiner* ranks the query object among all the objects by their probability densities estimated by a Gaussian kernel , it is interesting to consider using other kernel functions or outlierness degree measures proposed by outlier detection methods, such as *SOD* [67] and *LOF* [20]. Third, *OAMiner* discovers outlying aspects for a given object. Obviously selecting appropriate query points requires domain knowledge. Selecting appropriate background data sets for contrast against the query point also requires background knowledge. It is interesting to explore strategies for incorporating domain knowledge into outlying aspect mining. In practice, it may be the case that a user may want to study a set of objects. In addition, the current version of our method is capable of searching data sets with tens of dimensions. We may explore parallel computation approaches to improve the capability of *OAMiner* with data sets having hundreds, thousands, or even more dimensions; and extend *OAMiner* for mixed data containing both numerical and non-numerical values. Finally, it is also interesting to further

characterize different types of outlying aspects, such as maximal outlying aspects, and investigate how to measure the interpretability and the interestingness of an outlying aspect.

- *Contrast subspaces:* in Chapter 4, we work on the problem of mining contrast subspaces on numeric data. There are several interesting problems for future work. First, how to use contrast subspaces to characterize a given data set? Some features derived from contrast subspaces, such as the distribution of likelihood contrast and the dimensionality distribution of inlying/outlying contrast subspaces, can be used as interesting features to describe a data set. Second, how to use the contrast subspaces to improve the accuracy of supervised learning methods? For example, the concept of contrast subspaces can be used in the process of feature selection. Third, it is interesting to extend *CSMiner* for complex data sets, such as data sets with both nominal and numerical values and data sets with history records. Finally, it is important to explore parallel computation approaches to improve the efficiency of *CSMiner*.

- *Contextual outliers:* in Chapter 5, we propose the frame work of detecting contextual outliers on categorical data. There are several important and interesting problems for future work. For example, we mentioned the concept of outlier groups (Definition 5.5). From the user's point of view, it would be interesting to develop efficient and pay-as-you-go methods to compute a ranked list of outlier groups. This would also help to eliminate the need for setting an outlier degree threshold. As another example, it would be interesting to explore the correlation among outlier groups, reference groups and contexts more generally. This may lead to valuable insights into the inherent characteristics of high dimensional data. Last but not least, developing more efficient and scalable algorithms for contextual outlier detection is another challenge for future study.

- *Markov blanket based outlier:* in Chapter 6, we study the subspace outlier detection problem on categorical data by using Markov blanket based techniques. As future study, it is interesting to consider the sampling problem of a data set by using the Markov blanket based techniques developed in this chapter. Particularly, given a large data set, how can we choose the smallest number of data objects, which can reserve

the characteristics of the original data set as much as possible? This problem can be considered as the opposite problem of outlier detection problem in Chapter 6.

Bibliography

[1] Charu C. Aggarwal. *An Introduction to Outlier Analysis*. Springer, 2013. 11, 14

[2] Charu C. Aggarwal and Philip S. Yu. Outlier detection for high dimensional data. In *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data*, SIGMOD '01, pages 37–46, New York, NY, USA, 2001. ACM. 19

[3] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 487–499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc. 19

[4] Malik Agyemang, Ken Barker, and Rada Alhajj. A comprehensive survey of numeric and symbolic outlier mining techniques. *Intelligent Data Analysis*, 10:521–538, December 2006. 11

[5] Constantin F Aliferis, Alexander Statnikov, Ioannis Tsamardinos, Subramani Mani, and Xenofon D Koutsoukos. Local causal and markov blanket induction for causal discovery and feature selection for classification part i: Algorithms and empirical evaluation. *The Journal of Machine Learning Research*, 11:171–234, 2010. 133, 136

[6] John R. Anderson and Paul I. Kline. A learning system and its psychological implications. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI '79, pages 16–21, Tokyo, Japan, 1979. 117

[7] Fabrizio Angiulli, Fabio Fassetti, and Luigi Palopoli. Detecting outlying properties of exceptional objects. *ACM Transactions on Database Systems*, 34(1):7:1–7:62, April 2009. 17, 18, 28

[8] Fabrizio Angiulli, Fabio Fassetti, Luigi Palopoli, and Giuseppe Manco. Outlying property detection with numerical attributes. *CoRR*, abs/1306.3558, 2013. 17, 18, 28

[9] Fabrizio Angiulli and Clara Pizzuti. Fast outlier detection in high dimensional spaces. In *Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery*, PKDD '02, pages 15–26, London, UK, UK, 2002. Springer-Verlag. 13, 14

[10] Sakshi Babbar and Sanjay Chawla. On bayesian network and outlier detection. In *Proceedings of the 16th International Conference on Management of Data, 2010, Nagpur, India*, page 125, 2010. 24, 143

[11] K. Bache and M. Lichman. UCI machine learning repository, 2013. 51, 84, 147

[12] Daniel Barbará, Yi Li, Julia Couto, Jia-Ling Lin, and Sushil Jajodia. Bootstrapping a data mining intrusion detection system. In *Proceedings of the 2003 ACM Symposium on Applied Computing*, SAC '03, pages 421–425, New York, NY, USA, 2003. ACM. 15

[13] Stephen D. Bay and Michael J. Pazzani. Detecting group differences: Mining contrast sets. *Data Mining and Knowledge Discovery*, 5(3):213–246, 2001. 20

[14] Stephen D. Bay and Mark Schwabacher. Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, pages 29–38, New York, NY, USA, 2003. ACM. 13, 14

[15] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is "nearest neighbor" meaningful? In *Proceedings of the 7th International Conference on Database Theory*, ICDT '99, pages 217–235, London, UK, 1999. Springer-Verlag. 5

[16] Kevin Beyer and Raghu Ramakrishnan. Bottom-up computation of sparse and iceberg cube. In *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, SIGMOD '99, pages 359–370, New York, NY, USA, 1999. ACM. 16, 23

[17] Klemens Böhm, Fabian Keller, Emmanuel Müller, Hoang Vu Nguyen, and Jilles Vreeken. CMI: An information-theoretic contrast measure for enhancing subspace cluster and outlier detection. In *Proceedings of the 13th SIAM Int'l Conf. on Data Mining*, SDM '13, pages 198–206, 2013. 16, 19, 20, 24, 130, 134

[18] Richard J. Bolton and David J. H. Unsupervised profiling methods for fraud detection. In *Proc. Credit Scoring and Credit Control VII*, pages 5–7, 2001. 1

[19] R. Brause, T. Langsdorf, and M. Hepp. Neural data mining for credit card fraud detection. In *Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence*, ICTAI '99, pages 103–, Washington, DC, USA, 1999. IEEE Computer Society. 1

[20] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. LOF: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, SIGMOD '00, pages 93–104, New York, NY, USA, 2000. ACM. 13, 19, 116, 126, 129, 130, 139, 152

[21] Yi Cai, Hong-Ke Zhao, Hao Han, Raymond Y. K. Lau, Ho-Fung Leung, and Huaqing Min. Answering typicality query based on automatically prototype construction. In *Proceedings of the 2012 IEEE/WIC/ACM Int'l Joint Conf. on Web Intelligence and Intelligent Agent Technology - Volume 01*, pages 362–366, 2012. 21

[22] Antonio Cansado and Alvaro Soto. Unsupervised anomaly detection in large databases using bayesian networks. *Applied Artificial Intelligence*, 22(4):309–330, April 2008. 24

[23] Philip K. Chan, Matthew V. Mahoney, and Muhammad H. Arshad. A machine learning approach to anomaly detection. Technical report, Florida Institute of Technology, Melbourne, Florida, USA, 2003. 22

[24] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):15:1–15:58, July 2009. 11

[25] Lijun Chen and Guozhu Dong. Masquerader detection using OCLEP: One class classification using length statistics of emerging patterns. In *Proceedings of WAIM Workshops: International Workshop on Information Processing over Evolving Networks*, WINPEN '06, pages 5–, Washington, DC, USA, 2006. IEEE Computer Society. 20, 23

[26] Xi C. Chen, Karsten Steinhaeuser, Shyam Boriah, Snigdhansu Chatterjee, and Vipin Kumar. Contextual time series change detection. In *Proceedings of the 13th SIAM International Conference on Data Mining, SDM'13, May 2-4, 2013. Austin, Texas, USA*, pages 503–511, 2013. 1

[27] Anny Lai-mei Chiu and Ada Wai-chee Fu. Enhancements on local outlier detection. In *Database Engineering and Applications Symposium, 2003. Proceedings. Seventh International*, pages 298–307, July 2003. 13

[28] D.A Clifton, L. Clifton, S. Hugueny, D. Wong, and L. Tarassenko. An extreme function theory for novelty detection. *Selected Topics in Signal Processing, IEEE Journal of*, 7(1):28–37, Feb 2013. 1

[29] Elizabeth Cohen. Five commonly misdiagnosed diseases. *CNN News*, October 2007. 3

[30] Xuan Hong Dang, Ira Assent, Raymond T. Ng, Arthur Zimek, and Erich Schubert. Discriminative features for identifying and interpreting outliers. In *IEEE 30th International Conference on Data Engineering, Chicago, ICDE'14, IL, USA, March 31 - April 4*, pages 88–99, 2014. 1

[31] Kaustav Das and Jeff Schneider. Detecting anomalous records in categorical datasets. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, pages 220–229, New York, NY, USA, 2007. ACM. 23

[32] Luis M De Campos. A scoring function for learning bayesian networks based on mutual information and conditional independence tests. *The Journal of Machine Learning Research*, 7:2149–2187, 2006. 138

[33] Arthur Pentland Dempster, Nan M. Laird, and Donald Bruce Rubin. Maximum likelihood from incomplete data via the em algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39(1):1–38, 1977. 12

[34] Guozhu Dong and James Bailey. Overview of contrast data mining as a field and preview of an upcoming book. In *Proceedings of ICDM Workshops: Workshop on Contrast Data Mining and Applications*, pages 1141–1146, Washington, DC, USA, 2011. IEEE Computer Society. 23

[35] Guozhu Dong and James Bailey, editors. *Contrast Data Mining: Concepts, Algorithms, and Applications*. CRC Press, 2013. 20

[36] Guozhu Dong and Jinyan Li. Efficient mining of emerging patterns: discovering trends and differences. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '99, pages 43–52, New York, NY, USA, 1999. ACM. 20, 23, 71

[37] Levent Ertöz, Michael Steinbach, and Vipin Kumar. Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In *Proceedings of the Third SIAM International Conference on Data Mining, San Francisco, CA, USA, May 1-3, 2003*, pages 47–58, 2003. 14

[38] Eleazar Eskin. Anomaly detection over noisy data using learned probability distributions. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 255–262, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. 11

[39] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA*, pages 226–231, 1996. 14, 15

[40] Ronald Fagin, Ravi Kumar, and D. Sivakumar. Comparing top k lists. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '03, pages 28–36, Philadelphia, PA, USA, 2003. Society for Industrial and Applied Mathematics. 94

[41] Tom Fawcett and Foster Provost. Adaptive fraud detection. *Data Mining and Knowledge Discovery*, 1(3):291–316, 1997. 11, 22

[42] Advisor for Business Insights from Standard Life. Reduce healthcare fraud - the key is stakeholder collaboration. *Advisor For Business*, Winter 2013. 4

[43] A. Frank and A. Asuncion. UCI machine learning repository, 2010. 116, 117, 118

[44] Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1st edition, 1997. 23

[45] Jing Gao and Pang-Ning Tan. Converting output scores from outlier detection algorithms into probability estimates. In *Proceedings of the 2006 IEEE 6th International Conference on Data Mining*, ICDM '06, pages 212–221, Hongkong, China, Dec 2006. IEEE Computer Society. 12

[46] Amol Ghoting, Srinivasan Parthasarathy, and Matthew Eric Otey. Fast mining of distance-based outliers in high dimensional datasets. In *Proceedings of the Sixth SIAM International Conference on Data Mining, April 20-22, 2006, Bethesda, MD, USA*, pages 609–613, 2006. 13, 14

[47] Jim Gray, Surajit Chaudhuri, Adam Bosworth, Andrew Layman, Don Reichart, Murali Venkatrao, Frank Pellow, and Hamid Pirahesh. Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals. *Data Mining and Knowledge Discovery*, 1:29–53, 1997. 23

[48] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. ROCK: A robust clustering algorithm for categorical attributes. In *Proceedings of the 15th International Conference on Data Engineering, Sydney, Austrialia, March 23-26, 1999*, pages 512–521, 1999. 14

[49] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009. 126

[50] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2011. 11

[51] Wolfgang Härdle. *Smoothing Techniques: With Implementations in S*. Springer-Verlang, New York, 1990. 32

[52] Wolfgang Härdle, Axel Werwatz, Marlene Müller, and Stefan Sperlich. *Nonparametric and Semiparametric Modelss*. Springer Series in Statistics. Springer, 2004. 32

[53] Douglas M. Hawkins. *Identification of outliers*. Chapman and Hall London ; New York, 1980. 1

[54] Zengyou He, Xiaofei Xu, and Shengchun Deng. Discovering cluster-based local outliers. *Pattern Recognition Letters*, 24(9):1641–1650, 2003. 15

[55] Zengyou He, Xiaofei Xu, Zhexue Joshua Huang, and Shengchun Deng. Fp-outlier: Frequent pattern based outlier detection. *Computer Science and Information Systems/ComSIS*, 2(1):103–118, 2005. 19

[56] Paul Helman and Jessie Bhangoo. A statistically based system for prioritizing information exploration under uncertainty. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 27(4):449–466, 1997. 12

[57] Victoria Hodge and Jim Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22:85–126, October 2004. 11

[58] Ming Hua, Jian Pei, Ada Wai-chee Fu, Xuemin Lin, and Ho-Fung Leung. Top-k typicality queries and efficient query answering methods on large databases. *The VLDB Journal*, 18(3):809–835, 2009. 21

[59] Alexander Ihler, Jon Hutchins, and Padhraic Smyth. Adaptive event detection with time-varying poisson processes. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 207–216, New York, NY, USA, 2006. ACM. 1

[60] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceeding of the 30th annual ACM symposium on Theory of computing*, STOC '98, pages 604–613, New York, NY, USA, 1998. ACM.

[61] Harold S. Javitz and Alfonso Valdes. The sri ides statistical anomaly detector. In *Research in Security and Privacy, 1991. Proceedings., 1991 IEEE Computer Society Symposium on*, pages 316–326, May 1991. 11, 12

[62] Harold Jeffreys. *The Theory of Probability*. Oxford, 3rd edition, 1961. 67, 68

[63] Mon-Fong Jiang, Shian-Shyong Tseng, and Chih-Ming Su. Two-phase clustering process for outliers detection. *Pattern recognition letters*, 22(6-7):691–700, 2001. 15

[64] Robert E Kass and Adrian E Raftery. Bayes factors. *Journal of the american statistical association*, 90(430):773–795, 1995. 6

[65] Fabian Keller, Emmanuel Müller, and Klemens Böhm. Hics: High contrast subspaces for density-based outlier ranking. In *Proceedings of the 2012 IEEE 28th International Conference on Data Engineering*, ICDE '12, pages 1037–1048, Washington, DC, USA, 2012. IEEE Computer Society. 16, 19, 20, 24, 56, 60, 61, 129, 130, 134, 142

[66] Edwin M. Knorr and Raymond T. Ng. Algorithms for mining distance-based outliers in large datasets. In *Proceedings of the 24rd International Conference on Very Large Data Bases*, VLDB '98, pages 392–403, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. 13, 14, 139

[67] Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek. Outlier detection in axis-parallel subspaces of high dimensional data. In *Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, PAKDD '09, pages 831–838, Berlin, Heidelberg, 2009. Springer-Verlag. 16, 19, 60, 152

[68] Hans-Peter Kriegel, Peer Kroger, Erich Schubert, and Arthur Zimek. Outlier detection in arbitrarily oriented subspaces. In *Proceedings of the 2012 IEEE 12th International Conference on Data Mining*, ICDM '12, pages 379–388, Washington, DC, USA, 2012. IEEE Computer Society. 16

[69] Hans-Peter Kriegel, Matthias S hubert, and Arthur Zimek. Angle-based outlier detection in high-dimensional data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 444–452, New York, NY, USA, 2008. ACM. 19

[70] Solomon Kullback and Richard A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:49–86, 1951. 133

[71] Khaled Labib and Rao Vemuri. Nsom: A real-time network-based intrusion detection system using self-organizing maps. *Networks and Security*, pages 1–6, 2002. 15

[72] Laks V. S. Lakshmanan, Jian Pei, and Jiawei Han. Quotient cube: how to summarize the semantics of a data cube. In *Proceedings of the 28th International Conference on Very Large Data Bases*, VLDB '02, pages 778–789, Hong Kong, China, 2002. VLDB Endowment. 113

[73] Aleksandar Lazarevic, Levent Ertöz, Vipin Kumar, Aysel Ozgur, and Jaideep Srivastava. A comparative study of anomaly detection schemes in network intrusion detection. In *SDM*, pages 25–36, 2003. 1

[74] Song Lin and Donald E. Brown. An outlier-based data association method for linking criminal incidents. *Decision Support Systems*, 41:604–615, March 2006. 23

[75] Huan Liu and Hiroshi Motoda. *Computational Methods of Feature Selection (Chapman & Hall/Crc Data Mining and Knowledge Discovery Series)*. Chapman & Hall/CRC, 2007. xv, 132

[76] James MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. California, USA, 1967. 15

[77] Matthew V. Mahoney and Philip K. Chan. Learning rules for anomaly detection of hostile network traffic. In *Proceedings of the 3rd IEEE International Conference on Data Mining*, ICDM '03, pages 601–604, Melbourne, Florida, USA, 2003. 22

[78] Markos Markou and Sameer Singh. Novelty detection: a review – part 1: statistical approaches. *Signal Processing*, 83:2481–2497, December 2003. 11

[79] Markos Markou and Sameer Singh. Novelty detection: a review – part 2: neural network based approaches. *Signal Processing*, 83:2499–2521, December 2003. 11

[80] Barbora Micenková, Xuan-Hong Dang, Ira Assent, and Raymond T. Ng. Explaining outliers by subspace separability. In *Proceedings of the 2013 IEEE 13th International Conference on Data Mining*, ICDM '13, pages 518–527, Dallas, Texas, USA, Dec 2013. IEEE Computer Society. 17, 18

[81] Emmanuel Müller, Ira Assent, Uwe Steinhausen, and Thomas Seidl. Outrank: ranking outliers in high dimensional data. In *ICDE Workshops*, pages 600–603, 2008. 16

[82] Emmanuel Müller, Matthias Schiffer, and Thomas Seidl. Adaptive outlierness for subspace outlier ranking. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 1629–1632, New York, NY, USA, 2010. ACM. 129, 130, 132, 134

[83] Emmanuel Müller, Matthias Schiffer, and Thomas Seidl. Statistical selection of relevant subspace projections for outlier ranking. In *Proceedings of the 2011 IEEE 27th International Conference on Data Engineering*, ICDE '11, pages 434–445, Washington, DC, USA, 2011. IEEE Computer Society. 16, 24

[84] Kazuyo Narita and Hiroyuki Kitagawa. Detecting outliers in categorical record databases based on attribute associations. In *Proceedings of the Tenth Asia-Pacific Web Conference on Progress in WWW Research and Development*, APWeb '08, pages 111–123, Berlin, Heidelberg, 2008. Springer-Verlag. 22

[85] Hoang Vu Nguyen, Emmanuel Müller, and Klemens Böhm. 4s: Scalable subspace search scheme overcoming traditional apriori processing. In *BigData Conference*, pages 359–367, 2013. 16, 24, 142

[86] Petra Kralj Novak, Nada Lavrac, and Geoffrey I. Webb. Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining. *Journal of Machine Learning Research*, 10:377–403, 2009. 20

[87] M. Otey, S. Parthasarathy, A. Ghoting, G. Li, S. Narravula, and D. Panda. Towards nic-based intrusion detection. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, pages 723–728, New York, NY, USA, 2003. ACM. 15

[88] Themistoklis Palpanas, Dimitris Papadopoulos, Vana Kalogeraki, and Dimitrios Gunopulos. Distributed deviation detection in sensor networks. *SIGMOD Reccord*, 32(4):77–82, December 2003. 12

[89] Christos H. Papadimitriou and Mihalis Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43(3):425 – 440, 1991. 72

[90] Rohit Paravastu, Hanuma Kumar, and Vikram Pudi. Uniqueness mining. In *Proceedings of the 13th Int'l Conf. on Database Systems for Advanced Applications*, DASFAA '08, pages 84–94, 2008. 19

[91] Nicolas Pasquier, Yves Bastide, Rafik Taouil, and Lotfi Lakhal. Discovering frequent closed itemsets for association rules. In *Proceedings of the Seventh International Conference on Database Theory*, ICDT '99, pages 398–416, London, UK, 1999. Springer-Verlag. 113

[92] Animesh Patcha and Jung-Min Park. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks*, 51:3448–3470, 2007. 11

[93] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988. 24, 130, 131, 132, 136

[94] D. Pokrajac, A Lazarevic, and L.J. Latecki. Incremental local outlier detection for data streams. In *Computational Intelligence and Data Mining, 2007. CIDM 2007. IEEE Symposium on*, pages 504–515, March 2007. 1

[95] Manikantan Ramadas, Shawn Ostermann, and Brett Tjaden. Detecting anomalous network traffic with self-organizing maps. In *Recent Advances in Intrusion Detection*, pages 36–54. Springer, 2003. 15

[96] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD Int'l Conf. on Management of Data*, SIGMOD '00, pages 427–438, New York, NY, USA, 2000. ACM. 13, 14

[97] Stenphen J. Roberts. Extreme value statistics for novelty detection in biomedical signal processing. In *Advances in Medical Signal and Information Processing, 2000. First International Conference on (IEE Conf. Publ. No. 476)*, pages 166–172, 2000. 1

[98] Peter J. Rousseeuw and Annick M. Leroy. *Robust Regression and Outlier Detection*. John Wiley & Sons, Inc., New York, NY, USA, 1987. 139

[99] Ron Rymon. Search through systematic set enumeration. In *Proceedings 1992 Int. Conf. Principle of Knowledge Representation and Reasoning (KR'92)*, pages 539–550, Cambridge, MA, 1992. 35, 75

[100] Ali S Saber Tehrani, HeeWon Lee, Simon C Mathews, Andrew Shore, Martin A Makary, Peter J Pronovost, and David E Newman-Toker. 25-year summary of us malpractice claims for diagnostic errors 1986–2010: An analysis from the national practitioner data bank. *BMJ Quality & Safety*, 2013. 3

[101] Erich Schubert, Arthur Zimek, and Hans-Peter Kriegel. Generalized outlier detection with flexible kernel density estimates. In *Proceedings of the 14th SIAM International Conference on Data Mining (SDM), Philadelphia, PA*. SIAM, 2014.

[102] David W Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley Series in Probability and Statistics. Wiley, New York, 1992. 12, 32, 70

[103] Karlton Sequeira and Mohammed Zaki. Admit: Anomaly-based data mining for intrusions. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 386–395, New York, NY, USA, 2002. ACM. 1

[104] Walter A. Shewhart. Economic quality control of manufactured product1. *Bell System Technical Journal*, 9(2):364–389, 1930. 11, 12

[105] Bernard W Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall/CRC, London, 1986. 32, 69, 70, 98

[106] Helge Erik Solberg and Ari Lahti. Detection of outliers in reference distributions: performance of horns algorithm. *Clinical chemistry*, 51(12):2326–2332, 2005. 12

[107] Xiuyao Song, Mingxi Wu, Christopher Jermaine, and Sanjay Ranka. Conditional anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 19(5):631–645, May 2007. 21, 22

[108] Jian Tang, Zhixiang Chen, Ada Wai-Chee Fu, and David Wai-Lok Cheung. Enhancing effectiveness of outlier detections for low density patterns. In *Proceedings of the 6th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, PAKDD '02, pages 535–548, London, UK, UK, 2002. Springer-Verlag. 13

[109] Dallas Thornton, Roland M. Mueller, Paulus Schoutsen, and Jos van Hillegersberg. Predicting healthcare fraud in medicaid: A multidimensional data model and analysis techniques for fraud detection. *Procedia Technology*, 9(0):1252 – 1264, 2013. 1

[110] Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, 65(1):31–78, 2006. 143, 146

[111] Michal Valko, Branislav Kveton, Hamed Valizadegan, Gregory F. Cooper, and Milos Hauskrecht. Conditional anomaly detection with soft harmonic functions. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining*, ICDM '11, pages 735–743, Washington, DC, USA, 2011. IEEE Computer Society. 22

[112] Jianyong Wang, Jiawei Han, and Jian Pei. Closet+: searching for the best strategies for mining frequent closed itemsets. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, pages 236–245, New York, NY, USA, 2003. ACM. 113

[113] Lusheng Wang, Hao Zhao, Guozhu Dong, and Jianping Li. On the complexity of finding emerging patterns. *Theoretical Computer Science*, 335(1):15–27, 2005. 71

[114] Xiang Wang and Ian Davidson. Discovering contexts and contextual outliers using random walks in graphs. In *Proceedings of the 2009 Ninth IEEE International Conference on Data Mining*, ICDM '09, pages 1034–1039, Miami, Florida, USA, 2009. 22

[115] William Webber, Alistair Moffat, and Justin Zobel. A similarity measure for indefinite rankings. *ACM Transactions on Information Systems*, 28(4):20:1–20:38, November 2010. 94

[116] Li Wei, Weining Qian, Aoying Zhou, Wen Jin, and Jeffrey X. Yu. Hot: hypergraph-based outlier test for categorical data. In *Proceedings of the 7th Pacific-Asia conference on Advances in knowledge discovery and data mining*, PAKDD'03, pages 399–410, Berlin, Heidelberg, 2003. Springer-Verlag. 22

[117] Weng-Keen Wong, Andrew Moore, Gregory Cooper, and Michael Wagner. Rule-based anomaly pattern detection for detecting disease outbreaks. In *Proceedings of the 18th National Conference on Artificial Intelligence*, ENAI '02, pages 217–223, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence. 22, 23

[118] Weng-Keen Wong, Andrew Moore, Gregory Cooper, and Michael Wagner. Bayesian network anomaly pattern detection for disease outbreaks. In *In Proceedings of the Twentieth International Conference on Machine Learning*, pages 808–815. AAAI Press, 2003. 24

[119] Stefan Wrobel. An algorithm for multi-relational discovery of subgroups. In *Proceedings of the 1st European Symposium on Principles of Data Mining and Knowledge Discovery*, pages 78–87, 1997. 20

[120] Mingxi Wu and Christopher Jermaine. Outlier detection by sampling with accuracy guarantees. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 767–772, New York, NY, USA, 2006. ACM. 13, 14

[121] Shengli Wu and Fabio Crestani. Methods for ranking information retrieval systems without relevance judgments. In *Proceedings of the 2003 ACM Symposium on Applied Computing*, pages 811–816, New York, NY, USA, 2003. ACM. 94

[122] Guizhen Yang. The complexity of mining maximal frequent itemsets and maximal frequent patterns. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 344–353, New York, NY, USA, 2004. ACM. 116

[123] Xingwei Yang, Longin Jan Latecki, and Dragoljub Pokrajac. Outlier detection with globally optimal exemplar-based gmm. In *Proceedings of the 9th SIAM Int'l Conf. on Data Mining*, SDM '09, pages 145–154. SIAM, 2009. 11, 12

[124] He Zhang, Sanjay Mehotra, David Liebovitz, Carl A. Gunter, and Bradley Malin. Mining deviations from patient care pathways via electronic medical record system audits. *ACM Trans. Manage. Inf. Syst.*, 4(4):17:1–17:20, December 2013. 1

[125] Ji Zhang. Advancements of outlier detection: A survey. *EAI Endorsed Trans. Scalable Information Systems*, 1:e2, 2013. 11

[126] Arthur Zimek, Erich Schubert, and Hans-Peter Kriegel. A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining*, 5(5):363–387, October 2012. 11

Appendix A

List of Publications

A.1 Publications Related to This Thesis

- L. Duan, G. Tang, J. Pei, J. Bailey, A. Campbell, and C. Tang. "Mining Outlying Aspects on Numeric Data". To appear in Data Mining and Knowledge Discovery, Springer-Verlag.
- G. Tang, J. Pei, J. Bailey, and G. Dong. "Mining multidimensional contextual outliers from categorical relational data". To appear in Intelligent Data AnalysisAn International Journal, IOS Press.
- L. Duan, G. Tang, J. Pei, J. Bailey, G. Dong, A. Campbell, and C. Tang. "Efficient Discovery of Contrast Subspaces for Object Explanation and Characterization". To appear in Knowledge and Information Systems: An International Journal, Springer-Verlag.
- L. Duan, G. Tang, J. Pei, J. Bailey, G. Dong, A. Campbell, and C. Tang. "Mining Contrast Subspaces" (**Best paper award**). In Proceedings of the 18th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'14), Tainan, Taiwan, May 13-16, 2014.
- G. Tang, J. Pei, J. Bailey and G. Dong. "Mining Multidimensional Contextual Outliers from Categorical Relational Data". In Proceedings of the 25th International Conference on Scientific and Statistical Database Management (SSDBM'13), Baltimore, MA, USA, July 29-31, 2013.

A.2 Other Publications

- Y. Yang, Q. Lu, G. Tang, and J. Pei. "The Impact of Market Competition on Search Advertising". To appear in Journal of Interactive Marketing, Elsevier.
- G. Tang, J. Pei, and W-S. Luk. "Email Mining: Tasks, Common Techniques, and Tools". Knowledge and Information Systems: An International Journal, Volume 41, Issue 1, pages 1-31, October 2014, Springer-Verlag.
- G. Tang, Y. Yang, and J. Pei. "Price Information Patterns in Web Search Advertising: An Empirical Case Study on Accommodation Industry". In Proceedings of the 13th IEEE International Conference on Data Mining (ICDM'13), Dallas, TX, USA, December 7-10, 2013.
- Y. Cui, J. Pei, G. Tang, W-S. Luk, D. Jiang, and M. Hua. "Finding Email Correspondents in Online Social Networks". World Wide Web Journal, Volume 6, Issue 2, pages 195-218, March 2013, Springer-Verlag.
- L. Li, S. Petschulat, G. Tang, J. Pei, W-S. Luk. "Efficient and Effective Aggregate Keyword Search on Relational Databases". International Journal of Data Warehousing and Mining (IJDWM), Volume 7, Issue 4, pages 41-81, October 2012, Idea Group, Inc.
- Q. Jiang, A. Campbell, G. Tang, and J. Pei. "Multi-level Relationship Outlier Detection". The International Journal of Business Intelligence and Data Mining (IJBIDM), Volume 7, Number 4, pages 253-273, October 2012, InderScience Publishers.