Non-IID Outlier Detection with Coupled Outlier Factors

Guansong Pang

Supervisors:
Longbing Cao, Principal
Ling Chen
Huan Liu, Arizona State University

This thesis is presented as part of the requirements for the conferral of the degree:

Doctor of Philosophy

University of Technology Sydney
Faculty of Engineering and Information Technology

April 2019

To my beloved parents, Yuxiang Pang and Shangxing Li.
To my wife Lisha and my son Louis.

Certificate of Original Authorship

I, Guansong Pang, declare that this thesis, submitted in fulfillment of the requirements for the award of the degree: Doctor of Philosophy, in the Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise reference or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution. This research is supported by the Australian Government Research Training Program.

Guansong Pang
April 1, 2019

Acknowledgments

I am greatly indebted to my principal supervisor, Professor Longbing Cao, one of the best mentors I have ever met. Over the past three years, Longbing spent tremendous time on teaching me how to think critically, do solid research and write 'beautiful' technical papers. He has been very nice, thoughtful and supportive to me in that he gave me large freedom to exclusively explore interesting research problems and publish high-quality papers at the early stage of my candidature while prepared me for an independent academic career in my senior candidature by involving me in different professional activities (e.g., conference/workshop organization, program committee of leading conferences), conference tutorials, student mentoring, research proposals, and industry project meetings. Longbing has been very fruitful, critical and constructive in making comments on research. I will not forget his massive advice on my designs and manuscripts, which are extremely important and helpful for sharpening my research designs and paper writings. In addition to research, Longbing has been a great life mentor of me for sharing many wonderful and undesirable experiences to encourage me when I was down and to remind me pitfalls when I was in triumph. I cannot thank him enough for all his supports.

I would like to thank my co-supervisor Dr. Ling Chen and external supervisor Professor Huan Liu for their helpful and insightful comments on both of my research designs and papers. I would like to particularly thank Ling for her nice and warm greetings on many festivals and public holidays. Ling has been always very supportive and understanding for many aspects in my research and daily life I seek suggestions and help from her. I am very grateful to Huan for not only having nice and thorough discussions over my research whenever I met him, but also suggesting interesting directions for my work, offering me great advice on job applications and career planning.

Besides my supervisors, I would also like to thank the panel members of my candidature assessment, Dr. Wei Liu, Professor Jinyan Li, Dr. Haiyan Lu, and Professor Bogdan Gabrys, for their constructive comments and great encouragement.

I thank all my friends and colleagues in the Data Science Lab at AAi for being critical and insightful to my research in our weekly research meetings and being very nice to me whenever I met them. I particularly thank: Chengzhang Zhu for many wonderful discussions about our research and careers, Songlei Jian for the nice research discussions and collaborations, Shoujin Wang and Thac Do for their companion and story sharing in our quiet lab. I am very grateful to visiting professors Defu Lian, Wenpeng Lu, and Lizhen Wang for their constructive comments to my research and great help in my career planning. I really appreciate the collaboration with Hongzuo Xu. It was a pleasure to work with him and I learned many helpful mentoring skills from this experience. I would also like to thank the other friends in the lab who enrich my research life and/or give me consistent encouragement: Wei Wang, Liang Hu, Qi Zhang, Ke Liu, Longxiang Shi, Wenfeng Hou, Qing Liu, Usman Naseem, Yan Xing, Jingyu Shao, Frank Xu, Dr. Allen Lin, Professor Quangui Zhang, and Dr. Lei Gu. It was always a great time to celebrate every of our success in the lab and to gather together to have fun and enjoy the holidays in festivals.

I would like to express my special thanks to Professor Shengyi Jiang who introduced me to the academic community and was always there to give me advice whenever I seek help from him. I am also very thankful to Dr. Huidong Jin, Professor Kai Ming Ting, and Dr. David Albrecht for their enormous time and effort on my previous research which lays an important foundation to my PhD research. I also thank everyone who has inspired me and helped me in my research.

I would also like to thank the UTS Advanced Analytics Institute, School of Software, and Graduate Research School for their quality administration service and financial support for my domestic and international conference travels. Additionally, I thank Michele Mooney for her excellent proofreading of this thesis.

Last but not least, I wholeheartedly thank my beloved mother and father for their everlasting love, amazing encouragement and strong support throughout my life. I am also specially grateful to my brothers and sisters who are very supportive in taking care of all the family matters during my graduate study. Special thanks to my dear wife Lisha Zhang for being patient and extremely supportive during this tough period and for planning my career and making decisions on our future with me. I am so fortunate to have her in my life. I would also like to thank my son Louis (Zhizhong) Pang who colors my life with his cheerfulness, lovely and gorgeous smiles, and chaos of trouble. This thesis is dedicated to them.

Publications During Candidature

Papers that have been published or accepted for publication:

1. **Guansong Pang**, Longbing Cao, Ling Chen, and Huan Liu. "Learning Representations of Ultrahigh-dimensional Data for Random Distance-based Outlier Detection", In: *24th SIGKDD Conference on Knowledge Discovery and Data Mining (KDD-18).* London, UK. Accepted (long presentation).

2. Songlei Jian, **Guansong Pang**, Longbing Cao, Kai Lu, and Hang Gao. "CURE: Flexible Categorical Data Representation by Hierarchical Coupling Learning". *IEEE Transactions on Knowledge and Data Engineering (TKDE).* Accepted.

3. **Guansong Pang**, Longbing Cao, Ling Chen, Defu Lian and Huan Liu. "Sparse Modeling-based Sequential Ensemble Learning for Effective Outlier Detection in High-dimensional Numeric Data", In: *32nd AAAI Conference on Artificial Intelligence (AAAI-18).* AAAI Press, pp. 3892-3899. New Orleans, US.

4. **Guansong Pang**, Longbing Cao, Ling Chen and Huan Liu. "Learning Homophily Couplings from Non-IID Data for Joint Feature Selection and Noise-Resilient Outlier Detection". In: *26th International Joint Conference on Artificial Intelligence (IJCAI-17).* AAAI Press, pp. 2585-2591.

5. Songlei Jian, Longbing Cao, **Guansong Pang**, Kai Lu and Hang Gao. "Embedding-based Representation of Categorical Data by Hierarchical Value Coupling Learning". In: *26th International Joint Conference on Artificial Intelligence (IJCAI-17).* AAAI Press, pp. 1937-1943.

6. **Guansong Pang**, Hongzuo Xu, Longbing Cao and Wentao Zhao. "Selective Value Coupling Learning for Detecting Outliers in High-Dimensional Categorical Data". In: *26th ACM International Conference on Information and Knowledge Management (CIKM-17)*, Long paper track. ACM, pp. 807-816. Singapore.

7. **Guansong Pang**, Kai Ming Ting, David Albrecht and Huidong Jin. "ZERO++: Harnessing the Power of Zero Appearances to Detect Anomalies in Large-Scale Data Sets". *Journal of Artificial Intelligence Research (JAIR)* 57, pp. 593–620, 2016.

8. **Guansong Pang**, Longbing Cao, and Ling Chen. "Outlier Detection in Complex Categorical Data by Modeling Feature Value Couplings". In: *25th International*

Joint Conference on Artificial Intelligence (IJCAI-16). AAAI Press, pp. 1902–1908, 2016. New York City, US.

9. **Guansong Pang**, Longbing Cao, Ling Chen and Huan Liu. "Unsupervised Feature Selection for Outlier Detection by Modeling Hierarchical Value-Feature Couplings." In: *2016 IEEE International Conference on Data Mining (ICDM-16)*, Long paper track. IEEE, pp. 410-419. Barcelona, Spain.

10. **Guansong Pang**, Kai Ming Ting, and David Albrecht. "LeSiNN: Detecting anomalies by identifying Least Similar Nearest Neighbours". In: *2015 IEEE 15th International Conference on Data Mining Workshops (ICDMW-15)*. IEEE, pp. 623–630, 2015.

Papers that are under review:

1. **Guansong Pang**, Longbing Cao, and Ling Chen. "Outlier Detection with Non-IID Outlier Factors". *Data Mining and Knowledge Discovery (DMKD)*. Under review.

2. **Guansong Pang** and Longbing Cao. "Building Optimal Heterogeneous Univariate Outlier Ensembles". *IEEE Transactions on Neural Networks and Learning System (TNNLS)*. Under review.

Abstract

Outliers are data objects which are rare or inconsistent compared to the majority of objects. Outlier detection is one of the most important tasks in data mining due to its wide applications in various domains, such as finance, information security, healthcare and earth science. Most existing outlier detection methods assume that the outlier factor (i.e., outlierness scoring measure) of the entities (e.g., feature values, features, data objects) in a data set is Independent and Identically Distributed (IID), but this assumption is violated by many real-world applications where the outlierness of an entity is coupled with that of some other entities, leading to the failure of detecting sophisticated outliers. This issue is intensified in more challenging environments, e.g., noisy and/or high-dimensional data sets. To address this challenge, this thesis considers three key questions: what are the coupling relations between different outlier factors? how can we effectively and efficiently model these couplings? and how can we leverage these couplings to address challenging outlier detection problems?

Our explorations result in the following four key contributions. (i) This thesis introduces a *new outlier detection task*, non-IID outlier detection in multidimensional data, which opens a new research direction for tackling real-world complex outlier detection problems. (ii) We introduce the *first architecture* for the non-IID outlier detection task, which provides principled approaches to learn the outlierness interdependence at different levels from feature values, features, to data objects. The architecture breaks down the general coupling learning into a series of important finer-grained components: basic coupling relation, coupling capacity, coupling utility, and coupling passage manners, providing feasible ways to learn sophisticated couplings between outlier factors with efficient models. (iii) We propose principled frameworks and their instantiations under the non-IID outlier detection architecture to learn different types of couplings. Supported by extensive theoretical analysis and empirical experiments on diverse real-world data sets, these designs are shown to be scalable and effective in *addressing some notoriously challenging problems*, including outlier detection in non-IID data, data with many noisy features, or high-dimensional data. (iv) This thesis also introduces a set of *seminal work* on unsupervised feature selection for outlier detection in both categorical data and numeric data, including innovative feature selection methods that capture pairwise or full feature interactions and joint feature selection and outlier detection methods. Our proposed approaches are able to effectively compute the outlierness of features, which enables outlying feature selection and substantially improves the efficacy of subsequent outlier detection on data with high dimensionality or many noisy features.

Our extensive empirical results show that the average accuracy improvement of our non-IID outlier detectors over state-of-the-art IID outlier detectors ranges from 4% up to 18% on a large collection of real-world data sets; the maximum accuracy improvement on single data sets can be more than 50%, in which stat-of-the-art IID detectors only obtain an accuracy of being nearly equivalent to a random guess. This significant accuracy improvement can have great business value, e.g., the prevention of millions of dollars loss in credit card fraud detection, enabling safer digital environments by militating malicious programs or network intrusions, or saving life by having early detection of fatal diseases. This thesis also offers much more interpretable outlier detection solutions by enabling outlier detection in highly relevant and substantially smaller feature subsets.

Contents

List of Tables

List of Figures

Part I

Research Background and Foundation

Chapter 1

Introduction

This chapter introduces the motivation and the contributions of this research. The structure of this thesis is given at the end of this chapter.

1.1 Research Motivation

Outliers are data objects which are rare or inconsistent compared to the majority of objects [2, 27] . Outlier detection is regarded as one of the most important tasks in data mining due to its wide applications in various domains such as finance, information security, healthcare and earth science. Some application exemplars are credit card fraud detection, network attack detection, terrorist detection, and the early detection of diseases.

Numerous outlier detection methods have been introduced over the years. However, most existing outlier detection methods make a basic assumption that the *outlier factor* (outlierness scoring measure) of the entities in a data set (e.g., feature values, patterns or a combination of feature values, data objects) is Independent and Identically Distributed (IID), but the outlierness of the entities in many real-world applications is not independent, such as the following three important applications of outlier detection.

Example 1 (Disease Detection)**.** *Suppose people diagnosed with flu (influenza) are outliers. Since flu is a respiratory illness that (asynchronously or synchronously) causes multiple symptoms such as fever, headache, muscle pain, dry cough, sore throat, and runny or stuffy nose, the outlierness of having these symptoms (i.e., their relevance to flu) is not independent; rather their outlierness is coupled with each other. Similar phenomena may also be observed from the symptoms of other diseases, such as Type 2 diabetes and breast cancer.*

Example 2 (Fraud Detection)**.** *Suppose fraudulent users who write fake reviews on E-commerce products are outliers. In contrast to honest users, these fraudulent users write positive reviews to promote bad products while writing negative reviews on good products to damage the reputation of these products. As a result, the outlierness of a given user is dependent on the outlierness of some other users who have similar reviewing behaviors as the user.*

Example 3 (Malware Detection)**.** *Suppose malware are outliers. Computers that have been invaded by malicious programs are much more vulnerable than secured computers and are easily attacked by other malicious programs. This type of concurrence results in the interdependence between malicious programs.*

These sophisticated couplings between the outlier factors significantly challenge the methodology of existing methods as they violate the basic assumption of these methods. As a result, existing methods may fail to detect some important outliers, e.g., outliers that are too subtle to be identified without using the couplings between the outliers. This issue is much more severe in challenging outlier detection tasks, such as outlier detection in *noisy data* (i.e., data sets with many noisy features), high-dimensional outlier detection, and outlying feature selection.

To address these issues, we break them down into three key research questions: what are the relations between different outlier factors? how can we effectively model these couplings to build more faithful outlier detection models to address the complex outlier detection problems such as the those given in the aforementioned examples? how can we leverage these couplings to address other challenging outlier detection problems?

This thesis attempts to build a comprehensive methodology of modeling and leveraging coupled outlier factors to solve challenging outlier detection problems. Our main focuses are: (i) coupling learning, which explores and models the relations between the outlier factors of different entities, such as feature values, features, and data objects, and (ii) the applicability of the learned models, which links the coupling learning to the solutions of existing outlier detection challenges. We provide scalable algorithms with extensive theoretical supports and/or empirical results to justify our coupling-based designs, and we show how to use them to address some notoriously challenging outlier detection problems, including noise-resilient outlier detection, high-dimensional outlier detection, and feature selection for outlier detection.

1.2 Contributions

This thesis presents a series of explorations on learning the couplings of outlier factors to have reliable outlierness estimation in complex outlier detection problems. As shown in Figure 1.1, the couplings of outlier factors can occur at different levels, such as feature values, features, and data objects. Coupling learning involves the understanding of four elements: basic coupling relationship, such as binary, conditional, or mutual interdependence; the capacity of couplings, i.e., low-order or high-order interdependence; the utility of couplings, i.e., whether the couplings are fully or selectively relevant to the research problem at hand; and the passage of couplings, i.e., is the coupling relation passed on in a successive [1] or non-successive way. A pathway of linking these four elements results in novel outlierness estimation methods that are different from those of the other pathways, which provides flexibility in addressing different outlier detection challenges.

[1] In this thesis, *cascade* coupling refers to the successive passage of couplings with loops, while *sequential* coupling refers to the sequentially successive passage of couplings without loops.

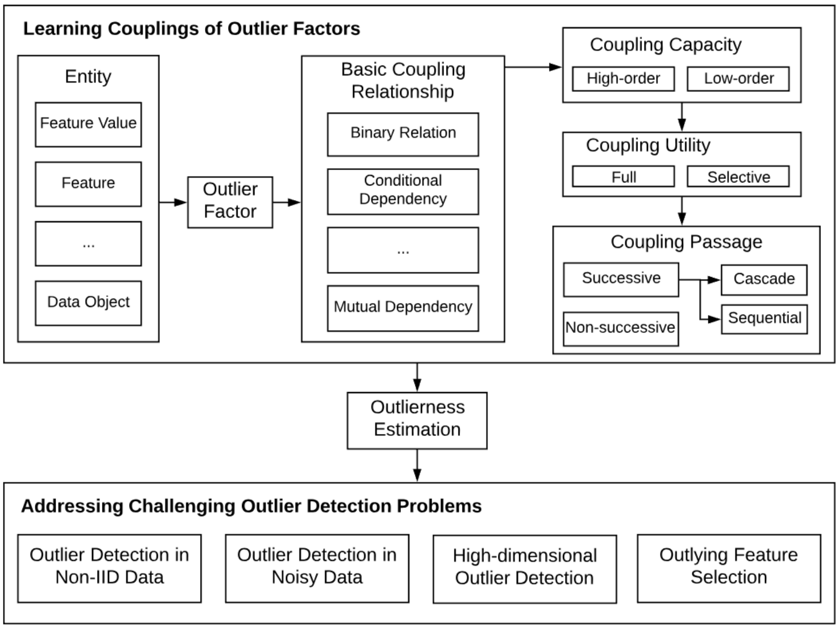


Figure 1.1: Conceptual Architecture of Outlier Detection with Coupled Outlier Factors.

We organize our research into two parts: value-level coupled outlier factors and feature/object-level outlier factors. The research problems and targeted challenges of each part are summarized in Table 1.1.

Table 1.1: Thesis Overview (Main Body)

Part	Research Problem	Targeted Challenge	Chapter
II: Value-level Coupled Outlier Factor	**Conditional cascade couplings**: how is the outlierness of one value influenced by that of other values?	Outlierness estimation of coupled values	4
	Selective conditional cascade: how can we only model useful interactions between outlier factors?	High-dimensional outlier detection in *categorical* data	5
	Binary cascade couplings: how can we quickly and accurately learn the cascade couplings?	Joint feature selection and outlier detection	6
	High-order cascade couplings: how can we efficiently learn the high-order interactions between outlier factors?	Noise-resilient outlier detection	7
III: Feature/object-level Coupled Outlier Factor	**Two-way couplings of features**: how is the outlierness of one feature influenced by that of the other features?	Outlying feature selection	8
	Sequential couplings of objects: how can we sequentially refine a given outlier factor?	High-dimensional outlier detection in *numeric* data	9

1.2.1 Value-level Coupled Outlier Factors

We explore the couplings between outlier factors from the finest level, the value level. Some interesting questions we ask include: how is the outlierness of one value influenced by that of the other values? how can we only model useful interactions between outlier factors?

how can we quickly and accurately learn the cascade couplings? how can we efficiently learn the high-order interactions between outlier factors? and more importantly, what are the benefits of those coupling learning? and how can they address some notoriously challenging problems in outlier detection? Our novel designs and findings w.r.t. these questions are briefed as follows:

- We first learn how the outlierness of feature values influences each other by modeling conditional cascade relations between the outlierness in Chapter 4. Our design enables an effective estimation of the outlierness of data with interdependent values, in which traditional methods are ineffective due to their underlying IID assumption. This work has been published in IJCAI-16 [91].

- We then consider the coupling utility problem in Chapter 5 and learn selective conditional cascade couplings of the value outlierness, which provides an effective approach for detecting high-dimensional outliers in categorical data. High-dimensional outlier detection poses significant challenges to most outlier detection methods due to the concentration of distances and the very sparse high-dimensional space [128]. By learning only selective couplings that are important to outlier detection, we work on highly relevant and condensed space, which enables a more accurate outlierness estimation than other methods. This exploration has resulted in a long paper published in CIKM-17 [94].

- In Chapter 6 we further propose approaches to efficiently learn the cascade couplings with trivial, or no, loss in the accuracy, which can be used to well support the computationally costly joint feature selection and outlier detection. Feature selection for outlier detection is notoriously difficult due to its unsupervised nature and the extremely imbalanced data distribution [2]. Limited work has been reported in this area. It is much more challenging for joint feature selection and outlier detection, which is similar to '*the chicken or the egg*' dilemma. Driven by the fast and effective estimation of value outlierness, we are able to quickly generate and thoroughly examine candidates of outlying features and outliers to achieve the joint optimization. This work has been published in IJCAI-17 [93].

- Instead of learning the low-order couplings in the previous three chapters, in Chapter 7 we learn to incorporate high-order coupling information into the cascade couplings for a more robust outlierness estimation, making our outlier detectors more resilient to noisy features. Noisy features can substantially mislead learning methods, particularly when the class labels are unavailable. Unfortunately, a large percentage of noisy features often presents in outlier detection tasks, because outliers are the minority objects and consequently the percentage of features that are relevant to outliers is often small. We leverage a multi-granularity high-order couplings between values to effectively compute the value outlierness in very noisy data. The extension of this work is under review by the DMKD journal.

1.2.2 Feature/object-level Coupled Outlier Factors

We then examine two types of couplings between outlier factors at the feature/object level to answer the following questions: how is the outlierness of one feature influenced by that of the other features? how can we sequentially refine a given outlier factor? and what is their applicability in addressing real-life outlier detection challenges? We introduce novel solutions for these questions as follows:

- In Chapter 8 we learn non-successive two-way couplings of feature-level outlier factors to define the outlierness of features, i.e., relevant to outlier detection, which can be used to perform feature selection for subsequent outlier detection. A novel parameter-free approach is introduced to efficiently capture pairwise feature interactions with a tight approximation guarantee. This exploration has resulted in a long paper published in ICDM-16 [96].

- We further explore the couplings of outlier factors at the data object level in Chapter 9, in which we learn sequentially interdependent outlier factors with sequential ensemble to mutually refine feature selection and outlier detection in high-dimensional numeric data. Sequential ensembles, such as Adaboost and its variants [43, 124], achieve state-of-the-art performance in classification, but the exploration of sequential ensembles for outlier detection is very limited due to the difficulty of obtaining reliable information to iteratively refine the base models of the ensembles. We introduce a novel way to leverage the correlation between feature selection and outlier detection to address this issue. Also, unlike the designs of feature selection in Chapters 7 and 8 that only capture pairwise interactions, this work captures the full feature interactions. The work has been published in AAAI-18 [95].

1.2.3 Summary

The overall contributions of this thesis are summarized as follows:

- This thesis formulates a *new outlier detection task*, non-IID outlier detection in multidimensional data, which opens a new research direction to the data mining and machine learning community for devising effective and scalable solutions to tackle real-world complex outlier detection problems.

- We introduce the *first architecture* for the non-IID outlier detection task, which provides principled approaches to learn the outlierness interdependence at different levels from feature values, features, to data objects. The architecture breaks down the general coupling learning into a series of important finer-grained components: basic coupling relation, coupling capacity, coupling utility, and coupling passage manner, providing feasible ways to learn sophisticated couplings between outlier factors with efficient models.

- We propose a collection of principled frameworks and their instantiations under the non-IID outlier detection architecture to learn different types of couplings, including four types of couplings of value-level outlier factors: conditional cascade couplings, selective conditional cascade, binary cascade couplings, and high-order cascade couplings, and two types of couplings of feature/object-level outlier factors: non-successive two-way couplings of features, and sequential couplings of objects (see Table 1.1 for the corresponding chapters). Supported by extensive theoretical and empirical justifications, these designs are shown to be scalable and effective in *addressing some notoriously challenging problems*, including non-IID outlier detection, outlier detection in noisy data, and high-dimensional outlier detection.

- This thesis also introduces a set of *seminal work* on unsupervised feature selection for outlier detection in both categorical data and numeric data, including innovative feature selection methods that capture pairwise or full feature interactions and joint feature selection and outlier detection methods. Our proposed approaches are able to effectively compute the outlierness of features, which enables outlying feature selection and substantially improves the accuracy and/or efficiency of subsequent outlier detection methods.

In addition to the above theoretical and/or algorithmic contributions, the potential business values of this thesis are summarized as follows:

- This thesis offers efficient and significantly higher outlier detection accuracy than existing stat-of-the-art IID outlier detectors, which would be of great importance to different applications, e.g., preventing the loss of millions of dollars by detecting more credit card frauds, building safer computer networks by militating more malicious programs and/or network attacks, or saving life by having early detection of fatal diseases. Particularly, the average accuracy improvement of our non-IID outlier detectors over the best IID outlier detector in different contexts ranges from 4% up to 18% on a large collection of real-world data sets; the maximum accuracy improvement on single data sets can be more than 50%, in which stat-of-the-art IID detectors only obtain an accuracy of being nearly equivalent to a random guess.

- We also provide much more interpretable outlier detection solutions for challenging real-life applications, e.g., noisy and/or high-dimensional business applications. This is supported by effectively determining the outlierness of features, which enables users to work on highly relevant and substantially smaller feature subsets and to have a better understanding of why a data object is identified as an outlier.

1.3 Organization

In the rest of this thesis, we introduce common and fundamental definitions or concepts in Chapter 2, including the common symbols, basic definitions, and experimental approach used throughout the thesis. This is followed by a literature review of research achievements that are related to two or more of the following chapters, including related work in IID outlier detection, non-IID outlier detection, and high-dimensional outlier detection in Chapter 3. We discuss additional related work as necessary in the corresponding chapters to provide a straightforward understanding of our research motivation.

We then present the main body of the thesis, including two parts: value-level coupled outlier factors and feature/object-level coupled outlier factors. As shown in Table 1.1, the part at the value level consists of Chapters 4, 5, 6, and 7. In Chapter 4, we explore a conditional relation between the outlierness of different values, aiming to answer the question: how is the outlierness of one value influenced by that of other values? This exploration lays an important foundation to the following chapters. Chapter 5 examines the utility of couplings between values and their implication in high-dimensional outlier detection in categorical data. Chapter 6 then explores a fast and effective coupling learning approach based on binary relations between the values, which shows significant applications in simultaneous feature selection and outlier detection. Lastly, Chapter 7 investigates how to incorporate high-order coupling information into the estimation of value outlierness and shows its importance in outlier detection in noisy data.

The part at the feature/object level consists of the explorations of higher-level coupled outlier factors in Chapters 8 and 9. In Chapter 8, we develop methods to capture the two-way feature interactions for outlying feature selection. This is followed by the examination of sequential couplings of object-level coupled outlier factors and its application in high-dimensional outlier detection in numeric data.

The last part of the thesis consists of Chapters 10 and 11. We summarize the thesis in Chapter 10, and then present the possible future research directions in Chapter 11.

Chapter 2

Preliminaries and Foundation

This chapter begins with a description of the most common symbols, and then introduces the main notations and definitions in the context of non-IID outlier detection that are important to understand the methods and algorithms proposed in this thesis. Lastly, we present the methods for preparing data sets, evaluating and understanding the accuracy of outlier detection in the experiments throughout this thesis.

2.1 Common Symbols

The common symbols and their descriptions are provided in Table 2.1. We define additional symbols as necessary in the following chapters.

Table 2.1: Common Symbols and Their Description. Caligraphic letters for sets. Bold capital letters for matrices. Lowercase bold letters for vectors.

Symbol	Description
$\mathcal{X}$	Multidimensional data set
$\mathcal{F}$, $D = \|\mathcal{F}\|$	Set of D features describing $\mathcal{X}$
$\mathbf{x} \in \mathcal{X}$ and $\mathbf{x} \in \mathbb{R}^D$	D-dimensional data object
$\mathsf{F} \in \mathcal{F}$	Feature in $\mathcal{F}$
$\mathcal{S} \subseteq \mathcal{F}$	Feature subspace
$\mathcal{O}$	Set of outlier candidates
$\mathcal{I} = \mathcal{X} \setminus \mathcal{O}$	Set of inlier candidates
$\mathcal{V}$	Set of categorical values
$\mathbf{r} \in \mathbb{R}^N$	List of outlier scores
G	Graph
$\mathbf{A}$	Adjacency matrix

2.2 Non-IID Outlier Detection

This section first introduces some basic definitions and then formally defines non-IID outlier detection.

Outlier factor is a function of an entity that determines the outlierness (or outlier score) of the entity. The entity can be feature values, features, combinations of multiple values, and data objects. For example, the inverse of the frequency of frequent patterns is a widely-used outlier factor in pattern-based outlier detection methods; k-th nearest-neighbor distance is a commonly used outlier factor in distance-based methods.

Outlierness vector is a vector in which each entry corresponds to an outlier score of an element of a collection of entities. For example, $\mathbf{r} \in \mathbb{R}^N$ is the outlierness vector for data objects in $\mathcal{X}$.

Coupling refers to any relationship or interaction that connects two or more entities [22]. For example, the interaction can be dependency, correlation, matching, or neighborhood-based relation.

Independent and identically distributed (IID) random variables refer to a sequence or other collection of random variables that are mutually independent and have the same probability distribution [119]. In contrast, **non-IID random variables** refer to a sequence or other collection of random variables, in which at least some random variables are coupled with each other and/or have different probability distributions.

Cascade of outlier factors refers to the phenomenon where the outlierness of an entity is coupled with the outlierness of its correlated entities, and the outlierness of these correlated entities are further dependent on that of their own correlated entities, and so on.

Non-IID outlier detection aims to learn the outlier score of a given entity by modeling the sophisticated couplings and heterogeneities among the outlier factors, which is formally defined as follows.

Definition 2.1 (Non-IID Outlier Detection). *Let* $\mathsf{X} \in \mathbb{R}^M$ *be a multivariate random variable or random vector with* M *outlier factors as its components. Then given an entity* e_i*, non-IID outlier detection methods define:*

$$\mathsf{X}_{e_i} \not\perp \mathsf{X}_{e_j},\ \exists j, 1 \leq j \leq M \ \&\ i \neq j, \tag{2.1}$$

or

$$\mathsf{X}_{e_i} \sim \mathcal{D}_i,\ \forall i, 1 \leq i \leq M. \tag{2.2}$$

where $\mathcal{D}_i$ *is an unknown distribution.*

Unlike most outlier detection methods that treat the outlier factors of the entities in an IID way, this definition considers the coupling relation between the outlier factor of one entity and that of the others in Eqn. (2.1) or the heterogeneous distributions taken by different outlier factors in Eqn.(2.2).

Coupled outlier factors are defined as the outlier factors that are not independent of each other, as shown in Eqn. (2.1). Instead of using traditional terms like ‘dependent’, we use the term '*coupled*' to indicate that $\not\perp\!\!\!\perp$ is not limited to the conditional probability-based dependence in statistics, but refers to any relationship or interaction that connect the outlier factors. The following chapters demonstrate how we effectively model different types of couplings between outlier factors, including conditional probability-based coupling, binary coupling, sequential coupling, and their cascade relations, at different levels from feature values, features, to data objects.

Since the independent assumption is often violated in many real-world applications, modeling the couplings between outlier factors helps build outlier detection models that are more genuine to the underlying data characteristics, and therefore, reduce the false positive/negative errors. Also, these couplings enable us to have a robust outlierness estimation of feature values, features, and data objects, making our detection models resilient to noisy features and addressing some notoriously challenging outlier detection problems.

Heterogeneous outlier factors consider the heterogeneous probability distributions taken by the outlier factors of different entities. In other words, one outlier factor that fits some entities may not work on other entities. Therefore, a set of heterogeneous outlier factors is required to identify such outlying entities. Due to the unsupervised nature of outlier detection, some major challenges are: (i) how to determine the effectiveness of a given outlier factor on the data set, (ii) how to devise a set of optimal outlier factors, and (iii) how to effectively unify these outlier factors.

While heterogeneous outlier factors are an important direction in non-IID outlier detection, this thesis focuses on coupled outlier factors. We plan to extend our work to the heterogeneity aspect in the future.

Point outlier is an individual data object that can be considered as outlying w.r.t. the majority of data objects. In addition to point outliers, there exist two other types of more complex outliers, namely contextual outliers and collective outliers [27]. While we focus on the simplest type of outliers, point outliers, to thoroughly understand the non-IID properties of the outliers, it is possible to extend our findings to the other types of outliers. Outliers are hereafter referred to as point outliers.

2.3 Experiment Approach

This section presents the common experiment components or operations in all our experiments, including data preparation and detection performance evaluation methods.

2.3.1 Data Preparation

Publicly available real-world data sets are one of the key drivers in promoting the evaluation and development of learning algorithms. Unfortunately, there exist far less such data sets for outlier detection, i.e., data sets with real outliers, than that for other tasks like classification, regression, and clustering. In our experiments, we use the following widely-used approaches to convert classification data sets into outlier detection data sets.

- **Downsampling approach** downsamples a small class such that the number of data objects in this class accounts for only a very small percentage of the entire data set (we adopt 2% throughout this thesis) [7, 21, 129]. The downsampled class is used as the outlier class and the large class(es) in the original data is treated as the normal class. This approach is used to convert data sets with relatively balanced class distributions. Note that this conversion may produce some features containing only one value in some data sets. We removed these features as they contain no useful information for outlier detection.
- **Rare class conversion** is used to transform extremely class-imbalanced data by treating rare classes as outliers versus the rest of the classes as normal class [7, 21, 73, 77, 121].

The above conversions guarantee that the outlier class chosen is a class with outlying semantics. Additionally, when our purpose is to detect outliers in categorical data, data sets with both numerical and categorical features are used with categorical features only. Categorical features are converted into numeric ones by 1-of-ℓ encoding [21] when applying outlier detection methods that are only applicable to numeric data. The above transformation methods may produce some features containing only one value in some data sets. We removed these features as they contain no information for outlier detection.

2.3.2 Detection Performance Evaluation

Given a data set, an outlier detection method yields a ranking of its data objects w.r.t. the outlier score, $\mathbf{r} \in \mathbb{R}^N$, i.e., the top-ranked objects are the most likely outliers. We evaluate the quality of the ranking by the area under the ROC curve (AUC), which can be directly calculated as follows [53]:

$$AUC = \frac{\sum_{i=1}^{N_{C_0}} [rank_i - i]}{N_{C_0} N_{C_1}}, \tag{2.3}$$

where N_{C_0} and N_{C_1} are the number of objects in the outlier and normal classes respectively, and $rank_i$ denotes the rank of the i-th outlier in an ascending object ranking.

AUC inherently considers the class-imbalance nature of outlier detection tasks, making it comparable across different data sets [21]. An AUC value close to 0.5 indicates a random ranking of the objects. A higher AUC indicates better detection performance. The consideration of the class imbalance and the easy interpretation make AUC one of the most widely-used performance measures in outlier detection. As an overall performance measure, AUC is used throughout all our experiments. In some cases, e.g., Chapter 6, we also use the precision at the top n positions, i.e., $P@n$, to evaluate our methods that attempt to optimize the outlier ranking at the top n positions (see Section 6.5.2 for detail).

For algorithms that involve sampling, their AUCs are the averaged results over 10 independent runs to deliver reliable performance.

The *Wilcoxon* signed rank test [35] is used by default to examine the significance of the AUC performance of our proposed methods against its counterparts, unless otherwise stated.

2.3.3 Data Indicator for Outlier Detection

Data indicator refers to measures that capture inherent characteristics of data sets, which is used to understand and quantify the underlying data characteristics that are sensitive to the performance of learning methods. This has been shown to be critical for the design and the evaluation of learning methods [21, 24, 39, 58, 74, 109]. A wide range of data indicators has been introduced to quantify data complexity at the feature and/or object levels for classification tasks or sequence analysis [24, 58, 74, 109], while little work has been done for outlier detection.

Two relevant studies are [21, 39]. In [21], a variety of k-nearest-neighbor-based outlier detection methods is employed to evaluate the complexity of many publicly accessible data sets via their detection performance. Two data indicators, *difficulty* and *diversity*, are defined based on agreements and conflicts in the performance of the detectors. This is very different from our work in that we quantify the data complexity from specific data aspects by designing various data indicators to capture different underlying data characteristics. A more related study is [39], which introduces three indicators, *point difficulty*, *clusteredness*, and *relative frequency*, to create benchmark data sets with different characteristics by varying these three indicators. These indicators are designed at the object level and are mostly proximity-based. In contrast, we define a set of indicators that span the value to object levels to capture more affluent data characteristics. However, it should be noted that having an accurate estimation of the data complexity itself is a very challenging task. We attempt to use these indicators to gain some insights into the data characteristics and our empirical results.

Value Coupling Complexity

We introduce a complexity measure κ_{vcc} to show how the value coupling relations affect the detection performance. We first define two concepts: noisy and positive value couplings. Noisy value couplings are the co-occurrence of multiple infrequent values contained by normal objects. Positive value couplings refer to the co-occurring infrequent values contained by outliers. Positive value couplings are positive because they are consistent with the outlier definition. Noisy value couplings are the opposite of the positive couplings and are therefore negative. Let N_{C_0} and N_{C_1} denote the number of data objects in the outlier class C_0 and the normal class C_1, respectively. N'_{C_0} and N'_{C_1} denote the number of outliers and normal objects that contain at least two values with a frequency no more than a low frequency threshold θ. The rate of noisy and positive value couplings can then be defined as $nvv = \frac{N'_{C_1}}{N_{C_1}}$ and $pvv = \frac{N'_{C_0}}{N_{C_0}}$, respectively. We then define κ_{vcc} as

$$\kappa_{vcc} = \frac{nvv}{pvv + nvv + \epsilon}, \tag{2.4}$$

where ϵ is a small constant used to avoid the zero probability problem.

When $nvv \gg pvv$, the noisy value couplings dominate a data set, leading to high data complexity. The outlier detection task is easy if $nvv \ll pvv$. $\kappa_{vcc} \in [0, 1]$, and a larger κ_{vcc} indicates a greater dominance of nvv over pvv and, therefore, has higher data complexity.

$\theta = 0.05$ and $\epsilon = 0.001$ are used in our evaluation.

Relevance of Feature Value Sets

κ_{rel} represents the relevance of a set of values w.r.t. the outlier class label. We use the probability of the outlier label given a single value to measure the relevance of the value. κ_{rel} is then defined as the average conditional probability of the outlier class label over all its values in a value set $\mathcal{U}$, i.e., $\kappa_{rel}(\mathcal{U}) = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}} P(C_0|u)$. High κ_{rel} indicates strong couplings between the outlier class and the values in $\mathcal{U}$.

Feature Noise Level

Due to the unsupervised nature of outlier detection, irrelevant features, in which outliers contain values with similar or higher frequencies than that of normal objects, mask normal objects as outliers. These features are therefore 'noise' to outlier detection. The presence of a large proportion of these features renders outlier detectors less effective. We accordingly define the feature noise level κ_{fnl} as the proportion of noisy features to characterize this difficulty. κ_{fnl} is defined as

$$\kappa_{fnl} = \frac{\sum_i \mathbb{I}(\mathsf{F}_i)}{D}, \tag{2.5}$$

where $\mathbb{I}(\mathsf{F}_i)$ is an indicator function, which returns one if feature F_i is considered a noisy feature and zero otherwise.

A feature is thought of as noise if its AUC-based feature efficiency is smaller than 0.5, i.e., outliers are more likely to be assigned with smaller outlier scores than normal objects along that feature in a random selection of object pairs. Per the definition of outliers, we first return an outlier ranking $\mathbf{r}$ using the inverse of the frequency of each object's value in a given feature F_i, and then compute the AUC for the feature F_i using Eqn. (2.3). $\kappa_{fnl} \in [0, 1]$ and a large κ_{fnl} means a high level of feature noise.

Feature Redundancy Level

Redundant features firstly need to be relevant features. Following the idea in Section 2.3.3, we defined relevant features as features whose corresponding AUC is more than 0.5. To examine whether a given relevant feature is redundant or not, we check every pair of the relevant features to compare the AUC by using pairwise feature combinations with that using individual features. One feature is thought to be redundant to another if the AUC difference is less than 0.01. We report the percentage of such combinations as κ_{rdn}.

Outlier Separability

One basic measure of the difficulty of outlier detection is the separability of outliers from normal objects. However, it is challenging to exactly compute this difficulty, since the separability varies significantly in different subspaces and the number of possible subspaces is 2^D. Rather than searching over such a huge space, existing studies focus on the separability in single features, as having strongly relevant features normally enables learning methods to achieve good accuracy [58, 74]. *Feature efficiency* is widely used for evaluating class separability in supervised classification problems [58, 74], in which the separability is defined by the range of non-overlapping values spanned by classes. However, this definition is not suitable for outlier detection, for which data has an extremely skewed class distribution. We are interested in the capability of ranking outliers prior to normal objects. Based on the definition of outliers, if there exists a feature where the outliers always contain more infrequent values than normal objects, the outliers can be easily separated from the normal objects. Such features are the most efficient. When outliers and normal objects contain the same values, or outliers contain more frequent values than normal objects, that feature is considered to be inefficient or noisy. The outlier separability of a feature F_i is defined as

$$\kappa_{sep}^{i} = performance(\mathbf{r}_i), \tag{2.6}$$

where $\mathbf{r}_i$ is the ranking list of data objects using the inverse of the frequencies of the values in feature F_i and $performance(\cdot)$ is a performance evaluation method. We instantiate $performance(\cdot)$ by computing the AUC based on $\mathbf{r}_i$. We then use the resulting AUC to denote the efficiency of the feature. Similar to [58, 74], the overall outlier separability κ_{sep} is represented by the maximum feature efficiency:

$$\kappa_{sep} = \arg\max_{i} \kappa_{sep}^{i}. \tag{2.7}$$

$\kappa_{sep} \in [0, 1]$ and a larger κ_{sep} indicates better outlier separability and lower data complexity. Below we use outlier inseparability defined as $\kappa_{ins} = 1 - \kappa_{sep}$. This is to have larger quantization values indicating higher data complexity, making this indicator work consistent with the other indicators.

Heterogeneity of Categorical Distribution

Most outlier detection methods implicitly assume that distributions taken by different features are homogeneous. Measuring the heterogeneity between the distributions of the features therefore offers one basic way to evaluate the problem difficulty in the given data. It is well known that location parameters convey key properties of probability distributions. Many statistical tests, e.g., the t-test or signed-rank test [35], are available for the distribution location test, but they are ineffective in evaluating the heterogeneity of distributions across the features, which is particularly true for categorical data. This is because the sample size in categorical distributions (i.e., the number of values per feature) varies significantly in different features and/or data sets, which often violates the sample size and distribution assumptions made in these tests. The problem is simplified as follows. We consider the mode as the key location parameter and use the difference of the frequencies of the modes across features to define the heterogeneity level. Specifically, the heterogeneity level κ_{het} is defined as the average difference in mode frequency over all possible feature pairs:

$$\kappa_{het} = \frac{2}{D(D-1)} \sum_{1 \leq k_i < k_j \leq D} \frac{\mathit{freq}(m_{k_i})}{\mathit{freq}(m_{k_j})}, \tag{2.8}$$

where $\{m_{k_1}, \cdots, m_{k_D}\}$ are the modes of all D features sorted based on their frequencies in descending order. $\kappa_{het} \in [1, \infty)$ and a large κ_{het} indicates strong heterogeneity.

Chapter 3

Literature Review

Outlier detection methods can be generally categorized into supervised methods, semi-supervised methods and unsupervised methods. Compared to supervised methods and semi-supervised methods, unsupervised methods are more applicable and widely used in industry, because obtaining accurately labeled data for most outlier detection applications often comes at a very high cost [2]. This work therefore focuses on unsupervised methods.

This chapter reviews the literature that is related to the general challenges we intend to solve throughout this thesis, including outlier detection with IID and non-IID outlier factors, and high-dimensional outlier detection. Some related work that is particularly related to a specific chapter will be discussed in the corresponding chapters. A summary of this review is provided at the end of this chapter.

3.1 Traditional Outlier Detection Methods

Most existing methods for outlier detection assume the outlier factors within a data set are IID. The methods for both numerical and categorical data are reviewed below.

3.1.1 Methods for Numeric Data

Most existing outlier detection methods focus on numeric data. Among these methods, proximity-based methods are arguably the most widely-used approach while ensemble methods have emerged as the leading approach in recent years. We discuss these two methods as follows. The other types of methods can be found in [2].

Proximity-based Methods

Proximity-based methods include distance-based methods, density-based methods, and clustering-based methods [2]. Some popular methods include: k-nearest-neighbor distance, kNN [66, 101]; local outlier factor, LOF [20, 105]; and clustering-based outlier factor, CBUID [63]. These methods generally assume that objects in the regions of low density are outliers. This assumption is underpinned by: the k-th or average k nearest-neighbor distance(s) in distance-based methods; relative lower density in neighborhood-based regions in density-based methods; and large distances to cluster centroids in clustering-based methods. To compute the outlier scores, the distance-based methods and density-based methods rely on expensive distance computations with a time complexity of $O(N^2)$, where N denotes the number of objects. This computational time may be reduced to $O(N \log N)$ if the objects are pre-indexed by an indexing scheme like R^*-tree [15] or k-d tree [16]. Some other strategies, such as pre-clustering or pruning rules [9, 14], have also been explored to improve the detection efficiency. By using efficient clustering methods to discover the predefined clusters, clustering-based methods like CBUID are efficient with a linear time complexity, but it is difficult for these clustering methods to find natural clusters with different shapes. Using sophisticated clustering methods like DBSCAN [106] may help solve this issue, but they have the expensive computation issue.

Ensemble Methods

In recent years, ensemble-based methods have shown state-of-the-art detection performance in a variety of data sets. This type of methods can be generally categorized into subsampling-based ensembles [92, 111, 116, 129] and subspace-based ensembles [7, 64, 73, 97]. This section discusses subsampling-based ensembles. The subspace-based methods are reviewed under a more related topic in Section 3.3.2.

Subsampling-based outlier ensembles compute the distances, local densities or other measures on small random subsamples to compute outlier scores. These methods substantially reduce the quadratic time complexity to a (nearly) linear time complexity w.r.t. data size, while at the same time achieving AUC performance that is comparable to, or better than, the same outlier factors that work on the full data set. For example, it is reported in [111] that the method named Sp, which uses the nearest neighbor distances in a very small *single* random subsample as outlier scores, can achieve significantly better AUC performance than kNN and several of its variants; at the same time, Sp runs orders-of-magnitude faster than kNN. Similar results are reported in [92, 116], in which a bagging ensemble of Sp, called LeSiNN, is introduced to provide more stable and better AUC performance at the expense of trivial computational time.

One of the most popular methods in this category is isolation-based methods [49, 77, 78], of which iForest (Isolation Forest) [77] is the most well-known method. iForest leverages the property that outliers are susceptible to isolation to build a set of isolation trees to identify outliers. Each tree is grown using a small random subsample until every data object is isolated, in which each tree node is built by randomly selected cut points on a randomly selected feature. To score a data object, the path length traversed from the root to a leaf node by the data object is used as the outlier score. Because outliers can be isolated using significantly fewer partitions than normal instances, outliers have a shorter path length than normal instances. iForest has a linear time complexity w.r.t. data size and dimensionality and obtains the best AUC performance in many widely-used outlier detection data sets [77].

The ensemble methods, such as Sp, LeSiNN, and iForest, are state-of-the-art methods in terms of both effectiveness and efficiency in general data sets. However, they have problems handling complex data sets. For example, they are ineffective for handing high-dimensional data sets. This is because the distances used in Sp and LeSiNN are not an effective measure in a high-dimensional space since the nearest and farthest neighbor distances have nearly no difference in this space [128]. For iForest, the volume of the high-dimensional space is huge, and as a result, iForest has a very rare chance of choosing the right regions in building their isolation trees since iForest randomly chooses the subsamples and cut points. More importantly, all these methods are based on the assumption that the outlier factors within the data are independent, which renders them less effective in data sets with non-IID entities.

3.1.2 Methods for Categorical Data

By contrast, significantly less research has been conducted on categorical data. Among the existing methods, most are pattern-based to address its discrete nature in categorical values. Additionally, there have been some efforts on transforming categorical data into numeric data to use the aforementioned proximity-based or ensemble methods.

Pattern-based Methods

Pattern-based methods typically search for infrequent/frequent patterns using approaches such as frequent pattern mining [57, 88, 108, 114], information-theoretic measures [7, 121], and probability tests [34, 120] and build pattern-based detection models. Those objects with infrequent patterns are considered to be outlying. Specifically, the most widely-used frequent pattern-based method, FPOF [57], employs frequent patterns as normal patterns and computes outlier scores based on the containment of the frequent patterns and their frequencies. FPOF is one of the most popular and effective outlier detectors for categorical data [47, 68, 121], performing better than the popular infrequent pattern-based method LOADED [88] that, in contrast, uses infrequent patterns to detect outliers. CompreX is a state-of-the-art information-theoretic-based method, which uses the data compression cost in a minimum description length [13] based feature partition space as outlier scores. It has powerful detection performance in data sets with different structures. Probability test-based methods perform statistical tests to obtain normal or abnormal patterns for detecting outliers. MarP [34] is a marginal probability-based probability test method, which uses the inverse of marginal probabilities to define outliers. Although it is a simple method, it performs better than, or comparably to, several other probability test methods such as Bayesian network-based methods [34].

However, all of these methods ignore the interdependence of patterns but calculate the outlier scores of individual patterns. As a result, they may fail to capture the genuine outlying degree of the patterns and overlook important outliers, especially in complex data. For example, in data sets with many noisy features, these methods identify a large proportion of misleading patterns. Since they treat potentially wrong and correct patterns independently, all patterns are scored in an identical way, which can mislead the outlier scoring process and incorrectly report many normal objects as outliers.

In practice, the pattern discovery-based detection [34, 57, 88, 108] has time and space complexities that are exponential to the number of features. Though a heuristic search was used in [7] to reduce the complexity from exponential to quadratic, the search is still computationally intensive in high dimensional data. Some other work accelerates the pattern discovery phase, e.g., by searching condensed representations of patterns [68] or pattern sampling [47]. However, they may overlook important outliers and/or perform unstably.

Also, it is non-trivial to tune the parameters (e.g., the minimum support and pattern length) involved in the pattern search, as the characteristics of patterns differ greatly between data sets [7, 47]. For example, in frequent pattern-based methods, a small minimum support generates a substantially large set of normal patterns, which may lead to false alarms and more expensive computations; while large minimum support results in an insufficient number of patterns, leading to a high false negative error.

Representation Learning-based Methods

Representation learning-based methods aim to transform categorical data into numeric data for the use of numeric data-based methods in subsequent outlier detection. The 1-of-ℓ encoding and IDF (Inverse Document Frequency) encoding [21] are the two most commonly-used categorical-to-numeric data transformation methods. 1-of-ℓ encoding, also known as one-hot encoding, converts a categorical feature with ℓ values into ℓ binary features, in which the values '1' and '0' indicate the presence and absence of a categorical value, whereas IDF encodes a feature value as $\mathit{IDF}(v) = \mathit{ln}(N/\mathit{freq}(v))$. These two methods are easy-to-implement and very efficient, but they do not capture much intrinsic data characteristics. Another main research line is similarity or metric learning which represents categorical data with an object-object similarity matrix. Many similarity measures have been introduced for categorical data over the years [5, 60, 61, 118, 126], which typically attempt to capture different types of interactions between values to produce a reliable similarity matrix. Since they work on the object-object similarity matrix, they often have quadratic time complexity, which impedes their applications to large-scale data sets. Another issue for similarity learning methods is that it is difficult to define a consistently effective similarity measure for data sets with different characteristics [2, 17]. Deep neural networks-based representation learning has been very popular and successful on image and text data, while limited work [122] has been done on categorical data because: (i) it is difficult to define proper loss functions to capture the sophisticated interactions between the values and (ii) the data size in many applications is not large enough to well train the neural networks. More efforts are required to explore whether deep learning-based methods can gain similar success on categorical data. Moreover, all these methods are mainly focused on capturing the regularity information for clustering tasks, thus, they may ignore some irregularity information that are important for outlier detection.

3.2 Non-IID Outlier Detection Methods

This section first reviews the research progress in non-IID learning and then discusses related work in non-IID outlier detection.

3.2.1 Non-IID Learning

Since the IID assumption generally does not hold in real-world applications, non-IID learning has received increasing attention. In terms of learning relevant data dependencies, one main relevant research line is to learn the spatial and/or temporal dependent data [33, 85, 110]. However, significantly less work has been reported on the complex hierarchical couplings and heterogeneities within a single data set, in which most existing learning methods cannot have a faithful modeling of these complexities due to their IID assumption [22, 23, 25]. Typical research models the explicit or implicit dependencies or relations. For example, the similarity between two observations (images patches) is influenced by the similarity of the neighbors of these two observations [125]; the sampling of observations is influenced by the observations (visual words) we sampled before and their distribution [32]; the similarity between two values is influenced by the similarities of the correlated values/patterns of these two values [118]. More recent efforts have been made on jointly learning explicit and implicit couplings, e.g., the explicit and implicit couplings between users (items) in very large-scale data [37].

On the other hand, learning different types of heterogeneities, such as heterogeneities between features, subspaces, views, objects, class labels, or graphs [23, 55, 112], using multi-view/task learning and heterogeneous information networks has been explored in a range of learning tasks. Another important research line is transfer learning approaches [90], which are devised to tackle the data distribution shift between training and testing data sets or between source and target domains.

3.2.2 Outlier Detection with Non-IID Outlier Factors

Limited work has been reported on non-IID outlier detection in static multidimensional data sets. Similar to non-IID learning, many studies focus on detecting outliers in data with explicit non-IID properties, e.g., outlier detection in graph or temporal data [6, 25, 50]. However, little research is available on leveraging implicit coupling/interdependent information to improve the aforementioned IID outlier detection methods, though it shows to be effective in various other domains [40, 44, 62, 118, 125]. A few related studies are [29, 82, 113], which exploit homophily couplings (a commonly observed relation in many real-world applications [42, 69, 115]) between the object-level outlier factors to identify abnormal events in fraud detection [82] and malware detection [29, 113]. These studies assume the misstated user accounts and malicious files have homophily couplings, respectively. Our work is very different from these studies in two major aspects below. (i) They incorporate domain knowledge (e.g., some labeled data objects) into their homophily learning models through semi-supervised learning, whereas our methods do not require labeled data. (ii) They focus on domain-specific problems using graph data, e.g., account-account graph or file-file graph, whereas we investigate the homophily at different levels of entities on generic multidimensional data, which has broader applications.

A few studies [102, 107] are also available for capturing heterogeneous outlier factors in outlier detection. These studies focus on how to construct heterogeneous ensembles of different outlierness scoring methods (or one method with different parameter settings) by leveraging the techniques of ensemble learning. Outlier detection in multi-view data [76, 123] or multi-source data [45] is also related to this topic. While this thesis focuses on the interdependence aspect of non-IID outlier detection, it is interesting to explore the use of heterogeneous outlier factors in complex data sets. We plan to investigate this problem in future work.

3.3 High-dimensional Outlier Detection

High-dimensional outlier detection is a significant challenge due to the curse of dimensionality. Existing solutions to this problem can be generally grouped into three categories: full-space-based methods, subspace-based methods, and feature selection-based methods. Full-space-based methods attempt to define outlierness measures that are more effective on a high-dimensional space, while subspace- and feature selection-based methods aim to select relevant feature subspace(s) for subsequent outlier detection methods to work on, which helps reduce the effect of the dimensionality curse.

3.3.1 Full-space-based Methods

As discussed above, traditional outlier detection methods like LOF, kNN and their numerous variants [21] rely on pairwise distances on the full data space to define outliers and they fail in high-dimensional data as the concept of distance becomes less meaningful with increasing dimensionality [128]. Some methods [11, 46, 70, 100] are dedicated to defining more robust outlierness measures than neighborhood-based measures for high-dimensional space. One representative state-of-the-art method is the angle-based methods called ABOD [70, 100], which uses the variance of the angles between a given data objects and the other objects to define outlier scores. A small angle variance indicates high outlierness. Although these methods successfully avoid to the direct use of pairwise distance in outlier scoring, their premises are dependent on the proximity concept in the original full space, thus, they are still biased by irrelevant features [128]. Also, these methods often require an input for the neighborhood size, which is heavily dependent on data size and data distribution and is difficult to be tuned as class labels are unavailable [97].

3.3.2 Subspace-based Methods

The number of possible feature subspaces increases exponentially with the dimensions. Therefore, heuristic search and random search methods are used to generate the subspaces for subsequent outlier detection to make the problem computationally tractable. In general, subspace-based outlier detection methods include deterministic and non-deterministic methods based on the way they generate the subspaces.

Deterministic Subspace-based Methods

Deterministic subspace methods includes local pattern-based methods [4, 10, 57], feature partition-based methods [7] and statistical dependence-based methods [64]. They are deterministic in the sense that they produce exactly the same subspaces/patterns that satisfy a given criterion. They normally first search occurrence frequency/local density, minimum description length or statistical dependence tests-based outlying subspaces/patterns, and then computes outlier scores in subspaces to avoid the inclusion of irrelevant features. However, their subspace/pattern search is still computationally costly (e.g., at least quadratic time complexity) in high-dimensional data. Also, the presence of irrelevant features may mislead the search to produce irrelevant subspaces, leading to false positive errors [91].

Non-deterministic Subspace-based Methods

In contrast to deterministic methods, non-deterministic methods [73, 77, 97, 104] work on randomly generated subspaces. These methods generally have substantially better efficiency than deterministic methods, since they do not require the costly subspace search and their random subspace generation is very fast. However, the random subspace generation may include many irrelevant features into subspaces while omit relevant features in high-dimensional data, where irrelevant features dominate over relevant features. Also, it is difficult to determine the size of these random subspaces, i.e., the number of features contained by each subspace.

3.3.3 Feature Selection-based Methods

Alternatively, feature selection-based methods aim to identify a single optimal feature subset that reveals the exceptional behaviors of all outliers. Although feature selection has shown effective in enabling clustering and classification for decades [75], there exists limited work on outlier detection because it is challenging to (i) define feature relevance to outlier detection given its unsupervised nature and (ii) find a single feature subset enabling the detection of all outliers.

Very limited feature selection methods have been designed for outlier detection, e.g., selecting features for imbalanced data classification or supervised outlier detection [12, 30, 80]. However, they are inapplicable for the context which has no class label information or where it is too costly to obtain class labels. Unfortunately, many real-world outlier detection applications fall into this scenario.

Some related work includes [56, 86, 98, 121]. In [56], a partial augmented Lagrangian method is introduced to co-select objects and features that are relevant to rare class detection. While the feature selection and rare class detection are shown effective in unsupervised settings, as pointed out by the authors, they assume that the objects in rare classes are strongly self-similar. This assumption does not apply to the nature of outlier detection, where many outliers may be isolated objects and distributed far away from each other in the data space. In [121], an unsupervised feature weighting for outlier detection on categorical data, denoted as ENFW, is introduced. This method employs an entropy-based measure to weight features and highlight strongly relevant features for subsequent outlier detection. However, it evaluates individual features without considering any feature interactions, and is thus very sensitive to noisy features. The method denoted as RegFS in [86, 98] defines the relevance of features by their correlation to the other features. The assumption is that outliers correspond to the violation of the dependency among normal objects and independent features are not useful in capturing such dependency/violation [2]. This assumption may be invalid since some features can be strongly relevant to outlier detection but not correlated to other features.

One shared problem for the above subspace/feature selection-based methods is a type of filter-based methods, which search feature subset(s) independently from the subsequent outlier detection methods, and they may consequently result in feature subset(s) that are suboptimal to the outlier detectors.

3.4 Summary

Most existing work on outlier detection in either numeric data or categorical data assume that the outlier factor of an entity in the data is independent from that of the other entities, whereas this thesis aims to learn different types of couplings between the outlier factors of the entities in *multidimensional data* at the different levels, which are of great importance to identify outliers in data sets with complex interdependence. There have been some interesting previous studies on non-IID outlier detection, but they are focused on data with explicit interdependence, such as temporal data and graph data, rather than multidimensional data. As far as we know, this thesis provides the first comprehensive exploration of non-IID outlier detection in multidimensional data.

Irrelevant features can mask outliers as normal objects, which is thus 'noise' to outlier detection. In addition, they are also one major cause of the curse of dimensionality issue. To reduce the effect of irrelevant features, many subspace-based methods are proposed to identify relevant subspaces for subsequent outlier detection in noisy data or high-dimensional data, but the search of relevant subspaces can be misled by these noisy features, and moreover, the search process is often computationally costly since the search space increases exponentially with the dimensionality. We show in this thesis that modeling the couplings of the outlier factors provides scalable and effective approaches to compute the outlierness in noisy data or high-dimensional data. This is one main benefit brought by non-IID outlier detection.

In addition, outlying feature selection is an alternative approach to outlier detection in noisy data or high-dimensional data, but limited work has been reported in this area due to the difficulty in computing the relevance of features to outlier detection.We show in this thesis that the rich interactions between entities at different levels in multidimensional data also enable us to reliably compute the feature relevance, which provides principled approaches for outlying feature selection. This is another main benefit brought by non-IID outlier detection.

Part II

Value-level Coupled Outlier Factors

Learning Couplings of Value-level Outlier Factors

Feature values are the foundation element of multidimensional data objects. The outlierness of values is more finer-grained than that of patterns in feature subspaces or the full space. Therefore, knowing the outlierness of values can provide important insights into challenging outlier detection problems.

Feature values of data objects are naturally coupled with each other in many application cases where the behaviors of the data objects occur asynchronously or synchronously. Therefore, considering the couplings between the values is required to have a reliable inference of the abnormality of the values. In this part, we examine four different types of couplings of value-level outlier factors to have a more accurate outlierness estimation of the values and their applications in addressing challenging outlier detection problems:

- **Conditional** cascade couplings, which enable the effective outlierness estimation of not independent values (Chapter 4);
- **Selective conditional** cascade couplings, which contribute to fast and effective high-dimensional outlier detection in categorical data (Chapter 5);
- **Binary** cascade couplings, which can be used to compute value outlierness in a closed-form and drive joint feature selection and outlier detection (Chapter 6);
- **High-order** cascade couplings, which enable more accurate value outlierness estimation in noisy data (i.e., data with many noisy features) (Chapter 7).

For each exploration, we provide motivations, generalize abstract frameworks, and introduce instances of the frameworks, followed by theoretical and empirical justifications of our frameworks' instantiations.

Chapter 4

Conditional Cascade of Outlier Factors

4.1 Introduction

How can we know the traits of a data object's behaviors, e.g., the outlierness? How much can we know about their traits from the traits of other data objects? This chapter focuses on these questions, and introduces approaches to explicitly capture the mutual influence of the outlierness between the behaviors, i.e., feature values of the data objects. Particularly, we are interested in their *homophily couplings*, which refer to the phenomenon that an entity tends to bind with other entities that have similar traits and consequently the entities have mutually positive influence on their traits [42, 83].

As quoted in many idioms like '*a man is known by the company he keeps*' and '*birds of a feather flock together*', this kind of coupling is common in our real life, e.g., the function of a protein can be inferred from its interacted proteins of known functions [87]; the happiness of people is influenced by the happiness of their surrounding friends [42]; the alcohol use, smoking, and aggressive and/or illegal behaviors of adolescence is influenced by their peers [18].

By having *homophily* couplings in outlier detection, we posit that the outlying behaviors are explicitly and/or implicitly coupled with each other, and the outlierness of one behavior is positively influenced by the outlierness of other behaviors. As a result, the outlierness of a behavior is proportional to the outlierness of its coupled behaviors, and the outlierness of these coupled behaviors are further proportional that of their own coupled behaviors, and so on. Such couplings form an iterative *cascade* influence on the outlierness of behaviors. Capturing these couplings helps to have a faithful measure of the behaviors' abnormality, e.g., the unexpected function of proteins and the risk of having depression.

However, most existing outlier detection methods for categorical data [7, 34, 57, 88, 108, 114, 121], take the IID assumption. Such methods identify a set of normal/outlying patterns from all possible patterns and compute the outlierness of the identified patterns independently. In doing this, they ignore the couplings between the patterns and cannot capture the above cascade influence. Consequently, they may meet with critical problems, e.g., they may treat the wrongly identified patterns as important as the genuine ones and result in high detection errors. Accordingly, properly modeling the outlying behaviors with homophily couplings is critical and can iteratively reinforce the outlierness of genuine outlying patterns, which may consequently reduce the impact of the erroneous patterns.

To detect outliers in categorical data with the coupled behaviors, this chapter introduces a novel framework, called Coupled Unsupervised OuTlier detection (CUOT), to capture the above cascade influence. CUOT estimates the outlierness of each value by modeling complex couplings between feature values. It uses *intra-feature value couplings* that consider the local context within a feature to compute semantically comparable initial outlierness for the feature values. On the other hand, CUOT uses the *inter-feature value couplings* to model the influence of different feature values on the outlierness of values. As shown in Figure 4.1, CUOT then integrates these two components into a *value-value graph*. It subsequently learns the outlierness of values by using off-the-shelf graph mining techniques to capture the cascade outlierness influence between the nodes. The value graph representation is used because it can support the flexible and effective modeling of the homophily couplings.

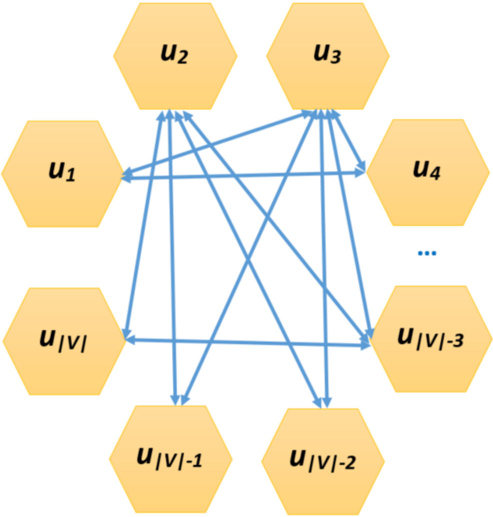


Figure 4.1: Conditional Cascade Couplings of Value-level Outlier Factors. u_i represents a value.

The defined outlierness of values can detect outliers in non-IID categorical data in two ways: (i) by directly computing the outlier scores of objects through consolidating the outlierness of their values; (ii) by first measuring the relevance of a feature to outliers through consolidating the outlierness of the values in the feature, i.e., features with high outlierness are considered to be *outlying features*, and then selecting important features for subsequent outlier detection.

CUOT is implemented by a method called Coupled Biased Random Walks (CBRW). CBRW defines an initial value outlierness via the feature mode-based normalization, and considers the mutual dependency of the outlierness of values from different features using the conditional probabilities of those values. The initial value outlierness and the outlierness influence between values are mapped onto a directed attributed value-value graph and modeled by biased random walks to estimate the outlierness of all values.

Accordingly, our contributions can be summarized as:

i. A novel coupled unsupervised outlier detection (CUOT) framework is introduced to estimate the outlier score of each *value* by modeling intrinsic intra- and inter-feature value couplings. Modeling value-level interactions provides an effective and efficient way to model sophisticated value interactions in complex categorical data. Moreover, the value-level outlier scores are more fine-grained and flexible than the pattern-level scores. This approach makes outlying feature selection possible in addition to direct outlier detection.

ii. CUOT is further instantiated to the CBRW method, which integrates the initial outlierness of values and the outlierness influence between values in a seamless manner. CBRW is guaranteed to be converge and perform stably with its only parameter.

Extensive experiments show that: (i) our CBRW-based outlier detection method significantly outperforms five state-of-the-art methods on 15 real-world data sets with different levels of non-IID values, outlier separability, and feature noise; (ii) the CBRW method runs substantially faster than pattern-based methods because the costly pattern searching is not required; (iii) the CBRW-based feature selection method can be used to significantly improve two different types of outlier detectors; (iv) the time complexity of CBRW is linear w.r.t. data size and nearly linear w.r.t. data dimensionality.

The rest of this chapter is organized as follows. The CUOT framework is detailed in Section 4.2. CBRW is introduced in Section 4.3. A theoretical analysis of CBRW is presented in Section 4.4. Section 4.5 gives the evaluation results. This chapter is then summarized in Section 4.6.

4.2 The Proposed CUOT Framework

The CUOT framework is shown in Figure 4.2. CUOT first defines intra- and inter-feature value coupling functions, δ and η, to capture the intrinsic data data characteristics w.r.t. the outlierness of values. δ is an outlier factor that computes the initial outlierness of each value based on the value couplings within individual features $\{\mathsf{F}_1, \mathsf{F}_2, \cdots, \mathsf{F}_D\}$. η considers the outlierness influence between values in a finite set of pairwise value couplings in feature subsets $\{\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_{D\times D}\}$, resulting a value coupling matrix $\mathbf{M}_\eta$. CUOT then maps these two components to a value-value graph G and further defines a graph-based scoring function ϕ to learn the final value outlierness.

CUOT detects outliers in a way fundamentally different from existing frameworks in terms of three major aspects. (i) CUOT learns *value* outlierness by modeling complex value interactions, whereas the existing frameworks rely on *pattern* outlierness and focus on the efficacy of searching for normal/outlying patterns. (ii) CUOT leverages intrinsic value couplings to capture the intra- and inter-feature outlierness, and further models the joint effects of the two different component. CUOT therefore obtains a more reliable outlierness estimation in real-world data with coupled behaviors, while the existing frameworks are mainly based on the inter-feature outlierness and treat the outlierness of different patterns

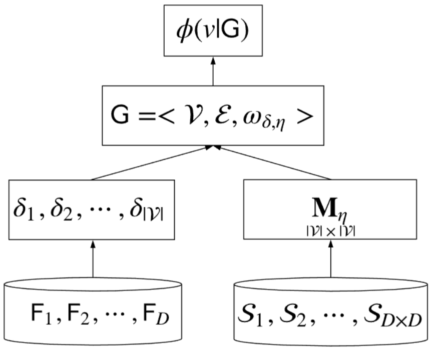

$\phi(v|\mathsf{G})$
$\mathsf{G} =< \mathcal{V}, \mathcal{E}, \omega_{\delta,\eta} >$
$\delta_1, \delta_2, \cdots, \delta_{|\mathcal{V}|}$
$\mathbf{M}_{\eta}$
$|\mathcal{V}| \times |\mathcal{V}|$
$\mathsf{F}_1, \mathsf{F}_2, \cdots, \mathsf{F}_D$
$\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_{D\times D}$

Figure 4.2: The Proposed CUOT Framework. F_i denotes an individual feature. $\mathcal{S}_j$ is a feature subset that contains a pair of features. $\mathcal{V}$ denotes the entire value set. δ computes an initial outlierness of feature values. η considers the inter-feature value couplings that highlight the homophily relations between outlying values. $\mathbf{M}_\eta$ is a $|\mathcal{V}| \times |\mathcal{V}|$ matrix whose entries are determined by η . G denotes the value-value graph, ω is an edge weighting function based on δ and η, and ϕ is the value outlierness learning function on the value graph.

in an independent way. (iii) CUOT produces value outlierness that can determine feature selection for subsequent outlier detection or directly identify outliers, whereas the existing frameworks are only aimed for direct outlier detection.

4.2.1 Value Outlierness Initialization

Specifically, the *initial value outlierness* examines the behavior of a value by considering the exceptional interactions of this value with other values from the same feature. Each feature can be treated as a random variable drawn from either a Bernoulli distribution for features with only two values or a categorical distribution for features with more than two values. Accordingly, the semantic of the frequency of values in different features differs significantly; this is even more the case for data with very imbalanced frequency distributions across features. We aim to obtain an initial value outlierness that is comparable in different features.

Let each feature $\mathsf{F} \in \mathcal{F}$ has a domain $dom(\mathsf{F}) = \{v_1, v_2, \cdots\}$, which consists of a finite set of possible feature values. Note that the semantic of the domain in different features is different from each other, since each feature has a different context. We therefore assume that the domains between features are distinct, i.e., $dom(\mathsf{F}_i) \cap dom(\mathsf{F}_j) = \emptyset, \forall i \neq j$. The entire set of feature values $\mathcal{V}$ is the union of all the feature domains: $\mathcal{V} = \cup_{\mathsf{F}\in\mathcal{F}} dom(\mathsf{F})$. $\mathcal{S} \subset \mathcal{F}$ is a feature subspace that is denoted by a Cartesian product set of k_n features with $1 \leq k_n \leq D - 1$, i.e., $\mathcal{S} = dom(\mathsf{F}_{k_1}) \times dom(\mathsf{F}_{k_2}) \times \cdots \times dom(\mathsf{F}_{k_n})$. Then, the initial outlierness based on intra-feature value couplings can then be defined as follows.

Definition 4.1 (Initial Outlierness)**.** *The initial outlierness w.r.t. a value $v \in dom(\mathsf{F})$ is defined as the outlierness aspect w.r.t. a reference value u from the same feature domain, denoted as $\delta(v|u)$.*

4.2.2 Outlierness Influence Between Values

The *outlierness influence vector* evaluates the behavior of a feature value by considering inter-feature value couplings, i.e., exceptional interactions of this value with values from other features. Here we focus on the homophily couplings between outlying values. Such homophily couplings indicate that the outlierness of a value is dependent on not only its own characteristics but also the outlierness of its correlated values. For example, a value has large outlierness if it has strong linkage to many outlying values. This is analogous to the homophily effects in social networks, where people with similar characteristics tend to connect with each other and have mutual influence [31, 42]. The pairwise value outlierness influence is defined below to facilitate the modeling of the homophily couplings in subsequent learning stages.

Definition 4.2 (Outlierness Influence Vector)**.** *The outlierness influence vector w.r.t. a value $v \in dom(\mathsf{F})$ is defined as a coupling vector* $\mathbf{q}$*, where each entry captures the interaction of the value v w.r.t. a specific value u in $\mathcal{V}$.*

$$\mathbf{q}_v = [\eta(v, u_1), \eta(v, u_2), \cdots, \eta(v, u_{|\mathcal{V}|})]^\intercal, \tag{4.1}$$

where $\eta(v, u_i)$ computes the outlierness of the value v w.r.t. the value u_i.

η only focuses on the inter-feature value couplings. If u and v are values of the same feature, we set $\eta(u, v) = 0$ as intra-feature value couplings are modeled by the δ function. For all the values in $\mathcal{V}$, we accordingly obtain a $|\mathcal{V}| \times |\mathcal{V}|$ outlierness influence matrix:

$$\mathbf{M}_\eta = [\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_{|\mathcal{V}|}]^\intercal. \tag{4.2}$$

4.2.3 Value Graph Construction

The *value-value graph* is then defined below to synthesis δ and $\mathbf{M}$. The value graph serves two purposes. (i) Learning from graph representations is a straightforward and effective way to capture homophily couplings, and many off-the-shelf graph mining techniques and theories can then be used to support such learning. (ii) A variety of graph representations, such as directed/undirected graphs and attributed/plain graphs, provides a multitude of options for fusing a collection of outlier factors.

Definition 4.3 (Value-value Graph)**.** *The value-value graph G is described by a three-dimensional tuple $\mathsf{G} =< \mathcal{V}, \mathcal{E}, \omega_{\delta,\eta} >$, where*

- *$\mathcal{V}$ represents the node set and each node $v \in \mathcal{V}$ represents a feature value.*
- *$\mathcal{E}$ denotes a set of edges connecting the nodes, i.e., $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$.*
- *$\omega : \mathcal{V} \times \mathcal{V} \mapsto \mathbb{R}$ is an edge weighting function based on δ and η.*

Since the graph is a value-value graph, the terms ‘value’ and ‘node’ are used interchangeably hereafter.

4.2.4 Value Outlierness Estimation

The *value outlierness estimation* is then performed on the value graph G to learn a final outlierness for each value, such that feature values that are positively related to outlier detection have larger outlier scores than the other values. The value graph-based outlierness estimation models the interactions between δ and η carried by each node of the graph. Specifically, the outlierness of a node influences the outlierness of its neighboring nodes, and the outlierness of the neighboring nodes further influences that of its own neighboring nodes, and so on and forth, forming a *cascade influence* of outlierness estimation.

Definition 4.4 (Value Outlierness Estimation)**.** *The outlier score of a feature value $v \in \mathcal{V}$ is learned by a function $\phi : \mathcal{V} \mapsto \mathbb{R}$ given the value-value graph G.*

There have broad applications of value outlierness, since the value is at the very low-level for evaluating data objects. For example, the value outlierness can characterize the importance of features for subsequent outlier detection. It can also measure the outlierness of data objects and facilitate direct outlier detection.

There are different ways to form the specifications. $\delta(v|u)$ may be specified based on the frequency difference or similarity between the values v and u. $\eta(v, u)$ may be instantiated based on the frequencies of co-occurring patterns between v and u, their conditional probabilities, pointwise mutual information, or other value dependency measures. ω may be specified to project the value-level outlier factors into a directed/undirected and attributed/plain graph. Lastly, ϕ may be specified to model the process of different types of random walks, subgraph discovery, or other graph mining techniques.

4.3 A CUOT Instance: CBRW

This section introduces an instantiation of CUOT, called Coupled Biased Random Walks (CBRW for short). CBRW works as follows. It computes an initial outlierness based on the deviation of the value's frequency from the mode's frequency, and then defines a conditional probability-based outlierness influence vector. CBRW further integrates the two components in a seamless manner via a directed and attributed value graph. It finally estimates the value outlierness according to the stationary probabilities of biased random walks [48] over the value graph. CBRW addresses the cascade homophily couplings of the outlier factors via the biased random walks on the value graph.

4.3.1 Mode-based Initial Outlierness

Per the definition of outliers, the outlierness of a feature value is dependent on its rarity. CBRW employs the frequency of the *mode* of a feature as a rarity comparison benchmark and examines the deviation of the value frequency to evaluate the intra-feature outlierness of a value.

Let $supp(v) = |\{\mathbf{x}_i \in X | x_{ij} = v\}|$, $v \in dom(\mathsf{F}_j)$, be the *support* of the value v. Each feature F is associated with either a *categorical distribution* or a *generalized Bernoulli distribution*, where F takes on one of the possible values $v \in dom(\mathsf{F})$ with a frequency

$freq(v) = \frac{supp(v)}{N}$. The intra-feature outlierness serves as an inital value outlierness and is specified as follows.

Definition 4.5 (Mode). *A* mode *of a categorical distribution of a feature $\mathsf{F} \in \mathcal{F}$, denoted as m, is defined as a value $v_i \in dom(\mathsf{F})$ such that $freq(v_i) = max(freq(v_1), \cdots, freq(v_O))$, where O is the number of possible values in F.*

Definition 4.6 (Mode-based Initial Outlierness). *The mode-based intra-feature outlierness of a feature value $v \in dom(\mathsf{F})$ is defined by the frequency of the mode and the extent that the value's frequency deviates from the mode's frequency*

$$\delta(v) = [\frac{1}{2}\,(base(m) + dev(v))]^1, \tag{4.3}$$

where $base(m) = 1 - freq(m)$ denotes the outlierness of the feature mode m and $dev(v) = \frac{freq(m)-freq(v)}{freq(m)}$ denotes the outlierness of value v compared to the mode.

As the location parameter (or the center) of a categorical distribution, the mode has the same semantic for different features. As shown in the following two key properties of function $\delta(\cdot)$, this specification not only guarantees the efficiency but also helps normalize the initial outlierness.

i. $\forall v \in \mathcal{V}$, $\delta(v) \in (0, 1)^2$.

ii. $\delta(\cdot)$ makes the intra-feature outlierness of values from features with different categorical distributions semantically comparable.

Since $base(m) \in (0, 1)$ and $dev(v) \in [0, 1)$, we have $\delta(v) \in (0, 1)$. For the second property, when two distributions are different in terms of their location parameters, the values drawn from these two distributions are not comparable without proper normalization. In $\delta(\cdot)$, the outlierness of the mode serves as a base, and the more the frequency of a feature value deviates from the mode frequency, the more outlying that value is. This results in a mode-based normalization, making the value outlierness comparable across features.

4.3.2 Conditional Probability-based Outlierness Influence

There is one critical condition for specifying function η in the outlierness influence vector to capture the homophily outlying couplings: η should be capable of contrasting the strong couplings between outlying values from the couplings between other values. Below, we show how the conditional probability-based η satisfies this condition.

Definition 4.7 (Conditional Probability-based Outlierness Influence Vector). *The outlierness influence vector of a value v due to the other values is defined as*

$$\mathbf{q}_v = [\eta(u_1, v), \eta(u_2, v), \cdots, \eta(u_{|\mathcal{V}|}, v)]^\intercal = [\frac{freq(u_1, v)}{freq(v)}, \frac{freq(u_2, v)}{freq(v)}, \cdots, \frac{freq(u_{|\mathcal{V}|}, v)}{freq(v)}]^\intercal, \tag{4.4}$$

[1] Since we only consider the coupling with the mode, $\delta(v, m)$ is hereafter simplified to $\delta(v)$ for brevity.

[2] We have ignored features with $freq(m) = 1$, as those features contain no useful information relevant to outlier detection.

where $\mathit{freq}(u_l, v) = \frac{\mathit{supp}(u_l,v)}{N}$ *with* $\mathit{supp}(u_l, v) = |\{\mathbf{x}_i \in X | x_{ij} = u_l \,\&\, x_{ik} = v\}|$, $u_l \in \mathit{dom}(\mathsf{F}_j)$ *and* $v \in \mathit{dom}(\mathsf{F}_k)$.

CBRW considers the interactions of the value v with all the values. Recall that $\eta(u, v) = 0$ if u and v are from the same features, so the vector $\mathbf{q}$ captures the outlierness influence based on inter-feature value couplings. Its entry, $\eta(u, v)$, is essentially the conditional probability of u given v, and it has three key properties.

i. $\eta(u, v) \in [0, 1]$.

ii. $\eta(u, v) \neq \eta(v, u)$ if $\mathit{freq}(u) \neq \mathit{freq}(v)$.

iii. $\eta(u, v) > 0$ if $\eta(v, u) > 0$; and $\eta(u, v) = 0$ if $\eta(v, u) = 0$.

Since $0 \leq \mathit{freq}(u, v) \leq \mathit{freq}(v) < 1$, we have $\eta(u, v) \in [0, 1]$. The second and third properties follow directly from the property statements.

We now analyze how Equation 4.4 captures the required value interactions that contrast the couplings between outlying values from that between other values. (i) A large η is expected when the inputs u and v are both *outlying values* (i.e., infrequent values contained by outliers). This is because outlying values have low frequency and they are presumed to be concurrent due to the homophily coupling assumption, resulting in larger conditional probabilities. (ii) A small η is expected when both of the inputs are *normal values* (i.e., frequent values contained by normal objects). Although normal values may have high co-occurrence frequency, its individual frequency is much higher, resulting in small η. (iii) A small η is expected when the two inputs are randomly distributed *noisy values* (i.e., infrequent values contained by normal objects). Such noisy values also have low individual frequencies, but they rarely co-occur together if they are randomly distributed. As a result, they have smaller η than outlying values. (iv) A small η is expected when one input value is an outlying value and the other input is a normal/noisy value. Normal or noisy values may also co-occur with outlying values. However, since normal or noisy values are mainly contained by normal objects while outlying values are mainly contained by outliers, the conditional probability of outlying values given normal or noisy values is smaller than that between outlying values.

4.3.3 Directed and Attributed Value Graph

It is challenging to properly integrate δ and $\mathbf{q}$ since they are of different lengths, e.g., $\delta(v)$ is a scalar while $\mathbf{q}_v$ is a $|\mathcal{V}|$-dimensional vector. CBRW tackles this challenge by mapping these two components onto an attributed value-value graph as follows.

Definition 4.8 (Attributed Value-value Graph). *The attributed value-value graph* G *is described by* $\mathsf{G} =< \mathcal{V}, \mathcal{E}, \omega_{\delta,\eta} >$*, where*

- $\mathcal{V}$ *represents the node set and each node* $v \in \mathcal{V}$ *represents a feature value.*
- $\mathcal{E}$ *denotes a set of edges connecting the nodes, i.e.,* $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$.

- $\delta(\cdot) : \mathcal{V} \mapsto (0, 1)$ *is a node property mapping function using the intra-feature outlier factor in Equation (4.3).*
- $\eta(\cdot, \cdot) : \mathcal{V} \times \mathcal{V} \mapsto [0, 1]$ *is an edge weighting function using the outlierness influence vector in Equation (4.4).*

It is easy to see that the graph G is a *directed* and *weighted* graph without self loops according to the properties of its edge weighting function η in Section 4.3.2. The edge weighting function η is different from the conventional methods that are built on similarities between the nodes. It is used because the conditional probabilities are simple and they fully capture the desired homophily relationships between outlying behaviors.

Note that although we do not explicitly specify the edge weighting function ω, but we show in Section 4.4.1 that ω is equivalent to a linear combination of the δ and η functions.

4.3.4 Biased Random Walks for Learning Value Outlierness

CBRW then builds *biased* random walks (BRWs) on the attributed value graph G to learn the value outlierness. Let $\mathbf{A}$ be an adjacency matrix of G, where $\mathbf{A}(u, v)$ denotes the outgoing edge weight from node u to node v. Then we have

$$\mathbf{A}(u, v) = \eta(u, v). \tag{4.5}$$

In building *unbiased* random walks (URWs), we can obtain a *walking (or transition) matrix* $\mathbf{W}$ by

$$\mathbf{W} = \mathbf{A} diag(\mathbf{A})^{-1}, \tag{4.6}$$

where $diag(\mathbf{A})$ denotes the diagonal matrix of $\mathbf{A}$ with its u-th diagonal entry $d(u) = \sum_{v \in V} \mathbf{A}(u, v)$. The entry $\mathbf{W}(u, v) = \frac{\mathbf{A}(u,v)}{d(u)}$ represents the probability of the transition from node u to node v, which satisfies $\sum_{v \in V} \mathbf{W}(u, v) = 1$.

However, URWs omit the δ-based intra-feature outlierness and only consider the η-based inter-feature outlierness influence only. Here, CBRW uses BRWs to introduce the intra-feature outlierness as a bias into the random walk process. This helps capture the intra-feature value couplings and the joint effects they may have with the inter-feature value couplings on the subsequent outlierness estimation. The entry of the corresponding transition matrix is defined as

$$\mathbf{W}^b(u, v) = \frac{\delta(v)\mathbf{A}(u, v)}{\sum_{v \in V} \delta(v)\mathbf{A}(u, v)}. \tag{4.7}$$

$\mathbf{W}^b(u, v)$ can be interpreted as that the transition from node u to node v has a probability proportional to $\delta(v)\mathbf{A}(u, v)$. Therefore, every random move is jointly determined by both the intra-feature outlierness and inter-feature outlierness influence.

CBRW essentially simulates an outlierness propagation process over the value graph to model the homophily couplings between outlying values. The inter-feature influence vector maintains the strength of homophily couplings between outlying values during the outlierness propagation process, while the intra-feature initial outlierness enables outlying values to attract more outlierness. As a result, if u and v are strongly coupled and they have large outlierness, the outlierness propagation from u to v would be large. v has large outlierness if there are many nodes having a similar relationship as u to v. Similarly, u has large outlierness if it is coupled with many outlying values. Such a cascade outlierness of each node can be effectively captured by the probability of the random walker visiting the node.

Let the vector $\boldsymbol{\pi}_t \in \mathbb{R}^{|\mathcal{V}|}$ denotes the *probability distribution* of the biased random walk at time step t, i.e., the probability of a random walker visiting any given node at the t-th step. Then we have

$$\boldsymbol{\pi}^{t+1} = \mathbf{W}^b \boldsymbol{\pi}^t. \tag{4.8}$$

$\boldsymbol{\pi}$ will converge to a stationary probability distribution $\boldsymbol{\pi}^*$ if the graph G is irreducible and aperiodic, i.e., $\boldsymbol{\pi}^* = \mathbf{W}^b \boldsymbol{\pi}^*$ (see Section 4.4 for more detail). This states that the stationary probabilities of the nodes are independent of the initialization of $\boldsymbol{\pi}$, and they are positively correlated to the incoming weights of the nodes. Motivated by this, we define the final value outlierness as follows.

Definition 4.9 (CBRW-based Value Outlierness)**.** *The outlierness of node v is defined by its stationary probability*

$$\phi(v) = \boldsymbol{\pi}^*(v), \tag{4.9}$$

where $\boldsymbol{\pi}^*(v)$ is the entry w.r.t. the value v in the stationary probability vector, $0 < \boldsymbol{\pi}^*(v) < 1$ *and* $\sum_{v \in V} \boldsymbol{\pi}^*(v) = 1$.

The value v has large outlierness iff it demonstrates outlying behaviors within the feature and co-occurs with many other outlying values. This is because $\boldsymbol{\pi}^*(v)$ is proportional to $\mathbf{W}^b(u, v)$, which is determined by $\delta(v)$ and $\eta(u, v)$.

4.3.5 The Algorithm and Its Time Complexity

The steps in CBRW are outlined in Algorithm 4.1. Steps 1-8 obtain the intra- and inter-feature value couplings. The matrix $\mathbf{W}^b$ is then generated based on Equations (4.3), (4.4), and (4.7).

Following [89], Step 12 introduces the damping factor α into Equation (4.8) to guarantee the convergence of the random walks

$$\boldsymbol{\pi}^{t+1} = (1 - \alpha)\frac{1}{|\mathcal{V}|}\mathbf{1} + \alpha \mathbf{W}^b \boldsymbol{\pi}^t. \tag{4.10}$$

In our experiments, we set $\alpha = 0.95$ directly rather than learning the parameter. There are two main reasons to do this. (i) The parameter α has an explicit meaning, so users can easily determine their own setting based on the application contexts. A detailed discussion on this issue is presented in Section 4.4.3. (ii) Our empirical results show that CBRW performs very stably with a wide range of values for α, i.e., $\alpha \in [0.85, 1)$. Therefore,

Algorithm 4.1 Coupled Biased Random Walk

```
Input: $\mathcal{X}$ - data objects, $\alpha$ - damping factor
Output: $\boldsymbol{\pi}^*$ - the stationary probability distribution
 1: for $i = 1$ to $D$ do
 2:   Compute $freq(v)$ for each $v \in dom(\mathsf{F}_i)$
 3:   Find the mode of $\mathsf{F}_i$
 4:   Compute $\delta(v)$
 5:   for $j = i + 1$ to $D$ do
 6:     Compute $freq(u, v)$, $\forall u \in dom(\mathsf{F}_j)$
 7:   end for
 8: end for
 9: Generate the matrix $\mathbf{W}^b$
10: Initialize $\boldsymbol{\pi}^*$ as a uniform distribution
11: repeat
12:   $\boldsymbol{\pi}^* \leftarrow (1 - \alpha)\frac{1}{|\mathcal{V}|}\mathbf{1} + \alpha \mathbf{W}^b \boldsymbol{\pi}^*$
13: until Convergence, i.e., $|\Delta\boldsymbol{\pi}^*| \leq 0.001$ or reach the maximum iteration $I_{max} = 100$
14: return $\boldsymbol{\pi}^*$
```

employing advanced procedures to learn the parameter may not have an obvious benefit in terms of detection performance.

CBRW requires $O(ND^2)$ to obtain the value couplings information in Steps 1-8. The generation of $\mathbf{W}^b$ requires at most $O(|\mathcal{E}|)$ in Step 9. The random walks in Steps 11-13 is linear to the maximum iteration step and the number of edges in the value graph, resulting in $O(|\mathcal{E}|I_{max})$. Therefore, the overall time complexity is $O(ND^2 + |\mathcal{E}|I_{max})$. N is often far larger than $|\mathcal{E}|$ and I_{max} is a constant. The time complexity is thus determined by $O(ND^2)$. Theoretically, CBRW is quadratic w.r.t. the data dimensionality, as two loops are required in Steps 1-8 to obtain the value co-occurrence information. However, the computation within the inner loop (i.e., Step 6) is a simple counting, which, in practice, leads to a nearly linear time complexity w.r.t. the number of features.

4.3.6 Applications of CBRW

Feature Weighting and Selection Using CBRW

In outlier detection, relevant features are the features where the outliers demonstrate outlying behaviors and are distinguishable from normal objects. Thus, the relevance or importance of a feature can be measured by consolidating the outlierness of each value of the feature as follows.

Definition 4.10 (Feature Relevance)**.** *The relevance of a feature* F *is defined as*

$$rel(\mathsf{F}) = 1 - \prod_{v \in dom(\mathsf{F})} [1 - \phi(v)]. \tag{4.11}$$

Since $\phi(v)$ denotes the value outlierness, $rel(\mathsf{F})$ in Equation (4.11) can be interpreted as the outlying likelihood of the feature F. A large $rel(\cdot)$ indicates high relevance of the feature to outlier detection. The top-ranked features are the most relevant features, while the bottom-ranked features are noisy or irrelevant. In addition to being used as a feature filter, these relevance weights can also be embedded in the outlier scoring function of an outlier detector as a feature weighting. One such example is shown in Equation (4.12).

As shown in Steps 1-3 in Algorithm 4.2, our CBRW-based feature selection method (denoted as CBRW$_{\mathrm{fs}}$) computes the weight of each feature using Equation (4.11). The top-ranked features for each data set are selected in Step 4. Outlier detectors can then work on the newly obtained data sets with the selected features.

Algorithm 4.2 Feature Selection

```
Input: $\mathcal{F}$ - feature set, $\phi$ - value outlierness estimation function, $\theta$ - a decision threshold
    (i.e., the number of features to be selected or a relevance threshold)
Output: $Subset_{\mathcal{F}}$ - a subset of features in $\mathcal{F}$
for $i = 1$ to $D$ do
    $rel(\mathsf{F}_i) \leftarrow 1 - \prod_{v \in dom(\mathsf{F}_i)}[1 - \phi(v)]$
end for
$Subset_{\mathcal{F}} \leftarrow filter(\mathcal{F})$: Select the features that meet the threshold $\theta$
return $Subset_{\mathcal{F}}$
```

Outlier Detection Using CBRW

As shown below, the value outlierness can also measure the outlierness of data objects by consolidating the outlierness of values contained by the objects.

Definition 4.11 (Object Outlierness). *The outlierness of an object $\mathbf{x}_i \in \mathcal{X}$ is defined as*

$$score(\mathbf{x}_i) = 1 - \prod_{j=1}^{D}[1 - \phi(x_{ij})]^{\omega(\mathsf{F}_j)}, \tag{4.12}$$

where $\omega(\mathsf{F}_j) = \frac{rel(\mathsf{F}_j)}{\sum_{j=1}^{D} rel(\mathsf{F}_j)}$ is a feature weighting component.

$score(\cdot)$ is used to evaluate the outlying likelihood of an object, with a relevance weighting factor to highlight the importance of highly relevant features. As shown in Steps 1-3 of Algorithm 4.3, our CBRW-based outlier detection method (denoted as CBRW$_{\mathrm{od}}$) employs Equation (4.12) to compute the outlying likelihood of each data object. Objects are then sorted by their outlierness. Outliers are data objects having large outlier scores.

Algorithm 4.3 Outlier Detection

```
Input: $\mathcal{X}$ - data objects, $\phi$ - value outlierness estimation function, $\boldsymbol{\omega}$ - feature weights
Output: $\mathbf{r}$ - an outlier ranking of objects in $\mathcal{X}$
for $i = 1$ to $N$ do
    $score(\mathbf{x}_i) \leftarrow 1 - \prod_{j=1}^{D}[1 - \phi(x_{ij})]^{\omega(\mathsf{F}_j)}$
end for
$\mathbf{r} \leftarrow$ Sort the objects in $\mathcal{X}$ in descending order
return $\mathbf{r}$
```

4.4 Theoretical Analysis

This section first presents the convergence analysis of CBRW, followed by a discussion on the homophily coupling modeling and how to tune the parameter α.

4.4.1 Convergence Analysis

Unbiased random walks (URWs) are easier to be analyzed than biased random walks (BRWs). Therefore, the equivalent URWs forms for CBRW are provided first to ease understanding.

Lemma 4.0.1 (Equivalence between BRWs and URWs)**.** *BRWs based on the adjacency matrix* $\mathbf{A}$ *and the bias* δ *is equivalent to URWs on a graph* G^b *with an adjacency matrix* $\mathbf{B}$*, in which*

$$\mathbf{B}(u,v) = \delta(u)\mathbf{A}(u,v)\delta(v),\ \forall u, v \in \mathcal{V}. \tag{4.13}$$

Proof. This lemma holds iff the transition matrix $\mathbf{T}$ of G^b satisfies: $\mathbf{T} \equiv \mathbf{W}^b$. Since $\mathbf{B}(u,v) = \delta(u)\mathbf{A}(u,v)\delta(v)$, we have

$$\begin{aligned}\mathbf{T}(u,v) &= \frac{\mathbf{B}(u,v)}{\sum_{v\in\mathcal{V}}\mathbf{B}(u,v)} = \frac{\delta(u)\mathbf{A}(u,v)\delta(v)}{\sum_{v\in V}\delta(u)\mathbf{A}(u,v)\delta(v)}\\ &= \frac{\mathbf{A}(u,v)\delta(v)}{\sum_{v\in V}\mathbf{A}(u,v)\delta(v)} = \mathbf{W}^b(u,v).\end{aligned}$$

□

In addition to the stated equivalence, Lemma 4.0.1 also shows that CBRW implicitly defines the edge weighting function ω as $\omega(u,v) = \delta(u)\eta(u,v)\delta(v)$.

Irreducibility and aperiodicity, as defined below, are the two necessities for the convergence of random walks.

Definition 4.12 (Irreducibility)**.** *The graph* G *is irreducible if* $\forall u, v, \exists t$ *s.t.* $\boldsymbol{\pi}_{u\to v}^{0\to t} > 0$*, where* $\boldsymbol{\pi}_{u\to v}^{0\to t} > 0$ *is the probability of visiting* v *in* t *steps from the initial state at* u*.*

Definition 4.13 (Aperiodicity)**.** *The graph* G *is aperiodic if* $\forall u, v$*,* $gcd\{t : \boldsymbol{\pi}_{u\to v}^{0\to t} > 0\} = 1$*, where gcd denotes the greatest common divider.*

Theorem 4.1 (Convergence of CBRW)**.** *If* G *is irreducible and aperiodic, CBRW will converge, i.e.,* $\boldsymbol{\pi}$ *converges to a unique stationary probability vector* $\boldsymbol{\pi}^*$ *such that* $\boldsymbol{\pi}^* = \mathbf{W}^b\boldsymbol{\pi}^*$*.*

Proof. If the graph G is irreducible and aperiodic, then based on the Perron–Frobenius Theorem [84], the URWs on G based on the adjacency matrix $\mathbf{A}$ will converge to a unique probability vector.

Since δ is always positive, the inclusion of δ into $\mathbf{A}$ does not change the graph's irreducibility and aperiodicity. In other words, $\mathbf{B}$ and $\mathbf{A}$ have the same irreducibility and aperiodicity. Therefore, if G is irreducible and aperiodic, so is G^b. We therefore have $\boldsymbol{\pi}^* = \mathbf{W}^b\boldsymbol{\pi}^*$.

□

However, different data sets may contain very different value couplings; therefore, the assumption that the graph G is irreducible and aperiodic may not always hold in practice. A common method to remedy this problem is to introduce a damping factor to transform the natural random walks into *teleporting random walks* [89].

Corollary 4.1.1 (Teleporting Random Walks Guaranteeing Convergence)**.** *By setting* $\mathbf{W}^b = (1-\alpha)\frac{1}{|\mathcal{V}|}\mathbf{1} + \alpha\mathbf{W}^b$*, where* $\alpha \in [0, 1)$*,* $\mathbf{W}^b\boldsymbol{\pi}$ *will always converge to a unique probability vector* $\boldsymbol{\pi}^*$*, i.e.,* $\boldsymbol{\pi}^* = \mathbf{W}^b\boldsymbol{\pi}^*$*.*

Proof. It is obvious that $\mathbf{W}^b$ becomes a real positive square matrix by the addition of $(1-\alpha)\frac{1}{|\mathcal{V}|}\mathbf{1}$. This guarantees that $\mathbf{W}^b$ is irreducible and aperiodic. We therefore will always have $\boldsymbol{\pi}^* = \mathbf{W}^b\boldsymbol{\pi}^*$. □

4.4.2 Modeling Homophily Outlying Behaviors

Random walks are one of the most popular and efficient methods for modeling homophily couplings [69]. Basically, as SDRW conducts random walks on the value graph, in which large edge weights indicate large outlierness of the corresponding nodes of the edge, the final outlierness of a node is determined by the outlierness associated with its direct neighbor nodes, and the outlierness of these neighbor nodes is governed by the neighbors of these nodes, and so on. This process models an iterative effect of outlierness propagation.

Definition 4.14 (Direct Neighbor)**.** *The direct neighbors of a node u is defined as*

$$\mathcal{N}(u) = \{v|\mathit{dist}(v, u) = 1,\ \forall v \in \mathsf{G}\}, \tag{4.14}$$

where $\mathit{dist}(v, u)$ *returns the shortest path from node* v *to node* u*.*

Proposition 4.1 (Homophily Coupling Modeling)**.** *Let* $\mathcal{N}(v)$ *be the direct neighbors of a node* $v \in \mathcal{V}$*. Then the outlierness of value* v *is linearly proportional to the outlierness of its direct neighbors and its coupling strength with these neighbors, i.e., in CBRW, we have*

$$\phi(v) \propto \sum_{u\in\mathcal{N}(v)} \phi(u)\delta(u)\eta(u, v)\delta(v). \tag{4.15}$$

Proof. According to Lemma 4.0.1, the biased random walks in CBRW can be represented by $\boldsymbol{\pi}^{t+1}(v) = \sum_{u\in\mathcal{N}(v)} \boldsymbol{\pi}^t(u)\frac{\delta(u)\mathbf{A}(u,v)\delta(v)}{\sum_{v\in V}\delta(u)\mathbf{A}(u,v)\delta(v)}$. Since the denominator is a constant for all the neighbors of node u under the same context, we can omit it and obtain $\boldsymbol{\pi}(v) \propto \sum_{u\in\mathcal{N}(v)} \boldsymbol{\pi}(u)\delta(u)\eta(u, v)\delta(v)$. Since $\phi(v) = \boldsymbol{\pi}(v)$, we achieve Equation (4.15). □

Proposition 4.1 states that the outlierness of a value is mainly determined by its coupling strength and the outlierness of its direct neighbors, in addition to its intra-feature outlierness. That is, a value has large outlierness if it is centered around outlying values. This captures exactly the homophily phenomenon of outlying behaviors - the tendency of a set of outlying behaviors to join together. Compared to IID methods that treat the outlierness scoring of outlying behaviors independently, i.e., $\phi(v)$ is independent of $\phi(u)$, our models can achieve a more effective outlierness estimation on data with homophily outlying behaviors.

4.4.3 Stability w.r.t. Parameter α

It is obvious from Equation (4.10) that the closer the α is to one, the more we respect the underlying structure of the value graph. On the other hand, if α is close to zero, changes in α will have a limited effect on the stationary probabilities. This idea is discussed in detail through the following theorem:

Theorem 4.2 (Effect of α [72]). *Let $\boldsymbol{\pi}^{*,\alpha}$ be the stationary probability vector obtained using damping factor $\alpha \in [0, 1)$. Then*

$$\left|\frac{d\boldsymbol{\pi}^{*,\alpha}(u)}{d\alpha}\right| \leq \frac{1}{1-\alpha}, \quad \forall u \in \mathcal{V}, \tag{4.16}$$

and

$$\left|\frac{d\boldsymbol{\pi}^{*,\alpha}}{d\alpha}\right| \leq \frac{2}{1-\alpha}. \tag{4.17}$$

Theorem 4.2 states that the gaps in the entries in the stationary probability vector (i.e., the value outlierness) are determined by $\frac{1}{1-\alpha}$. Using large α indicates a preference for a large difference in the outlierness of values, particularly the difference between the top-ranked values and and the bottom-ranked values. This helps CBRW to well distinguish outlying values from normal values.

Another interpretation is that a large α results in a large upper bound w.r.t. $\boldsymbol{\pi}^*$, which may lead to performance instability. However, this is based on the underlying assumption that the random walks are conducted on very sparse dynamic graphs, e.g., a web graph. Moreover, random walks are often proposed for applications that emphasize the performance on learning regularities. CBRW is different from these scenarios in the sense that: (i) we carefully set the edge weights to highlight the irregularities (i.e., the irregular behaviors of outlying values), and (ii) the graph structure is critical in capturing the outlying behavior of values. Hence, a large α is more appropriate in our design. Note that larger α may lead to slower convergence [72]. Therefore, we suggest using $\alpha = 0.95$ for CBRW to achieve a trade-off between effectiveness and efficiency.

4.5 Experiments and Evaluation

This section first gives the parameter settings of CBRW and its contenders, followed by the description of data sets, and then presents the performance of outlier detection and feature selection.

4.5.1 Outlier Detectors and Their Parameter Settings

The proposed method CBRW$_{\text{od}}$ with the default setting $\alpha = 0.95$ is evaluated against representative categorical data-oriented outlier detectors: FPOF [57], CompreX [7], MarP [34], and one numerical data-oriented method iForest [77]. They are used as our contenders because they have shown better or very comparable performance compared to other well-known methods and thus stand for state-of-the-art methods.

We also derive two variants of CBRW, named **CBRWia** and **CBRWie**, to have a comprehensive understanding of the performance of CBRW. The variants of CBRW can be easily obtained by weakening/neglecting either inter-feature value couplings or intra-feature value couplings, respectively. Specifically, CBRWia is obtained by using the CBRW with $\mathbf{A}(u, v) = 1$ iff $\mathbf{A}(u, v) > 0$, $\forall u, v \in \mathcal{V}$ in Equation (4.7), while CBRWie is obtained by the CBRW with $\delta(v) = 1$ iff $\delta(v) > 0$, $\forall v \in \mathcal{V}$.

CBRW and its two variants, FPOF, MarP, MarP$^+$ and iForest are implemented in JAVA in WEKA [52]. CompreX is obtained from the authors of [7] in MATLAB. All the experiments are performed at a node in a 3.4GHz Phoenix Cluster with 32GB memory.

4.5.2 Data Sets

We use the following two principles to filter out high-quality data sets: (i) the number of objects must be at least 1,000 to avoid potential bias caused by small data sets, and (ii) the data sets contain semantically meaningful outliers or have an extremely imbalanced class distribution to facilitate direct conversion from classification data into outlier detection data. Fifteen publicly available real-world data sets are finally adopted, which cover diverse domains, e.g., intrusion detection, text mining, image recognition, cheminfomatics and ecology, as shown in Table 4.1. *Probe* and *U2R* are derived from KDDCUP99 data sets using probe and user-to-root attacks as outliers against the normal class, respectively. Other data sets are transformed from extremely class imbalanced data using the rare class conversion method presented in Section 2.3.3.

The quantization results of the four data indicators, κ_{vcc}, κ_{het}, κ_{ins} and κ_{fnl} (see Section 2.3.3 for detail.), on 15 the data sets are reported in Table 4.1. We report the value of each data indicator per data set. Since the semantics of indicators differ significantly from each other, we rank the data sets and compute an average rank for the data to obtain an overall complexity quantization. The top-ranked data indicates high data complexity. Here we compute the unweighted average rank. If the importance of each indicator is known, a weighted average rank would be more preferable.

The top-10 ranked data sets are *BM*, *Census*, *AID362*, *w7a*, *CMC*, *APAS*, *CelebA*, *Chess* , *AD*and *SF*. They are often the top-ranked data sets in terms of individual indicator-based rankings. The complexity of these data sets are, to some extent, verified by the AUC results in Table 4.2, where all outlier detectors obtain substantially lower AUC results on these 10 data sets than on the bottom-ranked five data sets.

4.5.3 Outlier Detection Performance

We first present a summary of outlier detection performance on all data sets, and then analyze the detection performance on complex and simple data sets separately in the next two sections.

Table 4.1: A Summary of 15 Data Sets and Their Complexity Evaluation Results. The following acronyms are used for brevity: Bank Marketing = BM, aPascal = APAS, Internet Advertisements = AD, Contraceptive Method Choice = CMC, Solar Flare = SF, Reuters10 = R10, CoverType = CT, and Linkage = LINK. The data sets are ordered by the average rank in the last column.

			κ_{vcc}		κ_{het}		κ_{ins}		κ_{fnl}		
Data	N	Outliers	Value	Rank	Value	Rank	Value	Rank	Value	Rank	Avg. Rank
BM	41,188	yes	21.0%	6	2.028	2	0.373	3	90.0%	1	3.0
Census	299,285	50K+	41.9%	2	1.648	3	0.238	7	57.6%	4	4.0
AID362	4,279	active	32.4%	5	1.140	11	0.396	2	86.0%	2	5.0
w7a	49,749	yes	37.2%	3	1.059	12	0.407	1	48.0%	6	5.5
CMC	1,473	#child>10	3.8%	10	1.579	4	0.344	4	37.5%	7	6.3
APAS	12,695	train	33.0%	4	1.192	10	0.128	11	81.3%	3	7.0
CelebA	202,599	bald	12.1%	8	1.265	9	0.204	8	48.7%	5	7.5
Chess	28,056	zero	0.0%	14	2.242	1	0.264	6	33.3%	9	7.5
AD	3,279	ad.	46.4%	1	1.011	14	0.302	5	4.5%	12	8.0
SF	1,066	F	12.4%	7	1.564	5	0.178	9	9.1%	11	8.0
Probe	64,759	attack	1.3%	12	1.324	7	0.057	12	0.0%	13	11.0
U2R	60,821	attack	1.5%	11	1.285	8	0.015	15	16.7%	10	11.0
LINK	5,749,132	match	0.6%	13	1.392	6	0.021	14	0.0%	13	11.5
R10	12,897	corn	6.1%	9	1.010	15	0.132	10	0.0%	13	11.8
CT	581,012	cottonwood	0.0%	14	1.102	13	0.029	13	34.1%	8	12.0

Overall Performance

The AUC results of CBRW$_{\text{od}}$, CBRWie$_{\text{od}}$, CBRWia$_{\text{od}}$, MarP^{+}, MarP, FPOF, CompreX and iForest on the 15 data sets are presented in Table 4.2. The p-value results at the bottom are based on paired two-tailed t-test using the null hypothesis that the AUC results of our method and another detector come from distributions with equal means.

CBRW$_{\text{od}}$ achieves the best detection performance on four data sets, with six close to the best (having the difference in AUC no more than 0.03). The significance test results show that CBRW$_{\text{od}}$ significantly outperforms its two contenders FPOF and CompreX at the 95% confidence level and the other three contenders MarP^{+}, MarP and iForest at the 99% confidence level.

CBRWie$_{\text{od}}$ obtains the best AUC results on five data sets, with eight close to the best, and performs significantly better than iForest, while CBRWia$_{\text{od}}$ obtains the best AUC result on three data sets, with five close to the best.

It is clear that there exists a large gap of the AUC results between the top 10 data sets and the last five data sets. It is difficult for detectors to obtain a very good performance on the top 10 data sets, contrasting to the results on the last five data sets. We therefore break these data sets into two categories - *complex and simple data*, and discuss them in details in the next two subsections.

Handling Complex Data

We break down the analysis into four parts w.r.t. the four indicators in Table 4.1.

Results on Data Sets with Highly Complex Value Couplings. The top-10 data sets with the largest proportions of negative value couplings are *AD*, *Census*, *w7a*, *APAS*, *AID362*, *BM*, *SF*, *CelebA*, *R10* and *CMC* according to κ_{vcc} in Table 4.1. All these data sets fall in the category of complex data except *R10*. Although *R10* has over 6.1% negative couplings

Table 4.2: AUC Performance of CBRW$_{\mathrm{od}}$, its Two Variants and Five Contenders on the 15 Data Sets. '○' indicates out-of-memory exceptions, while '●' indicates that we cannot obtain the results within two months. The middle horizontal line roughly separates complex data from simple data based on average rank in Table 4.1. The best performance for each data set is boldfaced. The p-value of the null hypothesis rejected at the 1% or 5% confidence level is underlined.

Data	**CBRW$_{\mathrm{od}}$**	**CBRWie$_{\mathrm{od}}$**	**CBRWia$_{\mathrm{od}}$**	**MarP^{+}**	**MarP**	**FPOF**	**CompreX**	**iForest**
BM	0.6287	**0.6566**	0.5999	0.5778	0.5584	0.5466	0.6267	0.5762
Census	0.6678	0.6579	**0.6832**	0.6033	0.5899	0.6148	0.6352	0.5378
AID362	**0.6640**	0.6324	0.6034	0.6152	0.6270	○	0.6480	0.6485
w7a	0.6484	**0.7338**	0.4453	0.4565	0.4723	○	0.5683	0.4053
CMC	**0.6339**	0.6323	0.6179	0.5623	0.5417	0.5614	0.5669	0.5746
APAS	0.8190	0.8624	**0.8739**	0.6208	0.6193	○	0.6554	0.4792
CelebA	0.8462	**0.9108**	0.7135	0.7352	0.7358	0.7380	0.7572	0.6797
Chess	**0.7897**	0.4058	0.7766	0.6854	0.6447	0.6160	0.6387	0.6124
AD	0.7348	**0.8270**	0.7250	0.7033	0.7033	○	●	0.7084
SF	0.8812	0.8833	**0.8867**	0.8469	0.8446	0.8556	0.8526	0.7865
Probe	0.9906	**0.9907**	0.9434	0.9795	0.9800	0.9867	0.9790	0.9762
U2R	0.9651	0.9640	0.8817	0.8848	0.8848	0.9156	**0.9893**	0.9781
LINK	0.9976	0.9976	0.9976	0.9977	0.9977	**0.9978**	0.9973	0.9917
R10	**0.9905**	0.9903	0.9823	0.9866	0.9866	○	0.9866	0.9796
CT	0.9703	0.9703	0.9388	0.9770	**0.9773**	0.9772	0.9772	0.9364
Avg.(Top-10)	0.7314	0.7202	0.6925	0.6407	0.6337	0.6554	0.6610	0.6009
Avg.(All)	0.8152	0.8077	0.7779	0.7488	0.7442	0.7810	0.7770	0.7247
	CBRW$_{\mathrm{od}}$ vs.	0.7959	0.0392	0.0012	0.0008	0.0115	0.0147	0.0040
p-value		CBRWie$_{\mathrm{od}}$ vs.	0.4225	0.0969	0.0592	0.4316	0.3167	0.0446
			CBRWia$_{\mathrm{od}}$ vs.	0.1460	0.1223	0.2886	0.8490	0.0979

and high dimensionality, it has very simple data distributions, good outlier separability and contains no noisy features, and as a result, even simple outlier detectors like MarP can achieve very good performance.

CBRW$_{\mathrm{od}}$ achieves an average AUC improvement over MarP^{+} (12%), MarP (12%), FPOF (13%), CompreX (7%) and iForest (17%) on these 10 data sets, and CBRWie$_{\mathrm{od}}$ achieves more than 16%, 16%, 17%, 11% and 22% improvements, while CBRWia$_{\mathrm{od}}$ achieves about 6%, 7%, 8%, 2% and 12% improvements.

Most of these data sets contains over 10% negative value couplings. This can result in many misleading patterns and consequently substantially degrade the performance of traditional outlier detection methods (i.e., MarP^{+}, MarP, FPOF, CompreX and iForest). One key difference between positive value couplings and negative value couplings is that positive value couplings generally are much more stronger than the negative ones due to the rarity and homophily phenomenon of the positive value couplings. CBRW$_{\mathrm{od}}$ and CBRWie$_{\mathrm{od}}$ utilize the cascading outlierness propagation between feature values (i.e., the inter-feature outlier factor) to capture these properties, resulting in good robustness to those negative couplings. Although CBRWia$_{\mathrm{od}}$ focuses on the intra-feature outlier factor by weakening the effects of inter-feature outlier factors through setting $\eta(u, v) = 1$, $\forall u, v \in \mathcal{V}$, CBRWia$_{\mathrm{od}}$ can often perform well when positive value couplings dominate over negative value couplings.

Note that κ_{vcc} may underestimate the percentage of positive/negative value couplings, as it does not consider the length of the value couplings. We conjecture that the number of possible negative value couplings may outnumber that of positive couplings in some data sets like *w7a*, and as a result, CBRWia$_{\mathrm{od}}$ performs substantially worse than CBRW$_{\mathrm{od}}$ and

CBRWie$_{\text{od}}$ on those data.

Results on Data Sets with Strong Heterogeneity. The top-10 data sets with the strongest heterogeneity includes *Chess*, *BM*, *Census*, *CMC*, *SF*, *LINK*, *Probe*, *U2R*, *CelebA* and *APAS* according to κ_{het} in Table 4.1. Seven out of ten data sets are categorized into complex data. *LINK*, *Probe* and *U2R* are actually simple data as they have high outlier separability and simple value couplings.

Compared to MarP^{+}, MarP, FPOF, CompreX and iForest on these 10 data sets, on average, CBRW$_{\text{od}}$ obtains over 9%, 11%, 8%, 6% and 14% improvements, and CBRWie$_{\text{od}}$ achieves more than 6%, 7%, 4%, 3% and 10% improvements, while CBRWia$_{\text{od}}$ achieves about 6%, 7%, 5%, 3% and 10% improvements, respectively.

Data sets with large κ_{het} indicate diversified frequency distributions taken across their features, resulting in different semantics of the same frequencies in the features. However, all competitors of CBRW$_{\text{od}}$ ignore this characteristic and treat the same frequencies of values/patterns from different features/subspaces equally, leading to inaccurate outlier scoring of objects.

CBRW$_{\text{od}}$ and CBRWia$_{\text{od}}$ address this issue by modeling the intra-feature outlier factor and thus performs substantially better than their competitors. Although CBRWie$_{\text{od}}$ also neglects the heterogeneity, its advantage in handling complex value couplings complement its overall performance. Nevertheless, CBRWie$_{\text{od}}$ may perform poorly in data sets with very strong heterogeneity, such as *Chess*, *BM* and *CMC*.

Results on Data Sets with Low Outlier Separability. According to κ_{ins} in Table 4.1, the top-10 data sets with the lowest outlier separability are *w7a*, *AID362*, *BM*, *CMC*, *AD*, *Chess*, *Census*, *CelebA*, *SF* and *R10*. All these data sets are complex data except *R10*.

On average, CBRW$_{\text{od}}$ obtains an improvement over MarP^{+} (10%), MarP (11%), FPOF (14%), CompreX (7%) and iForest (14%). CBRWie$_{\text{od}}$ achieves more than 8%, 9%, 11%, 5% and 12% improvements, while CBRWia$_{\text{od}}$ achieves about 4%, 5%, 7%, 1% and 1% improvements.

It is interesting to note that the top-ranked data sets in terms of κ_{ins} are also top-ranked in terms of either κ_{vcc} or κ_{het}. In other words, the low outlier separability in those data sets are in part due to their underlying non-IID characteristics. CBRW$_{\text{od}}$ and its two variants model the intra- and/or inter-feature outlier factors to address the non-IID issue, and thus they perform better than their contenders. CBRW$_{\text{od}}$ addresses both heterogeneity and coupling issues while CBRWia$_{\text{od}}$ and CBRWie$_{\text{od}}$ handle one of these two issues only, so CBRW$_{\text{od}}$ obtains averagely better performance than its variants.

Results on Data Sets with High Feature Noise Level. The top-10 data sets with the highest level of feature noise are *BM*, *AID362*, *APAS*, *Census*, *CelebA*, *w7a*, *CMC*, *CT*, *Chess* and *U2R*. All of them are complex data except *U2R*, which is a 6-dimensional data with very high outlier separability and simple value couplings.

On average, CBRW$_{\text{od}}$ obtains an improvement over MarP^{+} (13%), MarP (14%), FPOF (7%), CompreX (8%) and iForest (18%). CBRWie$_{\text{od}}$ obtains more than 10%, 11%, 4%, 5% and 15% improvements. CBRWia$_{\text{od}}$ performs comparably well to FPOF and CompreX, and obtains over 6%, 7% and 10% than MarP$^+$, MarP and iForest.

CBRW$_{\text{od}}$ and its two variants employ the value outlierness propagation mechanism to distinguish outlying values from noisy values, so they often perform better than their competitors, which are misled by noisy features in pattern searching and have a high false positive error. CBRWie$_{\text{od}}$ performs substantially better than CBRW$_{\text{od}}$ and CBRWia$_{\text{od}}$ on *CelebA*. This may be because some noisy values in *CelebA* are more infrequent than outlying values, and as a result, the noisy values have higher intra-feature outlier scores than the outlying values. Consequently, CBRW$_{\text{od}}$ and CBRWia$_{\text{od}}$ fail to differentiate between outlying and noisy values, while CBRWie$_{\text{od}}$ assigns all values with the same intra-feature outlier and thus becomes insensitive to this problem.

Handling Simple Data

All seven detectors perform very well on the five simple data sets in Table 4.2. This is particularly true for the data sets *R10*, *Probe* and *LINK*, on which all the detectors, including the most simple detector MarP, obtain the AUC of (or nearly) one. Although some of these data sets (e.g., *R10*) are ranked slightly higher than some complex data sets w.r.t. one or two individual data indicators, they rank at the bottom in most cases, resulting in an overall low data complexity.

It is clear from the above results that CBRW$_{\text{od}}$, as an integration of CBRWie$_{\text{od}}$ and CBRWia$_{\text{od}}$, generally performs much better than both CBRWie$_{\text{od}}$ and CBRWia$_{\text{od}}$. This verifies the need of integrating of both intra- and inter-feature value couplings to handle complex data. We therefore focus on the analysis of CBRW hereafter.

4.5.4 Performance of Outlying Feature Selection

We evaluate the effectiveness of the feature selection method CBRW$_{\text{fs}}$ by examining whether it can reduce the complexity of data and how it affects the detection performance of subsequent outlier detection in next two subsections.

Similar to many existing feature selection methods, CBRW$_{\text{fs}}$ provides a feature relevance ranking only. Users are required to determine a relevance threshold or the number of selected features to filter out irrelevant features. Compared to methods that can automatically return a relevant feature subset, although users may be burdened with this parameter setting, this type of methods provides more flexibility for users to determine the final selected feature subset.

Our experiments show CBRW$_{\text{fs}}$ obtains stable performance on all 15 data sets in a wide range of relevance threshold options. We present the results on all our data sets using a consistent relevance threshold to demonstrate the general applicability of CBRW$_{\text{fs}}$ in practice. We observe that an outlier often demonstrates outlying behaviors in a few features only. Since the percentage of outliers is very small, it is reasonable to assume that only a small proportion of features are relevant to outlier detection. We therefore consider to use a small percentage as the relevance threshold. We found that different outlier detectors obtain substantially better performance on some data sets with only 10%

features (e.g., *CelebA*, *CMC*), and on most data sets with 30% features, compared to their performance on original data. They perform very stably and obtain much better AUC performance on all 15 data sets using 50% features. So we use 50% as the relevance threshold.

We compare CBRW_{fs} with its closely related feature-weighting competitor (denoted as ENFW) introduced in [121] and two baselines, including the performance on the full feature set (denoted as FULL) and a random feature selector (denoted as RADM) that randomly select 50% features.

Data Complexity Reduction

Table 4.3 shows the results of data complexity evaluation for each data indicator on data sets with selected feature subsets as well as full feature sets.

CBRW_{fs} considerably reduce the data complexity in most data indicators for all the data sets, achieving averagely more than 25%, 8% and 9% simplification in the indicators κ_{vcc}, κ_{het} and κ_{fnl}. ENFW obtains markedly larger simplification than CBRW_{fs} in κ_{fnl} and κ_{het}, while it substantially increases the outlier inseparability according to κ_{ins}. This is because ENFW evaluate the relevance of features without considering the interactions between features, and thus noisy features and highly relevant features are filtered out together. In other words, ENFW reduces the κ_{fnl}-based data complexity at the expense of increasing κ_{ins}-based data complexity. Also, ENFW is an entropy-based feature weighting method, which retains features with similar frequency distributions, and thus it obtains much larger simplification than CBRW_{fs} in κ_{het}. However, since it builds upon the feature independence assumption, it can remove features that are very relevant when combining with other features. In contrast, CBRW_{fs} considers the low-level intra- and inter-feature value couplings, which is sensitive to negative value couplings, value frequency distributions and noisy features, resulting in an outlier separability secured reduction of data complexity.

Performance of Different Subsequent Outlier Detectors

The effectiveness of the feature selection results determined by CBRW_{fs} is further verified by the detection performance of different subsequent outlier detectors. We use MarP and iForest, two simple and very different outlier detectors, to examine the performance of incorporating CBRW_{fs} into existing outlier detectors.

The AUC performance of MarP and iForest working on the data ses with feature subsets selected by CBRW_{fs} is shown in Table 4.4. Both of the CBRW-empowered MarP and iForest obtains substantial improvements (e.g., more than 12%, 17% and 7%) compared to that of ENFW, RADM and FULL, regardless of the difference working mechanisms of these two detectors. In particular, the CBRW-empowered MarP and iForest significantly outperform their counterparts empowered by ENFW and RADM at the 99% confidence level; and although they use 50% less features, they significantly outperforms MarP and iForest working on data with full feature sets at the 95% confidence level. The superiority of CBRW_{fs} is understandable, since it considerably reduces the levels of negative

Table 4.3: Data Complexity Evaluation Results of Data Sets with Feature Subsets Selected by CBRW_{fs} (denoted as CBRW) and ENFW, Using the Results on Original Data Sets as a Baseline. The last row 'Simp.' indicates the average simplification percentage compared to the baseline on the original data.

	κ_{vcc}			κ_{het}			κ_{ins}			κ_{fnl}		
Data	CBRW	ENFW	FULL	CBRW	ENFW	FULL	CBRW	ENFW	FULL	CBRW	ENFW	FULL
BM	**0.1930**	0.5010	0.2098	1.6979	**1.3030**	2.0278	**0.3731**	0.5244	**0.3731**	**0.8000**	1.0000	0.9000
Census	**0.4044**	0.5738	0.4194	1.8281	**1.1509**	1.6477	**0.2379**	0.3359	**0.2379**	0.6471	0.7647	**0.5758**
AID362	**0.2789**	0.3440	0.3245	1.0362	**1.0068**	1.1400	**0.3959**	0.4793	**0.3959**	0.9298	0.9649	**0.8596**
w7a	0.2016	**0.0983**	0.3719	1.0124	**1.0044**	1.0594	**0.4073**	0.4422	**0.4073**	0.2267	**0.0267**	0.4800
CMC	0.0376	**0.0000**	0.0376	1.2963	**1.2664**	1.5794	**0.3444**	0.3653	**0.3444**	**0.0000**	0.5000	0.3750
APAS	**0.2226**	0.3301	0.3301	1.0593	**1.0175**	1.1922	**0.1280**	0.2805	**0.1280**	**0.6562**	0.8750	0.8125
CelebA	**0.0810**	0.1213	0.1213	1.1572	**1.0544**	1.2647	**0.2039**	0.3233	**0.2039**	**0.2000**	0.4000	0.4872
Chess	0.0000	0.0000	0.0000	**1.2220**	2.0532	2.2416	0.2642	0.2642	0.2642	0.6667	**0.0000**	0.3333
AD	**0.2620**	0.3740	0.4639	1.0037	**1.0007**	1.0083	0.3378	0.4730	**0.3018**	0.0077	**0.0000**	0.0450
SF	0.1465	0.1522	**0.1242**	1.7204	**1.0804**	1.5639	**0.1779**	0.3027	**0.1779**	**0.0000**	0.1667	0.0909
Probe	0.0108	**0.0000**	0.0134	1.3562	**1.0361**	1.3243	**0.0573**	0.0675	**0.0573**	0.0000	0.0000	0.0000
U2R	0.0124	**0.0000**	0.0152	1.3457	**1.0032**	1.2851	**0.0154**	0.1455	**0.0154**	0.3333	**0.0000**	0.1667
LINK	**0.0000**	0.0060	0.0060	1.1880	**1.1831**	1.3916	0.0209	0.0209	0.0209	0.0000	0.0000	0.0000
R10	0.0123	**0.0028**	0.0610	1.0022	**1.0005**	1.0099	**0.1323**	0.4414	**0.1323**	0.0000	0.0000	0.0000
CT	0.0000	0.0000	0.0000	1.1715	**1.0033**	1.1018	**0.0291**	0.3177	**0.0291**	0.4545	**0.0000**	0.3409
Avg.	0.1242	0.1669	0.1665	1.2731	1.1443	1.3892	0.2084	0.3189	0.2060	0.3281	0.3132	0.3645
Simp. (%)	25.42%	-0.22%		8.35%	17.63%		-1.16%	-54.85%		9.97%	14.06%	

value couplings, heterogeneity and feature noise while at the same time retains the outlier separability (i.e., retain the most relevant features).

Table 4.4: AUC Performance of MarP and iForest Using Feature Selection Methods CBRW_{fs}, ENFW, RADM and Their Baseline FULL Using the Full Feature Set.

	MarP				**iForest**			
Data	CBRW	ENFW	RADM	FULL	CBRW	ENFW	RADM	FULL
BM	**0.5926**	0.4886	0.5181	0.5584	**0.5836**	0.5297	0.5544	0.5762
Census	**0.6258**	0.4525	0.5490	0.5899	**0.6106**	0.4403	0.5201	0.5378
AID362	**0.6620**	0.5909	0.6074	0.6270	**0.6525**	0.6155	0.6267	0.6485
w7a	0.7654	**0.8633**	0.4594	0.4748	0.7432	**0.8251**	0.3946	0.4053
CMC	**0.6474**	0.5082	0.5062	0.5417	**0.6607**	0.5288	0.5164	0.5746
APAS	**0.8569**	0.6346	0.5995	0.6193	**0.8426**	0.6372	0.5543	0.4792
CelebA	**0.8597**	0.7785	0.7102	0.7358	**0.8438**	0.7799	0.6764	0.6797
Chess	**0.7574**	0.6378	0.6076	0.6447	0.6138	**0.6241**	0.5829	0.6124
AD	**0.7624**	0.6603	0.6888	0.7033	**0.7620**	0.6592	0.6775	0.7084
SF	0.8157	0.6666	0.8181	**0.8446**	0.7667	0.6856	0.7660	**0.7865**
Probe	**0.9805**	0.9307	0.8951	0.9800	**0.9751**	0.8797	0.8990	0.9762
U2R	0.8846	0.8582	0.7911	**0.8848**	0.9776	0.7854	0.8168	**0.9781**
LINK	**0.9985**	0.9938	0.9723	0.9977	**0.9984**	0.9797	0.9636	0.9917
R10	**0.9893**	0.7648	0.9627	0.9866	**0.9926**	0.7566	0.9541	0.9796
CT	0.8570	0.8581	0.6154	**0.9773**	0.9072	0.8816	0.6374	**0.9364**
Avg.	**0.8037**	0.7125	0.6867	0.7444	**0.7954**	0.7072	0.6760	0.7247
Improvement (%)		12.80%	17.03%	7.97%		12.46%	17.66%	9.75%
p-value (CBRW vs.)		0.0012	0.0002	0.0435		0.0016	0.0008	0.0446

MarP and iForest using ENFW perform much worse than those working on the full feature set in almost all the used data sets. This is because, as discussed above, ENFW wrongly removes highly relevant features and degrades the outlier separability of the data sets, aggravating the detection performance of subsequent outlier detectors. It is interesting to note that MarP and iForest using ENFW perform much better than all their counterparts on *w7a*. This improvement is mainly because that ENFW removes almost 95% noisy features (against 53% achieved by CBRW_{fs}) while only loses little outlier separability on this data, as shown in Table 4.3. This indicates that although *w7a* contains many noisy features, these noisy features have less skewed frequency distributions than outlying features. As a result, simply examining the frequency distributions of individual features is probably the best way to clean up those noisy features.

4.5.5 Scalability Test

The scalability of CBRW w.r.t. data size is evaluated using five subsets of the largest data set *LINK*. The smallest subset contains 16,000 objects, and subsequent subsets are increased by a factor of four, until the largest subset which contains 4,096,000 objects.

The scaleup test results w.r.t. data size are presented in the left panel in Figure 4.3. As expected, all the five detectors have runtime linear to data size. The runtime of CompreX increases by a factor of more than 3,000 when the data size increases by a factor of 256; while that of CBRW increases by less than 60. Therefore, although CBRW and CompreX were implemented in different programming languages, the difference in *runtime ratio*[3] indicates that CBRW runs much faster than CompreX by a factor of more than 50. CBRW runs faster than iForest by a factor of more than 10, and is comparably fast to FPOF and MarP.

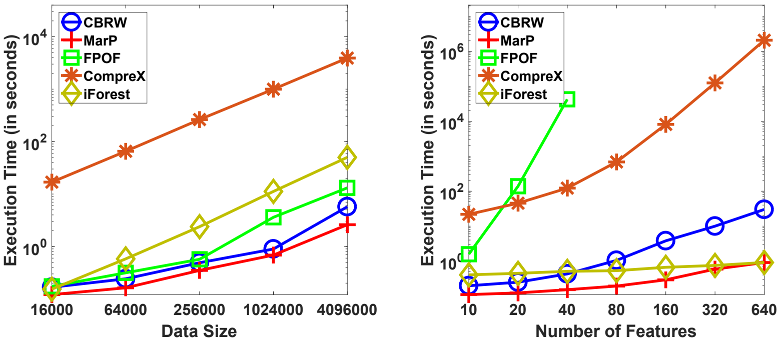


Figure 4.3: Scale-up Test Results of the Five Detectors w.r.t. Data Size and Dimensionality. Logarithmic scale is used in the vertical axis. Note that FPOF runs out-of-memory when the number of features reaches 80.

The scaleup test w.r.t. the number of features is conducted using seven synthetic data sets. The data sets have the same number of objects, i.e., 10,000 objects. The data set with the smallest number of features contains 10 features, and subsequent data sets are increased by a factor of two, until the data set with the largest number of features contains 640 features.

The results reported in the right panel in Figure 4.3 show that, as expected, CBRW has runtime nearly linear w.r.t. the number of features, which runs more than five orders of magnitude faster than FPOF. As indicated by runtime ratio, CBRW runs much faster than CompreX by a factor of more than 500. Since CBRW models much more complex underling data characteristics compared to MarP and iForest, it runs slower than these

[3] Since CompreX was implemented in a different programming language to the other methods, the runtime between CompreX and other methods is incomparable. Instead, we compare them in terms of runtime ratio, i.e., the runtime on a larger/higher-dimensional data set divided by that on a smaller/lower-dimensional data set, for a fairer comparison. Since the data size and the increasing factor of dimensionality are fixed, the runtime ratio is comparable across the methods in different programming languages.

two competitors by a factor of more than 30, but it significantly outperforms these two detectors in the AUC performance shown in Table 4.2.

4.5.6 Sensitivity Test w.r.t. the Damping Factor α

CBRW only has one parameter, the damping factor α. The use of α is to avoid the random walking getting stuck in isolated nodes by offering a small restart probability $(1-\alpha)$, which guarantees the algorithmic convergence while does not affect the effectiveness. $\alpha = 1.0$ is not recommended as this may break the convergence condition. Also, α should be sufficiently large, e.g., $\alpha \geq 0.85$, and the underlying graph structure is ignored otherwise. Below we examine the sensitivity of CBRW w.r.t. α in a wide range of values $[0.85, 0.99]$ by performing direct outlier detection. Figure 4.4 reports the AUC results w.r.t. α on all 15 data sets.

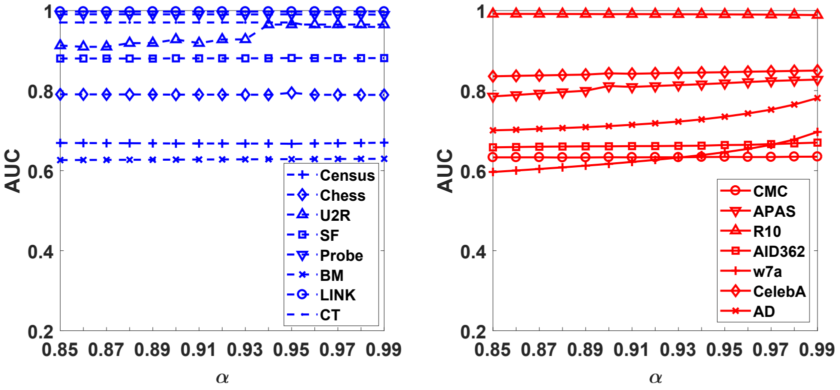


Figure 4.4: Sensitivity Test Results w.r.t. the Parameter α.

The results show that CBRW performs very stably over a large range of tuning options on most of the data sets, and a large α is more preferable than a small one. This is because (i) α is introduced to guarantee the convergence of the CBRW algorithm and it is data-insensitive in terms of effectiveness, which is different from some data-sensitive parameters in other detectors, such as the minimum support in FPOF and the subsampling size in iForest; and (ii) the graph structure and edges weights are carefully designed to highlight the outlying values, and we need to make use of this graph nature by setting a large α. A large α is needed to achieve the best performance on some data sets, e.g., *U2R*, *APAS*, *w7a* and *AD*. These data sets may contain some highly noisy values. A large α is required to increase the gap between the outlierness of outlying values and the highly noisy values. On the other hand, a medium α is needed to obtain the best performance on other data sets, like *CT*. This may be because some outlying values in these data sets cannot attract sufficiently large outlierness in the original graph structure, but rather rely on some outlierness propagated through restart probabilities. Therefore, we recommend using a relatively large α (e.g., $\alpha = 0.95$) to leverage both cases.

4.5.7 Convergence Test

The convergence rate of random walks is governed by two key graph properties - the graph diameter and the Cheeger constant [36, 41]. The runtime for computing the Cheeger constant is prohibitive for large graphs, so we replace this constant with clustering coefficients. The graph's diameter and clustering coefficients of the value graph for each data set are presented in Table 4.5. It is clear that all the value graphs has small graph diameter and large clustering coefficient. This is because a value in one feature often co-occurs with most, if not all, of the values in other features. Moreover, there exist linkages between values as long as the values co-occur together, resulting in a highly connected dense value graph. Fast convergence rates are expected for random walks on such graphs [36, 41].

The convergence test results in Figure 4.5 show that CBRW converges quickly on all 15 data sets, i.e., within 70 iterations. CBRW converges after about 10 iterations on 13 data sets, but takes about 70 iterations to converge on *Probe* and *U2R*. This is because these two data sets contain a large proportion of feature values having frequencies of less than three. This is particularly true for *Probe*. As a result, although their overall clustering coefficient is high, its Cheeger constant can be quite small, which leads to slower convergence.

Table 4.5: Two Key Properties of a Value Graph. Data is sorted by clustering coefficient. ‘◦’ indicates out-of-memory exceptions.

Data	Diameter	Coefficient
Census	2	0.76
Chess	2	0.79
U2R	2	0.80
SF	2	0.81
Probe	2	0.82
BM	2	0.85
LINK	2	0.86
CT	2	0.87
CMC	2	0.89
APAS	2	0.90
R10	2	0.91
AID362	2	0.92
w7a	2	0.93
CelebA	2	0.99
AD	◦	◦

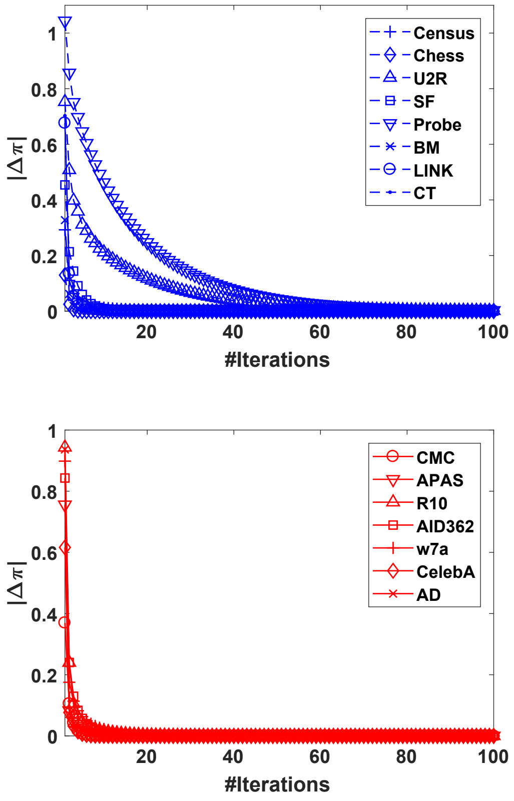


Figure 4.5: Convergence Test Results.

4.6 Summary

This chapter introduces a novel outlier detection framework (CUOT) and its instantiation (CBRW) for detecting outliers in data with interdependent feature values. Compared to traditional pattern-based methods, CUOT is data-driven, which learns from low-level intra- and inter-feature value couplings to estimate outlier scores of feature values. The outlier scores of feature values can determine the outlying ranking of both data objects and features. Motivated by the homophily phenomenon, CBRW models a value outlierness propagation process by a biased random walk on an attributed value-value graph to capture the cascade relation of value-level outlier factors. The effectiveness of CBRW is supported by significant AUC improvement over five state-of-the-art competitors on a large collection of 15 data sets with different complexities. CBRW is particularly superior in complex data, e.g., data sets contain sophisticated value couplings, high levels of noisy features, and/or low outlier separability.

CUOT may be extended to project the categorical values into a numeric low-dimensional outlier-resilient embedding space, in which each categorical value is represented by a low-dimensional vector, such that off-the-shelf numeric data-based learning methods (e.g., existing state-of-the-art classification, clustering and regression methods) can be applied to extract more sophisticated knowledge from categorical data while being outlier-resilient. Actually, CUOT has already projected the categorical values onto a one-dimensional new space where each categorical value is represented by a numeric outlier score, but it only attempts to capture the exceptional characteristics of the values. When CUOT captures more intrinsic data characteristics, the embedding of value would have better representation power and facilitate different outlier-resilient learning performance.

The instance CBRW only captures intra-feature value couplings and pairwise inter-feature couplings, which may omit long-length outlying patterns. On the other hand, simply using patterns obtained by pattern mining approaches fail to work effectively, in particular for data with noisy features. Considering the couplings between patterns in value outlierness estimation may help address this issue. Therefore, incorporating arbitrary-length patterns and their complex couplings into the CUOT framework may further improve the performance in data sets with long-length outlying patterns and sophisticated noisy features.

Chapter 5

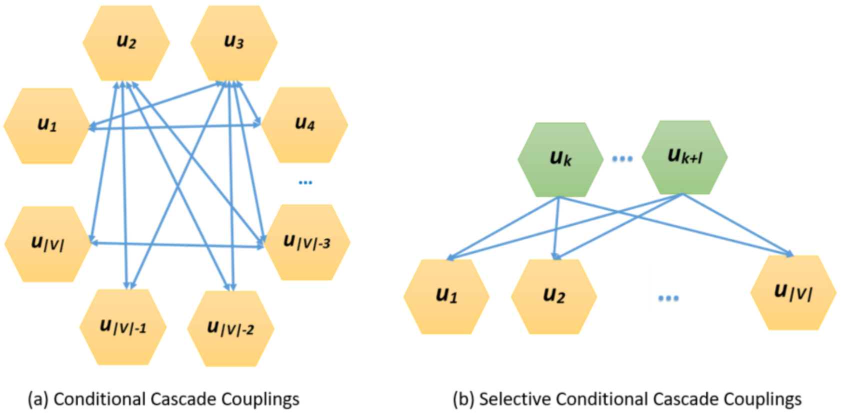
Selective Conditional Cascade of Outlier Factors

5.1 Introduction

In Chapter 4, we justified how the outlierness of a behavior can be influenced by its coupled behaviors, which provides important insights into the importance of considering the couplings between the outlier factors of values. In this chapter, we investigate whether all the couplings are relevant to the outlier detection tasks. Different from the previous chapter that considers the full couplings between the outlier factors, this chapter examines the use of selective couplings to build more cost-effective and efficient outlier detectors, which is particularly important for high-dimensional outlier detection.

This is due to two main challenges brought by high dimensionality. (i) High-dimensional data often contains a complex mixture of relevant and irrelevant features. The irrelevant features are 'noise' to outlier detection, since outliers are masked as normal objects by these features. Moreover, the sophisticated couplings within irrelevant features and between relevant and irrelevant features bring about substantially more 'noise' that impedes the separability of outliers from normal objects. (ii) It also presents a huge search space, i.e., 2^D, resulting in great difficulty in exploring the mixed couplings across the features. Most existing full space-based methods [46, 70, 100] can be largely biased by irrelevant features, particularly when the percentage of irrelevant features is large. This issue also applies to subspace/feature selection-based methods [7, 10, 57, 91, 96, 97] . This is because they need to search the outlying features/subspaces in the original data space independently from the subsequently outlier scoring, and subsequently may retain features that are irrelevant to the outlier scoring functions. Also, such search is very costly on high-dimensional data due to its huge search space.

The above analysis suggests that how to effectively and efficiently identify and model on a clean and condensed space from the original data space is the key to detecting high-dimensional outliers. Accordingly, this chapter proposes a novel high-dimensional outlier detection framework for categorical data by modeling *Selective Value Couplings* (the SelectVC framework for short), i.e., selective feature value interactions that are positively related to outlier detection. As shown in Figure 5.1(b), SelectVC aims to model the outlierness influence from only a set of the candidate outlying values $\{u_k, u_{k+1}, \cdots, u_{k+l}\}$ to all the values in $\mathcal{V}$ and iteratively update the outlierness of values and the candidate outlying value set, forming a selective cascade couplings of the outlier factors of the values. This is very different from the conditional cascade couplings in Figure 5.1(a) that capture the full value interactions. Since only a small percentage of behaviors are abnormal in real-life applications, learning the selective cascade couplings is more faithful in modeling the homophily couplings between outlying behaviors.



(a) Conditional Cascade Couplings

(b) Selective Conditional Cascade Couplings

Figure 5.1: Comparison of Selective Conditional Cascade Couplings and the Conditional Cascade Couplings Presented in Chapter 4.

We further instantiate the SelectVC framework to a method called POP. POP simulates Partial Outlierness Propagation from the value subset to the full value set to model the selective cascade couplings. The partial outlierness propagation is efficient for handling very high-dimensional data and is resilient to the large number of noisy features in those data.

Accordingly, this chapter makes the following two major contributions:

- The proposed SelectVC framework for outlier detection is novel for high-dimensional categorical data. Different from existing approaches that primarily work on the original full space and/or feature subsets identified independently from outlier scoring, SelectVC works on a clean and condensed data space composed by the couplings between the outlying value set and the full value set, by jointly optimizing outlying value selection and value outlierness scoring. This enables SelectVC to have a more reliable outlierness estimation on data with overwhelming irrelevant features.

- The performance of SelectVC is verified by its instance POP. POP models the contrasting couplings between outlying-to-outlying values and normal/noisy-to-outlying values by partial outlierness propagation. Our theoretical analysis shows that such outlierness propagation biases towards outlying behaviors, which assists POP to assign larger outlierness to outlying behaviors than non-outlying behaviors.

Extensive experiments show that POP (i) significantly outperforms five state-of-the-art full space- or subspace-based outlier detectors and their combinations with three feature selection methods (5%-39% AUC improvement) on 12 real-world high-dimensional data sets with different levels of irrelevant features; (ii) obtains good scalability w.r.t. data size and dimensionality; (iii) performs stably w.r.t. its only parameter k; and (iv) obtains fast convergence rate.

In the rest of this chapter, SelectVC is detailed in Section 5.2. POP is introduced in Section 5.3, followed by a theoretical analysis in Section 5.4. Empirical results are provided in Section 5.5. We conclude this chapter in Section 5.6.

5.2 The Proposed SelectVC Framework

SelectVC jointly optimizes value selection and value outlierness scoring, which is described as follows. Let $dom(\mathsf{F}) = \{v_1, v_2, \cdots\}$ be the domain of a feature F and $\mathcal{V}$ be the whole set of feature values in $\mathcal{F}$: $\mathcal{V} = \cup_{\mathsf{F}\in\mathcal{F}} dom(\mathsf{F})$, where $dom(\mathsf{F}) \cap dom(\mathsf{F}') = \emptyset, \forall \mathsf{F} \neq \mathsf{F}'$. As shown in Figure 5.2, given an initial value outlierness vector $\mathbf{q} \in \mathbb{R}^{|\mathcal{V}|}$, SelectVC first defines a value selection function $\psi(\mathbf{q})$ to select a set of outlying values, $\mathcal{U} \subset \mathcal{V}$. SelectVC further defines a value scoring function $\phi(\mathcal{U})$ that computes an outlier score for every single value in the full value set $\mathcal{V}$ by modeling the couplings between the single value and the values in the value subset $\mathcal{U}$. These two functions are iteratively reinforced until a stationary $\mathbf{q}$ is found. After obtaining value outlierness, given an object $\mathbf{x}$, we can integrate the outlierness of values contained by $\mathbf{x}$ to compute the object outlierness.

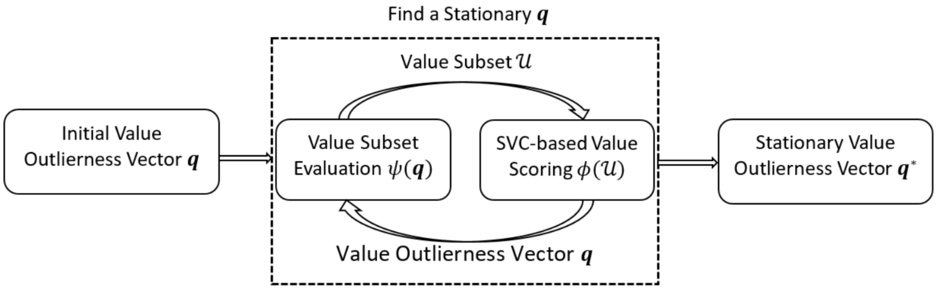

Find a Stationary q
Value Subset $\mathcal{U}$
Initial Value Outlierness Vector q
Value Subset Evaluation $\psi(q)$
SVC-based Value Scoring $\phi(\mathcal{U})$
Stationary Value Outlierness Vector q^*
Value Outlierness Vector q

Figure 5.2: The SelectVC Framework for Estimating Value Outlierness Based on Selective Value Couplings. The outlierness of data objects can then be obtained using value outlierness. SVC is short for Selective Value Couplings.

Outliers often demonstrate multiple outlying behaviors in high-dimensional data, i.e., outlying behaviors are often concurrent. Moreover, outlying behaviors have very low individual frequency. This results in strong mutual couplings between outlying behaviors. On the other hand, although outlying behaviors also co-occur with *non-outlying behaviors* (including *normal behaviors* and *noisy behaviors* - frequent and infrequent values which are mainly contained by normal objects, respectively), non-outlying behaviors are distributed very differently from outlying behaviors since they are manifested by respective normal objects and outliers. This results in weak couplings between non-outlying behaviors and outlying behaviors. The strength of couplings between outlying behaviors is therefore *contrasting* to that between non-outlying behaviors and outlying behaviors. SelectVC essentially models such contrasting couplings to iteratively assign larger outlierness to outlying values than normal/noisy values. The efficiency of SelectVC is mainly determined by the value selection function (ψ).

SelectVC is fundamentally different from existing frameworks in that: (i) SelectVC models the interactions with only the outlying behaviors. This avoids the interference from irrelevant couplings between irrelevant features, which significantly challenge full space-based approaches; and (ii) SelectVC unifies the two dependent tasks, value selection and outlier scoring, to optimize its outlier scoring, while existing subspace/feature selection-based approaches separate subspace/feature selection from outlier scoring and thus the subspaces/features retained by subspace/feature selection may be irrelevant to subsequent outlier detectors.

5.2.1 Value Subset Evaluation Function ψ

Since SelectVC aims to capture interactions of a value with only outlying values, function ψ is required to select a value subset $\mathcal{U}$ that consists of the most likely outlying values to facilitate the value outlier scoring in the next stage.

Definition 5.1 (Value Selection)**.** *Value subset evaluation function ψ is to select a value subset $\mathcal{U}$ that contains the most likely outlying values from all the possible $\binom{|\mathcal{V}|}{|\mathcal{U}|}$ subsets.*

The value selection here is similar as feature selection, but we work on the value level. Nevertheless, subset search methods for feature selection, such as sequential search, random search and complete search [75], can be used to select a proper value subset.

5.2.2 Selective Value Coupling-based Scoring Function ϕ

Outlying behaviors are often strongly bond together while they are weakly coupled with other behaviors [91]. For example, the abnormal symptoms of diseases (e.g., the suspected signs like frequent urination, tiredness, and excessive thirsty for diabetes) are often concurrent, whereas they have weak association with normal symptoms or misdiagnosed abnormal symptoms.

SelectVC exploits such contrasting couplings to compute value outlierness by modeling the selective value couplings with only the outlying value set $\mathcal{U}$.

Definition 5.2 (Value Scoring)**.** *The value scoring function $\phi : \mathcal{V} \mapsto \mathbb{R}$ exploits the couplings of a given value $v \in \mathcal{V}$ with the value subset $\mathcal{U}$ to compute the outlierness of the value v:*

$$\mathbf{q}(v) = \phi_v(\mathcal{U}) = \odot_{s\in\mathcal{U}}\eta(v, s), \tag{5.1}$$

where $\eta(\cdot,\cdot)$ captures the relation between the two values v and s, e.g., joint probability and conditional probability, and $\odot$ denotes one type of integration over η, e.g., first-order linear (or polynomial non-linear) summation and multiplication.

By working on the selective value couplings, SelectVC minimizes the interference from irrelevant features while captures the sufficient relevant information to assign larger outlierness to outlying values than normal/noisy values.

5.2.3 Stationary Criterion

The total number of possible value subsets is huge and different value subsets will result in very different value outlierness vectors. SelectVC aims to produce a stationary value outlierness vector to facilitate stable outier detection performance. Since we evaluate the convergence w.r.t. a vector, widely-used vector norms can be used. Let t be the iteration number, then a p-norm-based stationary criterion can be defined as follow.

$$\lim_{t\to\infty} ||\mathbf{q}_{t+1} - \mathbf{q}_t||_p \leq \epsilon, \tag{5.2}$$

where $p \geq 1$ and ϵ is a small constant.

5.3 The SelectVC Instance: POP

The SelectVC framework can be instantiated by specifying its three components: value scoring function ϕ, value subset evaluation function ψ, and the stationary criterion. The POP instance specifies these three components as follows. POP first specifies the functions ψ and ϕ by a top-k value selection function and a partial outlierness propagation-based value scoring function, respectively. POP then defines a stationary criterion using ℓ_1-norm.

5.3.1 Specifying ψ Using Top-k Outlying Value Selection

Given a value outlierness vector $\mathbf{q}$, POP defines a top-k outlying value selection function to select a value subset $\mathcal{U}$ containing a *k proportion* of the most outlying values from the full value set $\mathcal{V}$.

Definition 5.3 (Top-k Outlying Value Selection). *The top-k outlying value selection selects a value subset $\mathcal{U}$ with the cardinality $k|\mathcal{V}|$ from the full value set $\mathcal{V}$ as follows.*

$$\psi(\mathbf{q}) = \underset{\mathcal{U}\subset\mathcal{V}\ and\ |\mathcal{U}|=k|\mathcal{V}|}{\arg\max} \sum_{s\in\mathcal{U}} \mathbf{q}(s). \tag{5.3}$$

Since $\mathbf{q}$ contains the outlierness of all feature values, after using the entries in $\mathbf{q}$ to sort the values in a descending order, Equation (5.3) is equivalent to selecting the top-ranked $k|\mathcal{V}|$ values. This value selection can be done in linear time, which well guarantees the scalability of POP to very high-dimensional data.

Note that outlying value selection is nontrivial due to the presence of noisy values and the huge search space. Simply selecting the most infrequent values may include the noisy values and consequently downgrade the quality of value outlierness estimation. Therefore, in the next section, POP initializes the value selection based on the frequencies of individual values but jointly optimizes the value selection and value scoring to obtain reliable outlying value sets and value outlierness.

5.3.2 Specifying ϕ by Partial Outlierness Propagation

POP defines a partial outlierness propagation-based function ϕ to leverage the contrasting couplings between outlying values to the selected subset $\mathcal{U}$ and normal/noisy values to the subset $\mathcal{U}$.

POP first builds a $|\mathcal{V}| \times |\mathcal{U}|$ matrix to capture the selective couplings of the values in the full value set $\mathcal{V}$ with the values in $\mathcal{U}$ using conditional probability.

Definition 5.4 (Selective Coupling Matrix)**.** *The relation between the values in $\mathcal{V}$ and the values in $\mathcal{U}$ is captured by the selective coupling matrix $\mathbf{M} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{U}|}$ which is defined as:*

$$\mathbf{M} = \begin{bmatrix} \eta(v_1, s_1) & \dots & \eta(v_1, s_{|\mathcal{U}|}) \\ \vdots & \ddots & \vdots \\ \eta(v_{|\mathcal{V}|}, s_1) & \dots & \eta(v_{|\mathcal{V}|}, s_{|\mathcal{U}|}) \end{bmatrix}, \ v_i \in \mathcal{V}, s_j \in \mathcal{U}, \tag{5.4}$$

where $\eta(v_i, s_j) = P(s_j|v_i) = \frac{freq(v_i,s_j)}{freq(v_i)} \in [0,1]$ and freq denotes a frequency counting function.

Let u and u' be outlying and normal values, respectively. Given an outlying values $s \in \mathcal{U}$, since outlying values are often concurrent and the co-occurrence frequency is upper bounded by the frequency of s, we often have $freq(u,s) \gtrapprox freq(u',s)$. Moreover, per definition of outliers, $freq(u) \ll freq(u')$. Therefore, we normally obtain $\eta(u,s) > \eta(u',s)$ or $\eta(u,s) \gg \eta(u',s)$.

Let u'' be a noisy value. We may assume $freq(u) \approx freq(u'')$ as both u and u'' are infrequent. Since noisy values and outlying values are mainly contained by normal objects and outliers, respectively, u'' is presumed to have lower joint probabilities with the outlying values in $\mathcal{U}$, compared to the outlying value u. Thus, we also obtain $\eta(u,s) > \eta(u'',s)$. This demonstrates that the inherent asymmetrical property of conditional probability enables POP to effectively capture the aforementioned contrasting couplings.

POP further defines a partial outlierness propagation-based value scoring function ϕ by using $\mathbf{M}$ to propagate the outlierness of values in $\mathcal{U}$ to influence the scoring of values in $\mathcal{V}$.

Definition 5.5 (Partial Outlierness Propagation-based Value Scoring)**.** *The partial outlierness propagation-based value scoring function ϕ is defined as follows.*

$$\mathbf{q}_{t+1}(v) = \phi_v(\mathcal{U}_t) = \sum_{s\in\mathcal{U}_t} \tilde{\mathbf{M}}(v,s)\mathbf{q}_t(s), \tag{5.5}$$

where $\tilde{\mathbf{M}}$ denotes a column-wise normalization of $\mathbf{M}$, $\mathbf{q}_t$ is normalized into a ℓ_1-norm unit, and $t \in \mathbb{Z}^+$ is a positive integer.

Equation (5.5) models the selective value couplings by simulating to partially propagating the t-th step value outlierness to the outlierness scoring in the $(t+1)$-th step. Such partial outlierness propagation assits POP to iteratively enlarge the outlierness gap between the top-ranked values and the rest of values in the outlierness vector $\mathbf{q}$.

We initialize the vector $\mathbf{q}$ using a similar function as δ presented in Section 4.3.1 to produce a good initialization when the frequency distributions are very skewed across the features.

5.3.3 ℓ_1-Norm Stationary Criterion

A ℓ_1-norm-based stationary criterion is used in POP.

Definition 5.6 (ℓ_1-Norm Stationary Criterion)**.** *A value outlierness vector* $\mathbf{q}$ *is stationary when satisfying:*

$$\Delta = ||\mathbf{q}_{t+1} - \mathbf{q}_t||_1 = \sum_{v \in \mathcal{V}} |\mathbf{q}_{t+1}(v) - \mathbf{q}_t(v)| \leq \epsilon, \tag{5.6}$$

where $\epsilon = 10^{-4}$ *is used.*

Actually, since the matrix $\mathbf{M}$ is fixed, POP obtains the stationary status when the values and their ranks in $\mathcal{U}$ do not change.

5.3.4 The Algorithm and Its Time Complexity

Algorithm 5.1 presents the procedures of POP. Steps (1-6) are performed to obtain a $|\mathcal{V}| \times |\mathcal{V}|$ *full value coupling matrix* $\mathbf{M}'$. Since the conditional probabilities are fixed for all value pairs, we generate $\mathbf{M}'$ to facilitate quick access to the selective coupling matrix $\mathbf{M}$, which avoids re-scanning the data in the later iteration. Steps (7-11) performs the joint value selection and value scoring process to obtain the stationary $\mathbf{q}$. We then compute the outlierness of data objects in Steps (13-15). In Step (14), we compute the outlierness of an object $\mathbf{x}_i$ as the weighted outlierness summation of its values, in which $\omega_j = \sum_{v \in dom(\mathsf{F}_j)} \mathbf{q}(v)$. Such weighted outlierness integration highlights relevant features and facilitates a proper object outlierness estimation. We found that this object outlierness calculation achieves similar detection performance as the one used in CBRW in Eqn. (4.12). An object outlierness ranking $\mathbf{r}$ is finally returned in Step (16).

Algorithm 5.1 *POP-based Outlier Detection*

```
Input: $\mathcal{X}$ - data objects, $k$ - a proportion of the full value set
Output: $\mathbf{r}$ - an outlier ranking
 1: Initialize a $|\mathcal{V}| \times |\mathcal{V}|$ matrix $\mathbf{M}'$ for full value couplings
 2: for $v$ in $\mathcal{V}$ do
 3:     for $v'$ in $\mathcal{V}$ do
 4:         $\mathbf{M}'(v, v') \leftarrow \frac{freq(v,v')}{freq(v)}$
 5:     end for
 6: end for
 7: Initialize $\mathbf{q} \in \mathbb{R}^{|\mathcal{V}|}$
 8: repeat
 9:     $\mathcal{U} \leftarrow \underset{\mathcal{U} \subset \mathcal{V} \text{ and } |\mathcal{U}| = k|\mathcal{V}|}{\arg\max} \sum_{s \in \mathcal{U}} \mathbf{q}(s)$
10:     $\mathbf{q} \leftarrow \tilde{\mathbf{M}}_{|\mathcal{V}| \times |\mathcal{U}|} \times \mathbf{q}_{|\mathcal{U}| \times 1}(\mathcal{U})$
11: until Converge or reach the maximum iteration 200
12: Initialize $\mathbf{r} \in \mathbb{R}^{|\mathcal{X}|}$ as an outlierness vector for data objects
13: for $\mathbf{x}_i$ in $\mathcal{X}$ do
14:     $r_i \leftarrow \sum_{\mathsf{F}_j \in \mathcal{F}} \mathbf{q}^*(x_{ij})\omega_j$
15: end for
16: return $\mathbf{r}$
```

POP requires one scanning over the data objects to obtain $\mathbf{M}'$ in Steps (1-6), which has $O(|\mathcal{X}||\mathcal{V}|^2)$. The iterations in Steps (8-11) have $O(|\mathcal{V}||\mathcal{S}|)$ time complexity. The object outlierness scoring and sorting take $O(|\mathcal{X}||\mathcal{V}|)$ in Steps (12-16). Therefore, the overall time complexity of POP is linear w.r.t. the data size and quadratic w.r.t. the total number of values. Since the average number of values per feature is normally very small, POP also has quadratic time complexity w.r.t. the number of features.

5.4 Theoretical Analysis

This section analyzes the quality of the vector $\mathbf{q}^*$, the capability of POP in handling high-dimensional data, and the setting of k.

5.4.1 Quality of the Stationary Vector $\mathbf{q}^*$

We show below that $\mathbf{q}$ becomes stable when the values in the selected subset $\mathcal{U}$ have the largest total pointwise mutual information.

Theorem 5.1 (Stationary Vector)**.** *Let $pmi(\mathcal{W})$ be the total pointwise mutual information among the values in a value set $\mathcal{W}$, i.e., $pmi(\mathcal{W}) = \sum_{u \in \mathcal{W}} \sum_{u' \in \mathcal{W}} \log \frac{P(u,u')}{P(u)P(u')}$. Then, the value outlierness vector $\mathbf{q}$ converges to a vector $\mathbf{q}^*$ s.t. $\forall \mathcal{W} \subseteq \mathcal{V}$ and $|\mathcal{W}| = |\mathcal{U}^*|$, $pmi(\mathcal{U}^*) \geq pmi(\mathcal{W})$, where $\mathcal{U}^*$ is the stationary value subset.*

Proof. At each iteration of POP, the subset $\mathcal{U}$ is updated until convergence, while the value conditional probability matrix $\mathbf{M}$ is fixed. Therefore, $\mathbf{q}$ becomes stationary when $\mathcal{U}$ does not change, i.e., $||\mathbf{q}_{t+1} - \mathbf{q}_t||_1 \leq \epsilon$ if $\mathcal{U}_t \subseteq \mathcal{U}_{t+1}$ and $\mathcal{U}_{t+1} \subseteq \mathcal{U}_t$.

Since $\mathbf{q}$ is updated using the conditional probabilities of a given value $v \in \mathcal{V}$ on the value subset $\mathcal{U}$, $\mathbf{q}(v)$ is primarily determined by the probabilities of the values in $\mathcal{U}$ given value v. Therefore, $\mathbf{q}(v) \propto \sum_{s \in \mathcal{U}} P(s|v)$ and thus $\mathbf{q}(\mathcal{U}) = \sum_{s' \in \mathcal{U}} \mathbf{q}(s') \propto \sum_{s' \in \mathcal{U}} \sum_{s \in \mathcal{U}} P(s|s')$. We have $\mathbf{q}(\mathcal{U}) \propto \sum_{s' \in \mathcal{U}} \sum_{s \in \mathcal{U}} \frac{P(s|s')}{P(s)}$ after taking account of the way we initialize $\mathbf{q}$. We will obtain a value subset $\mathcal{U}^*$ which has the largest *pmi* by maximizing $\mathbf{q}(\mathcal{U})$, and subsequently obtain $\mathbf{q}^*$ based on the subset $\mathcal{U}^*$. $\mathcal{U}^*$ remains unchanged since $\psi(\mathcal{U}^*)$ is already maximized, and thus $\mathbf{q}^*$ becomes stationary.

□

It is well known in natural language processing that pointwise mutual information biases towards rare words [117], i.e., pointwise mutual information between concurrent rare words are generally much larger than commonly-used or frequent words. In our case, this implies that the top-ranked values in the stationary vector $\mathbf{q}^*$ are normally outlying values - values which are exceptionally rare and have mutual interactions. In other words, POP can often obtain a highly discriminate outlierness vector where outlying values have larger outlierness than normal and noisy values.

5.4.2 Handling Distance Concentration Effect

The *concentration of distances* is a major issue in the curse of dimensionality. The distance concentration effect states that the discrimination between the near and far neighbors of a data object diminishes with increasing dimensions, in particular when the increased dimensions are irrelevant features [128].

Since we focus on value outlierness estimation, in general, we expect $||\mathbf{q}_t(u) - \mathbf{q}_t(v)||_p$ to be sufficiently large if u and v are respective outlying values and normal/noisy values, and to be small otherwise. Let v to be a normal value, without loss of generality, there exists a normal value u as its nearest neighbor and an outlying value w as its farthest neighbor. For a given value set $\mathcal{W} \subseteq \mathcal{V}$, according to the concentration effect theory [128], however, we have

$$\lim_{|\mathcal{W}| \to \infty} \frac{max\_d - min\_d}{min\_d} = 0, \tag{5.7}$$

where $max\_d = || \sum_{w' \in \mathcal{W}} \tilde{\mathbf{M}}'(v, w')\mathbf{q}'_t(w') - \sum_{w' \in \mathcal{W}} \tilde{\mathbf{M}}'(w, w')\mathbf{q}'_t(w')||_p$ and $min\_d = || \sum_{w' \in \mathcal{W}} \tilde{\mathbf{M}}'(v, w')\mathbf{q}'_t(w') - \sum_{w' \in \mathcal{W}} \tilde{\mathbf{M}}'(u, w')\mathbf{q}'_t(w')||_p$ denote the largest and smallest distances to v, respectively.

As shown in [128], the concentration effect becomes more and more severe as the number of irrelevant features increases. Therefore, the larger the size of the value subset $\mathcal{W}$ is, we would be likely to have more severe concentration effect. The concentration effect is maximal when we use the full value couplings, i.e., to set $\mathcal{W} = \mathcal{V}$. POP substantially reduces such effect by working on a small value subset. POP could well overcome the concentration effect when setting k to be a sufficiently small value, but POP may lose relevant value couplings when k is too small. We will provide a general guideline for setting k in the next section.

5.4.3 Guidelines for Setting k

This section provides some guidelines for tuning the only parameter k, in particular for high-dimensional and small-sized data, based on three observations that (i) outliers typically account for only a small proportion of a data set; (ii) outliers often demonstrate their exceptional behaviors in only a small feature subset in high-dimensional data; and (iii) large k may lead to more severe distance concentration effect.

Theorem 5.2 (Maximum Number of Outlying Values)**.** *Let $\mathcal{O}$ be the set of outlier objects in the data set $\mathcal{X}$, I be the maximum number of outlying values contained by an outlier $\mathbf{o} \in \mathcal{O}$, and H be the total number of all possible outlying values in $\mathcal{X}$. Then*

$$H \leq I|\mathcal{O}|. \tag{5.8}$$

Proof. When all outliers in $\mathcal{O}$ manifest different outlying values, we have $H = I|\mathcal{O}|$. If there exists at least one $\mathbf{o} \in \mathcal{O}$ sharing the same outlying values with other outliers, then $H < I|\mathcal{O}|$. □

Corollary 5.2.1 (Upper Bound for k)**.** *Let $\mathcal{U}^*$ be the value subset containing exactly all the possible outlying values, i.e., $|\mathcal{U}^*| = H$ and $k^* = \frac{|\mathcal{U}^*|}{|\mathcal{V}|}$. In high-dimensional and* *small-size data, i.e.,* $|\mathcal{V}| > |\mathcal{F}| > |\mathcal{X}|$*, we have*

$$k^* \leq \frac{I|\mathcal{O}|}{|\mathcal{V}|} < \frac{I|\mathcal{O}|}{|\mathcal{X}|}. \tag{5.9}$$

According to Corollary 5.2.1, k^* is upper bounded by the outlier proportion $\frac{|\mathcal{O}|}{|\mathcal{X}|}$ and the number of outlying values contained per outlier I in a high-dimensional and small-size data set. In general, $k^* < 0.5$ is a good bound based on the above three observations. Since our goal is to select a reliable outlying value subset and to substantially reduce the concentration effect, $k < k^*$ is suggested. We show in Section 5.5.7 that POP with $k = 0.3$ obtains stable performance in data sets with diverse dimensions.

5.5 Experiments and Evaluation

We perform experiments to answer the following six questions:

- **Q1. Effectiveness in real-world data.** How accurately does POP detect outliers in real-world high-dimensional data with different levels of irrelevant features?
- **Q2. Significance of partial outlierness propagation.** How well does partial outlierness propagation perform compared to full outlierness propagation?
- **Q3. Significance of joint value selection and outlier scoring.** Can we replace POP with two independent successive modules: feature selection and outlier detection?
- **Q4. Scalability.** Does POP have good scalability?
- **Q5. Sensitivity.** How sensitive is POP to k?
- **Q6. Convergence.** How fast does POP converge?

5.5.1 Experiment Environment

POP and its competitors are implemented in JAVA. The implementations of all the competitors are obtained from their authors or the open-source platform ELKI [1]. All the experiments are executed at a node in a 3.4GHz Titan Cluster with 96GB memory.

5.5.2 Data Sets

Twelve publicly available real-world data sets are used, which cover diverse domains, e.g., Internet advertising, image object recognition, web page classification and text classification, as shown in Table 5.1. The data indicators are defined in Section 2.3.3. The four balanced data sets, *PCMAC*, *BASE*, *WebKB*, and *RELA*, are transformed into outlier detection data sets using the downsampling method described in Section 2.3.1, while the other eight highly imbalanced data sets are directly transformed using the rare class conversion method.

Table 5.1: A Summary of Data Sets Used and Indicator Quantization Results. $\kappa_{livc} = \frac{\kappa_{rel}(\mathcal{U})-\kappa_{rel}(\mathcal{V})}{\kappa_{rel}(\mathcal{V})}$ describes the level of irrelevant value couplings per data. The middle horizontal line roughly separates data sets with high κ_{livc} from that with low κ_{livc}.

Data Summary				Data Indicators			
Data	Acronym	$\lvert\mathcal{X}\rvert$	$\lvert\mathcal{F}\rvert$	$\kappa_{rel}(\mathcal{V})$	$\kappa_{rel}(\mathcal{U})$	κ_{livc}	κ_{sep}
w7a	-	49749	300	0.1490	0.4440	197.99%	0.5927
wap.wc	-	346	4229	0.0306	0.0866	183.01%	0.9713
Reuters8	R8	3974	9467	0.0358	0.0980	173.74%	0.9358
Caltech-16	CAL16	829	253	0.1099	0.2961	169.43%	0.9613
InternetAd	AD	3279	1555	0.1923	0.4370	127.25%	0.6982
Caltech-28	CAL28	829	727	0.0654	0.1465	124.01%	0.9780
CelebA	-	202599	39	0.0307	0.0665	116.61%	0.7961
PCMAC	-	1002	3039	0.0327	0.0638	95.11%	0.7721
BASEHOCK	BASE	1019	4320	0.0347	0.0613	76.66%	0.6292
WebKB	-	1658	6601	0.0303	0.0526	73.60%	0.7501
RELATHE	RELA	794	4080	0.0320	0.0554	73.13%	0.6365
Arrhythmia	Arrhy	452	64	0.2548	0.4287	68.25%	0.6293

5.5.3 Effectiveness in Real-world Data

Experimental Settings

POP is compared with five detectors: CBRW [91], ZERO [97], iForest [77], ABOD [70] and LOF [20] on the 12 real-world data sets to evaluate its effectiveness.

- *Subspace-based Competitors*: ZERO and iForest. Both ZERO and iForest are state-of-the-art non-deterministic subspace methods[1]. Their performance is taken average from 10 runs. iForest and ZERO are used with the recommended settings in [77, 97], respectively.

- *Full Space-based Competitors*: CBRW, ABOD and LOF. CBRW is a state-of-the-art outlier detector for categorical data and it is closely related to POP. ABOD is an angle-based method which is specially designed for high-dimensional data. LOF is one of the most popular methods that works on full dimensionality and it is used as a baseline competitor. As recommended in [91], $\alpha = 0.95$ is used in CBRW. ABOD is parameter-free. For LOF, small values are suggested for the neighborhood size *MinPts* in [20]. We performed LOF with a range of different *MinPts*, i.e., $\{1, 5, 10, 20, 40, 60, 80, 100\}$, and report the results with *MinPts* $= 5$ as LOF using *MinPts* $= 5$ performs more stably across the data sets.

POP uses $k = 0.3$ by default. We will compare POP with feature selection-enabled methods in Section 5.5.5. Note that categorical data is transformed into numeric data to allow iForest, ABOD and LOF to work on the same data. The data sets are transformed by using a commonly used method 1-of-l (or one-hot) encoding [21, 97].

[1] The computational time of deterministic subspace methods like FPOF [57] and Comprex [7] is prohibitive for high-dimensional data, and they run out of memory or cannot output the results for most of the used data sets within four weeks. Also, the empirical results in [91] show that CBRW significantly outperforms these methods. Thus, we focus on the comparison with CBRW and the other four competitors.

Findings - POP Performing Significantly Better Than Five State-of-the-art Outlier Detectors on Real-world High-dimensional Data

The AUC performance of POP and its five competitors: CBRW, ZERO, iForest, ABOD and LOF is reported in Table 5.2. POP performs better than all its five competitors on nine data sets, and significantly outperforms them at the 95% confidence level. On average, POP obtains more than 10%, 18%, 26%, 25% and 39% improvement over CBRW, ZERO, iForest, ABOD and LOF, respectively.

The data indicators $\kappa_{rel}(\mathcal{V})$ and$\kappa_{rel}(\mathcal{U})$ describe the coupling strength of the outlier class with the values in $\mathcal{V}$ and the values in $\mathcal{U}$, respectively. $\kappa_{livc} = \frac{\kappa_{rel}(\mathcal{U})-\kappa_{rel}(\mathcal{V})}{\kappa_{rel}(\mathcal{V})}$ therefore captures the level of *irrelevant value couplings* composed by the intersection of irrelevant value sets and the full value set. κ_{livc} is a fine-grained value-level indicator which also implies the amount of irrelevant features per data. Higher κ_{livc} indicates a larger percentage of irrelevant features a data set may contain. κ_{livc} is used below to further explore the performance of these six detectors in data sets with different levels of irrelevant value couplings (or irrelevant features).

(1) Handling Data Sets with High κ_{livc}. POP obtains the best performance on all the eight data sets with high κ_{livc} (e.g. $\kappa_{livc} > 90\%$) (i.e., *w7a*, *wap.wc*, *R8*, *CAL16*, *AD*, *CAL28* , *CelebA* and *PCMAC*), and it averagely achieves substantial AUC improvement over its five competitors CBRW, ZERO, iForest, ABOD, and LOF by more than 13%, 21%, 30%, 24%, and 66%, respectively.

Table 5.2: AUC Performance of POP, POP$^+$ and Their Competitors: Five Full Space- or Subspace-based Outlier Detectors. CBRW runs out of memory on high-dimensional data *R8* and *WebKB*. ABOD runs out-of-memory on large data *w7a* and *CelebA*.

	Our Methods		Competitors				
Data	POP	POP$^+$	CBRW	ZERO	iForest	ABOD	LOF
w7a	**0.8673**	0.8054	0.6460	0.5375	0.4053	NA	0.4996
wap.wc	**1.0000**	0.9666	0.7900	0.6552	0.5558	0.5243	0.5161
R8	**0.9479**	0.9324	NA	0.8827	0.8443	0.7856	0.8916
CAL16	0.9928	**0.9930**	0.9925	0.9878	0.9742	0.9766	0.3881
AD	**0.9290**	0.8300	0.7348	0.7062	0.7084	0.7023	0.5507
CAL28	0.9608	**0.9616**	0.9599	0.9538	0.9377	0.9268	0.4390
CelebA	0.8968	**0.8981**	0.8462	0.7595	0.6797	NA	0.4726
PCMAC	**0.6935**	0.6617	0.6332	0.5266	0.4767	0.4903	0.6198
BASEHOCK	0.6521	0.6329	0.6177	0.5287	0.4731	0.4883	**0.6639**
WebKB	0.7306	0.7266	NA	0.6950	0.6773	0.6701	**0.8250**
RELA	**0.7449**	0.7173	0.7014	0.6047	0.5578	0.5685	0.7432
Arrhy	0.6762	0.6890	**0.6910**	0.6644	0.6868	0.5948	0.6008
Average (Top-8)	**0.9110**	0.8811	0.8004	0.7512	0.6978	0.7343	0.5472
Average (All)	**0.8410**	0.8179	0.7613	0.7085	0.6648	0.6728	0.6009
P-value	**-**	0.0269	0.0098	0.0005	0.0010	0.0020	0.0122

The superiority of POP is mainly because POP computes the outlier scores based on only selective (relevant) value interactions, which substantially improves the resilience of POP to irrelevant value couplings. LOF performs poorly on all these data sets due to two major reasons: (i) the severe distance concentration effect caused by the presence of a large amount of irrelevant features and (ii) the heavy dependency on an optimal neighborhood size *MinPts*, which varies substantially in data with different data sizes and data distributions. Compared to LOF, the competitors ABOD, ZERO and iForest are less sensitive to the irrelevant couplings, as they use more robust measures to define outlierness (e.g., angle between data objects) or work on feature subspaces. CBRW models complex value couplings to enlarge the outlier score difference between outlying values and other values, which enables CBRW to obtain significant improvements over the other four competitors. Nevertheless, CBRW still works on the full value couplings, and its performance is significantly downgraded by the irrelevant couplings compared to POP.

It is interesting that the methods like CBRW, ZERO, iForest and ABOD can obtain very good AUC performance in some data sets with many irrelevant couplings, e.g., *CAL16* and *CAL28*. This may be due to their high outlier separability, e.g., *CAL16* with $\kappa_{sep} = 0.9613$ and *CAL28* with $\kappa_{sep} = 0.9780$. In other words, these data sets contain some highly relevant features which, to some extent, enable these methods to address the noise brought by irrelevant features.

(2) Handling Data Sets with Low κ_{livc}. As for the rest of the four data sets with low κ_{livc}, i.e., *BASE*, *WebKB*, *RELA* and *Arrhy*, POP obtains the best performance on one data set, with two close to the best (having the difference in AUC less than 0.02), which is comparable to the best performer LOF. This is understandable since POP may omit some relevant value couplings when data sets have only limited irrelevant couplings, whereas LOF works on the full value interactions and thus captures the relevant couplings better.

It is interesting that all outlier detectors obtain quite small AUC values on these four data sets. This may be because all the four data sets have rather low outlier separability, as shown by the indicator κ_{sep} in Table 5.1, and it is very challenging for learning methods to perform well on data sets without highly relevant features.

5.5.4 Significance of Partial Outlierness Propagation

Experimental Settings

POP is compared with its extreme variant called POP$^+$ which simulates full outlierness propagation by setting $k = 1.0$ to evaluate the significance of partial outlierness propagation in POP. Specifically, POP$^+$ computes value outlierness by $\mathbf{q}_{t+1}(v) = \sum_{u \in \mathcal{V}} \tilde{\mathbf{M}}'(v,u)\mathbf{q}_t(u)$, where $\mathbf{M}'$ is a $|\mathcal{V}| \times |\mathcal{V}|$ full value coupling matrix and $\tilde{\mathbf{M}}'$ is its column-wise normalization. Therefore, POP$^+$ is exactly the same as POP except that it uses the full value set $\mathcal{V}$ rather than the value subset $\mathcal{U}$ in POP.

Findings - POP Using Partial Outlierness Propagation Significantly Outperforming Its Counterpart Using Full Outlierness Propagation

The AUC performance of POP and POP$^+$ is reported in Table 5.2. Although POP uses more than two-thirds less information than POP$^+$, it obtains about 3% improvement over POP$^+$ and significantly outperforms POP$^+$ at the 95% confidence level. POP outperforms POP$^+$ on eight data sets, with the maximal improvement up to 11%, and it performs very comparably to POP$^+$ on the other four data sets.

POP$^+$ works on the original data space which contains much more irrelevant value couplings than the clean data space that POP works on, as indicated by the substantial difference between $\kappa_{rel}(\mathcal{V})$ and $\kappa_{rel}(\mathcal{U})$ in Table 5.1. As a result, even though POP$^+$ is operated on the data space that contains the condensed data space used by POP, its performance is significantly degraded due to two major reasons: (i) its distance concentration effect is more severe and (ii) its full outlierness propagation amplifies irrelevant couplings and makes negative propagation.

Note that although POP$^+$ underperforms POP, it substantially outperforms all the five competitors in Table 5.2. This may explain that the (either partial or full) outlierness propagation mechanism well captures contrasting couplings between outlying-to-outlying values and normal/noisy-to-outlying values and has better capability in handling high-dimensional data than the five competitors.

5.5.5 Significance of Joint Value Selection and Outlier Scoring

Experimental Settings

There are two major ways to replace POP with two independent successive modules: feature selection and outlier detection, which are described as follows.

- The value subset selected by POP can be used to perform feature selection. That is, for each data set, we create a corresponding new data set with a subset of features spanned by the values in the selected value subset. We denote this feature selection method as POFS. The existing outlier detectors can then be performed on the newly created data.

- Alternatively, existing outlier detectors can be combined with previously proposed feature selection methods which are designed for outlier detection. Two of the latest outlying feature selection methods: CBRW_FS (denoted by CBFS) [91] and DSFS [96] are used. CBFS only returns a feature ranking. CBFS is aligned with POFS and selects the top-ranked $|\mathcal{F}'|$ features, where $\mathcal{F}'$ denotes the feature subset selected by POFS. DSFS outputs a feature subset $\mathcal{F}''$ without any parameters.

The five outlier detectors with the same settings described in Section 5.5.3 are used with POFS, CBFS and DSFS to have a comprehensive comparison to POP. This enables us to examine how critical it is for the joint process of value selection and outlier scoring, compared to perform feature/value selection and outlier detection independently.

Findings - Joint Value Selection and Outlier Scoring Enabling POP to Obtain More Than 5% Improvement Over the Best Performer Among All the Successive Combinations of Three Outlying Feature/Value Selection Methods and Five State-of-the-art Outlier Detectors

The AUC performance of POP and all the 15 combinations of the three feature selection methods POFS, CBFS and DSFS and the five detectors CBRW, ZERO, iForest, ABOD and LOF is reported in Table 5.3. The results show that POP significantly outperforms all the 15 combinations and obtains over 5% to 50% improvements.

The POFS or CBFS-empowered CBRW, ZERO, iForest and ABOD substantially improve the AUC performance over its original editions, but they still perform significantly less effectively than POP. This is due to two major reasons: (i) POFS or CBFS selects features independently from the these outlier detectors and thus the selected features are not optimal to these detectors, in contrast to POP in which value selection and value outlierness scoring function are simultaneously optimized; and (ii) POP works on value subsets whereas its competitors operates on feature subsets, so POP captures more fine-grained value interactions than its counterparts. All three feature selection methods do not improve the performance of LOF. This is mainly because LOF needs to re-tune its neighborhood size *MinPts* to obtain desirable performance on the data sets with reduced feature sets due to its sensitivity to the data distribution.

Table 5.3: AUC Results of POP and the Combinations of CBRW, ZERO, iForest and LOF with Three Feature Selection Methods POFS, CBFS and DSFS on the 12 Data Sets. The performance of ABOD using POFS, CBFS or DSFS is similar to that of CBRW, ZERO, and iForest. We therefore omit the results of ABOD to fit the table well.

	POP	CBRW			ZERO			iForest			LOF		
Data	-	POFS	CBFS	DSFS	POFS	CBFS	DSFS	POFS	CBFS	DSFS	POFS	CBFS	DSFS
w7a	**0.8673**	0.8220	0.7738	0.5155	0.7701	0.7885	0.5155	0.5893	0.7674	0.5155	0.5661	0.6108	0.5010
wap.wc	**1.0000**	0.9026	0.8739	0.6387	0.7339	0.7429	0.5395	0.5902	0.6816	0.5121	0.6065	0.7161	0.4856
R8	**0.9479**	NA	NA	0.9249	0.8902	NA	0.8758	0.8370	NA	0.8426	0.8772	NA	0.7252
CAL16	0.9928	0.9930	0.9928	**0.9931**	0.9910	0.9900	0.9903	0.9828	0.9824	0.9811	0.4327	0.4428	0.2923
AD	**0.9290**	0.7845	0.7456	0.7432	0.7547	0.7587	0.7428	0.7345	0.7723	0.7435	0.5760	0.6652	0.5233
CAL28	**0.9608**	0.9603	0.9604	0.9599	0.9566	0.9584	0.9540	0.9488	0.9524	0.9421	0.2247	0.2393	0.3345
CelebA	**0.8968**	0.8901	0.8818	0.8502	0.8519	0.8511	0.7722	0.8038	0.8213	0.6973	0.5644	0.6051	0.5220
PCMAC	**0.6935**	0.6759	0.6678	0.6413	0.5952	0.5793	0.4959	0.5509	0.5425	0.4745	0.6605	0.6574	0.5988
BASE	0.6521	0.6294	0.6558	0.5760	0.5396	0.5897	0.4375	0.5096	0.5417	0.4233	0.6666	**0.6984**	0.6187
WebKB	0.7306	0.7449	NA	0.7251	0.7377	NA	0.6995	0.7292	NA	0.6891	0.4543	NA	**0.8246**
RELA	**0.7449**	0.7256	0.7352	0.6984	0.6580	0.6793	0.5987	0.6268	0.6459	0.5844	0.7141	0.7334	0.6965
Arrhy	**0.6762**	0.6095	0.6527	0.5625	0.6074	0.6540	0.5626	0.6065	0.6543	0.5624	0.6004	0.6230	0.5534
Average	**0.8410**	0.7943	0.7940	0.7357	0.7572	0.7592	0.6820	0.7091	0.7362	0.6640	0.5786	0.5992	0.5563
P-value	**-**	0.0098	0.0117	0.0010	0.0024	0.0020	0.0005	0.0005	0.0020	0.0005	0.0010	0.0098	0.0024

5.5.6 Scalability Test

Experiment Settings

We examine the scalability of POP w.r.t. both of data size and dimensionality.

We use six subsets of the largest data set *CelebA* to test the scalability w.r.t. data size. The smallest data subset contains 6,250 objects, and the sizes of subsequent subsets are increased by a factor of two until the largest subset containing 200,000 objects. All these data subsets contain the same number of features (i.e., 39).

In terms of scalability w.r.t. the number of features, four subsets of the data set with the largest number of features, *R8*, are used. The data subset with the lowest dimensionality contains 1,000 features, and subsequent data sets are created by increasing the dimensionality by a factor of 2, until the data set with highest dimensionality containing 8,000 features. All these four data subsets contain the same number of objects (i.e., 3,974).

Findings - POP Obtaining Good Scalability

As expected, POP is linear to the data size and quadratic to the number of features, as shown in Figure 5.3. In the left panel, POP runs comparably fast to CBRW, iForest and ZERO, and is two to four orders of magnitude faster than LOF and ABOD. In the right panel, POP and CBRW have similar runtime and they run considerably slower than the other four detectors, since both POP and CBRW model complex value interactions while the other four detectors ignore these interactions. Although POP and CBRW runs slower, they obtain significantly better AUC performance than their counterparts, as shown in Tables 5.2 - 5.3.

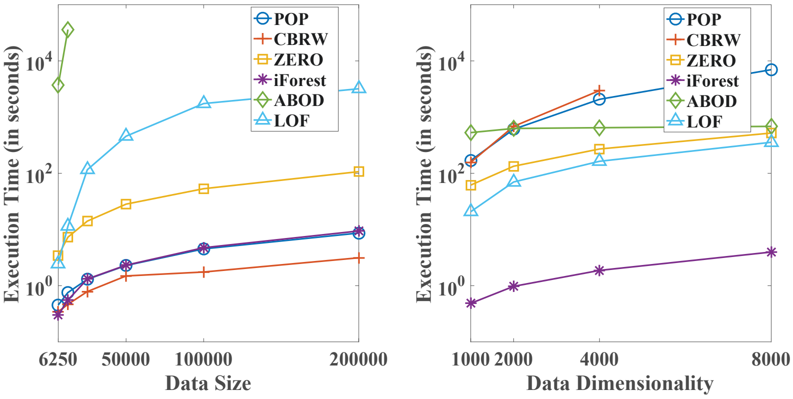


Figure 5.3: Scalability Test Results. ABOD and CBRW run out of memory when the number of objects reaches 25,000 and the number of features reaches 8,000, respectively.

5.5.7 Sensitivity Test

Experimental Settings

We investigate the sensitivity of POP w.r.t. its only parameter k on all the 12 data sets using a wide range of k, i.e., $\{0.1, 0.2, 0.3, 0.4, 0.5\}$.

Findings - POP Performing Stably w.r.t. k

The sensitivity test results of POP are shown in Figure 5.4. POP performs very stably w.r.t. k on all the data sets except *w7a* and *Arrhy* when k is chosen in $\{0.2, 0.3, 0.4\}$. This may be because POP is able to retain stable outlierness of the top-ranked outlying values in the value outlierness vector when the selected value subset mainly contains outlying values. We conjecture that the two data sets *w7a* and *Arrhy* may contain a larger proportion of outlying values, so a larger k is required to have a more effective modeling of the selective value couplings. In general, $k = 0.3$ is recommended in practice.

5.5.8 Convergence Test

Experimental Settings

We examine the ℓ_1-norm convergence, i.e., $\Delta = ||\mathbf{q}_{t+1} - \mathbf{q}_t||_1$, on the 12 data sets.

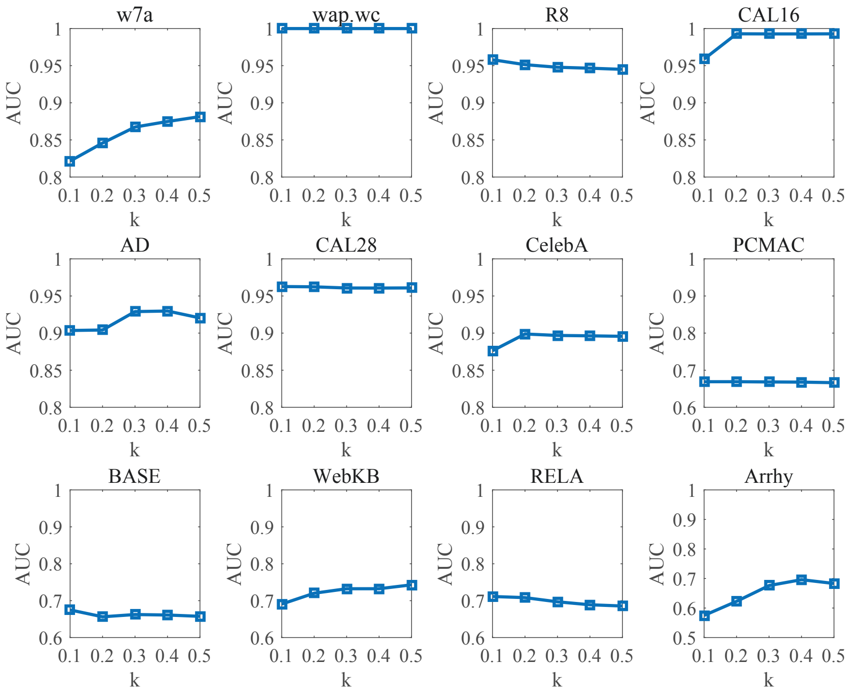


Figure 5.4: Sensitivity Test Results of POP w.r.t. k

Findings - POP Obtaining Rapid Convergence

The convergence test results are presented in Figure 5.5. As expected, POP converges on all the 12 data sets. POP converges within 100 iterations in most of the data sets. POP takes slight longer time to converge in a few data sets, e.g., *w7a*, *BASE*, *WebKB* and *Arrhy*. This may be because these data sets contain larger percentages of outlying values, or they contain many noisy values that behave quite similarly as outlying values. Nevertheless, POP converges within 160 iterations on these data sets.

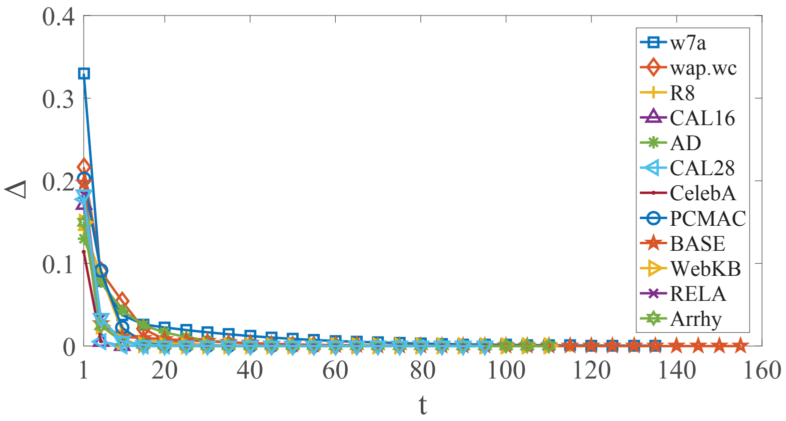


Figure 5.5: Convergence Test Results

5.6 Summary

A novel framework SelectVC is proposed to combine value selection with outlier scoring by iteratively learning selective value couplings to detect outliers in high-dimensional categorical data. SelectVC is further instantiated to a partial outlierness propagation-based method called POP. Our extensive empirical results show that (i) POP performs significantly better than 20 competitors, including five state-of-the-art full space- or subspace-based outlier detectors and their combinations with three outlying feature selection methods, on 12 real-world high-dimensional data with a variety of irrelevant features; (ii) The partial outlierness propagation enables POP to obtain about 3% AUC improvement, while the joint optimization enables POP to gain at least 5% AUC improvement; and (iii) POP obtains good scalability, stable performance w.r.t. the only parameter k and fast convergence rate. These results justify our key insight that modeling only selective value couplings enables us to well contrast outlying behaviors to non-outlying behaviors.

SelectVC and POP explores the binary coupling utility, i.e., focusing on only the couplings that are believed to be relevant to our tasks, which may not be able to fully capture the fine-grained utility of the couplings. One interesting extension to this work is to examine weighted functions to compute different weights for a more reliable modeling of different utilities.

Chapter 6

Binary Cascade of Outlier Factors

6.1 Introduction

In Chapters 4 and 5, we explored iterative methods to estimate the outlierness of values based on the conditional cascade couplings of the values. These methods are sufficiently efficient for single-run outlierness estimation, but they are too computationally costly when we need to repeatedly re-compute the outlierness to adapt to the change of the value graph. In this chapter, we focus on closed-form solutions to leverage complex couplings for value outlierness estimation. The closed-form solutions make use of the underlying intrinsic couplings but do not involve the iterations for outlierness approximation. Thus, they offer much faster yet effective outlierness estimation, which is of great importance in problems that require quick outlierness re-computation w.r.t. any changes of the value graph, e.g., joint outlier detection and feature selection, and streaming outlier detection.

Specifically, instead of working on the directed value graphs in Figure 6.1(a), we define a simplified undirected value graph by focusing on binary interactions between the values, as shown in 6.1(b). The use of the undirected value graph enables us to easily obtain efficient closed-form value outlierness estimation and to capture the binary cascade couplings between the outlier factors of values.

Here we show the significance of the closed-form outlierness estimation in the problem of joint outlier detection and feature selection. As discussed in previous chapters, outlying feature selection is very important for effective outlier detection in data with noisy features, while limited work has been done in this area. Moreover, the methods we introduced in [91, 96] are *filter*-based approaches [75] that select a feature subset independently from subsequent learning methods. Consequently, the relevant features they retain can be noisy w.r.t. subsequent outlier detection methods. In contrast to filter-based approaches, *wrapper*-based approaches choose an optimal feature subset w.r.t. the learning methods [67]. However, although wrapper-based feature selection is popular for classification and clustering [75], as far as we know, no such work has been reported on outlier detection.

This chapter proposes a novel Wrapper-based Outlier Detection framework (WrapperOD) to detect outliers in noisy data. WrapperOD unifies the outlier ranking quality with the feature subset relevance into one objective function, i.e., it measures the relevance of a feature subset by the quality of the outlier ranking produced in the feature subset, and

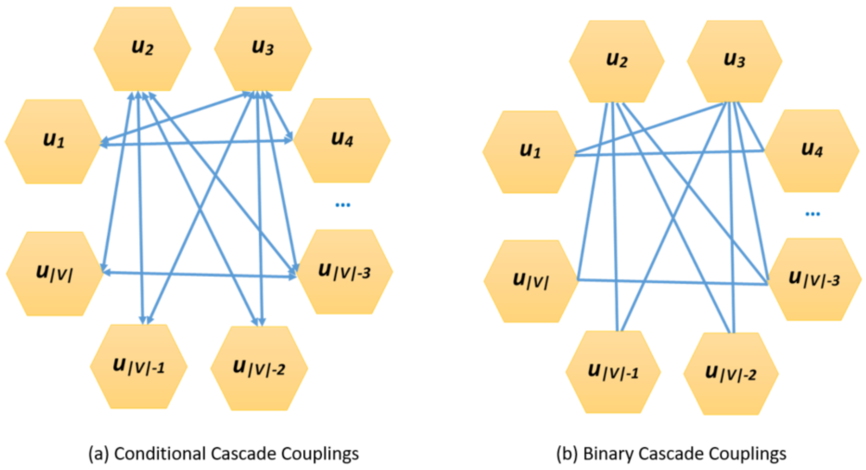

u2
u3
u1
u4
...
u|V|
u|V|-3
u|V|-1
u|V|-2
(a) Conditional Cascade Couplings
(b) Binary Cascade Couplings

Figure 6.1: Comparison of Binary Cascade Couplings and the Conditional Cascade Couplings Presented in Chapter 4.

makes a joint optimization. One big challenge lies in the efficiency of the optimization, since the number of possible feature subsets can be very large. Another challenge is how to properly evaluate the quality of the outlier ranking in an unsupervised way.

To address these two challenges, we instantiate WrapperOD to a Homophily cOupling-based oUtlieR detection method, called HOUR, for categorical data which has been insufficiently explored. HOUR first constructs a value graph with binary edge relations as shown in Figure 6.1(b) and then defines an efficient closed-form outlier scoring function based on the value graph. It further specifies the outlier ranking evaluation function to guide the joint optimization by maximizing the margin between the top-ranked k objects and the other objects.

Accordingly, this work makes the following two main contributions.

- We propose a novel WrapperOD framework to identify outliers in noisy data. In contrast to existing solutions that search feature subset(s) independently from outlier scoring, WrapperOD simultaneously optimizes its outlier scoring and feature selection, which enables its outlier scoring function to produce a much more reliable outlier ranking in noisy data.

- The performance of WrapperOD is verified by an instance HOUR. HOUR models homophily couplings between outlying behaviors to construct a noise-resilient outlier scoring function that empowers the joint optimization in WrapperOD. HOUR is built upon an efficient closed-form outlier scoring solution, which well guarantees the efficiency of the joint optimization.

Extensive experiments show that HOUR (i) significantly outperforms three state-of-the-art outlier detectors and their combination with two of the latest outlying feature selection methods in terms of AUC and/or $P@n$ on 15 real-world data sets with a diverse range of noise levels; (ii) performs stably w.r.t. k in most cases; and (iii) obtains good scalability: it is linear to data size and quadratic to the number of features.

In the rest of this chapter, WrapperOD is detailed in Section 6.2. HOUR is introduced in Section 6.3, followed by a theoretical analysis in Section 6.4. Empirical results are provided in Section 6.5. We conclude this chapter in Section 6.6.

6.2 The Proposed WrapperOD Framework

WrapperOD aims to perform outlier detection and feature selection simultaneously using a wrapper-based approach. The procedure of WrapperOD is presented in Figure 6.2. WrapperOD first defines an outlier scoring function $\phi_{\mathcal{S}}$ to compute object outlierness in a given feature subset $\mathcal{S} \subseteq \mathcal{F}$ and then sorts the objects based on their outlierness to obtain an outlier ranking $\mathbf{r}_{\phi_{\mathcal{S}}}$. WrapperOD further defines an outlier ranking evaluation function J to compute the quality of $\mathbf{r}_{\phi_{\mathcal{S}}}$ and uses this ranking quality as the relevance indicator of the subset $\mathcal{S}$. This means that the task of finding the best feature subset is equivalent to finding the best outlier ranking. WrapperOD iteratively performs function $\phi_{\mathcal{S}}$ and function J to obtain the best feature subset $\mathcal{S}^*$ and outlier ranking $\mathbf{r}^*_{\phi_{\mathcal{S}^*}}$. Overall, the problem can be formally stated as:

$$\arg\max_{\mathcal{S}} J(\mathbf{r}_{\phi_{\mathcal{S}}}). \tag{6.1}$$

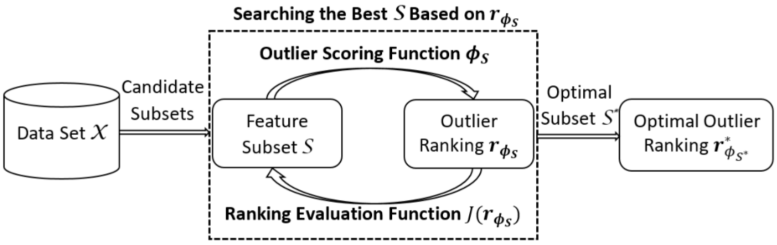

Searching the Best S Based on r_{ϕ_S}
Outlier Scoring Function ϕ_S
Data Set $\mathcal{X}$
Candidate Subsets
Feature Subset S
Outlier Ranking r_{ϕ_S}
Optimal Subset S^*
Optimal Outlier Ranking $r^*_{\phi_{S^*}}$
Ranking Evaluation Function $J(r_{\phi_S})$

Figure 6.2: The Proposed WrapperOD Framework

WrapperOD is very different from existing outlier detection and outlying feature selection frameworks in that: WrapperOD unifies the two correlated tasks, outlier detection and outlying feature selection, to simultaneously obtain the optimal outlier ranking and feature subset, while existing solutions treat these two tasks independently and are very sensitive to noisy features.

6.2.1 Fast Outlier Scoring Function

The scoring function can be defined as $\phi_{\mathcal{S}} : \mathcal{X}_{\mathcal{S}} \mapsto \mathbb{R}$, where $\mathcal{X}_{\mathcal{S}} \in \mathbb{R}^{N\times|\mathcal{S}|}$. That is, $\phi_{\mathcal{S}}$ computes the outlier scores of data objects in the feature subset $\mathcal{S}$ and outputs an outlier ranking $\mathbf{r} \in \mathbb{R}^N$. In general, $\phi_{\mathcal{S}}$ has to meet at least the two requirements: (i) being sufficiently resilient to noisy features, and it may opt for noisy features other than relevant features otherwise; and (ii) being very efficient as it will be repeatedly performed to evaluate a large number of feature subsets.

6.2.2 Outlier Ranking Evaluation

The J function takes the outlier ranking $\mathbf{r} \in \mathbb{R}^N$ as input and outputs a scalar to indicate the quality of the ranking $\mathbf{r}$. J is essentially an internal evaluation measure for a given outlier ranking, i.e., evaluating outlier rankings without class labels. Internal evaluation measures have been extensively studied for clustering tasks, while very little work has been done on outlier detection [81]. One related work is [81], which uses pseudo binary classification to evaluate the ranking quality. However, this method has $O(N^3)$ time complexity, which is computationally prohibitive to use here.

6.2.3 Feature Subset Generation

The last component is the feature subset generation that determines the feature subset the ϕ and J functions work on. Feature subset search methods, including complete search, sequential search, and random search [75], can be used to generate the feature subset $\mathcal{S}$. Although complete search outputs an optimal subset, it has exponential time complexity. Sequential search and random search may produce a suboptimal subset, but they are more practical than complete search as they run substantially faster.

6.3 A WrapperOD Instance: HOUR

We further instantiate WrapperOD for categorical data by proposing HOUR. HOUR specifies its three components by a homophily coupling-based outlier scoring function $\phi_{\mathcal{S}}$, a score margin-based outlier ranking evaluation function J, and a heuristic feature subset search method.

6.3.1 Specifying $\phi_{\mathcal{S}}$ with Homophily Couplings

Most outlier detectors are sensitive to noisy features and/or are computationally costly. HOUR exploits the homophily couplings between feature values to construct a fast and robust function $\phi_{\mathcal{S}}$. Let $dom(\mathsf{F}) = \{v_1, v_2, \cdots\}$ be the domain of a feature $\mathsf{F} \in \mathcal{S}$, which consists of a finite set of unordered feature values, and $\mathcal{V}$ be the whole set of feature values in $\mathcal{S}$: $\mathcal{V} = \cup_{\mathsf{F}\in\mathcal{S}} dom(\mathsf{F})$, where $dom(\mathsf{F}) \cap dom(\mathsf{F}') = \emptyset, \forall \mathsf{F} \neq \mathsf{F}'$.

Definition 6.1 (Outlierness Influence)**.** *The outlierness influence of a feature value $v \in \mathcal{V}$ is defined as follows.*

$$\tau(v) = \frac{\sum_{u\in\mathcal{N}_v} \delta(v)\delta(u)}{\sum_{v\in\mathcal{V}} \sum_{u\in\mathcal{N}_v} \delta(v)\delta(u)}, \tag{6.2}$$

where $\mathcal{N}_v$ denotes a set of values that co-occur with v and $\delta(\cdot) : \mathcal{V} \mapsto (0, 1)$ is an initial outlierness influence estimation of a value based on intra-feature frequency distribution.

Similar to the δ function in Section 4.3.1, we use $\delta(v) = \frac{1}{2}\left(\frac{freq(m)-freq(v)}{freq(m)} + \frac{1}{freq(m)}\right)$, $\forall v \in dom(\mathsf{F})$, m is a value that occurs most frequently in F (i.e., the mode) and $freq(\cdot)$ is a frequency counting function. Such mode absolute deviation helps $\delta(\cdot)$ address features with imbalanced frequency distributions.

Essentially, δ estimates the outlierness influence independently from the values of other features. τ further utilizes the binary homophily couplings between values from different features to have a better estimation of outlierness influence. This outlierness influence is then used to infer the value outlierness based on the coupling strength between feature values.

Definition 6.2 (Value Outlierness)**.** *The outlierness of a feature value $v \in \mathcal{V}$ is defined as follows.*

$$\psi(v) = \sum_{u \in \mathcal{N}_v} \rho(u, v)\tau(u), \tag{6.3}$$

where $\rho(u, v) = \log \frac{P(u,v)}{P(u)P(v)}$ is pointwise mutual information to measure the coupling strength between two values.

Similar to CBRW in Chapter 4, given an object $\mathbf{x}_i$, its outlierness is defined as a weighted product of value outlierness.

$$\phi_{\mathcal{S}}(\mathbf{x}_i) = 1 - \prod_{\mathsf{F}_j \in \mathcal{S}} [1 - \psi(x_{ij})]^{\omega_j}, \tag{6.4}$$

where $\omega_j = 1 - \prod_{v \in dom(\mathsf{F}_j)} [1 - \psi(v)]$ computes the weight of F_j.

Section 6.4.1 will discuss how this outlier scoring models the homophily couplings and why it is fast and noise-resilient.

6.3.2 Specifying J with Average Score Margin

We introduce a score margin-based outlier ranking evaluation measure. The measure has a linear time complexity, which helps guarantee the efficiency of the joint optimization.

$$J(\mathbf{r}_{\phi_{\mathcal{S}}}, k) = \frac{\Delta_{\mathcal{S}}}{|\mathcal{S}|} = \frac{1}{k|\mathcal{S}|} \sum_{\boldsymbol{x} \in \mathcal{O}} [\phi_{\mathcal{S}}(\boldsymbol{x}) - \phi_{\mathcal{S}}(\boldsymbol{x}')], \tag{6.5}$$

where $\mathcal{O}$ is a set of top-ranked k objects and $\phi_{\mathcal{S}}(\boldsymbol{x}')$ is the median outlierness in the remaining objects. $\Delta_{\mathcal{S}} = \frac{1}{k} \sum_{\boldsymbol{x} \in \mathcal{O}} [\phi_{\mathcal{S}}(\boldsymbol{x}) - \phi_{\mathcal{S}}(\boldsymbol{x}')]$ is the average score margin between the top-k objects and the center of the other objects, which also indicates the relevance of feature subset $\mathcal{S}$. So maximizing J finds an outlier ranking that jointly maximizes the object outlierness margin and the feature subset relevance.

6.3.3 Recursive Search of Feature Subset $\mathcal{S}$

A sequential search method, namely Recursive Backward Elimination (RBE), is used with the functions $\phi_{\mathcal{S}}$ and J to search for an approximately best subset. As shown in Algorithm 6.1, RBE recursively eliminates one feature at a time until no feature remains, and only retains the feature subset that results in the largest J. RBE is used because it helps guarantee a 2-approximate J to the optimal one (see Section 6.4.2).

Algorithm 6.1 *RBE ($\mathcal{F}$)*

```
Input: $\mathcal{F}$ - full feature set
Output: $\mathcal{S}$ - the feature subset selected
while $|\mathcal{F}| > 0$ do
    for $\mathsf{F} \in \mathcal{F}$ do
        Compute $J(\mathcal{F} \setminus \mathsf{F})$
    end for
    Remove the feature $\mathsf{F}$ that results in the largest $J(\mathcal{F} \setminus \mathsf{F})$
end while
return Return the subset with the largest $J(\cdot)$ as $\mathcal{S}$
```

6.3.4 The Algorithm and Its Time Complexity

Algorithm 6.2 presents the procedure of HOUR. Steps (1-3) evaluates the outlier ranking in the full feature set, followed by the evaluation of outlier rankings in feature subsets generated by RBE in Steps (4-14).

Algorithm 6.2 *HOUR($\mathcal{X}$, k)*

```
Input: $\mathcal{X}$ - data objects, $k$ - the number of targeted outliers
Output: $\mathbf{r}$ - an outlier ranking of objects, $\mathcal{S}$ - a feature subset
 1: $\psi(v) \leftarrow \sum_{u \in \mathcal{N}_v} \rho(u,v)\tau(u), \forall v \in \mathcal{V}$
 2: Compute $\phi_{\mathcal{F}}(\boldsymbol{x}), \forall \boldsymbol{x} \in \mathcal{X}$
 3: $\theta \leftarrow J(R_{\phi_{\mathcal{F}}}, k)$
 4: while $|\mathcal{F}| > 0$ do
 5:     for $i = 1$ to $|\mathcal{F}|$ do
 6:         Compute $\phi_{\mathcal{F} \setminus \mathsf{F}_i}(\boldsymbol{x}), \forall \boldsymbol{x} \in \mathcal{X}$
 7:         Compute $J_i(\mathbf{r}'_{\phi_{\mathcal{F}}}, k)$
 8:     end for
 9:     Find feature $\mathsf{F}_i$ with the largest $J_i(\mathbf{r}'_{\phi_{\mathcal{F}}}, k)$
10:     $\mathcal{F} \leftarrow \mathcal{F} \setminus \mathsf{F}_i$ and update $\psi(v)$ for all $v$ contained in $\mathcal{F}$
11:     if $J_i(\mathbf{r}'_{\phi_{\mathcal{F}}}, k) \geq \theta$ then
12:         $\mathbf{r} \leftarrow \mathbf{r}'$, $\mathcal{S} \leftarrow \mathcal{F}$ and $\theta \leftarrow J_i(\mathbf{r}'_{\phi_{\mathcal{F}}}, k)$
13:     end if
14: end while
15: return $\mathbf{r}$ and $\mathcal{S}$
```

Steps (1-2) require one database scan to perform ψ and $\phi_{\mathcal{S}}$ respectively, which is linear w.r.t. N. Step (3) needs to rank $\mathcal{X}$, which has $O(N \log N)$ in the worst case, and thus they have $O(N \log N)$. The two loops in Steps (4-14) result in $O(D^2)$ in the worst case, and the core computation within the loops performs outlier scoring and ranking, which has the same time complexity as the first three steps. Hence, the worst time complexity of HOUR is $O(D^2 N \log N)$.

6.4 Theoretical Analysis

6.4.1 Robustness w.r.t. Noisy Features

We analyze the robustness of HOUR from the value level to the feature level. At the value level, as per the definition of outliers, *outlying values* are infrequent values contained by outliers, while *noisy values* are also infrequent but contained by normal objects. In contrast, *normal values* are frequent values contained by both outliers and normal objects. In the following, we discuss how the outlier scoring function in HOUR can efficiently distinguish outlying values from normal and noisy values.

Theorem 6.1 (Closed-form Homophily Modeling). *The value influence estimation $\tau(v)$ in Eqn.(6.1) is equivalent to the stationary probability of visiting v in random walks on a strongly connected undirected value-value graph $\mathsf{G} =< \mathcal{V}, \mathcal{E}, \eta(\cdot,\cdot) >$, where a feature value v represents a graph node, $e(u,v) \in \mathcal{E}$ denotes an edge between two nodes u and v, and $\eta(u,v) = \delta(u)\mathbf{I}(u,v)\delta(v)$ ($\mathbf{I}(u,v) = 1$ if u and v have occurrences, and $\mathbf{I}(u,v) = 0$ otherwise) is the weight of edge $e(u,v)$, $\forall u, v \in \mathcal{V}$.*

Proof. Let $\pi^*(v)$ be the stationary probability, $P(u,v)$ be the transition probability from u to v, $d(v) = \sum_{u\in\mathcal{N}_v} \eta(v,u)$ be the weighted degree of v and $vol(G) = \sum_{v\in\mathcal{V}} d(v) = \sum_{v\in\mathcal{V}} \sum_{u\in\mathcal{N}_v} \delta(v)\delta(u)$ be the graph volume. Then we have:

$$\pi^*(v) = \sum_{u\in\mathcal{V}} \pi^*(u)P(u,v) = \sum_{u\in\mathcal{V}} \frac{d(u)}{vol(G)} \frac{\delta(u)\mathbf{I}(u,v)\delta(v)}{d(u)},$$

and we obtain:

$$\pi^*(v) = \frac{\sum_{u\in\mathcal{N}_v} \delta(v)\delta(u)}{\sum_{v\in\mathcal{V}} \sum_{u\in\mathcal{N}_v} \delta(v)\delta(u)} = \tau(v),$$

which completes the proof. □

Theorem 6.1 indicates that given $\forall u, v \in \mathcal{V}$, if value u has lower frequency and stronger couplings with other infrequent values compared to value v, i.e., $d(u) > d(v)$, then $\tau(u) > \tau(v)$. This essentially models the binary homophily couplings between outlying values. However, this homophily coupling modeling does not take account of the coupling strength between values. We further enhance the modeling by adding pointwise mutual information in Eqn. (6.3). There exist other ways to model homophily couplings. We use such a two-stage modeling because it has a closed-form solution which guarantees the efficiency of outlier scoring.

Outlying Values vs. Noisy Values. Noisy values have similarly low frequencies as outlying values, but they are supposed to co-occur randomly or follow a Gaussian distribution. Their homophily couplings are therefore weaker than those of outlying values. As a result, HOUR assigns smaller outlierness ψ to noisy values than outlying values.

Outlying Values vs. Normal Values. Normal values have much lower δ and τ than outlying values due to their high occurrence frequencies. Their high frequencies also result in weak couplings with infrequent values. As a result, they obtain substantially smaller outlierness ψ than outlying values.

At the feature level, HOUR prefers features that contain values of higher outlierness to maximize its objective function. Since outlying values have higher outlierness than normal or noisy values, the features HOUR iteratively eliminates are those containing normal and/or noisy values, resulting in a cleaned feature subset for its outlier scoring function.

6.4.2 Theoretical Bound

This section shows that HOUR is guaranteed to obtain an outlier ranking with the margin of at least half of the optimum value, provided that features are dependent on each other as in the homophily coupling modeling.

Theorem 6.2 (2-Approximation)**.** *Let* $\mathbf{r}$ *and* $\mathcal{S}$ *be the outlier ranking and feature subset returned by HOUR. Assume* Θ_{F} *be the contribution of feature* $\mathsf{F} \in \mathcal{S}$ *to the outlier ranking* $\mathbf{r}$ *by integrating its conjunctive functions with other features* $\theta(\mathsf{F} \wedge \mathsf{F}')$*, i.e.,* $\Theta_{\mathsf{F}} = \sum_{\mathsf{F}' \in \mathcal{S}} \theta(\mathsf{F} \wedge \mathsf{F}')$*, and* $\Delta_{\mathcal{S}} = \frac{1}{2} \sum_{\mathsf{F} \in \mathcal{S}} \Theta_{\mathsf{F}}$*. Then we have* $J(\mathbf{r}_{\phi_{\mathcal{S}}}, k) \geq \frac{1}{2} J_{opt}$*, where* J_{opt} *is the optimum value of* J*.*

Proof. Since J_{opt} is the optimum value of J, we have

$$J_{opt} = \frac{\Delta_{\mathcal{S}^*}}{|\mathcal{S}^*|} \geq \frac{\Delta_{\mathcal{S}^*} - \Theta_{\mathsf{F}}}{|\mathcal{S}^*| - 1}, \ \forall \mathsf{F} \in \mathcal{S}^*.$$

We obtain $\Theta_{\mathsf{F}} \geq J_{opt}$ after some replacements. Let $\mathsf{F} \in \mathcal{S}^*$ be the feature that HOUR removes first among those contained in $\mathcal{S}^*$ during the iteration of RBE and $\mathcal{T}$ be the feature set before F is removed, i.e., $\mathcal{S}^* \subset \mathcal{T}$. Since HOUR removes the least contributive feature at a time, we have $\Theta_{\mathsf{F}'} \geq \Theta_{\mathsf{F}}$,$\forall \mathsf{F}' \in \mathcal{T}$ when HOUR chooses to remove F, and thus $\Theta_{\mathsf{F}'} \geq J_{opt}$. As a result, we obtain $\sum_{\mathsf{F}' \in \mathcal{T}} \Theta_{\mathsf{F}'} \geq J_{opt}|\mathcal{T}|$, and thus $2\Delta_{\mathcal{T}} \geq J_{opt}|\mathcal{T}|$, resulting in $J(\mathbf{r}'_{\phi_{\mathcal{T}}}, k) = \frac{\Delta_{\mathcal{T}}}{|\mathcal{T}|} \geq \frac{J_{opt}}{2}$. Since HOUR retains $\mathcal{S}$ that results in the largest J and $\mathcal{T}$ is one of the candidates, we finally obtain $J(\mathbf{r}_{\phi_{\mathcal{S}}}, k) \geq J(\mathbf{r}'_{\phi_{\mathcal{T}}}, k) \geq \frac{J_{opt}}{2}$. □

6.5 Experiments and Evaluation

6.5.1 Data Sets

Fifteen real-world data sets are used, which cover diverse domains, e.g., bank marketing, image object recognition, network intrusion, and credit card fraud detection, as shown in Table 6.1. Most of the data sets are used in our previous chapters. We add some new data sets, including *SylvaA*, *SylvaP*, *CUP14*, *Alcohol*, *Turkiye*, and *Credit*, to examine the detection performance on data sets with a wide range of feature irrelevancy and outlier separability, which are respectively measured by the two data indicators, κ_{fnl} and κ_{sep} (see Section 2.3.3 for their definitions). These new data sets are transformed into outlier detection data sets using the rare class conversion method presented in Section 2.3.1.

6.5.2 Experiment Environment

HOUR is evaluated against three representative outlier detectors for categorical data: FPOF [57], CompreX [7] and CBRW [91]. FPOF is chosen because it is the most popular pattern-based method. CompreX is a state-of-the-art subspace method that captures arbitrary-length outlying behaviors. CBRW is a closely related value outlierness-based method. k in HOUR is set to the number of outliers by default. CompreX is parameter-free. FPOF and CBRW are used with their default settings.

We also compare HOUR to the combination of outlier detectors with two of our recently proposed outlying feature selection methods, CBFS [91] and DSFS [96]. CBFS returns a feature ranking. DSFS outputs a feature subset without any parameters. To have a fair comparison, CBFS selects the top-ranked $|\mathcal{S}|$ features so that CBFS and HOUR select the same number of features.

All methods are in Java in WEKA [52] except CompreX which is in MATLAB. All these methods are executed on a node in a 3.4GHz Phoenix Cluster with 32GB memory.

In terms of performance evaluation, the precision at n, i.e., $P@n$ (where we set n as the number of outliers in a data set), is used to evaluate the ability of the outlierness margin-based optimization objective in ranking outliers in the top positions, in addition to the AUC performance presented in Section 2.3.2. Higher $P@n$ indicates better performance.

6.5.3 Effectiveness in Real-world Data Sets

Obtaining Significantly Better Global or Top-n Outlier Ranking Than Other Outlier Detectors

We compare HOUR with CBRW, CompreX and FPOF in terms of AUC and $P@n$ in Table 6.1. In terms of AUC, HOUR obtains the best performance on 11 data sets; and on average, it obtains about 2%, 7% and 21% improvement over CBRW, CompreX and FPOF, respectively. HOUR significantly outperforms FPOF in AUC. In terms of $P@n$, HOUR performs significantly better than CBRW and CompreX and obtains more than 30%, 37% and 90% improvements over CBRW, CompreX and FPOF, respectively.

Table 6.1: AUC and $P@n$ Performance on 15 Data Sets. Data is sorted by κ_{fnl}. '∇' indicates the feature reduction rate of HOUR. FPOF runs out of memory in four high-dimensional data sets.

					AUC				$P@n$			
Data	N	$\|\mathcal{F}\|$	$\|\mathcal{S}\|(\nabla)$	κ_{fnl}	HOUR	CBRW	CompreX	FPOF	HOUR	CBRW	CompreX	FPOF
SylvaA	14,395	172	16(91%)	91%	**0.9829**	0.9353	0.8855	NA	**0.7483**	0.5914	0.3770	NA
BM	41,188	10	5(50%)	90%	**0.6939**	0.6287	0.6267	0.5466	**0.3265**	0.2474	0.2565	0.1369
AID362	4,279	114	8(93%)	86%	0.5147	**0.6640**	0.6480	NA	**0.0833**	0.0500	0.0167	NA
APAS	12,695	64	13(80%)	81%	**0.9065**	0.8190	0.6554	NA	0.0000	0.0000	0.0000	NA
SylvaP	14,395	87	15(83%)	78%	**0.9725**	0.9715	0.9537	NA	**0.6907**	0.6151	0.5700	NA
Census	299,285	33	3(91%)	58%	0.4867	**0.6678**	0.6352	0.6148	0.0616	**0.0677**	0.0675	0.0637
CelebA	202,599	39	12(69%)	49%	**0.8879**	0.8462	0.7572	0.7380	**0.2085**	0.1748	0.1533	0.1256
CUP14	619,326	7	3(57%)	43%	**0.9833**	0.9420	0.9398	0.6041	**0.6730**	0.2671	0.2671	0.0000
Alcohol	1,044	32	3(91%)	38%	**0.9365**	0.9254	0.8919	0.5468	**0.3889**	0.3333	**0.3889**	0.0556
CMC	1,473	8	4(50%)	38%	**0.6647**	0.6339	0.5669	0.5614	0.0345	0.0345	0.0345	**0.1034**
CT	581,012	44	3(93%)	34%	0.9688	0.9703	**0.9772**	0.9770	0.0499	0.0386	**0.0688**	0.0644
Chess	28,056	6	3(50%)	33%	**0.8507**	0.7897	0.6387	0.6160	0.0000	0.0000	0.0000	0.0000
Turkiye	5,820	32	21(34%)	25%	**0.5256**	0.5116	0.5101	0.4746	**0.0776**	0.0746	0.0687	0.0597
Credit	30,000	9	6(33%)	11%	**0.7204**	0.5804	0.6543	0.6428	**0.4875**	0.2215	0.3502	0.3333
Probe	64,759	6	2(67%)	0%	0.9661	**0.9906**	0.9790	0.9867	0.8440	**0.8579**	0.7928	0.8548
Average	128,022	44	8(69%)	50%	0.8041	0.7918	0.7546	0.6644	0.3116	0.2383	0.2275	0.1634
				p-value		0.1876	0.0730	0.0322		0.0068	0.0068	0.1055

Using outlier scoring results to guide outlying feature selection enables HOUR to remove most, if not all, of the noisy features while having little or no loss in outlier separability on most data sets, e.g., the 11 data sets on which HOUR obtains the best AUC performance (see the κ_{fnl} and κ_{sep} results of HOUR in Table 6.3). Hence, although HOUR works with 69% less features than its counterparts, it performs substantially better as it works on much cleaner data. Also, maximizing the margin of the top-k objects from the others helps rank more outliers in the top, resulting in significant improvement in $P@n$.

On the other hand, HOUR opts for strongly relevant features that help rank outliers in the top, so it may remove weakly relevant features that distinguish outliers from normal objects in other positions. As a result, HOUR may obtain worse AUC performance while comparable $P@n$ compared to its counterparts, e.g., the results on *AID362* and *Census*.

Outperforming the State-of-the-art Outlying Feature Selection Methods

HOUR is compared with a combination of CBRW and CompreX with outlying feature selection methods CBFS and DSFS in Table 6.2. The results show that, although the two feature selection methods largely improve CBRW and CompreX in terms of AUC and/or $P@n$, HOUR remains the best performer on most data sets. HOUR obtains significantly better performance than the combination of CBRW and CompreX with CBFS (i.e., CBRW† and CompreX† in Table 6.2) in AUC and significantly outperforms all the four different combinations in $P@n$.

Table 6.2: AUC and $P@n$ Performance Comparison between HOUR and the Combination of CBRW and CompreX with CBFS (Denoted by †) and DSFS (Denoted by ‡).

	AUC					$P@n$				
Data	HOUR	CBRW†	CBRW‡	CompreX†	CompreX‡	HOUR	CBRW†	CBRW‡	CompreX†	CompreX‡
SylvaA	**0.9829**	0.8793	0.9381	0.8726	0.8858	**0.7483**	0.5327	0.5948	0.4831	0.3781
BM	**0.6939**	0.6104	0.6114	0.6239	0.6239	**0.3265**	0.2259	0.2269	0.2567	0.2575
AID362	0.5147	0.4659	**0.6518**	0.4982	0.6342	**0.0833**	0.0000	0.0500	0.0000	0.0167
APAS	**0.9065**	0.6621	0.8807	0.6532	0.8771	0.0000	0.0000	0.0000	0.0000	0.0000
SylvaP	**0.9725**	0.9582	0.9707	0.9307	0.9628	**0.6907**	0.5553	0.5609	0.6140	0.5892
Census	0.4867	0.4844	0.6999	0.4841	**0.7135**	0.0616	0.0604	0.0732	0.0635	**0.0991**
CelebA	**0.8879**	0.8865	0.8502	0.8855	0.7594	0.2085	0.2098	0.1698	**0.2142**	0.1482
CUP14	**0.9833**	0.9821	0.9358	0.9821	0.9618	**0.6730**	0.6686	0.2671	0.6686	0.3224
Alcohol	**0.9365**	0.9264	0.9294	0.8919	0.8595	0.3889	0.3889	**0.4444**	0.3889	0.0556
CMC	**0.6647**	0.6366	0.6444	0.6475	0.6586	0.0345	0.0345	0.0345	0.0345	0.0345
CT	**0.9688**	0.9192	0.9673	0.9187	0.9670	**0.0499**	0.0000	0.0386	0.0000	0.0386
Chess	**0.8507**	0.7268	0.7649	0.7529	0.6305	0.0000	0.0000	0.0000	0.0000	0.0000
Turkiye	**0.5256**	0.5161	0.5108	0.5145	0.5119	**0.0776**	0.0716	0.0716	0.0746	**0.0776**
Credit	**0.7204**	0.5712	0.5712	0.6566	0.6566	**0.4875**	0.2131	0.2131	0.3531	0.3531
Probe	0.9661	0.9591	0.9591	**0.9794**	**0.9794**	**0.8440**	0.8397	0.8397	0.7672	0.7672
Average	0.8041	0.7456	0.7924	0.7528	0.7788	0.3116	0.2533	0.2390	0.2612	0.2092
p-value		0.0001	0.0730	0.0006	0.1070		0.0029	0.0269	0.0098	0.0029

The superiority of HOUR is because the wrapper-based feature selection scheme enables HOUR to remove substantially more truly noisy features than the filter-based methods CBFS and DSFS. This is verified by the κ_{fnl} and κ_{sep} differences between the full feature set and feature subsets selected by HOUR, CBFS and DSFS shown in Table 6.3. On average, HOUR removes over 57% of the noisy features, which is about triple and double more than that of CBFS (17%) and DSFS (32%), respectively; while at the same time, it obtains a very comparable outlier separability. In addition, we observe that filter-based methods like DSFS generally retain many more features than HOUR. These extra features contain noisy features as well as relevant features. This is why DSFS obtains a smaller noise reduction level but a better outlier separability than HOUR in Table 6.3. The extra relevant features retained by DSFS enable CBRW and CompreX to outperform HOUR in data sets where HOUR makes very aggressive feature reduction, e.g., on *AID362* and *Census*.

Table 6.3: Data Complexity Evaluation Results on $\mathcal{F}$, $\mathcal{S}$, $\mathcal{S}'$ and $\mathcal{S}''$. $\mathcal{F}$ is the original feature set. $\mathcal{S}$, $\mathcal{S}'$ and $\mathcal{S}''$ are feature subsets retained by HOUR, CBFS and DSFS, respectively.

	Feature Noise Level (κ_{fnl})				**Outlier Separability** (κ_{sep})			
Data	$\mathcal{F}$	$\mathcal{S}$ (∇)	$\mathcal{S}'$ (∇)	$\mathcal{S}''$ (∇)	$\mathcal{F}$	$\mathcal{S}$ (∇)	$\mathcal{S}'$ (∇)	$\mathcal{S}''$ (∇)
SylvaA	91%	13%(86%)	75%(18%)	91%(0%)	0.78	0.78(0%)	0.78(0%)	0.78(0%)
BM	90%	80%(11%)	80%(11%)	75%(17%)	0.63	0.63(0%)	0.63(0%)	0.63(0%)
AID362	86%	100%(-16%)	100%(-16%)	85%(1%)	0.60	0.49(19%)	0.47(23%)	0.60(0%)
APAS	81%	38%(53%)	85%(-4%)	50%(38%)	0.87	0.87(0%)	0.72(18%)	0.87(0%)
SylvaP	78%	0%(100%)	53%(32%)	71%(9%)	0.78	0.78(0%)	0.78(0%)	0.78(0%)
Census	58%	100%(-74%)	100%(-74%)	50%(13%)	0.76	0.49(35%)	0.49(35%)	0.76(0%)
CelebA	49%	0%(100%)	0%(100%)	50%(-3%)	0.80	0.78(2%)	0.78(2%)	0.80(0%)
CUP14	43%	0%(100%)	33%(22%)	50%(-17%)	0.92	0.92(0%)	0.92(0%)	0.92(0%)
Alcohol	38%	0%(100%)	0%(100%)	18%(53%)	0.91	0.91(0%)	0.91(0%)	0.91(0%)
CMC	38%	0%(100%)	0%(100%)	0%(100%)	0.66	0.66(0%)	0.66(0%)	0.66(0%)
CT	34%	0%(100%)	67%(-96%)	0%(100%)	0.97	0.97(0%)	0.97(0%)	0.97(0%)
Chess	33%	33%(0%)	6%(-100%)	25%(25%)	0.74	0.59(19%)	0.74(0%)	0.74(0%)
Turkiye	25%	14%(43%)	14%(43%)	21%(4%)	0.58	0.55(4%)	0.55(4%)	0.55(4%)
Credit	11%	0%(100%)	0%(100%)	0%(100%)	0.70	0.70(0%)	0.70(0%)	0.70(0%)
Probe	0%	0%(NA)	0%(NA)	0%(NA)	0.94	0.94(0%)	0.94(0%)	0.94(0%)
Average	50%	25%(57%)	44%(17%)	39%(32%)	0.78	0.74(5%)	0.74(5%)	0.77(0%)

6.5.4 Sensitivity Test

We examine the stability of HOUR w.r.t. k in Figure 6.3. HOUR shows stable performance in most of the 15 data sets. Here we selectively illustrate representative and interesting trends in its AUC performance w.r.t. a wide range of k on four data sets. HOUR performs very stably on *CelebA* and *CUP14*. It is very challenging to rank outliers in the top-k positions in data sets which contain only a very small proportion of outliers but have many noisy features (e.g., *CT*), as the outliers are easily masked as normal objects in those data. Due to these false negatives, HOUR requires a large k (e.g., 0.5% or 1.0%) to perform well on *CT*. On the other hand, HOUR can identify outliers more accurately using a smaller k in *Census* which contains a larger proportion of outliers, as the use of a large k in HOUR might lead to false positives. A general guideline is to set $k = 0.5\% \times N$ or $k = 1.0\% \times N$ to leverage the effect of false negatives and false positives.

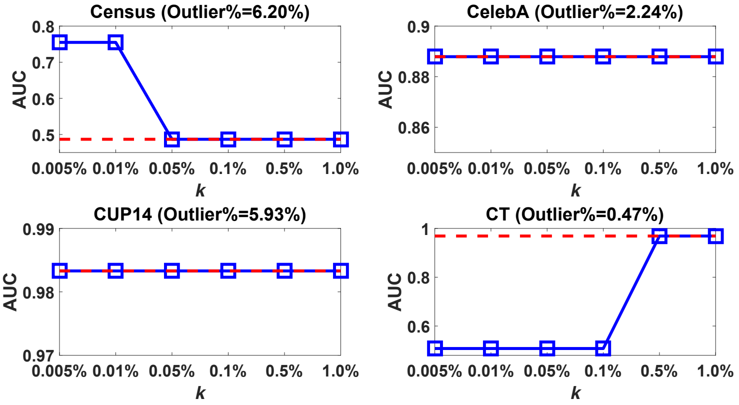


Figure 6.3: Representative AUC Performance of HOUR w.r.t. k. HOUR performs stably in most of the other data sets. The dashed line shows HOUR's performance with $k = outlier\%$.

6.5.5 Scalability Test

The scale-up test results are presented in Figure 6.4. As expected, HOUR is linear w.r.t. data size and quadratic w.r.t. dimensionality. HOUR runs comparably fast to CBRW and FPOF w.r.t. different data sizes. In the right panel, HOUR runs over five orders of magnitude faster than FPOF, while the iterative optimization process makes HOUR run considerably slower than CBRW. Nevertheless, HOUR is easy to parallelize. In future work we plan to reduce its time complexity to be nearly linear w.r.t. dimensionality by a parallel implementation of Steps (5-8) in Algorithm 6.2.

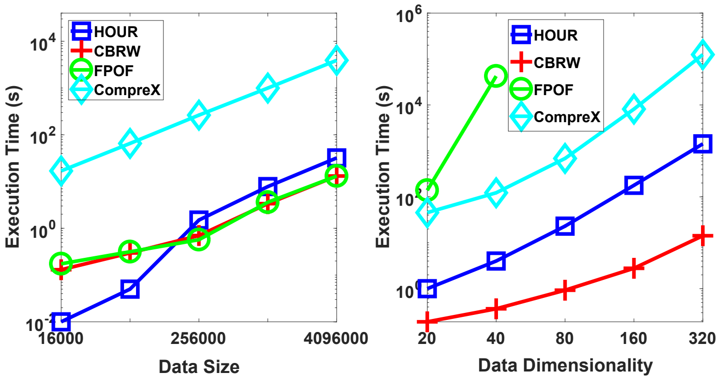


Figure 6.4: Scale-up Test w.r.t. Data Size and Dimensionality. FPOF runs out of memory when dimensionality reaches 80.

6.6 Summary

A wrapper-based outlier detection framework WrapperOD and its instance HOUR are introduced to joint top-k outlier detection with feature selection for handling data with noisy features. HOUR is more plausible than its counterparts: (i) it performs significantly better in global and/or local outlier ranking; and (ii) it obtains stable performance w.r.t. k and good scalability. The capability of returning the top k outliers with superior $P@n$ performance makes HOUR a good candidate for real-world applications, since investigation resources are often only sufficient for limited suspicious objects.

This is the first work to explore the applicability of wrapper approaches for outlier detection. One key challenge for WrapperOD is the efficiency, which is addressed in HOUR by a closed-form outlierness estimation based on binary cascade couplings between the outlier factors of values. However, this solution only works for categorical data. Similar efficient outlierness estimation methods are required to implement an instantiation of WrapperOD for numeric data. One potential solution is to design parallel implementations of existing outlier detectors to achieve desirable efficiency. Another challenge is the internal performance evaluation measures of outlier detection. HOUR defines an efficient outlierness margin-based measure, but it requires users to manually set the parameter, k, which is difficult to tune without proper prior knowledge. More easy-to-use internal evaluation measures are needed.

Chapter 7

High-order Cascade of Outlier Factors

7.1 Introduction

All the previous three chapters focus on low-order couplings of outlier factors, i.e., the outilerness coupling between pairs of nodes in the value graph. The solutions therein work very well for the cases that the noisy values do not interact with each other, and they may fail otherwise. This is because the noisy values may obtain outlierness that is comparable to, or even larger than, the outlying values when they are coupled with each other. Recall that both outlying values and noisy values have low occurrence frequency, but we assume the outlying values are successively coupled with each other while the noisy values only have very weak random couplings or do not have any couplings. However, this assumption may be violated in some challenging situations, e.g., when the total number of outlying values is small in data sets with a limited number of outliers, or when adversarial manipulations are done to generate a few strongly coupled noisy values.

In this chapter, we explore high-order couplings of value outlierness to address this issue. Instead of only considering the outlierness influence between pairs of values, we consider the outlierness influence among a set of values in the subgraphs of the value graph. As an example in Figure 7.1, the outlierness of a value u_2 is not only based on its first-order neighbors but also *locally and directly* dependent on the other higher-order neighbors in the relevant subgraphs. This is different from our previous methods that capture the outlierness influence from high-order neighbors in an *indirect* way, which are easily biased by the noisy values.

This idea motivates us to introduce a refined framework, called HOCOF, to have a noise-resilient outlierness estimation. HOCOF builds upon the CUOT framework presented in Chapter 4. HOCOF first leverages the value graph in CUOT to derive a set of its relevant subgraphs, and computes the outlierness of each value based on these subgraphs. These outlierness is then used to replace the initial value outlierness that is originally based on low-order intra-feature value couplings. HOCOF finally uses the stationary probability of the biased random walks as the value outlierness. The subgraph-based value outlierness is the key to capturing the high-order couplings. The cascade influence is still captured in

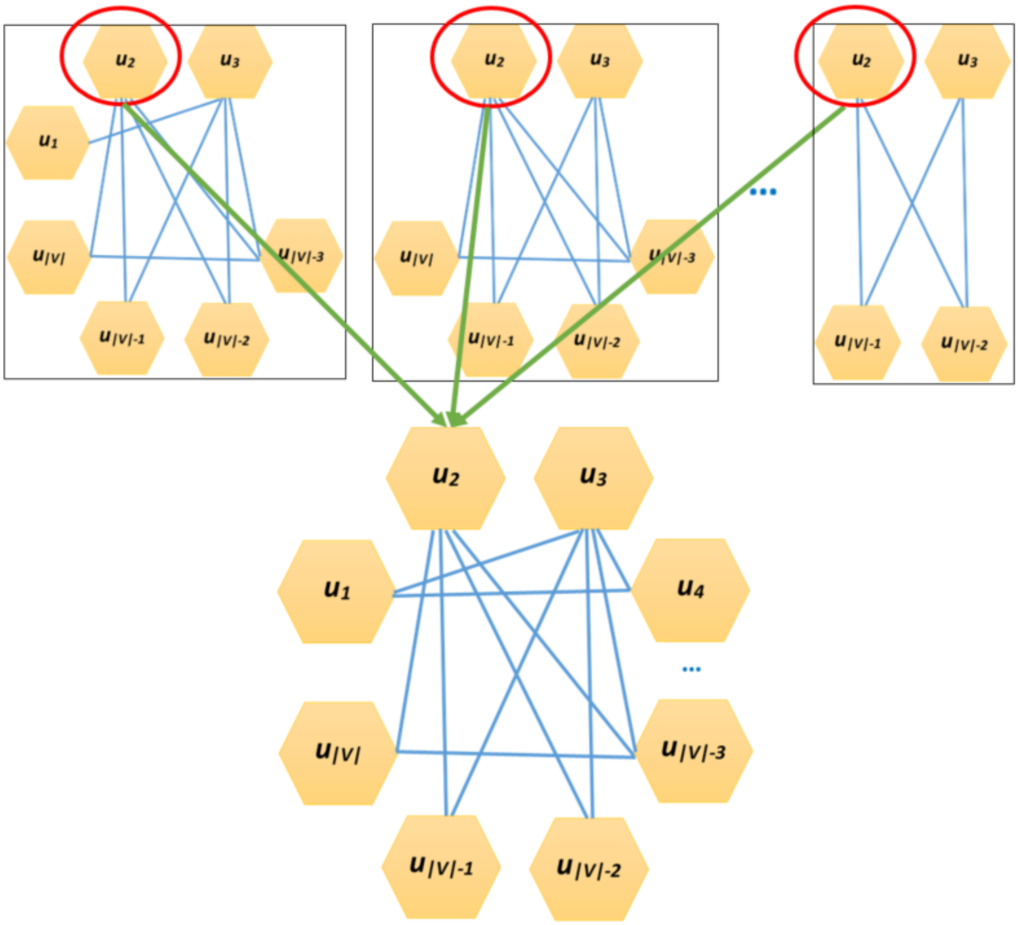


Figure 7.1: High-order Cascade Couplings. The top subgraphs denote a collection of high-order couplings between multiple values, capturing certain local interactions. We aim to use this type of local high-order couplings w.r.t. each node (e.g., u_2) or edge to augment the outlierness estimation in the original value graph that only captures pairwise low-order couplings

the random walks, but this cascade effect is built upon the high-order couplings.

The HOCOF framework is implemented by a method, called multiple-granularity Subgraph Densities augmented Random Walks (SDRW). SDRW defines mutual information-based outlierness influence vectors and works to capture the inter-feature value couplings, resulting in an *undirected* value graph that is different from the graph used by CBRW in Chapter 4 and HOUR in Chapter 6. More importantly, SDRW adds a new subgraph density-based outlier factor to capture high-order homophily couplings to further enhance its tolerance to noisy values.

Accordingly, this chapter makes two major contributions.

i. A novel coupled unsupervised outlier detection (HOCOF) framework estimates the outlier score of each value by incorporating high-order couplings into the cascade homophily modeling, which substantially improves its resilience to noisy values or adversarial manipulations.

ii. HOCOF is further instantiated by the SDRW method. Although SDRW involves high-order couplings, it has a similar efficiency as the low-order methods (e.g., CBRW) while it is much more tolerant to noisy data.

Extensive experiments show that: (i) our SDRW-based outlier detection method performs significantly better than five current state-of-the-art methods on 15 real-world data sets with different levels of feature noise, and it also significantly outperforms the CBRW-based detector, achieving more than 5% average improvement on complex data sets; and (ii) the SDRW-based feature selection method performs comparably well to the CBRW-based method, and both of them yield high-quality feature subsets, which help significantly improve the current state-of-the-art methods.

The rest of this chapter is organized as follows. The HOCOF framework is detailed in Section 7.2. The SDRW instance is introduced in Section 7.3. A theoretical analysis of SDRW is presented in Section 7.4. The evaluation results are given in Section 7.5. This work is then concluded in Section 7.6.

7.2 The Proposed HOCOF Framework

The HOCOF framework aims to incorporate the high-order coupling relationships into the outlierness propagation step. As shown in Figure 7.2, the whole procedure of HOCOF is exactly the same as CUOT in Chapter 4 except that HOCOF defines a subgraph-based value outlierness to replace the initial value outlierness before performing the outlierness propagation. Note that here we focus on incorporating the high-order information into each node, i.e., the refinement in the left, but we may also be able to use this high-order information to improve the outlierness influence matrix $\mathbf{M}$, i.e., the refinement in the right.

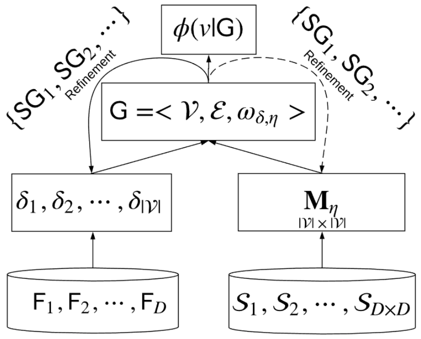

$\{SG_1, SG_2, \cdots\}$
Refinement
$\phi(v|G)$
$\{SG_1, SG_2, \cdots\}$
Refinement
$G =< \mathcal{V}, \mathcal{E}, \omega_{\delta,\eta} >$
$\delta_1, \delta_2, \cdots, \delta_{|\mathcal{V}|}$
$\underset{|\mathcal{V}|\times|\mathcal{V}|}{\mathbf{M}_\eta}$
$F_1, F_2, \cdots, F_D$
$\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_{D\times D}$

Figure 7.2: The Proposed HOCOF Framework. $\{\mathsf{SG}_1, \mathsf{SG}_2, \cdots\}$ is a set of subgraphs derived from the value graph G.

The high-order outlierness is defined as follows.

Definition 7.1 (High-order Outlier Factor)**.** *Let* $\{\mathsf{SG}_1, \mathsf{SG}_2, \cdots\}$ *be a set of subgraphs of* G *with no less than two nodes,* $\forall u \in \mathcal{V}$*,* $\phi(u|\mathsf{SG}_i)$ *denotes a outlierness measure that assigns a local outlierness to* u *under the context of* SG_i*. The high-order outlier factor of* u *is defined as the average of* $\{\phi(u|\mathsf{SG}_1), \phi(u|\mathsf{SG}_2), \cdots\}$*.*

Note that $\phi(u|\mathsf{SG}_i)$ computes the outlierness of u based on the whole context of SG_i, rather than the cascade of pairwise outlierness propagation over the whole value graph in CBRW, so $\phi(u|\mathsf{SG}_i)$ captures high-order information and its outlierness is also locally sensitive within the subgraph.

7.3 A HOCOF Instance: SDRW

This section introduces an instantiation of HOCOF, called multiple-granularity Subgraph Densities augmented Random Walks (SDRW). SDRW is motivated by CBRW, but it is a significantly enhanced instance compared to CBRW. Specifically, SDRW uses the same intra-feature mode-normalized initial outlierness as CBRW, but it replaces the conditional probability-based outlierness influence with pointwise mutual information-based influence. Although this change is minor, it effectively transforms the value graph into an undirected graph, which also captures more information than the binary coupling-based undirected graph in Chapter 6. Subsequently we can derive a more effective and parameter-free closed-form solution for learning value outlierness. More importantly, to enhance its tolerance to noisy features, SDRW uses a multiple-granularity dense subgraph mining to learn a more reliable bias into the biased random walks. A summary of the differences between CBRW and SDRW is provided in Table 7.1.

Table 7.1: Conceptual Comparison of CBRW and SDRW

	CBRW	**SDRW**
Initial Value Outlierness	Mode-based Normalization	
Outlierness Influence Vector	Conditional Probability	Pointwise Mutual Information
Value Graph	Directed	Undirected
Value Outlierness Learning	BRWs	Noise-tolerant BRWs
Closed-form Solution	No	Yes
Parameters	α	None
Time Complexity	$O(ND^2) + O(\lvert\mathcal{E}\rvert I_{max})$	$O(ND^2) + O(\lvert\mathcal{E}\rvert)$

To differentiate between the specifications for CBRW and SDRW, a superscript '′' is added to the notations in SDRW if the same notation is used in CBRW. Since SDRW and CBRW use the same method to compute the intra-feature initial outlierness, here we start with outlierness influence vectors (see Section 4.2.1 for the definition of the intra-feature initial outlierness).

7.3.1 Pointwise Mutual Information-based Outlierness Influence

Pointwise mutual information (PMI) is a widely-used measure to define the correlation between two values. PMI replaces the conditional probabilities in the outlierness influence vector as follows.

Definition 7.2 (PMI-based Outlierness Influence Vector). *The PMI-based outlierness influence vector of a value v due to all the other values is defined as*

$$\begin{aligned}\mathbf{q}'_v &= [\eta'(u,v), \cdots, \eta'(w,v)]^\intercal \\ &= [\frac{freq(u,v)}{freq(u)freq(v)}, \cdots, \frac{freq(w,v)}{freq(w)freq(v)}]^\intercal,\ \forall u, w \in \mathcal{V} \setminus v.\end{aligned} \tag{7.1}$$

where $\frac{freq(u,v)}{freq(u)freq(v)}$ is the pointwise mutual information between the values u and v with the logarithm removed.

Note that $-\infty \leq PMI(u;v) \leq min\{-\log P(u), -\log P(v)\}$. The logarithm in PMI is thus removed to guarantee that when $\mathbf{q}'_v$ is used to construct the value graph, the adjacency matrix of the graph is non-negative. The resulting inter-feature outlier factor has the following two key properties.

i. $\eta'(u,v) \in [0,1]$.

ii. $\eta'(u,v) = \eta'(v,u)$.

PMI captures more rigorous homophily couplings than conditional probabilities because it includes the individual frequencies of both values. Particularly, the difference obtained by $\eta'(u,v) - \eta'(w,z)$ is much larger than $\eta(u,v) - \eta(w,z)$ in CBRW, when both values u and v are outlying values and at least one of the values w and z is not an outlying value. This helps obtain a stronger correlation between outlying values, resulting in a higher contrast between the couplings of outlying values and that of other values.

7.3.2 Refining the Value Graph with Subgraph Densities

SDRW then also constructs an attributed value graph $\mathsf{G}' =< \mathcal{V}, \mathcal{E}, \Theta_{\delta',\eta'} >$ in the same way as CBRW built the graph G. Since SDRW and CBRW use the same mode-based initial outlierness, we have $\delta' = \delta$. Here the key difference between SDRW and CBRW is that G' is an undirected graph as $\eta'(u,v) = \eta'(v,u)$. while G is a directed graph.

Let $\mathbf{A}'$ be the adjacency matrix of G' with its entry $\mathbf{A}'(u,v) = \eta'(u,v)$. According to Lemma 4.0.1, the attributed value graph can be equivalently transformed to a plain graph with an adjacency matrix $\mathbf{C}$, in which its entry is

$$\mathbf{C}(u,v) = \delta'(u)\eta'(u,v)\delta'(v), \; \forall u,v \in \mathcal{V}. \tag{7.2}$$

One major problem with $\mathbf{C}(u,v)$ (or $\mathbf{B}(u,v)$ in CBRW) is that δ' (or δ) may mislead the subsequent value outlierness learning when u or v is a noisy value. This is because noisy values may have a lower frequency than outlying values. Consequently, noisy values have larger intra-feature outlierness δ than outlying values. When there are many such noisy values, this can downgrade the quality of the outlierness learning. One simple solution is to remove the term δ', but that would also remove important intra-feature value coupling information, making the solution less effective when outliers demonstrate obvious outlying behaviors in individual features (See the empirical results in Section 7.5.2).

Instead, SDRW learns a noise-tolerant term to replace δ' by aggregating the density of a collection of multiple-granularity dense subgraphs associated with a specific value. Our intuition is as follows. Due to the homophily couplings between outlying values, the neighbors of outlying values in the value graph are much more likely to be outlying values than noisy values. Since the edge weights convey the value outlierness, the outlying values are located in denser subgraphs than noisy values. We therefore define the following subgraph density-based outlier factor:

Definition 7.3 (Subgraph Density-based High-order Outlier Factor)**.** *Let the densest k subgraph SG_k be the densest subgraph of exactly k nodes in graph G', and let $\mathcal{G} =$*

$\{\mathsf{SG}_2, \mathsf{SG}_3, \cdots, \mathsf{SG}_{|\mathcal{V}|-1}\}$ be the complete set of the densest k subgraphs. The subgraph density-based outlier factor of a value v is defined as the average density of all the densest k subgraphs that contain v, i.e.,

$$ad(v) = \frac{1}{|\mathcal{G}_v|} \sum_{\mathsf{SG}^v \in \mathcal{G}_v} den(\mathsf{SG}^v), \tag{7.3}$$

where $\mathcal{G}_v$ is the set of the densest k subgraphs that contain v and the subgraph density is computed by

$$den(\mathsf{SG}^v) = \frac{\sum_{u \in \mathcal{V}_v} \sum_{v \in \mathcal{V}_v} \delta'(u)\eta'(u,v)\delta'(v)}{2|\mathcal{V}_v|}, \tag{7.4}$$

where $\mathcal{V}_v$ denotes the set of nodes contained in SG^v.

ad is built on the homophily couplings and is designed to capture the possible cascade relations of outlying values. This helps increase the outlierness of the outlying values that are surrounded by only a few direct outlying nodes, but their outlying neighbor nodes (or the neighbors of them, and so on) are coupled with many outlying values. However, finding single densest k subgraph has been proven to be an NP-hard problem [65]. We resort to a greedy method in Algorithm 7.1 to produce a set of dense subgraphs, $\mathcal{G}^+$, to approximate $\mathcal{G}$. Note that SG_1 and $\mathsf{SG}_{|\mathcal{V}|}$ are excluded from $\mathcal{G}$, since they provide no distinguishing information for computing the *ad* of each value.

Algorithm 7.1 Dense Subgraph Discovery

```
Input: $\mathcal{X}$ - data objects
Output: $\mathcal{G}^+$ - a set of dense subgraphs
Generate $\mathbf{C}$ using Equation (7.2)
Compute the weighted degree of each node $v \in \mathcal{V}$
Initialize $\mathcal{G}^+$ as an empty set
repeat
    Let $v \in \mathcal{V}$ be the node having the minimal weighted degree in $\mathsf{G}'$
    $\mathsf{G}' \leftarrow \mathsf{G}' \setminus v$
    $\mathcal{G}^+ \leftarrow \mathcal{G}^+ \cup \{\mathsf{G}'\}$
until Only one node left in $\mathsf{G}'$
return $\mathcal{G}^+$
```

Although Algorithm 7.1 cannot find the exact set of the densest k subgraphs, using $\mathcal{G}^+$ to compute *ad* guarantees that values with more outlying neighbors (i.e., with a larger weighted degree) obtain a larger *ad*. Moreover, it has linear time complexity w.r.t. $|\mathcal{V}|$, which enables SDRW to compute *ad* very efficiently. Other desirable properties of this algorithm include: (i) the densest subgraph in the subgraphs it produces has a $\frac{1}{2}$-approximation to the optimal densest subgraph without size constraints; and (ii) it is able to produce the densest subgraph with at least k nodes (a relaxed problem to the problem of finding the densest k subgraph) having $\frac{1}{3}$-approximation to the optimal solution [8, 65]. These two properties make the use of $\mathcal{G}^+$ obtain a good approximation to the exact *ad*.

We further replace δ with *ad* in Equation (7.2) and obtain

$$\mathbf{B}'(u,v) = ad(u)\eta'(u,v)ad(v),\ \forall u,v \in \mathcal{V}, \tag{7.5}$$

where ad can be seen as an enhanced δ based on the inter-feature outlierness function η for better tolerance to noisy values. Unlike η' that captures low-order pairwise value couplings, ad captures high-order homophily couplings.

7.3.3 Noise-tolerant Random Walks for Learning Value Outlierness

SDRW then performs random walks with the adjacency matrix $\mathbf{B}'$. The transition matrix is as follows:

$$\mathbf{T}'(u,v) = \frac{\mathbf{B}'(u,v)}{\sum_{v\in\mathcal{V}} \mathbf{B}'(u,v)} = \frac{ad(u)\mathbf{A}'(u,v)ad(v)}{\sum_{v\in\mathcal{V}} ad(u)\mathbf{A}'(u,v)ad(v)}. \tag{7.6}$$

This is equivalent to biased random walks with the following transition matrix

$$\mathbf{W}^{b\prime}(u,v) = \frac{ad(v)\mathbf{A}'(u,v)}{\sum_{v\in\mathcal{V}} ad(v)\mathbf{A}'(u,v)}, \tag{7.7}$$

in which the terms ad and $\mathbf{A}'(u,v)$ replace δ and $\mathbf{A}(u,v)$ in Equation 4.7, respectively. ad improves the tolerance to noisy values over δ, while $\eta'(u,v)$ improves the homophily coupling modeling over $\eta(u,v)$.

We can accordingly define the value outlierness as follows.

Definition 7.4 (SDRW-based Value Outlierness)**.** *The outlierness of node v is defined as*

$$\phi'(v) = \boldsymbol{\pi}^{*\prime}(v), \tag{7.8}$$

where $\boldsymbol{\pi}^{*\prime} = \mathbf{W}^{b\prime}\boldsymbol{\pi}^{*\prime}$ denotes the stationary probabilities of biased random walks with the transition matrix $\mathbf{W}^{b\prime}$.

As shown in Section 7.4.1, we can derive a closed-form of $\phi'(v)$ as

$$\phi'(v) = \frac{\sum_{u\in\mathcal{V}} ad(v)\eta'(u,v)ad(u)}{\sum_{v\in\mathcal{V}}\sum_{u\in\mathcal{V}} ad(v)\eta'(u,v)ad(u)}, \tag{7.9}$$

where the nominator is the weighted degree of node v and the denominator is the volume (i.e., total weighted degree) of the graph G'.

Similar to CBRW in Chapter 4, after obtaining the outlierness of values, we can respectively use Eqn. (4.11) and Eqn. (4.12) to compute the outlierness of features and data objects for outlying feature selection and outlier detection.

7.3.4 The Algorithm and Its Time Complexity

Algorithm 7.2 presents the procedures of SDRW. Step 1 computes the dense subgraph-based outlier factor ad, followed by the generation of $\mathbf{B}'$ in Step 2. Steps 3-5 further estimate the outlierness of each value using the closed-form of the stationary probability distribution of biased random walks.

Both the iterative removal of nodes in Algorithm 7.1 and the calculation of ad in Step 2 have a time complexity of $O(|\mathcal{E}|)$. Similar to CBRW, SDRW requires $O(ND^2)$ to

Algorithm 7.2 Subgraph Density Augmented Random Walks

```
Input: $\mathcal{X}$ - data objects
Output: $\boldsymbol{\pi}^{*\prime}$ - the stationary probability distribution
Compute $ad$ using Equation (7.3) with $\mathcal{G}^{+}$ returned by Algorithm 7.1
Obtain $\mathbf{B}'$ using Equation (7.5)
for $v \in \mathcal{V}$ do
    $\boldsymbol{\pi}'(v) \leftarrow \frac{\sum_{u\in\mathcal{V}} ad(v)\eta'(u,v)ad(u)}{\sum_{v\in\mathcal{V}}\sum_{u\in\mathcal{V}} ad(v)\eta'(u,v)ad(u)}$
end for
return $\boldsymbol{\pi}^{*\prime}$
```

obtain $\mathbf{B}'$ using Equation (7.5). The subsequent estimation of value outlierness has a time complexity of $O(|\mathcal{E}|)$. Therefore, SDRW has an overall time complexity of $O(ND^2 + |\mathcal{E}|)$.

7.4 Theoretical Analysis

7.4.1 Closed-form Solution

The closed-form solution to the value outlierness estimation function $\phi'(v)$ is proven as follows.

Theorem 7.1 (Closed-form Outlierness Estimation)**.** *Let* G' *be the graph with its adjacency matrix* $\mathbf{B}'$ *such that* $\mathbf{B}'(u,v) = ad(u)\eta'(u,v)ad(v)$, $\forall u, v \in \mathcal{V}$. *Then we have*

$$\boldsymbol{\pi}^{*\prime}(v) = \frac{d'(v)}{vol(\mathsf{G}')}, \ \ \forall v \in \mathcal{V}, \tag{7.10}$$

where $d'(v) = \sum_{u\in\mathcal{V}} \mathbf{B}'(u,v) = \sum_{u\in\mathcal{V}} ad(v)\eta'(u,v)ad(u)$ *denotes the weighted degree of node* v *and* $vol(\mathsf{G}') = \sum_{v\in\mathcal{V}} d'(v)$ *is the volume of* G'.

Proof. To prove Equation (7.10), we need to show that when $\boldsymbol{\pi}'(u) = \frac{d'(u)}{vol(\mathsf{G}')}$, $\forall u \in \mathcal{V}$, we have $\boldsymbol{\pi}' = \mathbf{W}^{b\prime}\boldsymbol{\pi}'$, i.e., $\boldsymbol{\pi}'$ becomes steady w.r.t. the time step.

First, the probability of visiting v in a time step is $\boldsymbol{\pi}'^{,t+1}(v) = \sum_{u\in\mathcal{V}} \boldsymbol{\pi}'^{,t}(u)\mathbf{W}^{b\prime}(u,v)$. We then have

$$\boldsymbol{\pi}'^{,t+1}(v) = \sum_{u\in\mathcal{V}} \boldsymbol{\pi}'^{,t}(u)\frac{\mathbf{B}'(u,v)}{\sum_{w\in\mathcal{V}} \mathbf{B}'_{u,w}}.$$

When $\boldsymbol{\pi}'^{,t}(u) = \frac{d'(u)}{vol(\mathsf{G}')}$, we have

$$\boldsymbol{\pi}'^{,t+1}(v) = \sum_{u\in\mathcal{V}} \frac{d'(u)}{vol(\mathsf{G}')}\frac{\mathbf{B}'(u,v)}{\sum_{w\in\mathcal{V}} \mathbf{B}'_{u,w}} == \sum_{u\in\mathcal{V}} \frac{d'(u)}{vol(\mathsf{G}')}\frac{\mathbf{B}'(u,v)}{d'(u)} = \sum_{u\in\mathcal{V}} \frac{\mathbf{B}'(u,v)}{vol(\mathsf{G}')}.$$

Since $\mathbf{B}'(u,v) = \mathbf{B}'(v,u)$, we further have

$$\boldsymbol{\pi}'^{,t+1}(v) = \sum_{u\in\mathcal{V}} \frac{\mathbf{B}'(u,v)}{vol(\mathsf{G}')} = \sum_{u\in\mathcal{V}} \frac{\mathbf{B}'(v,u)}{vol(\mathsf{G}')} = \frac{d'(v)}{vol(\mathsf{G}')}.$$

Therefore, we also have $\boldsymbol{\pi}'^{,t+1}(u) = \frac{d'(u)}{vol(\mathsf{G}')} = \boldsymbol{\pi}'^{,t}(u)$, and thus $\boldsymbol{\pi}'$ becomes steady.

□

Note that the above form is not necessarily the unique convergence form of the random walks. It becomes a unique convergence if, and only if, G' is irreducible and aperiodic [84]. However, this closed-form well captures the homophily outlying couplings. Hence, it is used to compute the outlierness of all values.

7.4.2 Handling Noisy Features

To handle data with noisy features, a fundamental requirement for value outlierness-based outlier detectors is to assign larger outlierness to outlying values (i.e., infrequent values contained by outliers) than noisy values (i.e., infrequent values contained by normal objects). The outlierness scoring methods in IID methods, such as $1 - \mathit{freq}(\cdot)$ or $\frac{1}{\mathit{freq}(\cdot)}$, assign similar outlierness to outlying and noisy values, since these two types of values often have a similarly low frequency. These methods therefore fail to distinguish outlying values/patterns from noisy ones and become ineffective in handling data with many noisy features. By contrast, modeling homophily couplings enables CBRW and SDRW to contrast the outlierness of outlying values from the noisy ones. We demonstrate this intuition with a straightforward example, as follows.

Let u' and w' be outlying and noisy values of the same frequency, respectively. Assume both u' and w' share the same direct neighbor set $\mathcal{N}$, in which $\mathcal{N}^o \subset \mathcal{N}$ is a set of outlying values and $\mathcal{N} \setminus \mathcal{N}^o$ is the set of normal values. Then, according to Proposition 4.1, we have

$$\phi(u') \propto \sum_{u \in \mathcal{N}^o} \phi(u)\delta(u)\eta(u, u')\delta(u') + \sum_{w \in \mathcal{N} \setminus \mathcal{N}^o} \phi(w)\delta(w)\eta(w, u')\delta(u'). \tag{7.11}$$

We can obtain a similar proportional form for $\phi(w')$ by replacing u' with w' in the above equation. Since u' and w' share the same direct neighbor set, the outcomes of the terms ϕ and δ on the right-hand side of $\phi(u')$ are the same as that in $\phi(w')$. We can therefore simplify them as $\phi(u') \propto \sum_{u \in \mathcal{N}^o} \eta(u, u') + \sum_{w \in \mathcal{N} \setminus \mathcal{N}^o} \eta(w, u')$ and $\phi'(w') \propto \sum_{u \in \mathcal{N}^o} \eta(u, w') + \sum_{w \in \mathcal{N} \setminus \mathcal{N}^o} \eta(w, w')$. When there exist homophily couplings among the outlying values while noisy values are randomly coupled with the outlying values, we have $\sum_{u \in \mathcal{N}^o} \eta(u, u') > \sum_{u \in \mathcal{N}^o} \eta(u, w')$. If the values u' and w' have similar co-occurrence patterns with the set of normal values $\mathcal{N} \setminus \mathcal{N}^o$, then $\sum_{w \in \mathcal{N} \setminus \mathcal{N}^o} \eta(w, u') \approx \sum_{w \in \mathcal{N} \setminus \mathcal{N}^o} \eta(w, w')$. Hence, $\phi(u') > \phi(w')$ holds in CBRW.

For SDRW, we can obtain the following equivalence due to Theorem 7.1

$$\phi'(u') = \sum_{u \in \mathcal{N}^o} \delta(u)\eta'(u, u')\delta(u') + \sum_{w \in \mathcal{N} \setminus \mathcal{N}^o} \delta(w)\eta'(w, u')\delta(u'), \tag{7.12}$$

where SDRW retains the δ function but changes the η function from conditional probabilities to PMI. Since the normal value w generally has a high frequency, $\eta'(w, u')$ is marginalized by $\mathit{freq}(w)$ and $\delta(w)$ is very small. Hence, the second term in Equation (7.12) can be generally left out. Similar to the cases in CBRW, we can omit δ. We therefore obtain $\phi'(u') = \sum_{u \in \mathcal{N}^o} \eta'(u, u')$ and $\phi'(w') = \sum_{u \in \mathcal{N}^o} \eta'(u, w')$. In such cases, we achieve $\phi'(u') > \phi'(w')$ in SDRW even when not using ad. Compared to CBRW that obtains $\phi(u') > \phi(w')$ under certain conditions, SDRW can achieve the same result without such constraints. This demonstrates the benefit of replacing η with η'.

However, many real-world data sets may demonstrate much tougher cases than the example as above. For example, the frequency of noisy values can be much lower than that of outlying values, and the values u' and w' have different direct neighbor sets. In such cases, if δ is still used in SDRW, we have $\delta(w') > \delta(u')$, and consequently $\phi'(w')$ can obtain larger outlierness from its normal value neighbors, compared to $\phi'(u')$. If w' randomly occurs with some outlying values while u' has limited outlying values in its direct neighbors, $\phi'(w')$ can also obtain larger outlierness from its outlying value neighbors, leading to the undesired result $\phi(u') < \phi(w')$. To tackle this problem, ad is introduced into SDRW to consider both direct neighbors and indirect neighbors. It is assured that $ad(u')$ is much larger than $ad(w')$ when the outlying values bond together, e.g., in the form of cascade. This enables SDRW to obtain much larger outlierness from the direct and indirect outlying value neighbors for the outlying value u', compared to the noisy value w'. As a result, replacing δ with ad in Equation (7.12) largely increases the ability of SDRW to assign larger outlierness to outlying values than noisy ones.

7.5 Experiments and Evaluation

This work uses the same data sets as that in Section 4.5. We also add our previous method CBRW [91] and another state-of-the-art method Sp [111] into our competing methods to perform more comprehensive empirical analyses.

The first three subsections provide the experimental evaluation results of outlier detection, algorithmic component justification, and outlying feature selection on the 15 benchmark data sets, respectively. The fourth subsection reports the scale-up test results.

7.5.1 Effectiveness of Outlier Detection

We first present a summary of detection performance on all data sets, and then analyze the detection performance on complex and simple data sets separately.

Overall Performance

The AUC results of SDRW$_{\text{od}}$, CBRW$_{\text{od}}$, MarP, FPOF, CompreX, iForest and Sp on the 15 data sets are presented in Table 7.2. SDRW$_{\text{od}}$ achieves the best detection performance on 10 data sets, with four close to the best (having the difference in AUC no more than 0.01). SDRW$_{\text{od}}$ obtains more than 5% improvement over CBRW$_{\text{od}}$ and 16%-28% improvement over the other detectors. The significance test results show that SDRW$_{\text{od}}$ significantly outperforms CBRW$_{\text{od}}$ and FPOF at the 95% confidence level and the other four contenders, MarP, CompreX, iForest and Sp, at the 99% confidence level.

Handling Complex Data

This analysis is separated into four parts based on the four indicators in Table 4.1.

Table 7.2: AUC Performance of SDRW$_{\text{od}}$ and Its Six Contenders on the 15 Data Sets. '○' indicates out-of-memory exceptions, while '●' indicates that we cannot obtain the results within two months. Simiar to the empirical analysis for CBRW in Chapter 4, we separate *complex* data from *simple* data sets based on average rank in Table 4.1. The best performance for each data set is boldfaced.

Data	**SDRW$_{\text{od}}$**	**CBRW$_{\text{od}}$**	**MarP**	**FPOF**	**CompreX**	**iForest**	**Sp**
BM	**0.6511**	0.6287	0.5584	0.5466	0.6267	0.5762	0.6006
Census	0.6371	**0.6678**	0.5899	0.6148	0.6352	0.5378	0.6175
AID362	0.6665	0.6640	0.6270	○	0.6480	0.6485	**0.6678**
w7a	**0.8059**	0.6484	0.4723	○	0.5683	0.4053	0.4517
CMC	**0.6415**	0.6339	0.5417	0.5614	0.5669	0.5746	0.5901
APAS	**0.8544**	0.8190	0.6193	○	0.6554	0.4792	0.7401
CelebA	**0.8845**	0.8462	0.7358	0.7380	0.7572	0.6797	0.7132
Chess	**0.8387**	0.7897	0.6447	0.6160	0.6387	0.6124	0.6410
AD	**0.8482**	0.7348	0.7033	○	●	0.7084	0.7183
SF	**0.8817**	0.8812	0.8446	0.8556	0.8526	0.7865	0.8434
Probe	0.9891	**0.9906**	0.9800	0.9867	0.9790	0.9762	0.9654
U2R	**0.9941**	0.9651	0.8848	0.9156	0.9893	0.9781	0.9886
LINK	**0.9978**	0.9976	0.9977	**0.9978**	0.9973	0.9917	0.9952
R10	0.9837	**0.9905**	0.9866	○	0.9866	0.9796	0.9870
CT	0.9703	0.9703	**0.9773**	0.9772	0.9772	0.9364	0.9601
Avg. (Top-10)	**0.7710**	0.7314	0.6337	0.6554	0.6610	0.6009	0.6584
Avg. (All)	**0.8430**	0.8152	0.7442	0.7810	0.7770	0.7247	0.7653
p-value	**SDRW$_{\text{od}}$** vs.	0.0245	0.0006	0.0117	0.0031	0.0001	0.0004
		CBRW$_{\text{od}}$ vs.	0.0004	0.0137	0.0067	0.0003	0.0020

Results on Data Sets with Highly Complex Value Couplings. The 10 data sets with the largest proportions of negative value couplings are *AD*, *Census*, *w7a*, *APAS*, *AID362*, *BM*, *SF*, *CelebA*, *R10* and *CMC* according to κ_{vcc}. On these 10 data sets, SDRW$_{\text{od}}$ achieves an average AUC improvement over MarP (17%), FPOF (18%), CompreX (12%), iForest (23%) and Sp (13%). SDRW$_{\text{od}}$ obtains more than 4% improvement over CBRW$_{\text{od}}$. Most of these data sets contain more than 10% negative value couplings. This can result in many misleading patterns and, consequently, substantially degrade the performance of traditional outlier detection methods (i.e., MarP, FPOF, CompreX, iForest and Sp). In contrast, the positive homophily couplings captured by SDRW$_{\text{od}}$ and CBRW$_{\text{od}}$ enable them to identify outliers more effectively in such adverse environments. Additionally, SDRW$_{\text{od}}$ incorporates the subgraph density outlier factor ad, which further enhances its ability to tackle the negative couplings over CBRW$_{\text{od}}$.

Results on Data Sets with Strong Heterogeneity. The 10 data sets with the strongest heterogeneity are *Chess*, *BM*, *Census*, *CMC*, *SF*, *LINK*, *Probe*, *U2R*, *CelebA* and *APAS* according to κ_{het}. SDRW$_{\text{od}}$ achieves an average AUC improvement over MarP (13%), FPOF (10%), CompreX (8%), iForest (16%) and Sp (8%). SDRW$_{\text{od}}$ and CBRW$_{\text{od}}$ perform very comparably in this case, with only a 1% difference in terms of the average AUC performance. Data sets with a large κ_{het} indicate diversified frequency distributions across their features, resulting in different semantics of the same frequency in the features. However, the five competitors ignore this characteristic, treating the same frequencies of values/patterns from different features/subspaces equally. This leads to inaccurate outlier scoring of objects. SDRW$_{\text{od}}$ addresses this issue by the intra-feature mode-normalized outlierness. Thus, they performs substantially better than their competitors. Since SDRW$_{\text{od}}$ and CBRW$_{\text{od}}$ use the same intra-feature initial outlierness, they have very similar performance on these 10 complex data sets.

Results on Data Sets with Low Outlier Separability. According to κ_{ins}, the 10 data sets with the lowest outlier separability are *w7a*, *AID362*, *BM*, *CMC*, *AD*, *Chess*, *Census*, *CelebA*, *SF* and *R10*. Compared to MarP, FPOF, CompreX, iForest and Sp, SDRW$_{\text{od}}$ achieves 16%, 19%, 12%, 20% and 14% average improvements. SDRW$_{\text{od}}$ obtains more than 4% improvement over CBRW$_{\text{od}}$ on these low separable data sets. It is interesting to note that the top-ranked data sets in terms of κ_{ins} are also top-ranked in terms of κ_{vcc}. In other words, the low outlier separability in these data sets is in part due to their underlying non-IID characteristics. SDRW$_{\text{od}}$ and CBRW$_{\text{od}}$ with coupled outlier factors may therefore perform better than the other five detectors.

Results on Data Sets with High Feature Noise Level. The 10 data sets with the highest level of feature noise are *BM*, *AID362*, *APAS*, *Census*, *CelebA*, *w7a*, *CMC*, *CT*, *Chess* and *U2R*. SDRW$_{\text{od}}$ achieves an average AUC improvement over MarP (19%), FPOF (11%), CompreX (12%), iForest (23%) and Sp (13%). SDRW$_{\text{od}}$ also gains more than 4% improvement over CBRW$_{\text{od}}$. One major reason for the tolerance of SDRW$_{\text{od}}$ and CBRW$_{\text{od}}$ to noisy features is due to their homophily coupling modeling, as discussed in Section 7.4.2. However, the use of the intra-feature initial outlierness δ in CBRW$_{\text{od}}$ is less effective on data sets, where outlying values are difficult to distinguish from noisy values. SDRW$_{\text{od}}$ replaces δ with the high-order outlier factor *ad* and is, therefore, more tolerant to such data sets. This is justified by the further 4% improvement obtained by SDRW$_{\text{od}}$ over CBRW$_{\text{od}}$.

Handling Simple Data

All seven detectors perform very well on the five simple data sets in Table 4.2. This is particularly true for *R10*, *Probe* and *LINK*, on which all the detectors, including the most simple detector MarP, obtain the AUC of (or nearly) one. Although some of these data sets (e.g., *R10*) are ranked slightly higher than some complex data sets w.r.t. one or two of the data indicators, they rank toward the bottom in most cases, resulting in an overall low data complexity.

7.5.2 Justification of Algorithmic Components

Similar to CBRW$_{\text{od}}$, SDRW$_{\text{od}}$ consists of three main components: an intra-feature initial value outlierness, an inter-feature outlierness influence, and a graph mining method that integrates the two components to learn value outlierness. This section presents empirical results to justify the contribution of each component to the value outlierness learning.

Specifically, we first derive a baseline method, called BASE, that assumes all features are completely independent and only uses the intra-feature outlier factor to obtain value outlierness. We then builds on two additional baselines: SDRWia$_{\text{od}}$/CBRWia$_{\text{od}}$ has a weakened inter-feature factor by setting $\eta(u, v) = 1$ iff u and v co-occur; SDRWie$_{\text{od}}$/CBRWie$_{\text{od}}$ uses the original η in SDRW/CBRW while ignores the intra-feature value couplings by setting all $\delta(\cdot)$ to one.

The AUC results for SDRW$_{\text{od}}$, CBRW$_{\text{od}}$ and their variants are shown in Table 7.3. The following four observations can be made from these results. (i) BASE substantially underperforms the other six methods on nearly all the data sets. This indicates that assuming the independence of features is often not desirable in practice. (ii) SDRWia$_{\mathrm{od}}$/CBRWia$_{\mathrm{od}}$ performs comparably to SDRWie$_{\mathrm{od}}$/CBRWie$_{\mathrm{od}}$ in terms of average AUC. This is because intra-feature or inter-feature outlier factor capture only partial value couplings of the data and it works well only when the outlier factor fits well the specific data set. Since each of the data sets has very different intrinsic characteristics, Significantly weakening intra-feature or inter-feature outlier factor therefore results in a considerable loss of the detection accuracy in the misfitted data sets. (iii) Although SDRW$_{\mathrm{od}}$/CBRW$_{\mathrm{od}}$ performs less effectively than its variants on a few data sets, it obtains averagely better performance and performs more stably. This indicates that the way SDRW$_{\mathrm{od}}$ and CBRW$_{\mathrm{od}}$ integrate the two outlier factors are generally reasonable, but a better method of integration is needed to improve the performance on the data sets like *Census*, *APAS*, and *CelebA*. (iv) Although SDRWia$_{\mathrm{od}}$ (SDRWie$_{\mathrm{od}}$) and CBRWia$_{\mathrm{od}}$ (CBRWie$_{\mathrm{od}}$) have very comparable average performance, SDRW$_{\mathrm{od}}$ demonstrates substantially large improvement over CBRW$_{\mathrm{od}}$. This indicates that the way SDRW$_{\mathrm{od}}$ integrates the two outlier factors is more faithful than CBRW$_{\mathrm{od}}$.

Table 7.3: AUC Performance of SDRW$_{\mathrm{od}}$, CBRW$_{\mathrm{od}}$ and Their Variants Created by Removing One or Two Components. The best performance within CBRW/SDRW is boldfaced.

Data	**BASE**	**CBRWia$_{\mathrm{od}}$**	**CBRWie$_{\mathrm{od}}$**	**CBRW$_{\mathrm{od}}$**	**SDRWia$_{\mathrm{od}}$**	**SDRWie$_{\mathrm{od}}$**	**SDRW$_{\mathrm{od}}$**
BM	0.5778	0.5999	**0.6566**	0.6287	0.5988	**0.6698**	0.6511
Census	0.6033	**0.6832**	0.6579	0.6678	**0.7259**	0.6231	0.6371
AID362	0.6152	0.6034	0.6324	**0.6640**	0.6572	0.6307	**0.6665**
w7a	0.4744	0.4477	**0.7363**	0.6484	0.6106	0.8002	**0.8059**
CMC	0.5623	0.6179	0.6323	**0.6339**	0.6075	0.6373	**0.6415**
APAS	0.6208	**0.8739**	0.8624	0.8190	0.6660	**0.8604**	0.8544
CelebA	0.7352	0.7135	**0.9108**	0.8462	0.7367	**0.8998**	0.8845
Chess	0.6854	0.7766	0.4058	**0.7897**	0.7692	0.2322	**0.8387**
AD	0.7033	0.7250	**0.8270**	0.7348	0.6600	0.8426	**0.8482**
SF	0.8469	**0.8867**	0.8833	0.8812	0.8650	0.8809	**0.8817**
Probe	0.9795	0.9434	**0.9907**	0.9906	0.9807	0.9854	**0.9891**
U2R	0.8848	0.8817	0.9640	**0.9651**	0.8793	**0.9949**	0.9941
LINK	0.9977	0.9976	0.9976	0.9976	0.9976	0.9976	**0.9978**
R10	0.9866	0.9823	0.9903	**0.9905**	**0.9874**	0.9837	0.9837
CT	0.9770	0.9388	**0.9703**	**0.9703**	0.9607	0.9581	**0.9703**
Avg.	0.7500	0.7781	0.8078	0.8152	0.7802	0.7998	0.8430

7.5.3 Outlying Feature Selection Performance

This section presents the results of data complexity reduction by feature selection, followed by the AUC performance of two outlier detectors on the reduced data.

Data Complexity Reduction

Table 7.4 shows the results of the data complexity evaluation for each data indicator on the data sets with selected feature subsets as well as full feature sets.

SDRW$_{\mathrm{fs}}$ and CBRW$_{\mathrm{fs}}$ considerably reduce the data complexity in most data indicators on all data sets. Specifically, SDRW$_{\mathrm{fs}}$ reduces the complexities of κ_{vcc}, κ_{het} and κ_{fnl} by

25%, 7% and 19% respectively; and CBRW$_{\text{fs}}$ achieves respective 25%, 8% and 10% simplification in the indicators κ_{vcc}, κ_{het} and κ_{fnl}. ENFW obtains markedly large simplification in κ_{fnl} and κ_{het}, whereas it substantially increases the outlier inseparability according to κ_{ins}. This is because ENFW evaluate the relevance of features without considering their interactions. Thus, noisy features and highly relevant features may be filtered out together. In other words, ENFW reduces the data complexity in terms of κ_{fnl} at the expense of increasing the data complexity in terms of κ_{ins}. Also, ENFW is an entropy-based feature weighting method, which retains features with similar frequency distributions. As a result, ENFW can simply the data far more than SDRW$_{\text{fs}}$ and CBRW$_{\text{fs}}$ in terms of κ_{het}. However, since it builds upon the feature independence assumption, it can remove features that are very relevant when combining with other features. By contrast, both SDRW$_{\text{fs}}$ and CBRW$_{\text{fs}}$ consider the low-level intra- and inter-feature value couplings, which are sensitive to negative value couplings, value frequency distributions and noisy features, resulting in an outlier separability secured reduction of data complexity.

Table 7.4: Complexity Quantification of Data Sets with Feature Subsets Selected by SDRW$_{\text{fs}}$, CBRW$_{\text{fs}}$, ENFW and FULL. The last row shows the percentage of the average complexity reduction compared to the baseline FULL. We use SD = SDRW$_{\text{fs}}$, CB = CBRW$_{\text{fs}}$, EN = ENFW, and FU = FULL to concisely present the results.

	$\boldsymbol{\kappa}_{vcc}$				$\boldsymbol{\kappa}_{het}$				$\boldsymbol{\kappa}_{ins}$				$\boldsymbol{\kappa}_{fnl}$			
Data	**SD**	**CB**	**EN**	**FU**	**SD**	**CB**	**EN**	**FU**	**SD**	**CB**	**EN**	**FU**	**SD**	**CB**	**EN**	**FU**
BM	0.28	**0.19**	0.50	0.21	1.91	1.70	**1.30**	2.03	**0.37**	**0.37**	0.52	**0.37**	0.80	**0.80**	1.00	0.90
Census	0.41	**0.40**	0.57	0.42	1.85	1.83	**1.15**	1.65	**0.24**	**0.24**	0.34	**0.24**	0.71	0.65	0.76	**0.58**
AID362	**0.28**	0.28	0.34	0.32	1.01	1.04	**1.01**	1.14	**0.40**	**0.40**	0.48	**0.40**	0.93	0.93	0.96	**0.86**
w7a	0.13	0.20	**0.10**	0.37	1.01	1.01	**1.00**	1.06	**0.41**	**0.41**	0.44	**0.41**	**0.01**	0.23	0.03	0.48
CMC	0.04	0.04	**0.00**	0.04	1.30	1.30	**1.27**	1.58	**0.34**	**0.34**	0.37	**0.34**	**0.00**	**0.00**	0.50	0.38
APAS	0.25	**0.22**	0.33	0.33	1.06	1.06	**1.02**	1.19	**0.13**	**0.13**	0.28	**0.13**	0.69	**0.66**	0.88	0.81
CelebA	**0.08**	**0.08**	0.12	0.12	1.20	1.16	**1.05**	1.26	**0.20**	**0.20**	0.32	**0.20**	**0.15**	0.20	0.40	0.49
Chess	0.00	0.00	0.00	0.00	**1.22**	**1.22**	2.05	2.24	**0.26**	0.26	0.26	0.26	0.67	0.67	**0.00**	0.33
AD	**0.26**	0.26	0.37	0.46	1.01	1.00	**1.00**	1.01	**0.30**	0.34	0.47	**0.30**	0.01	0.01	**0.00**	0.05
SF	**0.11**	0.15	0.15	0.12	1.72	1.72	**1.08**	1.56	**0.18**	**0.18**	0.30	**0.18**	**0.00**	**0.00**	0.17	0.09
Probe	0.00	0.01	**0.00**	0.01	1.42	1.36	**1.04**	1.32	**0.06**	**0.06**	0.07	**0.06**	0.00	0.00	0.00	0.00
U2R	0.00	0.01	**0.00**	0.02	1.37	1.35	**1.00**	1.29	**0.02**	**0.02**	0.15	**0.02**	**0.00**	0.33	**0.00**	0.17
LINK	**0.00**	**0.00**	0.01	0.01	1.19	1.19	**1.18**	1.39	0.02	0.02	0.02	0.02	0.00	0.00	0.00	0.00
R10	0.03	0.01	**0.00**	0.06	1.00	1.00	**1.00**	1.01	0.34	**0.13**	0.44	**0.13**	0.00	0.00	0.00	0.00
CT	0.00	0.00	0.00	0.00	1.17	1.17	**1.00**	1.10	**0.03**	**0.03**	0.32	**0.03**	0.45	0.45	**0.00**	0.34
Avg.	0.12	**0.12**	0.17	0.17	1.30	1.27	**1.14**	1.39	0.22	0.21	0.32	**0.21**	**0.29**	0.33	0.31	0.36
$\triangledown$ (%)	25	25	-0.2	-	7	8	18	-	-7	-1	-55	-	19	10	14	-

Performance of Different Subsequent Outlier Detectors

The effectiveness of SDRW$_{\text{fs}}$ is further verified by the AUC performance of different subsequent outlier detectors using their resultant feature subsets. Two very different outlier detectors, MarP and iForest, are used here.

The AUC performance of MarP and iForest working on the data ses with feature subsets is shown in Table 7.5. SDRW$_{\text{fs}}$-empowered MarP and iForest obtains substantial improvements than ENFW (12%), RADM (17%) and FULL (7%), regardless of the difference working mechanisms of MarP and iForest. The SDRW$_{\text{fs}}$-empowered MarP and iForest significantly outperform their counterparts empowered by ENFW and RADM at the 99% confidence level. Although they use 50% less features, they significantly outperform MarP and iForest working on data with full feature sets at the 95% confidence level.

SDRW_{fs}-based MarP and iForest do not show significantly improvement over that using CBRW_{fs}, this may be because MarP and iForest do not capture high-order information, and as a result, the high-order couplings reserved by SDRW_{fs} are not really caught up by the two outlier detectors. It is interesting to note that MarP and iForest using SDRW_{fs} and ENFW perform much better than all their counterparts on *w7a*. This improvement is mainly because SDRW_{fs} and ENFW remove more than 95% of the noisy features with little or no loss to the outlier separability in this data, as shown in Table 4.3.

Table 7.5: AUC Performance of MarP and iForest Using SDRW_{fs}, CBRW_{fs}, ENFW, RADM, and FULL.

	MarP					**iForest**				
Data	SDRW_{fs}	CBRW_{fs}	ENFW	RADM	FULL	SDRW_{fs}	CBRW_{fs}	ENFW	RADM	FULL
BM	0.5627	**0.5926**	0.4886	0.5181	0.5584	0.5618	**0.5836**	0.5297	0.5544	0.5762
Census	0.6052	**0.6258**	0.4525	0.5490	0.5899	0.5801	**0.6106**	0.4403	0.5201	0.5378
AID362	0.6612	**0.6620**	0.5909	0.6074	0.6270	**0.6641**	0.6525	0.6155	0.6267	0.6485
w7a	0.8413	0.7654	**0.8633**	0.4594	0.4748	0.8084	0.7432	**0.8251**	0.3946	0.4053
CMC	0.6474	**0.6474**	0.5082	0.5062	0.5417	**0.6609**	0.6607	0.5288	0.5164	0.5746
APAS	0.8454	**0.8569**	0.6346	0.5995	0.6193	0.8385	**0.8426**	0.6372	0.5543	0.4792
CelebA	**0.8652**	0.8597	0.7785	0.7102	0.7358	0.8388	**0.8438**	0.7799	0.6764	0.6797
Chess	**0.7574**	**0.7574**	0.6378	0.6076	0.6447	**0.6859**	0.6138	0.6241	0.5829	0.6124
AD	**0.8256**	0.7624	0.6603	0.6888	0.7033	**0.8206**	0.7620	0.6592	0.6775	0.7084
SF	0.8343	0.8157	0.6666	0.8181	**0.8446**	0.7838	0.7667	0.6856	0.7660	**0.7865**
Probe	**0.9837**	0.9805	0.9307	0.8951	0.9800	**0.9842**	0.9751	0.8797	0.8990	0.9762
U2R	**0.9937**	0.8846	0.8582	0.7911	0.8848	**0.9879**	0.9776	0.7854	0.8168	0.9781
LINK	**0.9985**	**0.9985**	0.9938	0.9723	0.9977	**0.9986**	0.9984	0.9797	0.9636	0.9917
R10	0.8705	**0.9893**	0.7648	0.9627	0.9866	0.8705	**0.9926**	0.7566	0.9541	0.9796
CT	0.8570	0.8570	0.8581	0.6154	**0.9773**	0.9122	0.9072	0.8816	0.6374	**0.9364**
Avg.	**0.8099**	0.8037	0.7125	0.6867	0.7444	**0.7998**	0.7954	0.7072	0.6760	0.7247
p-value	SDRW vs.	0.6772	0.0009	0.0016	0.0340	-	0.2890	0.0005	0.0013	0.0262
	CBRW vs.		0.0023	0.0005	0.0113	-	-	0.0023	0.0004	0.0131

7.5.4 Scalability Test

Both SDRW-based outlier detection and feature selection are linear consolidation of the value outlierness. Hence, they have similar scalability. Here, we show the scalability of SDRW_{od}.

The scalability of SDRW_{od} w.r.t. data size is evaluated using four subsets of the largest data set *LINK*. The smallest subset contains 64,000 objects, and subsequent subsets are increased by a factor of four, until the largest subset which contains 4,096,000 objects.

The scaleup test results w.r.t. data size are presented in the left panel in Figure 7.3. As expected, all the seven detectors have runtime linear w.r.t. data size. SDRW_{od} runs faster than iForest and Sp by a factor of more than 20 and 30, respectively. SDRW_{od} runs slightly faster than CBRW_{od}, since SDRW_{od} requires no iteration to obtain the value outlierness. Nevertheless, CBRW_{od} runs faster than iForest and Sp. Both SDRW_{od} and CBRW_{od} are slightly slower than MarP but comparably fast to FPOF.

The scaleup test w.r.t. the number of features is conducted using seven synthetic data sets. The data sets have the same number of objects, i.e., 10,000 objects. The data set with the smallest number of features contains 10 features, and subsequent data sets are increased by a factor of two, until the data set with the largest number of features contains 640 features.

The results reported in the right panel in Figure 7.3 show that, as expected, SDRW_{od}

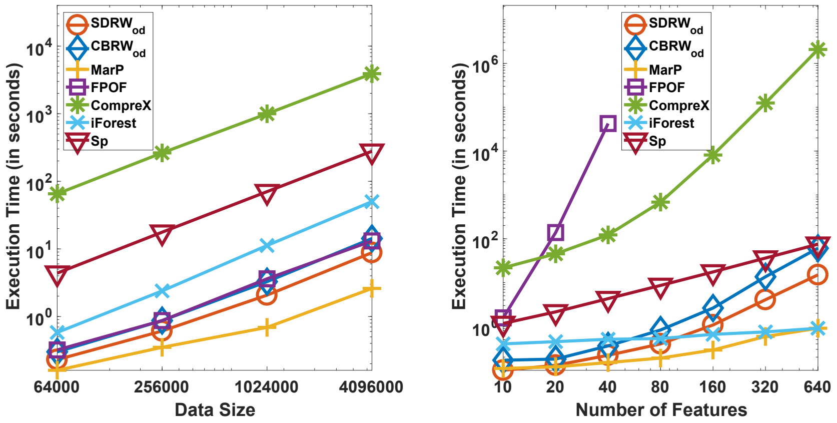


Figure 7.3: Scale-up Test Results of the Seven Detectors w.r.t. Data Size and Dimensionality. Logarithmic scales are used in both axes. Note that FPOF runs out-of-memory when the number of features reaches 80.

and CBRW$_{od}$ have runtime nearly linear w.r.t. the number of features, which run more than five orders of magnitude faster than FPOF. SDRW$_{od}$ and CBRW$_{od}$ run much faster than CompreX by a factor of more than 600 and 250 in terms of runtime ratio[1], respectively. Compared to Sp, SDRW$_{od}$ and CBRW$_{od}$ run faster on data sets with lower dimensions, but they may become slower on data sets with higher dimensions. This is because the runtime of SDRW$_{od}$ and CBRW$_{od}$ increase at a much faster rate than Sp. Since SDRW$_{od}$ and CBRW$_{od}$ model much more complex data characteristics than MarP and iForest, they run substantially slower than these two competitors, but with significantly better accuracy in terms of AUC, as shown in Table 4.2.

7.6 Summary

This chapter introduces an outlier detection framework, HOCOF, which extends the CUOT framework by incorporating high-order value coupling relations. HOCOF is further implemented by the SDRW method that uses the densities of multi-granularity subgraphs of the value graph to capture the high-order coupling information. Extensive experiments show that (i) the SDRW-based outlier detector performs significantly better than six state-of-the-art detectors - CBRW, MarP, FPOF, CompreX, iForest and Sp, at the 95% confidence level; (ii) our SDRW-based outlying feature selection method considerably reduces the complexities of different data sets while retains their outlier separability, which enables two different outlier detectors to significantly outperform the competing feature selection methods; and (iii) SDRW has linear or nearly linear time complexity w.r.t. data size and the number of features.

Compared to CBRW, SDRW is significantly better in terms of overall AUC performance, computational time, parameter tuning effort and tolerance to noisy features.

[1] Since CompreX was implemented in a different programming language to the other methods, the runtime between CompreX and other methods is incomparable. Instead, we compare them in terms of runtime ratio, i.e., the runtime on a larger/higher-dimensional data set divided by that on a smaller/lower-dimensional data set, for a fairer comparison. Since the data size and the increasing factor of dimensionality are fixed, the runtime ratio is comparable across the methods in different programming languages.

Therefore, SDRW is generally recommended when the complexity of a given problem is unknown, or when the problem has high requirements on computational cost, users' inputs, or robustness to feature noise. However, SDRW seems to less effective than CBRW to identify outliers in cleaner data sets, e.g., data with only a few noisy features and some strongly relevant features like *Census* and *R10*, since SDRW reduces the effect of δ by replacing δ with ad. Therefore, CBRW is recommended for outlier detection in cleaned data sets.

SDRW uses the value subgraphs to capture the couplings of multiple values. Since the value graph is built upon pairwise value interactions, the value subgraphs may not be able to capture the couplings involving the concurrence information of multiple values. Frequent/infrequent patterns derived from pattern mining may be helpful for capturing this type of information, but using pattern mining-based methods may be too computationally costly. More advanced methods are needed to efficiently and effectively to capture more sophisticated high-order value coupling information.

Part III

Feature/object-level Coupled Outlier Factors

Learning Couplings of Feature- or Object-level Outlier Factors

In addition to the outlierness estimation of data objects, quantifying the outlierness of features is another important goal in outlier detection, since it can determine relevant features for subsequent outlier detection methods, or explain the reasons why a data object is reported as an outlier.

The interactions between feature values intrinsically contribute to the interactions between features or between data objects. In this part, we examine the higher-level (feature or object level) couplings of outlier factors to understand the abnormality of features or objects, and their ability in addressing challenging outlier detection problems:

- **Two-way feature**-level couplings, which enable a parameter-free outlying feature selection with approximation guarantees (Chapter 8);
- **Sequential** couplings at the **object level**, which provide manners for mutual refinement of feature selection and outlier detection that is important in high-dimensional outlier detection (Chapter 9);

For each exploration, we have the same structure as the chapters in Part II, including motivation, abstract framework, instantiation, and theoretical and empirical justifications.

Chapter 8

Two-way Couplings of Feature-level Outlier Factors

8.1 Introduction

How can we know the outlierness of a feature, i.e., its relevance to outlier detection? and how is the outlierness of one feature influenced by that of the other features? In this chapter, we explore methods to estimate the outlierness of features for different important outlier detection applications, e.g., feature selection for subsequent outlier detection methods or outlier explanation.

However, it is very challenging to determine the relevance of features to outlier detection because (i) we often do not have class labels due to its unsupervised nature, (ii) there are complex interactions between noisy/redundant features and relevant features, and (iii) the data distribution is extremely imbalanced. We demonstrated in Chapter 4 that the outlierness of a feature can be effectively measured by the consolidation of the outlierness of values contained by the feature, but the outlierness of a feature is computed independently from that of the other features therein, which may be ineffective for data with interdependent features. In an attempt to address this issue, this chapter explores the two-way couplings of the feature outlierness. Motivated by the success of value outlierness-based estimation of feature outlierness in Chapter 4, as shown in Figure 8.1, here we define the feature outlierness using the feature interactions in a feature-feature graph derived from the value graph, which helps capture more reliable feature outlierness than the method introduced in Chapter 4 when handling data with coupled features.

Feature selection is of great importance to outlier detection. This is because outliers are easily masked as normal objects in irrelevant/noisy features - features for which outliers do not demonstrate any suspected behaviors. For example, in loan fraud detection, suspects may be spotted by partial features, such as marital status and income level, while they may fake themselves as normal with other features, such as education and profession. In addition, many data sets contain a large number of *redundant features* - weakly relevant features that contribute a very limited capability, or none, to outlier detection when combined with other features, e.g., property holdings to income level.

Eliminating noisy and redundant features may therefore substantially improve the

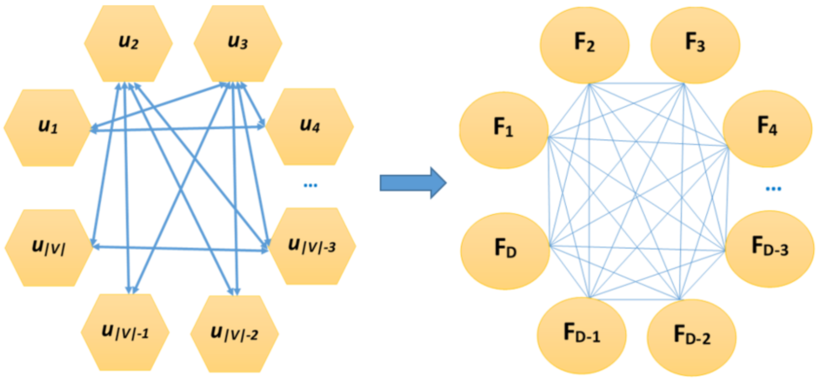


Figure 8.1: Two-way Couplings of Feature-level Outlier Factors. The outlierness of features is inferred from the value-level outlierness interdependence.

effectiveness and efficiency of subsequent outlier detection. This is particularly true for outlier detection methods for categorical data (e.g., [7, 34, 57, 108, 114, 121]), which are mainly pattern-based methods. These methods search for outlying/normal patterns and employ pattern frequency as a direct outlierness measure. However, these methods fail to perform effectively and efficiently in data sets that have the aforementioned characteristics for two main reasons: (i) many noisy features mislead the pattern search and result in a large proportion of faulty patterns and a high ‘false positive’ rate; and (ii) feature redundancy results in numerous redundant patterns and considerably downgrades the efficiency of the pattern search and outlier detection.

In this chapter, by utilizing hierarchical value-feature couplings, we propose a novel Coupled Unsupervised Feature Selection framework (CUFS) to filter out noisy and redundant features for outlier detection in categorical data. CUFS first estimates the outlierness of feature values by modeling the low-level intra- and inter-feature value couplings. These value couplings reflect the intrinsic data characteristics and facilitate the differentiation between relevant and other features. We further incorporate the value-level outlierness into feature outlierness by learning value-to-feature interactions. This *value-to-feature outlierness* is then mapped onto graph representations, on which existing graph mining techniques and theories are used to identify the desirable relevant feature subset.

We further instantiate CUFS to a Dense Subgraph-based Feature Selection method (DSFS), which synthesizes the advantages of hierarchical couplings captured in CUFS and the dense subgraph search theories. DSFS computes value outlierness by integrating intra-feature value frequency deviation and inter-feature value correlation, and obtains feature outlierness by a linear combination of value outlierness. The feature subset max-relevance criterion and sequential search strategy are then used to identify the most relevant feature subset, which is equivalent to the discovery of the densest subgraph of a feature graph.

Accordingly, this chapter makes the following two main contributions.

i. We propose a novel and flexible coupled unsupervised feature selection (CUFS) framework for detecting outliers in categorical data, in which relevant features are highly mixed with noisy and redundant features. CUFS captures two-way feature interactions by modeling the outlierness (relevance) of features w.r.t. hierarchical intra- and inter-feature couplings, which distinguish relevant features from noisy and redundant features.

ii. The CUFS framework is instantiated to a *parameter-free* feature subset selection method DSFS. We prove that the feature subset selected by DSFS has a *2-approximation* to the optimal subset. This demonstrates the flexibility of CUFS in enabling state-of-the-art graph mining techniques to tackle the feature selection challenge in unlabeled and imbalanced categorical data.

Extensive experiments show that (1) DSFS obtains a large average feature reduction rate (48%) on 15 data sets with a variety of complexities, including different levels of noisy and redundant features, and greatly improves three different types of pattern-based outlier detectors in AUC and/or runtime performance; (2) DSFS substantially outperforms its feature weighting-based contender (maximally 94% improvement on a data set); and (3) DSFS achieves good scalability w.r.t. data size (linear to data size, completing execution within one second for a data set with over one million objects) and the number of features (completing the execution within 20 seconds for a data set with over 1000 features).

The rest of this chapter is organised as follows. CUFS is detailed in Section 8.2. DSFS is introduced in Section 8.3. A theoretical analysis of DSFS is presented in Section 8.4. Empirical results are provided in Section 8.5. This work is summarized in Section 8.6.

8.2 The Proposed CUFS Framework

In this section, we introduce the CUFS framework. CUFS builds and integrates two-level hierarchical couplings, i.e., feature value couplings and feature couplings, toward a proper estimation of the feature relevance to outlier detection. Specifically, it learns the intra- and inter-feature value couplings to compute outlierness on the feature value level and constructs a *value graph* with the outlierness being the edge weights. We then feed the value graph to the feature-level coupling analysis and construct a *feature graph* by aggregating the value-level outlierness. Our coupled feature selection framework for unsupervised outlier detection (i.e., CUFS) is shown in Figure 8.2.

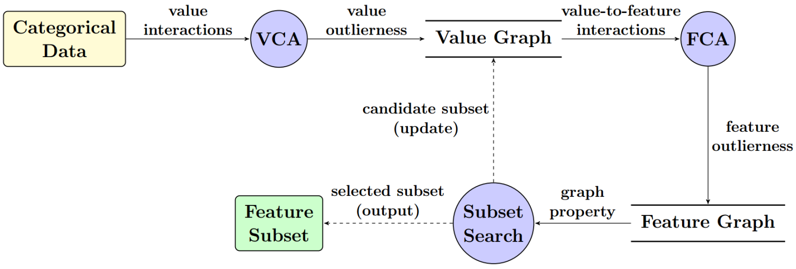


Figure 8.2: The Proposed CUFS Framework. VCA and FCA are short for Value Coupling Analysis and Feature Coupling Analysis, respectively.

The value coupling analysis captures the intrinsic interactions between the values of data objects, which enables a proper estimation of the value outlierness in data and helps distinguish outlying values from noisy values. As the features build their capability on their values, feature outlierness is thus modeled by aggregating value outlierness in terms of the value-to-feature interactions. Such feature couplings distinguish useful features from noisy and redundant features.

As a result of these factors, CUFS builds on the deep understanding of intrinsic data characteristics in outlying data, and effectively combines the advantages of data-driven complex feature relation analysis with unsupervised feature selection and graph theories for outlier detection. It has the graph properties and a feature subset search strategy as input to search and select a feature subset for outlier detection.

8.2.1 Value Graph Construction

The outlying behaviors of a feature value are captured by intra-feature and inter-feature value couplings. Accordingly, we define *value couplings* and *value graph* as follows.

Definition 8.1 (Value Coupling)**.** *The couplings in a value v of feature F are represented by a three-dimensional tuple $VC = (\mathsf{F}, \delta(\cdot), \eta(\cdot, \cdot))$, where*

- $\mathsf{F} \in \mathcal{F}$*, where $\mathcal{F}$ is the feature space.*
- $\delta(\cdot)$ *captures the outlying behaviors of the value v w.r.t. the value interactions within feature F. For example, $\delta(\cdot)$ may be a function of deviations of value frequencies from the mode frequency or value similarities, etc.*
- $\eta(\cdot, \cdot)$ *captures the outlying behaviors of the value v w.r.t. interactions with the values in the rest of the features in $\mathcal{F}$. For example, $\eta(\cdot, \cdot)$ may be a function of value co-occurrence frequency, conditional probabilities or other value correlation quantization methods.*

With the value couplings of all feature values, a *value graph* can be built to present their relationship.

Definition 8.2 (Value Graph)**.** *The value graph G is defined as $\mathsf{G} =< \mathcal{V}, \mathbf{A}, g(\delta(\cdot), \eta(\cdot, \cdot)) >$, where a value $v \in \mathcal{V}$ represents a node, the entry of the weighted adjacency matrix $\mathbf{A}(v, v')$ (i.e., edge weight) is determined by function $g(\cdot, \cdot)$, which is a joint function of $\delta(v)$ and $\eta(v, v')$, $\forall v, v' \in \mathcal{V}$.*

The graph G can be an undirected or directed graph depending on how the edge weight is defined.

One major benefit of mapping the value couplings to the value graph is that we can utilize the value graph properties (e.g., ego-network, shortest path, node centrality, or random walk distance [26]) to infer deeper value interactions and to further explore feature interactions by building the following feature graph.

8.2.2 Feature Graph Construction

The feature couplings are derived from the value couplings to capture the value-to-feature interactions.

Definition 8.3 (Feature Coupling)**.** *The couplings within a feature* F *are described as a three-dimensional tuple* $FC = (dom(\mathsf{F}), \delta^*(\cdot), \eta^*(\cdot,\cdot))$*, where*

- $dom(\mathsf{F})$ *is the domain of the feature* F*, which consists of a finite set of possible feature values contained in* F*.*
- $\delta^*(\cdot)$ *computes the outlying degree of* F *based on its value outlierness* $\delta(\cdot)$*. For example,* $\delta^*(\mathsf{F})$ *may be a linear or non-linear function for combining all* $\delta(v)$*,* $\forall v \in dom(\mathsf{F})$*.*
- $\eta^*(\cdot,\cdot)$ *captures the outlying degree of* F *w.r.t. its value interactions with other features in* $\mathcal{F}$*. Specifically, given* $\forall \mathsf{F}' \in \mathcal{F} \setminus \mathsf{F}$*,* $\eta^*(\mathsf{F},\mathsf{F}')$ *may be a linear or non-linear function for incorporating* $\eta(v,v')$ *for* $\forall v \in dom(\mathsf{F})$ *and* $\forall v' \in dom(\mathsf{F}')$*.*

These couplings are then mapped into a feature graph G^*.

Definition 8.4 (Feature Graph)**.** *The feature graph* G^* *is defined as* $\mathsf{G} =< \mathcal{F}, \mathbf{A}^*, h(\delta^*(\cdot), \eta^*(\cdot,\cdot)) >$*, where a feature* $\mathsf{F} \in \mathcal{F}$ *represents a node and the entry of the weighted adjacency matrix* $\mathbf{A}^*(\mathsf{F},\mathsf{F}')$ *is determined by* $h(\cdot,\cdot)$*, a function combining* $\delta^*(\mathsf{F})$ *and* $\eta^*(\mathsf{F},\mathsf{F}')$ *for* $\forall \mathsf{F}, \mathsf{F}' \in \mathcal{F}$*.*

With the feature graph, existing graph mining algorithms and theories (e.g., dense subgraph discovery, graph partition and frequent graph pattern mining [26]) can then be applied to identify the most relevant feature subset for outlier detection. As presented in Section 8.3, by utilizing dense subgraph discovery theories, the CUFS instance can efficiently retain a 2-approximation feature subset.

8.2.3 Feature Subset Selection

Our goal here is to find a feature subset, i.e., a subgraph of the feature graph, which reserves feature nodes with high outlierness while at the same time reduces redundancy between the reserved features.

The feature subset search contains two major ingredients: *search strategy* and *objective function* (i.e., subset evaluation criteria) [79]. Typical search strategies include complete search, sequential forward or backward search, and random search. *Complete search* can obtain an optimal feature subset, but its runtime is prohibitive for high-dimensional data. *Sequential search* and *random search* are heuristic and result in a suboptimal subset, but they are more practical than complete search as they have much better efficiency.

A generic objective function for this context is:

$$\arg\max_{\mathcal{S}} \; J(\mathcal{S}), \tag{8.1}$$

where $J(\cdot)$ is a function evaluating the outlierness in the feature subset $\mathcal{S}$, which needs to be specified based on the chosen search strategy.

As illustrated in Figure 8.2, we may need to iteratively update the value graph and feature graph during the subset searching, e.g., when adding or removing features in sequential search, before obtaining an optimal subset.

8.3 A CUFS Instance: DSFS

The CUFS framework can be instantiated by first specifying the three functions δ, η and g for constructing the value graph and the other three functions δ^*, η^* and h for building the feature graph. A subset search strategy can then be formed by utilizing the graph properties of the feature graph to identify the desired feature subset.

We illustrate the instantiation of CUFS by identifying the dense subgraph of the feature graph, i.e., DSFS. DSFS uses the recursive backward elimination search with the subgraph density as the objective function.

8.3.1 Specifying Functions δ, η and g for the Value Graph

Per the definition of outliers, the frequencies of values are closely related to the degree of outlierness. Hence, the outlierness of feature values is dependent on its intra-feature frequency distribution and inter-feature value co-occurrence frequencies. Motivated by this, we specify the intra- and inter-feature value outlierness in terms of frequency deviation and confidence values.

Definition 8.5 (Intra-feature Value Outlierness δ)**.** *The intra-feature outlierness $\delta(v)$ of a feature value $v \in dom(\mathsf{F})$ is defined as the extent to which its frequency deviates from the frequency of the mode:*

$$\delta(v) = \frac{freq(m) - freq(v) + \epsilon}{freq(m)}, \tag{8.2}$$

where m is the mode of the feature F, $freq(\cdot)$ is a frequency counting function and $\epsilon = \frac{1}{N}$.

In Equation (8.2), the mode frequency is used as a benchmark, and the more the frequency of a feature value deviates from the mode frequency, the more outlying the value is. We use $\epsilon = \frac{1}{N}$ to estimate the outlierness of the mode, which is proportional to the data size. $\delta(\cdot)$ makes the outlierness of values from different frequency distributions more comparable, which differs from many existing studies [7, 57, 108] in which the outlierness of each pattern is measured without considering its associated frequency distributions.

Definition 8.6 (Inter-feature Value Outlierness η)**.** *The inter-feature outlierness $\eta(v, v')$ of a value $v \in dom(\mathsf{F})$ and another value $v' \in dom(\mathsf{F}')$ is defined as follows:*

$$\eta(v, v') = \delta(v) conf(v, v') \delta(v'), \tag{8.3}$$

where $conf(v, v') = \frac{freq(v,v')}{freq(v')}$.

$\eta(v, v')$ models a simple outlierness diffusion effect. That is, a value has high outlierness if it has a strong correlation with outlying values. For example, a person experiencing both weight loss and frequent urination is more suspected of having health problems than someone who has the symptoms of weight loss and normal urination, assuming weight loss and frequent urination are outlying symptoms.

Definition 8.7 (Edge Weighting Function g for Value Graph G)**.** *The edge weight of the value graph* G*, i.e., the entry* (v, v') *of the weight matrix* $\mathbf{A}$*, is defined as follows:*

$$\mathbf{A}(v, v') = g(v, v') = \begin{cases} \delta(v), & v = v' \\ \eta(v, v'), & otherwise \end{cases}. \tag{8.4}$$

We have $\delta(\cdot) \in (0, 1)$ and $\eta(\cdot, \cdot) \in [0, 1)$ according to Equations (8.2) and (8.3), and thus $g(\cdot, \cdot) \in [0, 1)$. That is, the edge weight would be zero iff two distinctive nodes v and v' have no association.

Note that although the two cases in Equation (8.4) are in slightly different ranges, they are used independently in the next section to avoid incomparable issues. We also discuss in Section 8.4 how this function helps to distinguish noisy features from relevant features.

Overall, the value graph G has the following properties.

i. G is a directed graph with self loops, as there exists $\mathbf{A}(v, v') \neq \mathbf{A}(v', v)$ and $\mathbf{A}(v, v) \neq 0$.

ii. Its adjacency matrix $\mathbf{A}$ is a value outlierness matrix, representing the outlying degree of individual values and pairs of distinctive values. The larger a matrix entry is, the higher the outlierness is.

8.3.2 Specifying Functions δ^*, η^* and h for the Feature Graph

For simplicity and the consideration of common scenarios, we assume that the intra-feature and inter-feature value outlierness measures are linearly dependent. Accordingly, we estimate the intra- and inter-feature outlierness of a feature and their integration for feature-level outlierness by simply summing its associated δ and η values.

Definition 8.8 (Intra-feature Outlierness δ^*)**.** *The intra-feature outlierness of a feature* $\mathsf{F} \in \mathcal{F}$ *is specified below:*

$$\delta^*(\mathsf{F}) = \sum_{v \in dom(\mathsf{F})} \delta(v). \tag{8.5}$$

Definition 8.9 (Inter-feature Outlierness η^*)**.** *The inter-feature outlierness of a feature* F *w.r.t. feature* F' *is quantified as:*

$$\eta^*(\mathsf{F}, \mathsf{F}') = \sum_{v \in dom(\mathsf{F}), v' \in dom(\mathsf{F}')} \eta(v, v'). \tag{8.6}$$

Similar to g, we specify the function h using intra-feature outlierness as diagonal entries and inter-feature outlierness as off-diagonal entries in the weight matrix $\mathbf{A}^*$.

Definition 8.10 (Edge Weighting Function h for Feature Graph G^*)**.** *The edge weight* $\mathbf{A}^*(\mathsf{F}, \mathsf{F}')$ *of the feature graph* G^**, i.e., the entry* $(\mathsf{F}, \mathsf{F}')$ *of* $\mathbf{A}^*$*, is measured as:*

$$\mathbf{A}^*(\mathsf{F}, \mathsf{F}') = h(\mathsf{F}, \mathsf{F}') = \begin{cases} \delta^*(\mathsf{F}), & \mathsf{F} = \mathsf{F}' \\ \eta^*(\mathsf{F}, \mathsf{F}'), & otherwise \end{cases}. \tag{8.7}$$

Note that, to make the entries in $\mathbf{A}^*$ comparable, δ^* and η^* are normalized into the same range $[0,1]$ for further use in feature subset searching.

The feature graph G^* has the following key properties.

i. G^* is a complete graph with self loops, as $\delta^*(\cdot) > 0$ and $\eta^*(\cdot,\cdot) > 0$.

ii. G^* is an undirected graph, as we always have $\mathbf{A}^*(\mathsf{F},\mathsf{F}') = \mathbf{A}^*(\mathsf{F}',\mathsf{F})$ for $\forall \mathsf{F}',\mathsf{F} \in \mathcal{F}$.

iii. Its adjacency matrix $\mathbf{A}^*$ is a feature outlierness matrix, representing the outlying degree of features and their combinations. Larger values in $\mathbf{A}^*$ indicate higher outlierness.

iv. The total edge weight of a feature node F is large if both of its intra- and inter-feature outlierness are high.

8.3.3 The Search Strategy

Our target is to find a subset of features with the highest relevance to outlier detection, i.e., with the highest outlierness. A feature has high outlierness if it has large edge weights in the feature graph G^*, according to the properties (3) and (4) of G^*. However, simply selecting the top-ranked k features does not necessarily obtain the best feature subset, since the outlierness of a feature also depends on its coupled features. This distinguishes our design from existing methods that overlook feature interactions.

Motivated by the *max-relevance* idea in [99], the following *max-relevance objective function* is designed to search for the most relevant feature subset $\mathcal{S}$:

$$\arg\max_{\mathcal{S}} \frac{1}{|\mathcal{S}|} \sum_{\mathsf{F}\in\mathcal{S}} \sum_{\mathsf{F}'\in\mathcal{S}} \mathbf{A}^*(\mathsf{F},\mathsf{F}'). \tag{8.8}$$

In other words, we specify $J(\cdot)$ in Equation (8.1) as $J(\mathcal{S}) = \frac{1}{|\mathcal{S}|}\sum_{\mathsf{F}\in\mathcal{S}}\sum_{\mathsf{F}'\in\mathcal{S}} \mathbf{A}^*(\mathsf{F},\mathsf{F}')$.

Searching the exact $\mathcal{S}$ is computationally intractable for high dimensional data, as the search space is 2^D. A heuristic sequential search strategy, namely Recursive Backward Elimination (RBE), is used to search for an approximately best subset. RBE conducts an iterative search as shown in Algorithm 8.1. In the next section, we prove that the resultant subset is a 2-approximation to the optimum.

Algorithm 8.1 *RBE ($\mathcal{F}$)*

```
Input: $\mathcal{F}$ - full feature set
Output: $\mathcal{S}$ - the feature subset selected
while $|\mathcal{F}| > 0$ do
    for $\mathsf{F} \in \mathcal{F}$ do
        Compute $J(\mathcal{F} \setminus \mathsf{F})$
    end for
    Remove the feature $\mathsf{F}$ that results in the largest $J(\mathcal{F} \setminus \mathsf{F})$
end while
return Return the subset with the largest $J(\cdot)$ as $\mathcal{S}$
```

8.3.4 The Algorithm and Its Time Complexity

Algorithm 8.2 presents the procedures of the proposed instantiation DSFS. Steps (1-7) and (8-13) construct the value graph G and the feature graph G^*, respectively. Steps (14-19) obtain the feature subset $\mathcal{S}$. As proved in Lemma 8.0.2, Steps (16-17) are equivalent to Steps (2-5) in RBE in Algorithm 8.1.

DSFS requires only one database scan to compute the intra- and inter-feature value outlierness in Steps (1-7), and thus has $O(N)$. DSFS has $O(D^2)$, as inner loops are required in order to generate the adjacency matrices of the value graph and the feature graph. However, the computation within the inner loop, i.e., Steps (5) and (11), is a very simple multiplication and value assignment, enabling it to complete the execution quickly in high dimensional data. Hence, DSFS has good scalability w.r.t. data size and the number of features.

Algorithm 8.2 *DSFS ($\mathcal{X}$)*

```
Input: $\mathcal{X}$ - data objects
Output: $\mathcal{S}$ - the feature subset selected
 1: Initialize $\mathbf{A}$ as a $|V| \times |V|$ matrix
 2: for $\mathsf{F} \in \mathcal{F}$ do
 3:    Compute $\delta(v)$ for each $v \in dom(\mathsf{F})$
 4:    for $\mathsf{F}' \in \mathcal{F}$ do
 5:       $A(v, v') \leftarrow g(v, v'), \forall v' \in dom(\mathsf{F}')$
 6:    end for
 7: end for
 8: Initialize $\mathbf{A}^*$ as a $|D| \times |D|$ matrix
 9: for $\mathsf{F} \in \mathcal{F}$ do
10:    for $\mathsf{F}' \in \mathcal{F}$ do
11:       $\mathbf{A}^*(\mathsf{F}, \mathsf{F}') \leftarrow h(\mathsf{F}, \mathsf{F}')$
12:    end for
13: end for
14: Set $\mathcal{S} \leftarrow \mathcal{F}$ and $s \leftarrow den(\mathbf{A}^*)$
15: for $i = 1$ to $D$ do
16:    Find $\mathsf{F}$ that has the smallest weighted degree in $\mathbf{A}^*$
17:    $\mathcal{F} \leftarrow \mathcal{F} \setminus \mathsf{F}$ and update $\mathbf{A}^*$
18:    $\mathcal{S} \leftarrow \mathcal{F}$ and $s \leftarrow den(\mathbf{A}^*)$ if $s \leq den(\mathbf{A}^*)$
19: end for
20: return $\mathcal{S}$
```

8.4 Theoretical Analysis

Theoretical analysis is provided for DSFS in the first subsection and we then discuss why DSFS can handle noisy and redundant features in the remaining two subsections.

Approximation

Following the definition of subgraph density for *unweighted graphs* in [28, 65], we define the subgraph density for *weighted graphs* by replacing the total number of edges with the total weight defined in our graph.

Definition 8.11 (Subgraph Density). *The density of an undirected weighted subgraph $\mathcal{S}$ is its average weighted degree:*

$$den(\mathcal{S}) = \frac{vol(\mathcal{S})}{|\mathcal{S}|}, \tag{8.9}$$

where $vol(\mathcal{S}) = \frac{\sum_{\mathsf{F}\in\mathcal{S}}\sum_{\mathsf{F}'\in\mathcal{S}}\mathbf{A}^*(\mathsf{F},\mathsf{F}')}{2}$ is the volume of $\mathcal{S}$.

With Equations (8.8) and (8.9), we have the following lemma.

Lemma 8.0.1 (Equivalence to the Densest Subgraph Discovery). *Equation (8.8) is equivalent to calculating the maximum of $den(\mathcal{S})$, i.e., the densest subgraph of the feature graph G^*.*

Proof. It is easy to see that Equation (8.8) is equivalent to maximizing $2den(\mathcal{S})$, and thus the densest subgraph of G^* is the exact solution $\mathcal{S}$ to Equation (8.8). □

We show below that the RBE search with quadratic time complexity can be simplified to an equivalent procedure with linear time complexity. Following theorems of dense subgraph discovery in unweighted graphs [28, 65], we further prove that the RBE search on the weighted graph G^* achieves a feature subset with a 2-approximation to the optimum.

Lemma 8.0.2 (Search Strategy Equivalence). *Steps (2-5) of RBE in Algorithm 8.1 are equivalent to the removal of the feature node F with the smallest weighted degree.*

Proof. If the feature node F has the smallest weighted degree, $\sum_{\mathsf{F}'\in\mathcal{F}\setminus\mathsf{F}}\sum_{\mathsf{F}''\in\mathcal{F}\setminus\mathsf{F}}\mathbf{A}^*(\mathsf{F}',\mathsf{F}'')$ is the largest in the current iteration. Since $\frac{1}{|\mathcal{F}\setminus\mathsf{F}'|}$ is the same $\forall\mathsf{F}'\in\mathcal{F}$, the removal of F results in the largest $J(\cdot)$. □

Instead of recursively computing $J(\cdot)$ for each feature in each iteration, we therefore remove the feature node with the smallest weighted degree to achieve the same result, which avoids the inner loop and has linear time complexity.

Theorem 8.1 (2-Approximation). *The feature subset $\mathcal{S}$ created by the RBE search is a 2-approximation to the optimal subset.*

Proof. Let $\mathcal{S}_{opt}$ be the set of feature nodes in the densest subgraph. According to Lemma 8.0.1, below we show $den(\mathcal{S}) \geq \frac{den(\mathcal{S}_{opt})}{2}$ to prove the theorem.

Since $\mathcal{S}_{opt}$ forms the densest subgraph, we have

$$den(\mathcal{S}_{opt}) = \frac{vol(\mathcal{S}_{opt})}{|\mathcal{S}_{opt}|} \geq \frac{vol(\mathcal{S}_{opt}) - d(\mathsf{F})}{|\mathcal{S}_{opt}| - 1},\ \forall\mathsf{F}\in\mathcal{S}_{opt},$$

where $d(\mathsf{F}) = \sum_{\mathsf{F}'\in\mathcal{S}_{opt}}\mathbf{A}^*(\mathsf{F},\mathsf{F}')$ denotes the weighted degree of a feature node. After some replacements, we have $d(\mathsf{F}) \geq den(\mathcal{S}_{opt})$, $\forall\mathsf{F}\in\mathcal{S}_{opt}$, i.e., every node in $\mathcal{S}_{opt}$ has a weighted degree of at least $den(\mathcal{S}_{opt})$.

Let $\mathcal{T}_i$ be the set of feature nodes left after the i-th node is removed. Considering the iteration of RBE, let $\mathcal{T}_j$ be the set of remaining nodes when the first node F contained in the optimal subset $\mathcal{S}_{opt}$ is removed, so $\mathcal{T}_{j-1}$ is the set of remaining nodes before node F is removed, which indicates that $d(\mathsf{F}') \geq den(\mathcal{S}_{opt})$, $\forall \mathsf{F}' \in \mathcal{T}_{j-1}$, according to Lemma 8.0.2. Since G^* is a complete graph, we have

$$2vol(\mathcal{T}_{j-1}) \geq den(\mathcal{S}_{opt})|\mathcal{T}_{j-1}|.$$

We then have

$$den(\mathcal{T}_{j-1}) = \frac{vol(\mathcal{T}_{j-1})}{|\mathcal{T}_{j-1}|} \geq \frac{den(\mathcal{S}_{opt})}{2}.$$

Since RBE returns the feature subset $\mathcal{S}$ with the largest subgraph density over all iterations and $\mathcal{T}_{j-1}$ is one of the feature subset candidates, $den(\mathcal{S})$ has at least $\frac{den(\mathcal{S}_{opt})}{2}$. □

Handling Noisy Features

According to Equation (8.4), a value node has high outlierness if δ and η are high. Given a noisy feature value that occurs infrequently but is contained by normal objects, since it has low frequency, its intra-feature value outlierness δ is high. However, since these noisy values tend to be more frequent or only contained by normal objects, they are presumed to have stronger couplings with normal values versus weak/no couplings with outlying values. On the other hand, truly outlying values have high outlierness in terms of both δ and η, because the frequency is low and the couplings with other outlying values are strong, and thus the overall value outlierness is often much higher than that of noisy feature values. Since the intra- and inter-feature outlierness is linearly correlated to intra- and inter-feature value outlierness respectively, the intra- and inter-feature outlierness of outlying features is also higher than that of noisy features. As a result, the noisy features are removed during the iterative procedure in RBE, while the relevant features are reserved in order to maximize $J(\cdot)$.

Handling Redundant Features

Redundant features refer to features that are weakly relevant when evaluating the features individually while having very limited or no capability for outlier detection when they are combined with strongly relevant features [67]. In other words, redundant features have quite high intra-feature outlierness, but their inter-feature outlierness is low. This results in a low overall feature outlierness, and consequently these features are not retained in $\mathcal{S}$ since all the features in $\mathcal{S}$ have high outlierness.

8.5 Experiments and Evaluation

8.5.1 Data Sets

Fifteen publicly available real-world data sets are used, which cover diverse domains, e.g., intrusion detection, image object recognition, advertising and marketing, population and ecological informatics, as shown in Table 8.1. The two data sets, *Probe* and *U2R*, are derived from the KDDCUP99 data sets which integrates multiple types of *probing* and *user-to-root* attacks as outliers; we transform two balanced classification data sets, *Mushroom* and *Optdigits* with classes '1' and '7', by the downsampling method described in Section 2.3.1. The other 11 data sets are directly transformed from highly imbalanced data using the rare class conversion method.

8.5.2 Baselines and Settings

We first evaluate the feature selection method DSFS by examining its capability to improve the effectiveness and efficiency of unsupervised outlier detectors. Three different types of representative pattern-based outlier detection methods, MarP [34], CompreX [7] and FPOF [57], are compared. MarP and CompreX are parameter-free. Following [57], FPOF is set with the minimum *support* threshold $supp = 0.1$ and the maximum pattern length $l = 5$.

We further compare DSFS with the entropy-based feature weighting method (denoted by ENFW) [121] for outlier detection using the above three detectors. Feature weighting methods only assign relevance weights to features and require a decision threshold to select a feature subset. To have a fair comparison, the top-ranked D' features are selected, where D' is the number of features in the feature subset selected by DSFS.

The scalability of DSFS w.r.t. data size and the number of features is evaluated on six subsets of the two UCI data sets *LINK* and *AD*, which have the largest number of objects and features in our data sets. For LINK, the smallest subset contains 1,000 objects, and subsequent subsets are increased by a factor of four until the largest subset which contains 1,024,000 objects. For AD, the data with the smallest feature subset contains 40 features, and subsequent subsets are increased by a factor of two, until the largest feature subset which contains 1,280 features.

DSFS, ENFW, FPOF and MarP are implemented in JAVA in WEKA [52]. CompreX is obtained from the authors of [7] in MATLAB. All the experiments are performed at a node in a 3.4GHz Phoenix Cluster with 32GB memory.

8.5.3 Feature Reduction Rate

We record the number of selected features by DSFS, D', and the *reduction rate*, *RED*. The reduction rate is defined as the rate of the reduced number of features in the feature subset selected by DSFS to that in the full feature set, which is shown in the last column in Table 8.1. The results show that DSFS leads to a significant reduction rate, ranging from 13% up to 97% across the 15 data sets. On average, DSFS obtains 48% reduction rate.

The two data indicators κ_{fnl} and κ_{rdn} demonstrate that nearly all data sets have a large proportion of noisy or redundant features. These noisy and redundant features make the three types of pattern-based outlier detectors less effective and efficient. We show in the next section that proper feature selection is essential to enable the detectors to handle the data complexities.

8.5.4 Performance of Different Subsequent Outlier Detectors

The AUC performance and runtime of the three detectors: MarP, CompreX and FPOF compared with their editions by incorporating DSFS: MarP*, CompreX* and FPOF* are presented in Table 8.3[1]. On average, MarP*, CompreX* and FPOF* obtain 6%, 4% and 3% AUC improvements respectively while they only use 52% of the features compared to their counterparts. In particular, the maximal improvement that MarP* achieves is 42% on aPascal, CompreX* achieves 33% on aPascal, and FPOF* gains 18% on Census. It is interesting to see that less improvement is made on UCI data sets, which is understandable as the UCI data sets tend to be highly manipulated and simpler.

Table 8.1: Feature Selection Results on Data Sets with Different Characteristics. The data sets are sorted by κ_{fnl}. The middle horizontal line roughly separates data sets with many noisy features (i.e., $\kappa_{fnl} > 35\%$) from the other data sets. $RED = \frac{D-D'}{D}$ (%) denotes the reduction rate by DSFS.

Data	**Acronym**	κ_{fnl}	κ_{rdn}	N	D	D'	**RED**
BankMarketing	BM	90%	0%	41188	10	4	60%
aPascal	-	81%	0%	12695	64	20	69%
Sylva	-	78%	0%	14395	87	66	24%
Census	-	58%	0%	299285	33	10	70%
CelebA	-	49%	4%	202599	39	34	13%
CMC	-	38%	4%	1473	8	5	38%
CoverType	CT	34%	22%	581012	44	5	89%
Chess	-	33%	0%	28056	6	4	33%
U2R	-	17%	7%	60821	6	3	50%
SolarFlare	SF	9%	0%	1066	11	8	27%
Optdigits	DIGIT	8%	26%	601	64	46	28%
Mushroom	MRM	5%	2%	4429	22	13	41%
Advertisements	AD	5%	78%	3279	1555	49	97%
Probe	-	0%	7%	64759	6	2	67%
Linkage	LINK	0%	0%	5749132	5	4	20%
Avg.		34%	10%	470986	131	18	48%

With regard to efficiency, MarP*, CompreX* and FPOF* run orders of magnitude faster than their counterparts as they work on the highly reduced feature subsets. For example, FPOF* runs six orders of magnitude faster than FPOF on *CT*. DSFS enables CompreX and FPOF to perform outlier detection on high dimensional data, such as *Sylva* with 87 features and *AD* with 1555 features, where these detectors are otherwise prohibitive in terms of runtime and/or space requirements.

A more *straightforward benefit* is that the simplest detector MarP empowered by DSFS can obtain the AUC performance that is the same as, or very competitive with, that of the two other complex detectors CompreX and FPOF, while at the same time saving several orders of magnitude in runtime. In other words, only simple detectors are needed to obtain the desired efficacy with the premise of DSFS.

Next two subsections further explore the performance of these three detectors in data sets with many noisy or redundant features, respectively.

[1] All runtime refers to the runtime of the detectors only, excluding that of DSFS, but our empirical results show that the runtime of DSFS is within one second in most data sets which is almost negligible in practice.

Table 8.2: AUC Performance of the Three Detectors with or without DSFS. The three baseline detectors are MarP, CompreX and FPOF. Their editions using DSFS are MarP*, CompreX* and FPOF*, respectively. IMP indicates the AUC improvement of the detectors combined with DSFS. '○' indicates out-of-memory exceptions. '●' indicates that we cannot obtain the results within four weeks, i.e., 2,419,200 seconds.

Data	MarP	MarP*	IMP	CompreX	CompreX*	IMP	FPOF	FPOF*	IMP
BM	0.56	0.59	5%	0.63	0.62	-2%	0.55	0.58	5%
aPascal	0.62	0.88	42%	0.66	0.88	33%	○	0.88	○
Sylva	0.96	0.96	0%	0.95	0.96	1%	○	○	○
Census	0.59	0.69	17%	0.64	0.71	11%	0.61	0.72	18%
CelebA	0.74	0.74	0%	0.76	0.76	0%	0.74	0.75	1%
CMC	0.54	0.66	22%	0.57	0.66	16%	0.56	0.65	16%
CT	0.98	0.97	-1%	0.98	0.97	-1%	0.98	0.97	-1%
Chess	0.64	0.64	0%	0.64	0.63	-2%	0.62	0.61	-2%
U2R	0.88	0.92	5%	0.99	0.99	0%	0.92	0.97	5%
SF	0.84	0.85	1%	0.85	0.86	1%	0.86	0.86	0%
DIGIT	0.95	0.95	0%	0.97	0.97	0%	0.96	0.94	-2%
MRM	0.89	0.89	0%	0.93	0.94	1%	0.91	0.91	0%
AD	0.70	0.74	6%	●	0.75	●	○	0.74	○
Probe	0.98	0.98	0%	0.98	0.98	0%	0.99	0.98	-1%
LINK	1.00	1.00	0%	1.00	1.00	0%	1.00	1.00	0%
Avg.			6%			4%			3%

Table 8.3: Runtime of the Three Detectors with or without DSFS. Three baseline detectors are MarP, CompreX and FPOF. Their editions using DSFS are MarP*, CompreX* and FPOF*, respectively. SU indicates the runtime speedup of the detectors combined with DSFS.

Data	MarP	MarP*	SU	CompreX	CompreX*	SU	FPOF	FPOF*	SU
BM	0.17	0.15	1	212.46	170.43	1	0.85	0.57	1
aPascal	0.31	0.12	3	451.36	41.00	11	○	53.29	○
Sylva	0.21	0.20	1	1137.07	498.59	2	○	○	○
Census	1.62	0.51	3	18174.49	12878.14	1	30790.78	75.23	409
CelebA	0.89	0.82	1	1647.47	1169.27	1	159377.51	50188.65	3
CMC	0.14	0.01	11	5.14	2.42	2	0.10	0.06	2
CT	3.14	0.36	9	3914.33	341.98	11	410016.55	1.09	377547
Chess	0.12	0.08	1	95.35	49.30	2	0.42	0.18	2
U2R	0.28	0.13	2	318.95	255.28	1	0.39	0.22	2
SF	0.02	0.01	1	6.33	4.40	1	0.39	0.09	4
DIGIT	0.04	0.03	1	217.10	111.51	2	10196.85	31.99	319
MRM	0.07	0.07	1	48.72	32.18	2	19.32	2.70	7
AD	0.85	0.10	9	●	126.35	●	○	54088.52	○
Probe	0.28	0.11	3	576.08	456.00	1	0.47	0.20	2
LINK	2.74	2.27	1	6365.26	5203.67	1	23.56	17.93	1
Avg.			3			3			31525

Substantially Enhancing both AUC and Runtime on Data Sets with High Feature Noise Level

In data with many noisy features, e.g., *BM* (90% w.r.t. κ_{fnl}), *aPascal* (81%), *Sylva* (78%), *Census* (58%), *CelebA* (49%) and *CMC* (38%) (see Table 8.1), on average, DSFS removes 45% features and enables MarP, CompreX and FPOF to respectively obtain 14%, 10% and 10% AUC improvements as shown in Table 8.3, compared to their counterparts. This is because DSFS successfully removes many noisy features from these highly noisy data, and enables pattern-based detectors to work on much cleaner data, and thus perform more effectively.

In other data sets (e.g., *Sylva* and *CelebA*) where feature reduction rates are smaller, resulting in a number of noisy features retained in the selected feature subset, it is very difficult to separate them from the relevant features. As a result, the detectors make very limited, or none, AUC improvements. This shows that such tough noisy features are deeply mixed with the outlier-discriminative features, and generate higher outlierness than truly outlying features. In these cases, it is too difficult for DSFS to distinguish them from outlying features.

In addition to the AUC improvement, the DSFS-enabled detectors can also have a significant speedup due to the significant feature reduction rate, e.g., FPOF runs 409 times slower than FPOF* on *Census*.

Achieving a Substantial Speedup on Data Sets with High Feature Redundancy

In data sets with a high feature redundancy level, e.g., *CT* (22% w.r.t. κ_{rdn}) and *AD* (78% w.r.t. κ_{rdn}), DSFS generates very aggressive feature reduction, removing 89% and 97% features, respectively. Although this massive feature reduction might result in little loss in terms of AUC, e.g., 1% on *CT*, the outlier detectors can obtain up to six orders of magnitude speedup by working on a substantially smaller feature set, e.g., FPOF on *CT* and CompreX on *AD*. On the other hand, MarP using DSFS obtains 6% AUC improvement on *AD* even if it works on the data with only 3% original features left.

For data sets such as *U2R*, *SF*, *MRM*, *Probe* and *LINK*, the reduction rates are more than the sum of κ_{fnl} and κ_{rdn}. It should be noted that we only have a conservative estimation of κ_{fnl} and κ_{rdn}, so the true feature noise and redundancy levels might be much higher than the estimated values. This explains why the three detectors empowered by DSFS can still perform equally well or very competitively on these data sets, compared to their counterparts not using DSFS.

8.5.5 Comparison to Feature Weighting-based Contenders

A comparison between two feature selection methods ENFW and DSFS via the performance of the three detectors on data with selected feature sets is shown in Table 8.4. On average, MarP, CompreX and FPOF using DSFS obtain 24%, 25% and 24% AUC improvements, compared to MarP, CompreX and FPOF using ENFW, respectively. Impressively, the maximal improvement that the DSFS-empowered MarP gains is 91% on aPascal, the DSFS-empowered CompreX makes 94% on CT, and the DSFS-empowered FPOF achieves 91% on aPascal, compared to their ENFW-empowered counterparts.

We further explore the power of DSFS on noisy data. As shown in 8.4, DSFS generally performs much better than ENFW on almost all data sets that contain noisy features. This is mainly because ENFW evaluates features independently and wrongly takes noisy features as relevant features. However, DSFS estimates the outlierness of features based on the intra- and inter-feature couplings embedded within/between features, thus can much better filter out noisy features than ENFW.

The exceptional cases are on *CelebA* and *Chess*, where DSFS and ENFW perform equally well. This is because both DSFS and ENFW cannot remove a sufficient number of noisy features, and as a result the three detectors not using DSFS and ENFW obtain equally good performance as their counterparts using either DSFS or ENFW. This also

Table 8.4: AUC Performance Comparison of the Three Detectors Using ENFW and DSFS respectively. IMP denotes the improvement of DSFS over ENFW.

	MarP			**CompreX**			**FPOF**		
Data	ENFW	DSFS	IMP	ENFW	DSFS	IMP	ENFW	DSFS	IMP
BM	0.53	0.59	11%	0.56	0.62	11%	0.53	0.58	9%
aPascal	0.46	0.88	91%	0.46	0.88	91%	0.46	0.88	91%
Sylva	0.82	0.96	17%	0.82	0.96	17%	○	○	○
Census	0.43	0.69	60%	0.43	0.71	65%	0.46	0.72	57%
CelebA	0.74	0.74	0%	0.76	0.76	0%	0.75	0.75	0%
CMC	0.50	0.66	32%	0.52	0.66	27%	0.51	0.65	27%
CT	0.51	0.97	90%	0.50	0.97	94%	0.51	0.97	90%
Chess	0.64	0.64	0%	0.63	0.63	0%	0.61	0.61	0%
U2R	0.86	0.92	7%	0.83	0.99	19%	0.86	0.97	13%
SF	0.81	0.85	5%	0.82	0.86	5%	0.83	0.86	4%
DIGIT	0.93	0.95	2%	0.95	0.97	2%	0.93	0.94	1%
MRM	0.89	0.89	0%	0.93	0.94	1%	0.90	0.91	1%
AD	0.56	0.74	32%	0.56	0.75	34%	0.56	0.74	32%
Probe	0.93	0.98	5%	0.88	0.98	11%	0.93	0.98	5%
LINK	1.00	1.00	0%	1.00	1.00	0%	1.00	1.00	0%
Avg			24%			25%			24%

shows the challenge of identifying intrinsic characteristics and sophisticated interactions between features for outlier detection.

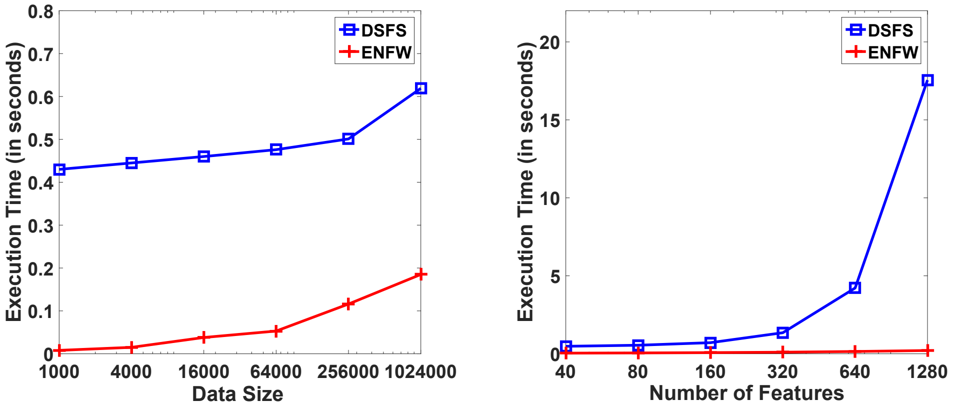


Figure 8.3: Scale-up Test Results of DSFS against ENFW w.r.t. Data Size and the Number of Features.

8.5.6 Scalability Test

The scalability test results of DSFS against ENFW as a baseline are illustrated in Figure 8.3. As expected, DSFS has linear time complexity with respect to data size and is quadratic to the number of features. Although DSFS runs slower than ENFW, it still has quite good scalability with respect to both data size and the number of features, given that DSFS completes its execution within one second for the largest data set with 1,024,000 objects and less than 20 seconds for the high-dimensional data with 1,028 features.

8.6 Summary

This chapter introduces a novel and flexible unsupervised feature selection framework for outlier detection (CUFS). Unlike existing feature selection and unsupervised outlier detection, CUFS effectively captures the low-level hierarchical interactions embedded in relevant features which are mixed with noisy and redundant features. We further introduce a parameter-free instantiation (DSFS) of the CUFS framework. DSFS combines the advantage of CUFS with graph-based strategies. We prove that the feature subset selected by DSFS achieves a 2-approximation to the optimum.

Our extensive evaluation results show that, on average, (i) DSFS obtains 48% feature reduction rate on 15 real-world data sets with different levels of noisy features and redundant features, and (ii) DSFS enables three different types of pattern-based outlier detectors (i.e., MarP, CompreX and FPOF) to respectively obtain 6%, 4% and 3% AUC improvements compared to their counterparts not using DSFS. On data sets with a high noise level, in particular, DSFS is able to remove a large proportion of noisy features, resulting in more than 10% improvement for all three detectors. Moreover, by working on data sets with significantly smaller feature subsets, CompreX and FPOF, which have at least quadratic time complexity w.r.t. the number of features, perform orders of magnitude faster than on the original full feature set. Compared to its feature selection contender ENFW, DSFS performs substantially better on most data sets with noisy features. On average, all three DSFS-based detectors obtain more than 20% AUC improvements compared to ENFW.

This work showcases the applicability of leveraging value-level coupled outlier factors to infer the feature-level coupled outlier factors. One main implication here is that the value-level coupled outlier factors explored in Chapters 4, 5, 6, and 7 may provide important hints for designing methods to capture the rich couplings of outlier factors at the feature/object level.

Chapter 9

Sequential Couplings of Object-level Outlier Factors

9.1 Introduction

The previous chapters focus on exploiting the interdependence of value/feature outlierness based on pairwise or high-order outlierness influence in a graph representation of the data. In this chapter, we work on a very different type of couplings to extend the scope of this thesis. Particularly, we are interested in exploring a sequential coupling of the outlierness at the data object level to successively refine the outlierness scoring results, and demonstrate its importance in the problem of high-dimensional outlier detection.

The major challenge in high-dimensional outlier detection is due to the curse of dimensionality [128]. Therefore, most existing high-dimensional outlier detection solutions are based on subspace/feature selection methods, which search for relevant feature subset(s) to apply off-the-shelf outlier detection methods on these relevant feature subset(s) to alleviate the dimensionality curse or bias brought by irrelevant features. However, we often do not have supervision information to guide the feature subset search, and thus, it is very difficult to identify the truly relevant feature subset(s) in a single pass, leading to inaccurate outlierness scoring of data objects in the identified feature subset(s).

To address this issue, we leverage a sequential collection of models to successively refine the outlierness scoring of the objects. As shown in Figure 9.1, we use the outlierness obtained by a scoring function ϕ in Step $t-1$ to guide the feature selection in Step t, which enables us to select a feature subset $\mathcal{S}$ that is specifically optimal to the ϕ scoring function, and thus, we expect to obtain improved outlierness scoring results when performing ϕ on the subset $\mathcal{S}$ that is tailored for it. The improved outlierness scoring results in Step t in turn refine the feature selection in Step $t+1$. This mutual refinement results in more effective high-dimensional outlier detection.

We introduce a novel SparsE Modeling-based Sequential Ensemble learning (*SEMSE*) framework based on the this idea, which is focused on outlier detection in high-dimensional *numeric* data. Specifically, SEMSE first uses a given *outlier scoring* method to compute the outlier scores of data objects, and defines an *outlier thresholding* function to identify a set of outlier candidates. SEMSE then performs *sparse regression* on the outlier candidate

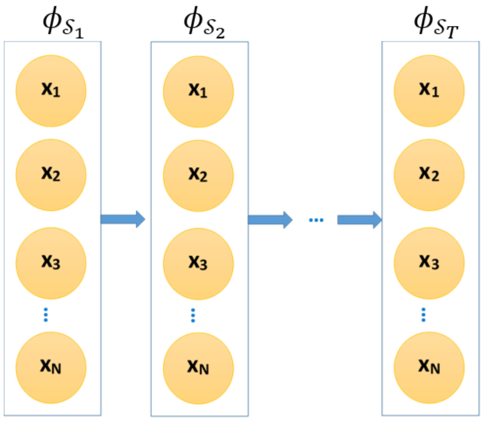


Figure 9.1: Sequential Couplings of Object-level Outlier Factors. $\phi_{\mathcal{S}}$ denotes the outlierness scoring performed on the feature subset $\mathcal{S}$. Therefore, the sequential couplings indicate that the feature subset $\mathcal{S}_t$ is successively determined by the outlierness scoring function ϕ on the feature subset $\mathcal{S}_{t-1}$, which helps iteratively enhance the feature subset selection for the ϕ function.

set by treating the outlier scores as a target feature and the original features as predictors to select the most relevant features w.r.t. the outlier scores. This process is referred to as *fragmentary* sparse modeling to highlight that the sparse regression is built on a small data subset (i.e., *the outlier candidate set*) rather than the full data set. SEMSE finally applies the same given outlier detector to the data with the selected features to produce a refined outlier scoring. These three steps are iteratively performed to produce a set of outlier scores until the loss function of the sparse regression does not decrease.

Essentially, this learning procedure integrates the two correlated tasks: feature selection and outlier detection, and obtains a set of *sequentially coupled* outlier detection (or outlying feature selection) models which are commonly known as *sequential ensemble* [43]. This enables SEMSE to produce feature subsets that are tailored for the outlier scoring method. A single sequential ensemble may perform unstably in data sets with many noisy features. We therefore have a boostrap aggregating (i.e., *bagging*) [19] of the sequential ensembles (i.e., an ensemble of sequential ensembles) to further enhance its capability and stability.

We further implement SEMSE by defining a *Cantelli*'s INequality-based Fragmentary lassO, termed CINFO, to build the sequential ensembles. Specifically, CINFO first defines a *Cantelli*'s inequality [38] based outlier thresholding function to select the outlier candidates, and further applies lasso-based fragmentary sparse regression on the outlier candidate set to obtain the relevant feature subset. Two diverse subsampling-based outlier scoring methods, namely LeSiNN [92] and iForest [77] that respectively work on the full space and random subspaces of the input data, are respectively used to obtain the outlier scores to demonstrate the flexibility of SEMSE.

Unlike the well-established ensemble methods for clustering and classification, outlier ensemble learning has attracted wide attention only in recent years [3, 127]. Most existing outlier ensembles [73, 77, 92, 111] are in the parallel ensemble learning paradigm that constructs a set of independent base models. In contrast, sequential ensembles construct dependent base models by using the results of the current base model to refine the next one.

It is very difficult to construct sequential ensembles for outlier detection as class labels are often assumed to be unavailable. As far as we know, the method called CARE in [103] is the only work of this kind, which intends to reduce the masking and swamping effects [51] by iteratively removing potential outliers to refine the base models. This does not help in addressing the aforementioned issues in high-dimensional space. SEMSE is fundamentally different from CARE, as we explore how to iteratively eliminate noisy features to mutually refine feature selection and outlier scoring.

Accordingly, this chapter makes two main contributions:

i. We introduce a novel sequential ensemble learning framework SEMSE for identifying outliers in high-dimensional numeric data. SEMSE defines a recurrent fragmentary sparse modeling process to build the sequential ensembles, in which feature selection and outlier scoring are iteratively and mutually refined. It results in more reliable outlier scores on data with many noisy features, compared to existing subspace/feature selection-based solutions.

ii. SEMSE is further instantiated to CINFO, a method that introduces a *Cantelli*'s inequality-based fragmentary lasso to learn the sequential ensembles. The *Cantelli*'s inequality provides a false positive upper bound for outlier thresholding with no specific probability distribution assumption on the outlier scores, which well guarantees the refinement of feature selection and outlier scoring in the later stage of sequential ensembles.

A series of empirical results shows that (i) the CINFO-enabled LeSiNN and iForest perform significantly better than three state-of-the-art competitors and the bare versions of LeSiNN and iForest on 11 real-world high-dimensional data sets; (ii) CINFO has much better resilience to noisy features than its competitors; and (iii) CINFO has linear time complexity w.r.t. data size and data dimensionality.

In the rest of this chapter, SEMSE is detailed in Section 9.2. CINFO is introduced in Section 9.3, followed by a theoretical analysis in Section 9.4. Empirical results are provided in Section 9.5. We conclude this chapter in Section 9.6.

9.2 The Proposed SEMSE Framework

The SEMSE framework builds a set of sequential ensembles to mutually refine outlier scoring and feature selection. As shown in Figure 9.2, SEMSE works as follows. At the t-th iteration, given a set of N data objects $\mathcal{X}=\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ described by a set of D features (i.e., $\mathbf{x}_i=\{x_{i1}, x_{i2}, \cdots, x_{iD}\}$) and their outlier score vector $\mathbf{y}^{t-1} \in \mathbb{R}^N$ obtained in the previous iteration, SEMSE first defines an outlier thresholding function η^t to yield a set of L^t outliers $\mathcal{R}^t \in \mathbb{R}^{L^t \times (D+1)}$. $\mathcal{R}^t$ contains $D+1$ dimensions as it concatenates the original D dimensions and $\mathbf{y}^{t-1}$. SEMSE further treats $\mathbf{y}^{t-1}$ as the target feature and the other D features as predictors, and applies a sparse regression model ψ^t on $\mathcal{R}^t$ to produce a new data set $\mathcal{S}^t$ with a set of M^t optimal features w.r.t. $\mathbf{y}^{t-1}$, i.e., $\mathcal{S}^t \in \mathbb{R}^{N \times M^t}$, together with an empirical error mse^t. SEMSE then uses an outlier scoring function ϕ^t on $\mathcal{S}^t$ to re-compute an outlier score vector $\mathbf{y}^t$. SEMSE repeats these recurrent steps to yield a set of outlier score vectors until $mse^{t+1} > mse^t$. These recurrent steps compose a sequential ensemble model. SEMSE finally performs bagging to aggregate a set of sequential ensemble models to obtain the final outlier scores.

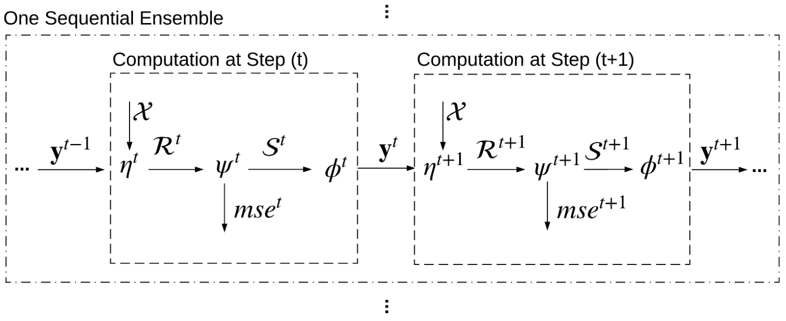


Figure 9.2: Our SEMSE Framework. $\mathbf{y}$ contains outlier scores of all data objects. η, ψ and ϕ are functions for outlier thresholding, fragmentary sparse modeling, and outlier scoring, respectively.

Essentially, SEMSE uses the pseudo target feature $\mathbf{y}^{t-1}$ to generate $\mathcal{S}^t$ with a feature subset that is mostly correlated to the outlier scores produced by the scoring function ϕ^{t-1}. Since ϕ^t works on $\mathcal{S}^t$ with the selected features that are tailored for it, SEMSE likely obtains an enhanced score vector $\mathbf{y}_t$, and it can in turn yield a better feature subset in $\mathcal{S}^{t+1}$ in the next iteration. This cycle enables SEMSE to obtain more reliable outlier scores compared to that computed on the original feature space. The sequential ensembles help SEMSE reduce the learning bias while the final bagging stage helps reduce the learning variance [2].

SEMSE has good generalizability since it can be instantiated to a specific sequential ensemble method by specifying its three components η, ψ and ϕ. We introduce an instance of SEMSE in the next section and then verify its performance by theoretical and empirical analyses.

9.3 A SEMSE Instance: CINFO

CINFO instantiates SEMSE by a *Cantelli*'s inequality-based outlier thresholding function η, a lasso-based fragmentary sparse regression function ψ, and a subsampling-based outlier scoring function ϕ. After building a sequential ensemble with these three functions, bagging is performed to obtain a set of such sequential ensembles and combine their outlier scores to well identify high-dimensional outliers.

9.3.1 Building a Sequential Ensemble

Outlier Thresholding η with *Cantelli*'s Inequality.

The outlier thresholding function η is to identify a set of most likely outliers. We define a *Cantelli*'s inequality-based η as follows, which provides an upper bound for false positives.

Definition 9.1 (Outlier Thresholding)**.** *Given an outlier score vector* $\mathbf{y} \in \mathbb{R}^N$*, in which large scores indicate high outlierness, and let* μ *and* δ^2 *be its expected value and variance,* *then the outlier candidate set $\mathcal{R}$ is defined as follows:*

$$\mathcal{R} = \{(\mathbf{x}_i, y_i) | \eta(y_i, a) \geq 0\}, \ \forall \mathbf{x}_i \in \mathcal{X}, y_i \in \mathbf{y}, \tag{9.1}$$

where $\eta(y_i, a) = y_i - \mu - a\delta$ and a is user-defined.

This outlier thresholding is equivalent to selecting the outlier candidates with a false positive upper bound of $\frac{1}{1+a^2}$ based on *Cantelli*'s inequality (see our theoretical support in the next section).

Fragmentary Sparse Modeling ψ with Lasso.

CINFO performs fragmentary sparse modeling on the data subset $\mathcal{R} \in \mathbb{R}^{L\times(D+1)}$. $\mathcal{R}$ is the newly created data set with reduced objects at the outlier thresholding stage, in which L represents the number of data objects identified as outliers by the η function and thus $L \ll N$. Specifically, CINFO conducts a univariate sparse regression learning as follows:

$$\psi(\mathcal{R}, \lambda) = \arg\min_{\boldsymbol{\omega}} \left(\frac{1}{2L} \sum_{i=1}^{L} (y_i - \mathbf{x}_i^{\intercal}\boldsymbol{\omega})^2 + \lambda ||\boldsymbol{\omega}||_1 \right), \tag{9.2}$$

where $\boldsymbol{\omega}$ is the coefficient vector and λ is a regularization parameter. When λ is large, solving Eqn. (9.2) obtains a shrinking solution to the least squares model, resulting in a number of zero-coefficient features that are not correlated to the outlier score $\mathbf{y}$. We then obtain another newly created data set $\mathcal{S} \in \mathbb{R}^{N\times M}$ with reduced features (i.e., $M < D$):

$$\mathcal{S} = \{\mathbf{x}_{\cdot i} | \omega_i \neq 0, \ 1 \leq i \leq D\}, \tag{9.3}$$

The parameter λ is critical to the performance of lasso. Inappropriate λ will lead to overfitting or underfitting. To address this issue, we use 10-fold cross validation on $\mathcal{R}$ to choose the best λ that minimizes the mean square error *mse*.

CINFO performs fragmentary sparse modeling for two major reasons. (i) Restricting the sparse modeling only on the outlier candidate set $\mathcal{R}$ enables CINFO to select features that are mostly relevant to outlier identification. Since outliers are normally a minority of the data, sparse modeling on the full data set can be dominated by normal objects and fail to obtain outlier-sensitive features. (ii) It helps tune the parameter λ much more efficiently. Since $L \ll N$, performing the cross validation on $\mathcal{R}$ is substantially much faster than on the full data set $\mathcal{X}$.

Subsampling-based Outlier Scoring ϕ.

To demonstrate the flexibility of SEMSE, we use the two very different subsampling-based outlier scoring methods LeSiNN and iForest to specify ϕ, respectively.

LeSiNN is a subsampling-based ensemble of the nearest-neighbor outlier detector using the *full dimensionality* of the input data $\mathcal{S}$. Given a data object $\mathbf{x}_i \in \mathcal{S}$, its outlier score is computed as the average of the nearest neighbor distances in l subsamples:

$$y_i = \phi(\mathbf{x}_i) = \frac{1}{l}\sum_{j=1}^{l} nn\_dist(\mathbf{x}_i|\mathcal{M}_j), \tag{9.4}$$

where $\mathcal{M}_j \subset \mathcal{S}$ is a random data subsample and *nn_dist* returns the nearest neighbor distance of $\mathbf{x}_i$ in $\mathcal{M}_j$.

iForest posits that outliers are susceptible to isolation and builds isolation trees on random *subspaces* in $\mathcal{S}$ to identify outliers. Each tree is grown by using a random subsample until every data object is isolated, where the feature and cut-point at each tree node are randomly selected. The inverse of the path length traversed from the root to a leaf node by $\mathbf{x}_i$ is used as its outlier score:

$$y_i = \phi(\mathbf{x}_i) = (2^{-\frac{E(h(\mathbf{x}_i))}{c(|\mathcal{M}|)}})^{-1}, \tag{9.5}$$

where $h(\mathbf{x}_i)$ denotes the path length of $\mathbf{x}_i$ in a subsample $\mathcal{M}$, $E(h(\mathbf{x})) = \frac{1}{l}\sum\limits_{j=1}^{l} h_j(\mathbf{x}_i|\mathcal{M}_j)$ is the average path length of $\mathbf{x}_i$ from a set of l isolation trees, and $c(|\mathcal{M}|)$ is the expected path length given the subsample size $|\mathcal{M}|$.

The use of subsampling results in the linear time complexities in LeSiNN and iForest, which is critical to the efficiency of CINFO. LeSiNN and iForest are the state-the-of-art detectors and they are expected to yield fairly good outlier scores to ensure that there are at least some outliers in $\mathcal{R}$ output by η.

Combination of Outlier Scores.

CINFO performs the aforementioned three recurrent components η, ψ and ϕ until the mean squared error *mse* produced by ψ does not further decrease. Assume the sequential ensemble learning terminates after T iterations, i.e., $t = \{1, 2, \cdots, T\}$, we obtain a set of T outlier score vectors and their associated *mse*. We employ the commonly-used weighted summation [43] to combine the outlier score vectors with *mse* as weights, and define an outlier score for each data object in the sequential ensemble as follows:

$$seq\_score(\mathbf{x}_i) = \frac{1}{T}\sum_{t=1}^{T} w^t \tau(y_i^t), \tag{9.6}$$

where w^t is a normalized weight by $w^t = \frac{\mathbb{Z}_{mse} - mse^t}{\sum_{t=1}^{T}[\mathbb{Z} - mse^t]}$ with $\mathbb{Z} = \sum_{t=1}^{T} mse^t$, and $\tau(y_i^t) = \frac{y_i^t}{||\mathbf{y}^t||_1}$ is a vector normalization function that normalizes the vector $\mathbf{y}$ into a unit norm to address the heterogeneity of the outlier scores from heterogeneous feature subsets.

Note that the initial outlier score vector $\mathbf{y}_0$ is not integrated into the above weighted combination. This is because $\mathbf{y}_0$ is obtained from the original full feature space with noisy features and is thus not as reliable as the later score vectors.

9.3.2 Aggregating a Set of Sequential Ensembles

Using single sequential ensemble may produce high detection errors, when the initial outlier score vector $\mathbf{y}_0$ happens to mislead the subsequent outlier scoring in the sequential ensemble. We therefore further aggregate a set of sequential ensembles to address this issue by bagging. Bagging is a representative approach for building a set of base models independently, which can largely reduce the generalization error [19]. Specifically, the final outlier score of a given object is the average over its outlier scores obtained from a set of independent sequential ensembles:

$$\mathit{score}(\mathbf{x}_i) = \frac{1}{m}\sum_{j=1}^{m} \mathit{seq\_score}_j(\mathbf{x}_i), \tag{9.7}$$

where m is the number of sequential ensembles we built.

9.3.3 The Algorithm and Its Time Complexity

Algorithm 9.1 presents the procedure of CINFO. Given a data set $\mathcal{X}$, Step 2 obtains the initial outlier scores. Steps 4-10 use the three recurrent functions η, ψ and ϕ to build a sequential ensemble for the iterative refinement of the selected feature subset in $\mathcal{S}$ and outlier scores $\mathbf{y}$. The lasso problem in Step 7 is implemented by alternating direction method of multipliers (ADMM), and $\boldsymbol{\omega}$, mse and λ are obtained by 10-fold cross validation on $\mathcal{R}$. The outer loop in Steps 1-12 builds a set of independent sequential ensembles by bagging, followed by the average combination of the outlier scores from these sequential ensembles in Step 13. CINFO then returns an outlier ranking based on the outlier scores.

Algorithm 9.1 *CINFO*

```
Input: $\mathcal{X}$ - data objects, $a$ - outlier thresholding parameter, $m$ - bagging size
Output: $\mathbf{r}$ - an outlier ranking of objects
 1: for $j = 1$ to $m$ do
 2:     $\mathbf{y}^0 \leftarrow \phi(\mathcal{X})$
 3:     $\mathit{mse}^0 = 1,\ t = 0$
 4:     repeat
 5:         $t \leftarrow t + 1$
 6:         $\mathcal{R}^t \leftarrow \eta(\mathbf{y}^{t-1}, a)$
 7:         $\boldsymbol{\omega}^t,\ \mathit{mse}^t \leftarrow \psi(\mathcal{R}^t, \lambda^t)$
 8:         $\mathcal{S}^t \leftarrow \{\mathbf{x}_{\cdot i} | \omega_i^t \neq 0,\ 1 \leq i \leq D\}$
 9:         $\mathbf{y}^t \leftarrow \phi(\mathcal{S}^t)$
10:     until $\mathit{mse}^t > \mathit{mse}^{t-1}$
11:     $\mathit{seq\_score}_j(\mathcal{X}) = \frac{1}{T}\sum_{t=1}^{T}(\mathbf{w}^t)^\intercal \tau(\mathbf{y}^t)$
12: end for
13: $\mathit{score}(\mathcal{X}) = \frac{1}{m}\sum_{j=1}^{m} \mathit{seq\_score}_j(\mathcal{X})$
14: $\mathbf{r} \leftarrow$ Sort $\mathcal{X}$ w.r.t. $\mathit{score}$
15: return $\mathbf{r}$
```

The sequential ensemble learning in Steps 4-10 often terminates after a few iterations (e.g., within 10). The bagging in the outer loop typically converges quickly, say after about 10-30 iterations. Therefore, the time complexity of CINFO is determined by the complexity of the three functions η, ψ and ϕ. Obviously, η has a linear time complexity w.r.t. data size and the number of features. The LeSiNN/iForest-based ϕ function has a similar linear time complexity [77, 92]. Moreover, a linear convergence rate is expected for ADMM-based lasso implementation according to [59]. We therefore expect that the overall time complexity of CINFO is linear w.r.t. data size and dimensionality size.

9.4 Theoretical Analysis

The following three subsections present some theoretical support for the specifications of the three functions η, ψ and ϕ in CINFO, respectively.

9.4.1 Upper Bound for Outlier Thresholding

Corollary 9.0.1 (False Positive Bound). *Assume the scores in* $\mathbf{y}$ *have the expected value* μ *and variance* δ^2*. Let* $y_i \in \mathbf{y}$*, the outlier thresholding function* $\eta(y_i, a) = y_i - \mu - a\delta$ *then has a false positive upper bound of* $\frac{1}{1+a^2}$.

Proof. We have $P(y_i \geq \mu + \alpha) \leq \frac{\delta^2}{\delta^2+\alpha^2}$ per *Cantelli*'s inequality. By replacing $\alpha = a\delta$, we obtain

$$P(y_i \geq \mu + a\delta) \leq \frac{1}{1+a^2}. \tag{9.8}$$

This states that the values in $\mathbf{y}$ have a maximum probability of $\frac{1}{1+a^2}$ being greater than $\mu + a\delta$. Since large y_i indicates high outlierness, this inequality implies that the probability that we could wrongly identify normal objects as outliers is up to $\frac{1}{1+a^2}$ when we define the threshold as $\mu + a\delta$. □

Cantelli's inequality is a one-sided *Chebyshev*'s inequality. Similar to *Chebyshev*'s inequality, it makes no assumption on specific probability distributions. It holds for a wide class of probability distributions that have statistical mean and variance. This property enables η to be data-dependent and to perform well for $\mathbf{y}$ following different distributions.

9.4.2 Optimal Feature Subsets w.r.t. Outlier Scoring ϕ

Since the sparse modeling in Eqn. (9.2) is a convex problem [54], the feature subset in $\mathcal{S}$ is expected to be globally optimal w.r.t. the target $\mathbf{y}$ on the outlier candidate set $\mathcal{R}$. In other words, the selected features are customized to the outlier scoring function ϕ that produces the score vector $\mathbf{y}$. This enables ϕ to work on a more reliable feature set when re-computing the outlier scores by using $\mathcal{S}$, resulting in refined outlier scores compared to that in the previous iteration.

In the best case, the outlier scoring or feature selection is iteratively refined. In another extreme, when the outlier scores are poor, e.g., no true outliers are in the outlier candidates, it can mislead the feature selection and does not help improve the successive outlier scoring. The next section analyzes the use of subsampling to obtain quality outlier scores.

9.4.3 Obtaining Good Outlier Scores by Subsampling

In addition to substantial speedup, using subsampling can well guarantee the outlier scoring quality, which is supported by theoretical results from the perspectives of, e.g., density estimation [129], data distribution [111] and variance reduction [2]. We provide the following analysis to further complement these existing theoretical results.

Following [129], for two data objects $\mathbf{x}_1$ and $\mathbf{x}_2$, their expected kNN distance k_dist in $\mathcal{X}$ can be respectively approximated by $E(k\_dist(\mathbf{x}_1|\mathcal{X})) = r\left(\frac{k}{N_1}\right)^{\frac{1}{D}}$ and $E(k\_dist(\mathbf{x}_2|\mathcal{X})) = r\left(\frac{k}{N_2}\right)^{\frac{1}{D}}$, where N_1 and N_2 are the number of objects uniformly distributed in the r-radius sphere of $\mathbf{x}_1$ and $\mathbf{x}_2$, respectively; and their expected k_dist in a random subsample $\mathcal{R}$ of size L can then be given by $E(k\_dist(\mathbf{x}_1|\mathcal{R})) = r\left(\frac{k}{N_1\frac{L}{N}}\right)^{\frac{1}{D}}$ and $E(k\_dist(\mathbf{x}_2|\mathcal{R})) = r\left(\frac{k}{N_2\frac{L}{N}}\right)^{\frac{1}{D}}$. After some transformation, we can obtain:

$$\frac{E(k\_dist(\mathbf{x}_1|\mathcal{R})) - E(k\_dist(\mathbf{x}_2|\mathcal{R}))}{E(k\_dist(\mathbf{x}_1|\mathcal{X})) - E(k\_dist(\mathbf{x}_2|\mathcal{X}))} = (\frac{N}{L})^{\frac{1}{D}}. \tag{9.9}$$

Eqn. (9.9) implies that the contrast between the kNN-based densities in the subsamples and those in the full data set are enlarged and are inversely proportional to the subsampling size. This indicates that subsampling helps enhance the contrast between kNN/density-based outlier scores. Moreover, it also guarantees a ranking-stable result, i.e., $E(k\_dist(\mathbf{x}_1|\mathcal{R})) > E(k\_dist(\mathbf{x}_2|\mathcal{R}))$ if $E(k\_dist(\mathbf{x}_1|\mathcal{X})) > E(k\_dist(\mathbf{x}_2|\mathcal{X}))$. These two properties enable the subsampling-based scoring to yield better outlier scores [71].

Numerous existing outlier scoring methods including LeSiNN assume that outliers are data objects in low-density regions. Therefore, the above results are widely applicable, and subsampling is recommended for the specification of ϕ in CINFO when using this type of methods.

9.5 Experiments and Evaluation

9.5.1 Data Sets

As shown in Table 9.1, 11 real-world data sets are used, which cover diverse domains, e.g., intrusion detection, molecular bioactivity detection, Internet advertising and image object recognition. Some data sets like *AD*, *AID362*, *Probe*, *U2R* and *Thrombin* contain semantically real outliers. For the other data sets, we use the rare class conversion method in Section 2.3.1 to transform them into outlier detection data sets.

9.5.2 Experiment Environment

CINFO and its competitors are implemented in MATLAB. All the experiments are executed at a node in a 3.4GHz Phoenix cluster with 32GB memory. In all our experiments, CINFO uses $a = 1.732$ (i.e., the upper bound for false positives in η is 25%) and $m = 30$

[1]; and the number of subsamples l and subsampling size $|\mathcal{M}|$ for LeSiNN and iForest are set as the recommended settings of their authors.

9.5.3 Effectiveness in Real-world Data

Experimental Settings.

We compare the CINFO-enabled LeSiNN and iForest with their bare versions to evaluate whether CINFO can eliminate irrelevant features and retain (or improve) the performance of these two detectors.

Findings - CINFO Significantly Improves Different Types of Outlier Detectors.

Table 9.1 demonstrates the feature reduction and AUC performance of CINFO-based LeSiNN and iForest, compared to LeSiNN and iForest performing in the original feature space. CINFO-enabled LeSiNN and iForest work with about 10% (e.g., on *AID362* and *BM*) to over 95% (e.g., on *Isolet*, *SECOM* and *Thrombin*) less features, while their performance is substantially better than, or roughly the same as, their bare versions. On average, CINFO enables LeSiNN and iForest to gain about 4% and 7% improvement, respectively. Our significance test shows that CINFO enables LeSiNN and iForest to achieve significantly better AUC performance at the 95% and 99% confidence levels, respectively.

Table 9.1: Feature Reduction and AUC Performance of CINFO-enabled LeSiNN and iForest (denoted by LeSiNN* and iForest*). D is the original feature number. D' and D'' are the average numbers of features retained by LeSiNN* and iForest*, respectively. The average iteration for sequential ensembles per data is 2 to 5.

Data Info.		**Feature Reduction**			**AUC Performance**			
Data	N	D	D'	D''	LeSiNN	LeSiNN*	iForest	iForest*
AD	3279	1555	197	245	0.7107	**0.8666**	0.6830	**0.7907**
AID362	4279	117	106	94	0.6704	**0.6710**	0.6461	**0.6658**
aPascal	12695	64	34	46	0.7308	**0.8554**	0.6755	**0.7963**
BM	41188	62	54	52	0.6854	**0.7100**	0.7316	**0.7678**
Caltech16	829	253	59	50	0.9861	**0.9869**	0.9636	**0.9684**
Census	299285	500	399	422	0.6344	**0.6620**	0.6276	**0.6616**
Isolet	730	617	27	28	1.0000	1.0000	0.9996	**1.0000**
Probe	64759	34	27	25	0.9975	**0.9978**	0.9899	**0.9908**
SECOM	1567	590	27	18	0.5316	**0.5867**	0.5448	**0.6506**
U2R	60821	36	28	30	0.9879	**0.9890**	0.9908	**0.9922**
Thrombin	1909	139351	114	58	**0.8997**	0.8916	0.8843	**0.9044**
			Average		0.8031	**0.8379**	0.7943	**0.8353**

CINFO uses the sequential ensemble learning to mutually improve its feature selection and outlier scoring, which enables CINFO to safely remove noisy features in these high-dimensional data sets. As a result, CINFO-enabled LeSiNN and iForest work on much cleaner data sets and thus can achieve significant performance improvement.

[1] CINFO performs very stably when $m \geq 30$. $m = 30$ is thus used.

9.5.4 Comparison to State-of-the-art Competitors

Experimental Settings.

CINFO is compared with three state-of-the-art competitors: feature bagging (FB for short) [73], RegFS [98], and CARE [103] from three different but relevant research lines.

- *Subspace-based method - FB.* FB is a framework for enabling outlier detectors to handle high-dimensional data by using feature bagging, i.e., working on a set of random feature subsets of size between $\lfloor \frac{D}{2} \rfloor$ and $(D-1)$. It can also be seen as a random feature selection ensemble.
- *Feature selection-based competitor - RegFS.* RegFS only returns a feature relevance ranking. For a thorough comparison, RegFS selects the top-ranked $\lceil rD \rceil$ features with a wide range of r, i.e., $r = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. We report the results of $r = 0.7$, with which the used detectors obtain the best performance.
- *Sequential outlier ensemble - CARE.* CARE attempts to iteratively refine detection models by removing outlier candidates. It uses feature bagging to introduce diversity and handle high-dimensional data.

Findings - CINFO Significantly Outperforms Three State-of-the-art Competitors.

The AUC performance of CINFO, RegFS, FB, and CARE is reported in Table 9.2. The CINFO-enabled LeSiNN and iForest obtain the best performance on eight data sets, with three very close to the best (having the difference in AUC less than 0.01), and they obtain about 4%-7% improvement over their respective competitors. The improvement is significant at the 95% (w.r.t. RegFS and FS) or 90% (w.r.t. CARE) confidence level.

Unlike FB and RegFS which ignore the outlier scoring methods when they perform feature selection, CINFO couples these two dependent tasks to iteratively refine their performance by sequential ensembles. This enables CINFO to substantially reduce its detection errors and obtain more than 4%-22% AUC improvement over its competitors in tough data sets like *AD*, *aPascal*, *Census*, and *SECOM*, which likely contain a large proportion of noisy features.

CINFO and CARE are two very different sequential ensemble methods. CARE builds sequential ensembles horizontally, which iteratively removes likely outliers for identifying some outliers that are otherwise masked by the removed outliers. In contrast, CINFO works in a vertical manner, which iteratively remove noisy features. Although feature bagging is incorporated into CARE, the FB method itself has limited capability in handling noisy features. CINFO therefore obtains similarly large AUC improvement (i.e., 6%-24%) over CARE on the aforementioned noisy data sets.

Table 9.2: AUC Performance of CINFO, RegFS, FB, and CARE Empowered LeSiNN and iForest. 'NA' indicates the execution cannot be completed in two weeks.

	LeSiNN				**iForest**			
Data	CINFO	RegFS	FB	CARE	CINFO	RegFS	FB	CARE
AD	**0.8666**	0.7058	0.7111	0.6934	**0.7907**	0.6832	0.6892	0.6989
AID362	0.6710	0.6371	0.6704	**0.6767**	0.6658	0.6421	0.6659	**0.6752**
aPascal	**0.8554**	0.7464	0.7319	0.7349	**0.7963**	0.7085	0.6642	0.6829
BM	**0.7100**	0.6943	0.6879	0.6818	**0.7678**	0.7328	0.7440	0.7444
Caltech16	0.9869	**0.9874**	0.9869	**0.9874**	0.9684	**0.9728**	0.9670	0.9691
Census	**0.6620**	0.6112	0.6340	0.6198	**0.6616**	0.5638	0.6290	0.6416
Isolet	1.0000	1.0000	1.0000	1.0000	**1.0000**	0.9995	**1.0000**	**1.0000**
Probe	**0.9978**	0.9943	0.9974	0.9970	0.9908	0.9900	0.9908	**0.9941**
SECOM	**0.5867**	0.5343	0.5294	0.5282	**0.6506**	0.5636	0.5533	0.5589
U2R	**0.9890**	0.9645	0.9877	0.9853	**0.9922**	0.9717	0.9904	0.9903
Thrombin	0.8916	NA	0.8995	**0.9023**	**0.9044**	NA	0.9024	0.9034
Average	0.8379	0.7875	0.8033	0.8006	0.8353	0.7828	0.7997	0.8053
p-value	-	0.0078	0.0273	0.0840	-	0.0098	0.0078	0.0840

9.5.5 Resilience to Noisy Features

Experiment Settings.

Following [128], we create a collection of 100-dimensional synthetic data sets with different percentages of relevant features (or noisy features). In this data, normal objects are from a Gaussian distribution and outliers lie at two standard deviations of the distribution in relevant features, and the other features are from a uniform distribution and used as noisy features. For each noise level, we generate 10 data sets with the same number of noisy features and average AUC over them to have more reliable results.

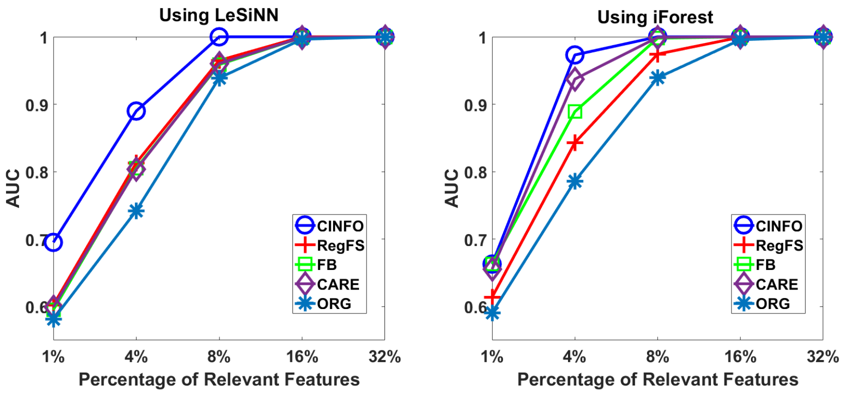


Figure 9.3: AUC Performance on Data with Different Levels of Noisy Features. 'ORG' denotes the bare LeSiNN/iForest. All methods obtain AUC of one with more than 32% relevant features.

Findings - CINFO Greatly Enhances the Resilience of the Outlier Detectors w.r.t. Noisy Features, Especially for Very Noise-Sensitive Detectors.

The AUC performance on the synthetic data sets is shown in Figure 9.3. CINFO-enabled LeSiNN and iForest perform consistently better than their four other versions in a wide range of noise levels. The advantage of CINFO is much more obvious in enabling LeSiNN than iForest. This may be due to the fact that LeSiNN works on the full space of the input data while iForest operates on feature subspaces, and as a result, LeSiNN is much more sensitive to the noisy features retained by the feature subset selection methods and is more difficult to enhance compared to iForest. The substantially better performance of the CINFO-enabled LeSiNN over its competitors highlights its superiority in eliminating noisy features and upgrading very noise-sensitive detectors.

9.5.6 Scalability Test

Experiment Settings.

We generate data sets by varying the data dimension w.r.t. to a fixed data size (i.e., 1000), as well as varying the data size while fixing the data dimension (i.e., 50), respectively.

Findings - CINFO Obtains Linear Time Complexity w.r.t. Data Size and Dimensionality.

The runtime of the five versions of LeSiNN is shown in Figure 9.4. In the left panel, all the methods have linear time complexity. CINFO is comparably fast to RegFS and CARE. These three methods are slower than FB and the bare LeSiNN, since they incorporate more sophisticated components to enhance the accuracy of LeSiNN. In the right panel, the CINFO/FB/CARE-enabled and the bare LeSiNN have linear time complexity, and they run considerably faster than RegFS that has a quadratic complexity.

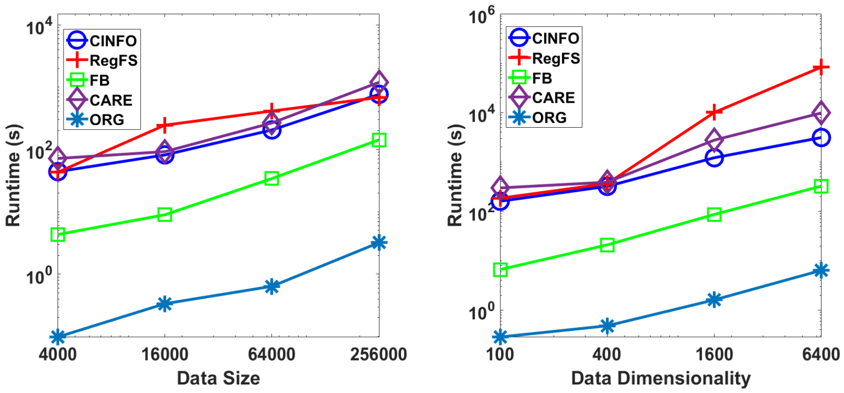


Figure 9.4: Runtime of CINFO and Its Competitors Using LeSiNN. 'ORG' denotes the bare LeSiNN. Logarithmic scales are used. Similar trends can be expected when using iForest as the outlier detector, since LeSiNN and iForest have similar time complexities.

9.6 Summary

This chapter introduces a sequential ensemble-based high-dimensional outlier detection framework SEMSE and its instance CINFO. They perform an iterative mutual refinement of feature selection and outlier scoring, and can efficiently obtain reliable outlier scores in high-dimensional numeric data with many noisy features. Although CINFO works on considerably smaller feature subsets, it obtains significantly better AUC performance in 11 real-world high-dimensional data sets, substantially better resilience to noisy features, compared to its four competitors. CINFO also has linear time complexity w.r.t. data size and dimensionality.

SEMSE may be instantiated into other instances by using lasso-based sparse modeling with other sparse constraints to capture different types of feature interactions. Another possible extension to SEMSE is that, instead of performing univariate regression, we can include the outlier scores in all the $t-1$ stages to perform multivariate regression at the t-th stage, which helps capture much richer sequential couplings between the object-level outlier factors.

Different from the other chapters that focus on categorical data, this work focuses on the data object level in numeric data. It would be interesting to explore whether the value/feature-level coupled outlier factors in the previous chapters could be similarly defined to enable more effective outlier detection in complex non-IID numeric data.

Part IV

Conclusions and Future Directions

Chapter 10

Conclusion

The detection of outliers provides important insights into numerous real-world applications in various domains, ranging from fault detection in mechanical engineering and manufacturing, and fraud detection and insider detection in business and government management, to disease detection in healthcare, and the discovery of new stars/planets in astronomy.

Unlike most outlier detection methods that assume the independence between the outlier factors of the data entities, this thesis formulates the task of non-IID outlier detection in multidimensional data and examines different types of coupling relationships between the outlier factors of the data entities at different levels from feature values, features, to data objects, and leverages these coupling modelings to address challenging outlier detection problems. Our explorations result in a principled architecture for learning complex interactions between the outlier factors at different levels. Under this non-IID outlier detection architecture, in each chapter of Chapters 4-9, we show that each pathway of the architecture can be further formalized into a flexible framework and the framework can be implemented to be a scalable and effective method for addressing a targeted challenging outlier detection problem. This demonstrates the flexibility and applicability of our proposed architecture for complex real-world outlier detection tasks. Although feature selection for outlier detection is known to be extremely difficult, we show that the rich intrinsic couplings underlying the data can be harnessed to effectively select relevant features for subsequent outlier detection, resulting in a set of seminal work on unsupervised outlying feature selection. A detailed conclusion of this thesis is provided as follows.

10.1 Learning Couplings of Outlier Factors

To learn different types of complex interactions between the value-level outlier factors, we provided flexible and principled frameworks and their instantiations with scalable and theoretically sound graph mining techniques for addressing the following four issues:

- "**How is the outlierness of one value influenced by that of other values?**": We introduced the CUOT framework and its instance CBRW that incorporates interactions of feature values within and between the features into the modeling of the outlierness couplings between the values. Our experiments on a large collection of real-world data sets demonstrate significant AUC improvement over several state-of-the-art traditional outlier detectors, indicating strong couplings among the feature values. One key implication is that the feature values are the finest elements in multidimensional data, and thus, this level couplings can contribute the couplings at the higher levels, such as features, feature subspaces, and data objects.

- "**How can we only model useful interactions between outlier factors?**": We posited and justified that only a subset of the couplings between the outlier factors are important while the rest of the couplings are redundant or noisy using the proposed selective coupling learning framework, SelectVC, and its instance POP. By considering such kinds of coupling utility, we achieve state-of-the-art AUC performance in high-dimensional outlier detection in categorical data.
- "**How can we quickly and accurately learn the cascade couplings?**": We contributed the joint feature selection and outlier detection framework, WrapperOD, and its instance HOUR. Joint optimization is often computationally expensive, which is particularly true for simultaneously optimizing the feature subset and data subset since it involves an exponential combination of the features/objects. The proposed method HOUR is driven by the proposed binary cascade coupling learning that can effectively and efficiently compute outlierness in an efficient closed-form.
- "**How can we efficiently learn the high-order interactions between outlier factors?**": We introduced the high-order coupling learning framework, HOCOF, and its instance SDRW. SDRW extends our CBRW method by incorporating a granularity of subgraph-based density outlier factors. The subgraph density considers the interactions of a set of values to capture the high-order interdependence. We leverage the state-of-the-art dense subgraph discovery techniques to guarantee the efficiency of our method by identifying relevant subgraphs in linear time.

To learn the sophisticated couplings between the higher-level outlier factors, we devised novel scalable and principled frameworks and their instantiations that draw methods from subgraph discovery and ensemble learning to find answers to the following two questions:

- "**How is the outlierness of one feature influenced by that of the other features?**": We introduced the CUFS framework and its instance DSFS to capture non-successive two-way interactions between the outlierness of features for outlying feature selection. The DSFS method is parameter-free and achieves a 2-approximation guarantee. Moreover, it is scalable and enables different types of pattern-based outlier detectors to obtain substantial AUC improvement and/or significant speedup.
- "**How can we sequentially refine a given outlier factor?**": We contributed the SEMSE framework and its instance CINFO to capture the sequentially coupled outlier factors by mutually refining feature selection and outlier detection, which is shown powerful in enabling high-dimensional outlier detection in a large collection of real-world numeric data. The proposed framework provides principled approaches for capturing the full feature interactions in joint feature selection and outlier detection.

10.2 Significance of Non-IID Outlier Detection

The significance of the proposed non-IID outlier detection task is demonstrated by the significantly better detection performance of our methods in addressing challenging outlier detection problems compared to existing state-of-the-art outlier detectors. Specifically, the challenging contexts we address include outlier detection in interdependent multidimensional data, data with many noisy features, and data with high dimensionality. Under these contexts, in some particular cases, existing state-of-the-art IID outlier detectors can only obtain an accuracy of being nearly equivalent to random guess results, while our non-IID methods obtain significantly better performance, achieving more than 50% AUC improvement; on average, our non-IID approach achieves 4%-18% AUC improvement over the best outlier detector in any of the aforementioned three challenging contexts. This improvement has two main implications: to the academic community, these results imply new research directions of devising non-IID outlier detectors to well identify sophisticated outliers in real life applications; to the industry, this significant improvement may mean the prevention of millions of dollars loss by fraud detection or life saving due to a successful early detection of fatal diseases.

Chapter 11

Vision and Future Work

While we showed in the previous chapters that our proposed non-IID outlier detection methods are significantly more effective than traditional IID methods in handling many real-world data sets, to build *easy-to-use and effective non-IID outlier detection systems*, several more interesting directions require further exploration and are discussed as follows.

11.1 Broadening Non-IID Outlier Detection

This thesis explored only several types of coupling relationships between the outlier factors. To have a systematic understanding and a complete theory of non-IID outlier detection, more explorations are required in the following important research directions.

11.1.1 Further Exploration of Coupled Outlier Factors

We explored several types of couplings between the outlier factors at different levels. One next step is to examine: whether the couplings of outlier factors that are effective in one level (e.g., the value level) are also applicable to the other levels (e.g., the feature or object level), and how to have a hierarchical consolidation of the couplings at different levels to build more powerful non-IID outlier detectors. Another interesting direction is to learn coupling relationships beyond the ones we examined here for the same targeted outlier detection challenges or other challenges.

11.1.2 Heterogeneous Outlier Factors

We only examined the not-independent aspect of the non-IID outlier detection. Another important aspect for future research is heterogeneous outlier factors. Many complex data sets may require heterogeneous outlier factors in that different outliers may be generated from different mechanisms. Some interesting questions include: how can we determine what types of outlier factors are suitable for a given data? how can we make use of the complementary and consensus information of a set of heterogeneous outlier factors while avoiding the negative effects due to their conflicts?

11.1.3 Exploration of Coupled Heterogeneous Outlier Factors

A more challenging area is to simultaneously capture both the not-independent and the not-identically-distributed aspects of non-IID outlier detection. This might be the ultimate goal of unlocking the full ability of non-IID outlier detectors. Once we have solutions for coupled outlier factors and heterogeneous outlier factors, we are then interested in combining them in a compatible and semantically reasonable way. Particularly, we may need to determine which levels of coupled/heterogeneous outlier factors to include in the combination and how to properly model these coupled and heterogeneous outlier factors.

11.2 Selection of IID/non-IID Outlier Detection Methods

Having obtained a pool of IID and non-IID outlier detection methods, a natural question to ask is: how can we determine whether we use IID methods or non-IID methods for a given data set? A possible solution to this challenging problem is to define a series of data indicators to measure the underlying data characteristics of the given data set and their association with the accuracy performance of specific types of outlier detection methods. These indicators then serve as the key to the selection or combination of the IID and/or non-IID outlier detection methods.

11.2.1 Data Indicators for Measuring the IID/Non-IID Information

We defined some data indicators in Section 2.3.3 to provide insights into the detection performance of our proposed methods on real-world data sets at a post-detection stage, but they are insufficient for the above purpose. First, since we focus on unsupervised outlier detection, all data indicators are supposed to be unsupervised. Second, the data indicators also need to be linked to specific types of outlier detectors, in addition to the data characteristics.

11.2.2 Automatic Selection or Combination of IID/Non-IID Methods

We then need to learn the correlation between a set of data indicators and the detection performance of outlier detectors to achieve the goal of automatic selection or combination of the IID and non-IID outlier detection methods. This enables the use of advanced outlier detection methods without any domain expertise requirements, which would largely promote the deployment of outlier detection systems in different domains. However, this would be very challenging since we do not have class labels to guide the learning. We may address this issue by leveraging a limited amount of labeled outliers to devise highly discriminative data indicators and learn a reliable correlation.

Appendix A

Codes and Data Sets

To promote the research reproducibility, the source codes of all our algorithms in this thesis are made publicly available at https://sites.google.com/site/gspangsite/sourcecode; and the implementations of most of the competing methods are available in two well-known open-source data mining projects, WEKA [52] and ELKI [1].

All the 33 data sets used in this thesis are from real-world applications. We provide the sources of these data sets to acknowledge the contributors. Specifically, 15 data sets, including Bank Marketing (BM), Internet Advertisements (AD), Contraceptive Method Choice (CMC), Solar Flare (SF), CoverType (CT), Linkage (LINK), Chess, Arrhythmia (Arrhy), Alcohol, Turkiye, credit card (Credit), Mushroom (MRM), Optical digits (DIGIT), SECOM, and Isolet, are available from the UCI Machine Learning Repository at http://archive.ics.uci.edu/ml/. The sources of the other 18 data sets are listed in Table A.1. Some data sets, including AD, CMC, Arrhy, Credit, U2R, Probe, CUP14, and Thrombin, contain real outliers. The other data sets are originally used for classification evaluation, which are transformed into outlier detection data sets using the downsampling or rare class conversion method presented in Section 2.3.1.

Table A.1: Data Sources.

Data	**Acronym**	**Source**
U2R	-	http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
Probe	-	http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
CelebA	-	http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html
aPascal	APAS	http://vision.cs.uiuc.edu/attributes/
Reuters10	R10	http://sci2s.ugr.es/keel/
w7a	-	https://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets/
BASEHOCK	BASE	http://featureselection.asu.edu
PCMAC	-	http://featureselection.asu.edu
RELATHE	RELA	http://featureselection.asu.edu
CalTech-16	CAL16	https://people.cs.umass.edu/∽marlin/data.shtml
CalTech-28	CAL28	https://people.cs.umass.edu/∽marlin/data.shtml
wap.wc	-	http://tunedit.org/repo/data/text-wc
KDD CUP 2014	CUP14	https://www.kaggle.com/c/kdd-cup-2014-predicting-excitement-at-donors-choose
WebKB	-	http://ana.cachopo.org/datasets-for-single-label-text-categorization
Reuters8	R8	http://ana.cachopo.org/datasets-for-single-label-text-categorization
Sylva Agnostic	SylvaA	http://www.agnostic.inf.ethz.ch/
Sylva Prior	SylvaP	http://www.agnostic.inf.ethz.ch/
Thrombin	-	http://pages.cs.wisc.edu/∼dpage/kddcup2001/

Bibliography

[1] Elke Achtert et al. “Interactive data mining with 3D-parallel-coordinate-trees”. In: *SIGMOD*. 2013, pp. 1009–1012.

[2] Charu C Aggarwal. *Outlier analysis*. Second. Springer, 2017.

[3] Charu C. Aggarwal. “Outlier ensembles: position paper”. In: *ACM SIGKDD Explorations Newsletter* 14.2 (2013), pp. 49–58.

[4] Charu Aggarwal and S Yu. “An effective and efficient algorithm for high-dimensional outlier detection”. In: *The VLDB Journal* 14.2 (2005), pp. 211–221.

[5] Amir Ahmad and Lipika Dey. “A method to compute distance between two categorical values of same attribute in unsupervised learning for categorical data set”. In: *Pattern Recognition Letters* 28.1 (2007), pp. 110–118.

[6] Leman Akoglu, Hanghang Tong, and Danai Koutra. “Graph based anomaly detection and description: A survey”. In: *Data Mining and Knowledge Discovery* 29.3 (2015), pp. 626–688.

[7] Leman Akoglu et al. “Fast and reliable anomaly detection in categorical data”. In: *CIKM*. ACM. 2012, pp. 415–424.

[8] Reid Andersen and Kumar Chellapilla. “Finding dense subgraphs with size bounds”. In: *Algorithms and Models for the Web-Graph* (2009), pp. 25–37.

[9] Fabrizio Angiulli and Fabio Fassetti. “Dolphin: An efficient algorithm for mining distance-based outliers in very large datasets”. In: *ACM Transactions on Knowledge Discovery from Data* 3.1 (2009), p. 4.

[10] Fabrizio Angiulli, Fabio Fassetti, and Luigi Palopoli. “Detecting outlying properties of exceptional objects”. In: *ACM Transactions on Database Systems* 34.1 (2009), p. 7.

[11] Fabrizio Angiulli and Clara Pizzuti. “Outlier mining in large high-dimensional data sets”. In: *IEEE Transactions on Knowledge and Data Engineering* 17.2 (2005), pp. 203–215.

[12] Fatemeh Azmandian et al. “GPU-accelerated feature selection for outlier detection using the local kernel density ratio”. In: *ICDM*. IEEE. 2012, pp. 51–60.

[13] Andrew Barron, Jorma Rissanen, and Bin Yu. “The minimum description length principle in coding and modeling”. In: *IEEE Transactions on Information Theory* 44.6 (1998), pp. 2743–2760.

[14] Stephen D Bay and Mark Schwabacher. “Mining distance-based outliers in near linear time with randomization and a simple pruning rule”. In: *KDD*. ACM. 2003, pp. 29–38.

[15] Norbert Beckmann et al. “The R*-tree: An efficient and robust access method for points and rectangles”. In: *SIGMOD*. ACM, 1990, pp. 322–331.

[16] Jon Louis Bentley. “Multidimensional binary search trees used for associative searching”. In: *Communications of the ACM* 18.9 (1975), pp. 509–517.

[17] Shyam Boriah, Varun Chandola, and Vipin Kumar. “Similarity measures for categorical data: A comparative evaluation”. In: *SDM*. SIAM. 2008, pp. 243–254.

[18] Whitney A Brechwald and Mitchell J Prinstein. “Beyond homophily: A decade of advances in understanding peer influence processes”. In: *Journal of Research on Adolescence* 21.1 (2011), pp. 166–179.

[19] Leo Breiman. “Bagging predictors”. In: *Machine Learning* 24.2 (1996), pp. 123–140.

[20] Markus M Breunig et al. “LOF: Identifying density-based local outliers”. In: *ACM SIGMOD Record* 29.2 (2000), pp. 93–104.

[21] Guilherme O Campos et al. “On the evaluation of unsupervised outlier detection: Measures, datasets, and an empirical study”. In: *Data Mining and Knowledge Discovery* (2016), pp. 1–37.

[22] Longbing Cao. “Coupling learning of complex interactions”. In: *Information Processing & Management* 51.2 (2015), pp. 167–186.

[23] Longbing Cao. “Non-iidness learning in behavioral and social data”. In: *The Computer Journal* 57.9 (2014), pp. 1358–1370.

[24] Longbing Cao, Xiangjun Dong, and Zhigang Zheng. “e-NSP: Efficient negative sequential pattern mining”. In: *Artificial Intelligence* 235 (2016), pp. 156–182.

[25] Longbing Cao, Yuming Ou, and Philip S Yu. “Coupled behavior analysis with applications”. In: *IEEE Transactions on Knowledge and Data Engineering* 24.8 (2012), pp. 1378–1392.

[26] Deepayan Chakrabarti and Christos Faloutsos. “Graph mining: Laws, generators, and algorithms”. In: *ACM Computing Surveys* 38.1 (2006), p. 2.

[27] Varun Chandola, Arindam Banerjee, and Vipin Kumar. “Anomaly detection: A survey”. In: *ACM Computing Surveys* 41.3 (2009), p. 15.

[28] Moses Charikar. “Greedy approximation algorithms for finding dense components in a graph”. In: *Approximation Algorithms for Combinatorial Optimization*. 2000, pp. 84–95.

[29] Duen Horng Chau et al. “Polonium: Tera-scale graph mining and inference for malware detection”. In: *SDM*. SIAM. 2011, pp. 131–142.

[30] Xuewen Chen and Michael Wasikowski. "FAST: A ROC-based feature selection metric for small samples and imbalanced data classification problems". In: *KDD*. ACM. 2008, pp. 124–132.

[31] Nicholas A Christakis and James H Fowler. "The spread of obesity in a large social network over 32 years". In: *New England Journal of Medicine* 357.4 (2007), pp. 370–379.

[32] Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid. "Approximate fisher kernels of non-iid image models for image categorization". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.6 (2016), pp. 1084–1098.

[33] Noel Cressie and Christopher K Wikle. *Statistics for spatio-temporal data*. John Wiley & Sons, 2015.

[34] Kaustav Das and Jeff Schneider. "Detecting anomalous records in categorical datasets". In: *KDD*. ACM. 2007, pp. 220–229.

[35] Janez Demšar. "Statistical comparisons of classifiers over multiple data sets". In: *The Journal of Machine Learning Research* 7 (2006), pp. 1–30.

[36] Persi Diaconis and Daniel Stroock. "Geometric bounds for eigenvalues of Markov chains". In: *The Annals of Applied Probability* 1.1 (1991), pp. 36–61.

[37] Trong Dinh Thac Do and Longbing Cao. "Coupled Poisson Factorization Integrated with User/Item Metadata for Modeling Popular and Sparse Ratings in Scalable Recommendation". In: *AAAI*. 2018.

[38] Devdatt P Dubhashi and Alessandro Panconesi. *Concentration of measure for the analysis of randomized algorithms*. Cambridge University Press, 2009.

[39] Andrew F Emmott et al. "Systematic construction of anomaly detection benchmarks from real data". In: *KDD Workshop*. ACM. 2013, pp. 16–21.

[40] Xuhui Fan, Richard Yi Da Xu, and Longbing Cao. "Copula Mixed-Membership Stochastic Blockmodel". In: *IJCAI*. 2016, pp. 1462–1468.

[41] James Allen Fill. "Eigenvalue Bounds on Convergence to Stationarity for Nonreversible Markov Chains, with an Application to the Exclusion Process". In: *The Annals of Applied Probability* 1.1 (1991), pp. 62–87.

[42] James H Fowler and Nicholas A Christakis. "Dynamic spread of happiness in a large social network: longitudinal analysis over 20 years in the Framingham Heart Study". In: *BMJ: British Medical Journal* 337 (2008), a2338.

[43] Yoav Freund and Robert E Schapire. "A desicion-theoretic generalization of on-line learning and an application to boosting". In: *COLT*. Springer. 1995, pp. 23–37.

[44] Murat Can Ganiz, Cibin George, and William M Pottenger. "Higher order Naive Bayes: A novel non-IID approach to text classification". In: *IEEE Transactions on Knowledge and Data Engineering* 23.7 (2011), pp. 1022–1034.

[45] Jing Gao et al. "A spectral framework for detecting inconsistency across multi-source object relationships". In: *ICDM*. IEEE. 2011, pp. 1050–1055.

[46] Amol Ghoting, Srinivasan Parthasarathy, and Matthew Eric Otey. "Fast mining of distance-based outliers in high-dimensional datasets". In: *SDM*. SIAM, 2006.

[47] Arnaud Giacometti and Arnaud Soulet. "Anytime algorithm for frequent pattern outlier detection". In: *International Journal of Data Science and Analytics* (2016), pp. 1–12.

[48] Jesús Gómez-Gardeñes and Vito Latora. "Entropy rate of diffusion processes on complex networks". In: *Physical Review E* 78.6 (2008), p. 65102.

[49] Sudipto Guha et al. "Robust Random Cut Forest Based Anomaly Detection On Streams". In: *ICML*. 2016, pp. 2712–2721.

[50] Manish Gupta et al. "Outlier detection for temporal data". In: *Synthesis Lectures on Data Mining and Knowledge Discovery* 5.1 (2014), pp. 1–129.

[51] Ali S Hadi and Jeffrey S Simonoff. "Procedures for the identification of multiple outliers in linear models". In: *Journal of the American Statistical Association* 88.424 (1993), pp. 1264–1272.

[52] Mark Hall et al. "The WEKA data mining software: An update". In: *ACM SIGKDD Explorations Newsletter* 11.1 (2009), pp. 10–18.

[53] David J Hand and Robert J Till. "A simple generalisation of the area under the ROC curve for multiple class classification problems". In: *Machine Learning* 45.2 (2001), pp. 171–186.

[54] Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical learning with sparsity: The lasso and generalizations*. CRC Press, 2015.

[55] Jingrui He. "Learning from Data Heterogeneity: Algorithms and Applications". In: *IJCAI*. 2017, pp. 5126–5130.

[56] Jingrui He and Jaime Carbonell. "Coselection of features and instances for unsupervised rare category analysis". In: *Statistical Analysis and Data Mining* 3.6 (2010), pp. 417–430.

[57] Zengyou He et al. "FP-outlier: Frequent pattern based outlier detection". In: *Computer Science and Information Systems* 2.1 (2005), pp. 103–118.

[58] Tin Kam Ho and Mitra Basu. "Complexity measures of supervised classification problems". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.3 (2002), pp. 289–300.

[59] Mingyi Hong and Zhi-Quan Luo. "On the linear convergence of the alternating direction method of multipliers". In: *Mathematical Programming* 162.1-2 (2017), pp. 165–199.

[60] Dino Ienco, Ruggero G Pensa, and Rosa Meo. "From context to distance: Learning dissimilarity for categorical data clustering". In: *ACM Transactions on Knowledge Discovery from Data* 6.1 (2012), p. 1.

[61] Hong Jia, Yiu-ming Cheung, and Jiming Liu. “A new distance metric for unsupervised learning of categorical data”. In: *IEEE Transactions on Neural Networks and Learning Systems* 27.5 (2016), pp. 1065–1079.

[62] Songlei Jian et al. “Embedding-based Representation of Categorical Data by Hierarchical Value Coupling Learning”. In: *IJCAI*. 2017, pp. 1937–1943.

[63] Shengyi Jiang et al. “A clustering-based method for unsupervised intrusion detections”. In: *Pattern Recognition Letters* 27.7 (2006), pp. 802–810.

[64] Fabian Keller, Emmanuel Muller, and Klemens Bohm. “HiCS: High contrast subspaces for density-based outlier ranking”. In: *ICDE*. 2012, pp. 1037–1048.

[65] Samir Khuller and Barna Saha. “On finding dense subgraphs”. In: *Automata, Languages and Programming*. 2009, pp. 597–608.

[66] Edwin M Knox and Raymond T Ng. “Algorithms for mining distance based outliers in large datasets”. In: *VLDB*. Citeseer. 1998, pp. 392–403.

[67] Ron Kohavi and George H John. “Wrappers for feature subset selection”. In: *Artificial Intelligence* 97.1 (1997), pp. 273–324.

[68] Anna Koufakou, Jimmy Secretan, and Michael Georgiopoulos. “Non-derivable itemsets for fast outlier detection in large high-dimensional categorical data”. In: *Knowledge and Information Systems* 29.3 (2011), pp. 697–725.

[69] Danai Koutra et al. “Unifying guilt-by-association approaches: Theorems and fast algorithms”. In: *ECMLPKDD* (2011), pp. 245–260.

[70] Hans-Peter Kriegel and Arthur Zimek. “Angle-based outlier detection in high-dimensional data”. In: *KDD*. 2008, pp. 444–452.

[71] Hans-Peter Kriegel et al. “Interpreting and unifying outlier scores”. In: *SDM*. 2011, pp. 13–24.

[72] Amy N Langville and Carl D Meyer. “Deeper inside pagerank”. In: *Internet Mathematics* 1.3 (2004), pp. 335–380.

[73] Aleksandar Lazarevic and Vipin Kumar. “Feature bagging for outlier detection”. In: *KDD*. ACM. 2005, pp. 157–166.

[74] Enrique Leyva, Antonio González, and Raul Perez. “A set of complexity measures designed for applying meta-learning to instance selection”. In: *IEEE Transactions on Knowledge and Data Engineering* 27.2 (2015), pp. 354–367.

[75] Jundong Li et al. “Feature Selection: A Data Perspective”. In: *ACM Computing Surveys* 50.6 (2017), p. 94.

[76] Sheng Li, Ming Shao, and Yun Fu. “Multi-view low-rank analysis for outlier detection”. In: *SDM*. SIAM. 2015, pp. 748–756.

[77] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. “Isolation-based anomaly detection”. In: *ACM Transactions on Knowledge Discovery from Data* 6.1 (2012), 3:1–3:39.

[78] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. "On detecting clustered anomalies using SCiForest". In: *ECMLPKDD.* Springer, 2010, pp. 274–290.

[79] Huan Liu and Lei Yu. "Toward integrating feature selection algorithms for classification and clustering". In: *IEEE Transactions on Knowledge and Data Engineering* 17.4 (2005), pp. 491–502.

[80] Sebastián Maldonado, Richard Weber, and Fazel Famili. "Feature selection for high-dimensional class-imbalanced data sets using Support Vector Machines". In: *Information Sciences* 286 (2014), pp. 228–246.

[81] Henrique O Marques et al. "On the internal evaluation of unsupervised outlier detection". In: *SSDBM.* ACM. 2015.

[82] Mary McGlohon et al. "SNARE: a link analytic system for graph labeling and risk detection". In: *KDD.* ACM. 2009, pp. 1265–1274.

[83] Miller McPherson, Lynn Smith-Lovin, and James M Cook. "Birds of a feather: Homophily in social networks". In: *Annual review of sociology* 27.1 (2001), pp. 415–444.

[84] Carl D Meyer. *Matrix analysis and applied linear algebra.* SIAM, 2000.

[85] Mehryar Mohri and Afshin Rostamizadeh. "Rademacher complexity bounds for non-iid processes". In: *NIPS.* 2009, pp. 1097–1104.

[86] Keith Noto, Carla Brodley, and Donna Slonim. "FRaC: a feature-modeling approach for semi-supervised and unsupervised anomaly detection". In: *Data Mining and Knowledge Discovery* 25.1 (2012), pp. 109–133.

[87] Stephen Oliver. "Proteomics: guilt-by-association goes global". In: *Nature* 403.6770 (2000), p. 601.

[88] Matthew Eric Otey, Amol Ghoting, and Srinivasan Parthasarathy. "Fast distributed outlier detection in mixed-attribute data sets". In: *Data Mining and Knowledge Discovery* 12.2-3 (2006), pp. 203–228.

[89] Lawrence Page et al. "The PageRank citation ranking: Bringing order to the Web". In: *WWW Conference.* 1998, pp. 161–172.

[90] Sinno Jialin Pan and Qiang Yang. "A survey on transfer learning". In: *IEEE Transactions on knowledge and data engineering* 22.10 (2010), pp. 1345–1359.

[91] Guansong Pang, Longbing Cao, and Ling Chen. "Outlier detection in complex categorical data by modelling the feature value couplings". In: *IJCAI.* 2016, pp. 1902–1908.

[92] Guansong Pang, Kai Ming Ting, and David Albrecht. "LeSiNN: Detecting anomalies by identifying Least Similar Nearest Neighbours". In: *ICDM Workshop.* IEEE. 2015, pp. 623–630.

[93] Guansong Pang et al. "Learning Homophily Couplings from Non-IID Data for Joint Feature Selection and Noise-Resilient Outlier Detection". In: *IJCAI.* 2017, pp. 2585–2591.

[94] Guansong Pang et al. "Selective Value Coupling Learning for Detecting Outliers in High-Dimensional Categorical Data". In: *CIKM*. ACM. 2017, pp. 807–816.

[95] Guansong Pang et al. "Sparse Modeling-based Sequential Ensemble Learning for Effective Outlier Detection in High-dimensional Numeric Data". In: *AAAI*. AAAI Press, 2018, pp. 3892–3899.

[96] Guansong Pang et al. "Unsupervised feature selection for outlier detection by modelling hierarchical value-feature couplings". In: *ICDM*. IEEE, 2016, pp. 410–419.

[97] Guansong Pang et al. "ZERO++: Harnessing the power of zero appearances to detect anomalies in large-scale data sets". In: *Journal of Artificial Intelligence Research* 57 (2016), pp. 593–620.

[98] Heiko Paulheim and Robert Meusel. "A decomposition of the outlier detection problem into a set of supervised learning problems". In: *Machine Learning* 100.2-3 (2015), pp. 509–531.

[99] Hanchuan Peng, Fuhui Long, and Chris Ding. "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.8 (2005), pp. 1226–1238.

[100] Ninh Pham and Rasmus Pagh. "A near-linear time approximation algorithm for angle-based outlier detection in high-dimensional data". In: *KDD*. ACM. 2012, pp. 877–885.

[101] Sridhar Ramaswamy, R Rastogi, and K Shim. "Efficient algorithms for mining outliers from large data sets". In: *ACM SIGMOD Record* (2000), pp. 427–438.

[102] Shebuti Rayana and Leman Akoglu. "Less is more: Building selective anomaly ensembles". In: *ACM Transactions on Knowledge Discovery from Data* 10.4 (2016), p. 42.

[103] Shebuti Rayana, Wen Zhong, and Leman Akoglu. "Sequential ensemble learning for outlier detection: A bias-variance perspective". In: *ICDM*. IEEE. 2016, pp. 1167–1172.

[104] Saket Sathe and Charu C Aggarwal. "Subspace outlier detection in linear time with randomized hashing". In: *ICDM*. IEEE, 2016, pp. 459–468.

[105] Erich Schubert, Arthur Zimek, and Hans-Peter Kriegel. "Local outlier detection reconsidered: A generalized view on locality with applications to spatial, video, and network outlier detection". In: *Data Mining and Knowledge Discovery* 28.1 (2014), pp. 190–237.

[106] Erich Schubert et al. "DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN". In: *ACM Transactions on Database Systems* 42.3 (2017), p. 19.

[107] Erich Schubert et al. "On evaluation of outlier rankings and outlier scores". In: *SDM*. SIAM. 2012, pp. 1047–1058.

[108] Koen Smets and Jilles Vreeken. “The Odd One Out: Identifying and Characterising Anomalies”. In: *SDM*. 2011, pp. 109–148.

[109] Michael R Smith, Tony Martinez, and Christophe Giraud-Carrier. “An instance level analysis of data complexity”. In: *Machine Learning* 95.2 (2014), pp. 225–256.

[110] Ingo Steinwart and Andreas Christmann. “Fast learning from non-iid observations”. In: *NIPS*. 2009, pp. 1768–1776.

[111] Mahito Sugiyama and Karsten Borgwardt. “Rapid Distance-Based Outlier Detection via Sampling”. In: *NIPS* (2013), pp. 467–475.

[112] Yizhou Sun and Jiawei Han. “Mining heterogeneous information networks: principles and methodologies”. In: *Synthesis Lectures on Data Mining and Knowledge Discovery* 3.2 (2012), pp. 1–159.

[113] Acar Tamersoy, Kevin Roundy, and Duen Horng Chau. “Guilt by association: large scale malware detection by mining file-relation graphs”. In: *KDD*. ACM. 2014, pp. 1524–1533.

[114] Guanting Tang et al. “Mining multidimensional contextual outliers from categorical relational data”. In: *Intelligent Data Analysis* 19.5 (2015), pp. 1171–1192.

[115] Jiliang Tang et al. “Exploiting homophily effect for trust prediction”. In: *WSDM*. ACM. 2013, pp. 53–62.

[116] Kai Ming Ting et al. “Defying the gravity of learning curve: A characteristic of nearest neighbour anomaly detectors”. In: *Machine Learning* 106.1 (2017), pp. 55–91.

[117] Peter D. Turney and Patrick Pantel. “From frequency to meaning: Vector space models of semantics”. In: *Journal of Artificial Intelligence Research* 37 (2010), pp. 141–188.

[118] Can Wang et al. “Coupled Attribute Similarity Learning on Categorical Data”. In: *IEEE Transactions on Neural Networks and Learning Systems* 26.4 (2015), pp. 781–797.

[119] Larry Wasserman. *All of statistics: A concise course in statistical inference*. Springer Science & Business Media, 2013.

[120] Weng-Keen Wong et al. “Bayesian network anomaly pattern detection for disease outbreaks”. In: *ICML*. 2003, pp. 808–815.

[121] Shu Wu and Shengrui Wang. “Information-Theoretic Outlier Detection for Large-Scale Categorical Data”. In: *IEEE Transactions on Knowledge and Data Engineering* 25.3 (2013), pp. 589–602.

[122] Weinan Zhang, Tianming Du, and Jun Wang. “Deep learning over multi-field categorical data”. In: *ECIR*. Springer. 2016, pp. 45–57.

[123] Handong Zhao et al. “Consensus Regularized Multi-View Outlier Detection”. In: *IEEE Transactions on Image Processing* 27.1 (2018), pp. 236–248.

[124] Zhi-Hua Zhou. *Ensemble methods: foundations and algorithms*. CRC press, 2012.

[125] Zhi-Hua Zhou, Yu-Yin Sun, and Yu-Feng Li. “Multi-instance learning by treating instances as non-iid samples”. In: *ICML*. ACM. 2009, pp. 1249–1256.

[126] Chengzhang Zhu et al. “Heterogeneous metric learning of categorical data with hierarchical couplings”. In: *IEEE Transactions on Knowledge and Data Engineering* 30.7 (2018), pp. 1254–1267.

[127] Arthur Zimek, Ricardo J G B Campello, and Jörg Sander. “Ensembles for Unsupervised Outlier Detection: Challenges and Research Questions”. In: *ACM SIGKDD Explorations Newsletter* 15.1 (2013), pp. 11–22.

[128] Arthur Zimek, Erich Schubert, and Hans-Peter Kriegel. “A survey on unsupervised outlier detection in high-dimensional numerical data”. In: *Statistical Analysis and Data Mining* 5.5 (2012), pp. 363–387.

[129] Arthur Zimek et al. “Subsampling for efficient and effective unsupervised outlier detection ensembles”. In: *KDD*. ACM. 2013, pp. 428–436.