

© 2021 Dawei Zhou

HARNESSING RARE CATEGORY TRINITY FOR COMPLEX DATA

BY

DAWEI ZHOU

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois Urbana-Champaign, 2021

Urbana, Illinois

Doctoral Committee:

Associate Professor Jingrui He, Chair

Professor Jiawei Han

Professor Heng Ji

Associate Professor Leman Akoglu, Carnegie Mellon University

ABSTRACT

In the era of big data, we are inundated with the sheer volume of data being collected from various domains. In contrast, it is often the rare occurrences that are crucially important to many high-impact domains with diverse data types. For example, in online transaction platforms, the percentage of fraudulent transactions might be small, but the resultant financial loss could be significant; in social networks, a novel topic is often neglected by the majority of users at the initial stage, but it could burst into an emerging trend afterward; in the Sloan Digital Sky Survey, the vast majority of sky images (e.g., known stars, comets, nebulae, etc.) are of no interest to the astronomers, while only 0.001% of the sky images lead to novel scientific discoveries; in the worldwide pandemics (e.g., SARS, MERS, COVID19, etc.), the primary cases might be limited, but the consequences could be catastrophic (e.g., mass mortality and economic recession). Therefore, studying such complex rare categories have profound significance and longstanding impact in many aspects of modern society, from preventing financial fraud to uncovering hot topics and trends, from supporting scientific research to forecasting pandemic and natural disasters.

In this thesis, we propose a generic learning mechanism with trinity modules for complex rare category analysis: **(M1) Rare Category Characterization** - characterizing the rare patterns with a compact representation; **(M2) Rare Category Explanation** - interpreting the prediction results and providing relevant clues for the end-users; **(M3) Rare Category Generation** - producing synthetic rare category examples that resemble the real ones. The key philosophy of our mechanism lies in “all for one and one for all” - each module makes unique contributions to the whole mechanism and thus receives support from its companions. In particular, M1 serves as the *de-novo* step to discover rare category patterns on complex data; M2 provides a proper lens to the end-users to examine the outputs and understand the learning process; and M3 synthesizes real rare category examples for data augmentation to further improve M1 and M2. To enrich the learning mechanism, we develop principled theorems and solutions to characterize, understand, and synthesize rare categories on complex scenarios, ranging from static rare categories to time-evolving rare categories, from attributed data to graph-structured data, from homogeneous data to heterogeneous data, from low-order connectivity patterns to high-order connectivity patterns, etc. It is worthy of mentioning that we have also launched one of the first visual analytic systems for dynamic rare category analysis, which integrates our developed techniques and enables users to investigate complex rare categories in practice.

This thesis is dedicated to my parents, Jun Zhou and Huichun Gu, for their love and support.

ACKNOWLEDGMENTS

First and foremost, I feel greatly indebted to my advisor Dr. Jingrui He, who is the best advisor I can hope for. She is a true inspiration to me in research and sets me to a high standard of being a scholar. Her passion for research and her dedication to science have deeply influenced me and shaped my research mindset. Moreover, from Jingrui, I learned that the best way of mentoring is to be supportive. She is always open-minded and encourages me to get involved in a variety of academic activities, including presenting in conferences, reviewing papers, giving guest lectures, mentoring students, serving conference program committees, and organizing tutorials/workshops. Over the past years, she prepared me not only as a qualified Ph.D. student but also as an independent scholar. To me, Jingrui is my life-long role model, and I will follow her steps to be a true scholar and also spread her spirits to the next generation. I would like to sincerely appreciate Dr. Hanghang Tong for being my great mentor and dear friend. He provided immense help for my job search by sharing lots of his valuable experiences, encouraging me to maintain a positive attitude towards setbacks, and providing pertinent feedback on my application materials. I also would like to express my gratitude to Dr. Jiawei Han, Dr. Heng Ji, Dr. Leman Akoglu for serving on my thesis committee. Your valuable comments, insightful questions, and generous support have helped me a lot in improving my thesis work. In particular, I would like to thank Jiawei for advising me during my job hunting process so that I could avoid taking detours when making the critical career decisions.

I would like to thank my fantastic group of collaborators: Lecheng Zheng, Si Zhang, Dongqi Fu, Yada Zhu, Jiawei Han, Jiebo Luo, Jinjie Gu, Ross Maciejewski, Hasan Davulcu, Nan Cao, K. Selcuk Candan, Wendi Heinzelman, Hongxia Yang, Jianbo Li, Yu Cao, Jaesun Seo, and Henry Kautz. It was Yada and Jianbo who introduced me to the area of AI for Finance and brought me the “real data and real problems”. It was my best research companions, Lecheng, Dongqi, and Si who influenced me with their precise on research and always inspired me with novel ideas.

I also would like to thank all the lab members and my wonderful friends: Pei Yang, Arun Reddy Nelakurthi, Yao Zhou, Xu Liu, Lecheng Zheng, Jun Wu, Dongqi Fu, Yikun Ban, Haonan Wang, Ziwei Wu, Wenxuan Bao, Yunzhe Qi, Liangyue Li, Chen Chen, Xing Su, Si Zhang, Boxin Du, Jian Kang, Qinghai Zhou, Lihui Liu, Baoyu Jing, Yuchen Yan, Zhe Xu, Shweta Jain. I really appreciate and enjoy the time spending with them, from springs to winters, from Arizona to Illinois, from students to scholars.

At last, I would like to give my special gratitude to my parents (Jun Zhou and Huichun Gu), who believe in me and support me throughout my life; my wife (Kangyang Wang), who is an angel lighting up my life and calming my soul; my daughter (Joyce) and son (Jasper), who are naive but also my ultimate origin of strength and confidence.

TABLE OF CONTENTS

CHAPTER 1	INTRODUCTION	1
1.1	Background	2
1.2	Overview	3
1.3	Main Modules	5
1.4	Contribution and Impact	8
1.5	Organization	10
1.6	General Notation	10
CHAPTER 2	LITERATURE REVIEW	13
2.1	Mining Rare Categories in Attributed Data.	13
2.2	Mining Rare Categories in Static Graphs.	14
2.3	Mining Rare Categories in Time-Series Data	15
2.4	Mining Rare Categories in Temporal Graphs	16
CHAPTER 3	MULTI-VIEW RARE CATEGORY CHARACTERIZATION	18
3.1	Overview and Motivation	18
3.2	Related Work	19
3.3	Algorithm	20
3.4	Experimental Evaluation	24
3.5	Summary	29
CHAPTER 4	RARE CATEGORY CHARACTERIZATION ON TIME-EVOLVING GRAPHS	31
4.1	Overview and Motivation	31
4.2	Related Work	32
4.3	Algorithm	34
4.4	Query Dynamics	43
4.5	Experimental Evaluation	48
4.6	Summary	54
CHAPTER 5	BI-LEVEL RARE TEMPORAL PATTERN CHARACTERIZATION	55
5.1	Overview and Motivation	55
5.2	Related Work	57
5.3	Algorithm	58
5.4	Experimental Evaluation	68
5.5	Summary	73

CHAPTER 6	HIGH-ORDER RARE CATEGORY CHARACTERIZATION	74
6.1	Overview and Motivation	74
6.2	Related Work	76
6.3	Preliminaries	78
6.4	Algorithm	80
6.5	Generalizations and Applications	90
6.6	Experimental Evaluation	92
6.7	Summary	101
CHAPTER 7	DOMAIN ADAPTIVE RARE CATEGORY CHARACTERIZATION	102
7.1	Overview and Motivation	102
7.2	Related Work	104
7.3	Preliminaries	105
7.4	Algorithm	107
7.5	Experimental Evaluation	112
7.6	Summary	117
CHAPTER 8	RARE CATEGORY REPRESENTATION LEARNING	118
8.1	Overview and Motivation	118
8.2	Related Work	120
8.3	Preliminaries	122
8.4	Algorithm	123
8.5	Experimental Evaluation	131
8.6	Summary	138
CHAPTER 9	VISUAL ANALYTIC TOOL FOR RARE CATEGORY EXPLA- NATION	139
9.1	Overview and Motivation	139
9.2	Related Work	141
9.3	Preliminaries	143
9.4	System Design	146
9.5	System Evaluation	156
9.6	Discussion	160
9.7	Summary	161
CHAPTER 10	TEMPORAL INTERACTION NETWORK GENERATION	162
10.1	Overview and Motivation	162
10.2	Related Work	164
10.3	Preliminaries	165
10.4	Algorithm	167
10.5	Experimental Evaluation	174
10.6	Summary	180

CHAPTER 11	FAIR GENERATION FOR RARE CATEGORIES ON GRAPHS . .	182
11.1	Overview and Motivation	182
11.2	Related Work	184
11.3	Preliminaries	185
11.4	Algorithm	188
11.5	Experimental Evaluation	196
11.6	Summary	201
CHAPTER 12	CONCLUSION AND FUTURE WORK	202
12.1	Conclusion	202
12.2	Vision and Future Work	202
REFERENCES	207

CHAPTER 1: INTRODUCTION

The success of modern artificial intelligence has been partially attributed to the big data and the advanced data management techniques. However, in contrast to the sheer volume of data being collected, it is often the rare categories (i.e., the minority classes with scarce observations) that are of great importance in many high-impact domains. For example, in online transaction platforms, the percentage of fraudulent transactions might be small, but the resultant financial loss could be significant; in social networks, a novel topic is often neglected by the majority of users at the initial stage, but it could burst into an emerging trend afterward; in the Sloan Digital Sky Survey (Figure 1.1), the vast majority of sky images (e.g., known stars, comets, nebulae, etc.) are of no interest to the astronomers, while only 0.001% of the sky images could lead to novel scientific discoveries; in the worldwide pandemics (e.g., SARS, MERS, COVID19, etc.), the primary cases might be limited, but the consequences could be catastrophic (e.g., mass mortality and economic recession). Given the profound significance, the main focus of my research aims to develop artificial intelligence and machine

learning models and techniques that can systematically investigate the minority examples and help the end-users gain a better understanding of these rare categories on complex data. In particular, our work contributes to a wide spectrum of application domains with diverse data formats, ranging from static rare categories [2, 3, 4] to time-evolving rare categories [5, 6, 7, 8, 9, 10, 11], from attributed data [2] to graph-structured data [4, 9, 12, 13],

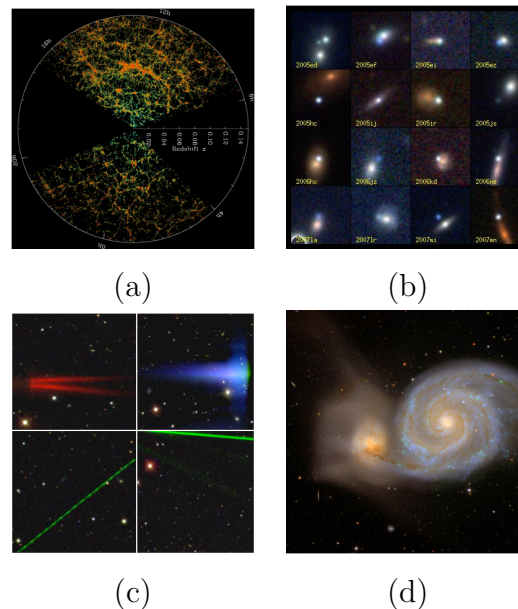


Figure 1.1: An illustration of Sloan Digital Sky Survey [1]. (a) The universe map, where each dot is a galaxy. (b) Known galaxies in the universe. (c) Uninteresting anomalies (e.g., diffraction spikes of satellite trails or the artifacts of the telescope). (d) Interesting anomalies that lead to discovery extraordinary objects (e.g., the bright spiral galaxy M51 and its fainter companion). It is revealed that 99% of the anomalies are uninteresting patterns, such as diffraction spikes shown in (c), and only 1% are interesting patterns that are worthy of future research and may lead to the discovery of extraordinary objects like (d).

from homogeneous data [5, 6, 7] to heterogeneous data [2, 14], from low-order connectivity patterns [9, 15] to high-order connectivity patterns [10, 16, 17, 18], etc.

1.1 BACKGROUND

The branch of data mining concerned with identifying rare events has a longstanding history. Backtracking to 1980, Douglas M. Hawkins firstly proposed the definition of outliers [19] in Def. 1.1. Following the Hawkins’ definition of outliers, the problem of anomaly detection or outlier detection has been generalized and studied in various contexts, such as high-dimensional numerical data [20], sequential data [21], time-series data [22], graph data [23], financial data [24, 25], and thus resulted in many domain-specific names for outliers and anomalies, such as novelties, events, surprising changes, fraud, outbreaks, etc.

Definition 1.1. Hawkins’ Definition of Outliers [19]

An outlier is an observation that differs so much from other observations as to arouse suspicion that it was generated by a different mechanism.

Despite the tremendous success of anomaly detection methods in a variety of domains, it is commonly agreed that not all anomalies are necessarily useful or relevant to the actual events of interest. In fact, most anomalies are uninteresting data points, which are drawn from the known distribution of noise or correspond to the combinations of features that are less valuable to the downstream applications [26]. Recalling the illustrative example in Figure 1.1, we present a set of sky images captured by ground-based telescopes in the program of Sloan Digital Sky Survey (SDSS). According to the analytics from SDSS, 99.9% of the captured sky images (Figure 1.1 (b)) by SDSS can be well explained based on the known phenomena of the universe (e.g., discovered galaxies, stars, comets, nebulae, etc.) and only 0.1% of the images (the bottom row of Figure 1.1) are anomalies. Moreover, within the anomalies, 99% of the images (Figure 1.1 (c)) are of no interest to astronomers and are caused by the diffraction spikes of satellite trails or the artifacts of the telescope; and only 1% of the abnormal instances (a minuscule 0.001% of the whole SDSS database) are useful, and the patterns are of interest, which can correspond to unknown objects and may lead to new scientific discoveries (e.g., Figure 1.1 (d)). Here, we refer to the anomalies that are not only statistically significant but also interesting as the rare category examples. Meanwhile, the problem of studying the rare category examples is referred to rare category analysis. In traditional rare category analysis, we are given an imbalanced data set \mathcal{D} that consists of C distinct classes (majority classes and minority classes). The goal is to identify examples

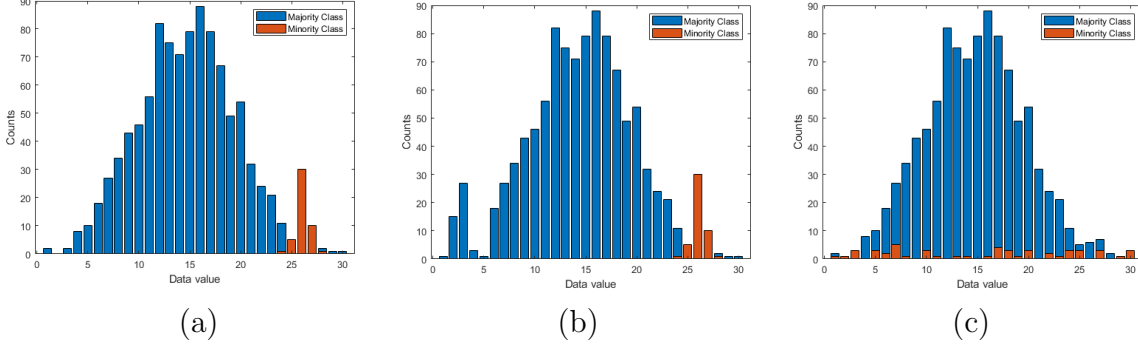


Figure 1.2: The support regions of a majority class and a minority class in a one-dimensional synthetic data set. (a) Both Assumption 1.1 & 1.2 hold. (b) Only Assumption 1.1 holds. (c) Only Assumption 1.2 holds.

from the minority classes with high accuracy and recall. In general, we make the following assumptions regarding the support region of the majority classes and the minority classes.

Assumption 1.1. Smoothness Assumption for Majority Class. Given a highly skewed data set \mathcal{D} , the distribution of the support region of each majority class is sufficiently smooth.

Assumption 1.2. Compactness Assumption for Minority Class. Given a highly skewed data set \mathcal{D} , the minority class examples can be represented as a compact cluster in the feature space.

These assumptions are made for the purpose that the rare categories are identifiable and meaningful. To be more clear, let us first look at the example in Figure 1.2 (a), where the majority class (colored in blue) has a Gaussian distribution with a large variance on the left while the minority class (colored in orange) corresponds to a peak with a small variance on the right. If the distribution of the majority class is not smooth and violates Assumption 1.1 (e.g., the majority class in Figure 1.2 (b) consists of multiple narrow and sharp peaks just as the minority class), then the minority class cannot be identified with a clear clue; if the minority class is not compact and violates Assumption 1.2 (e.g., the minority class in Figure 1.2 (c) is uniformly distributed in the feature space), then no algorithm can outperform the random-sampling approach.

1.2 OVERVIEW

This thesis aims to provide a *generic learning mechanism for rare category analysis on complex data*. As shown in Figure 1.3, it consists of three key modules: **(M1) Rare Cat-**

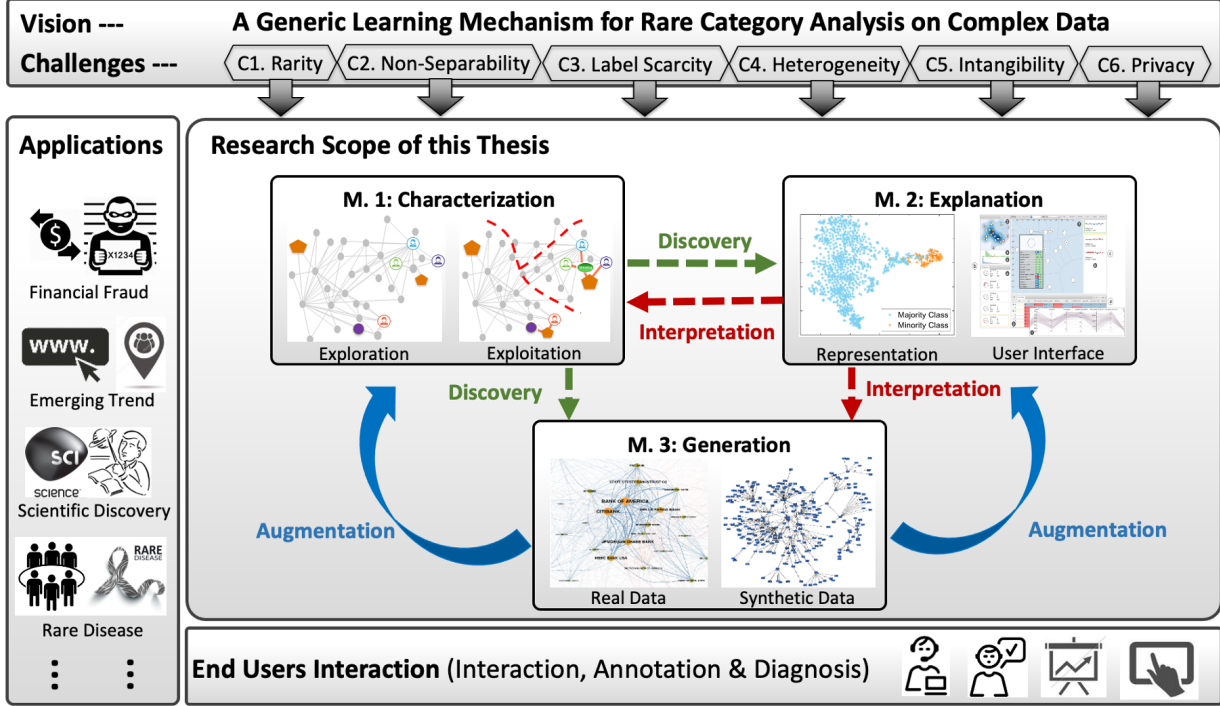


Figure 1.3: Complex rare category analysis.

egory Characterization - how to characterize the rare patterns with a compact representation? **(M2) Rare Category Explanation** - how to interpret the prediction results and provide relevant clues for the end-users? **(M3) Rare Category Generation** - how to produce synthetic rare category examples that resemble the real ones? In addition, the proposed mechanism operates through a *mutually beneficial synergy* (shown in Figure 1.3) among these three modules, where the rare category characterization module serves as the *de-novo* step to characterize and identify potential rare category examples without or with limited annotated data; the rare category explanation module aims to provide a proper lens (e.g., from the right/relevant data sources, in the subspace spanned by the right/relevant attributes, at the right/relevant time steps) to examine and interpret the outputs from M1 and M3; the rare category generation module incorporates the raw data as well as the discoveries from M1 and M2 to mimic the underlying data distribution and thus enable data augmentation.

Complex rare category analysis is confronted with several unique challenges as follows.

- (C1) *rarity* - the target signals are often extremely rare compared to the massive background signals.
- (C2) *non-separability* - the target signals (the minority classes) are often non-separable

from background signals (the majority classes) in the given feature space.

- (C3) *label scarcity* - it is often expensive, or sometimes infeasible, to collect labels of the target signals (the minority classes).
- (C4) *heterogeneity* - the real-world scenarios often exhibit data and task heterogeneity, e.g., the multi-modal representation of examples and the analysis of similar rare categories across multiple related tasks.
- (C5) *intangibility* - due to the “black-box” nature, many advanced machine learning models are capable of memorizing complex concepts between input features and output labels, but lack an intuitive and interpretable way to make the underlying process transparent to the end-users.
- (C6) *privacy* - With increasing demand for AI systems as service providers in an expanding list of domains (e.g., finance, healthcare), massive data containing sensitive information are generated (e.g., sex, income, age), which poses a substantial security challenge in releasing and sharing them.

The conventional machine learning tools, especially those requiring large, clean, and annotated data, may fail in practice with rare examples, which motivates the studies of rare category analysis in the past decades. Nevertheless, the previous work on rare category analysis mostly focuses on addressing the challenges associated with the nature rare category examples (C1, C2, C3), while neglecting the emerging challenges (C4, C5, C6) arising in the era of big data.

1.3 MAIN MODULES

Our proposed framework focuses on characterizing, interpreting, and even generating rare category patterns for complex data (e.g., multi-view data, multi-resolution time series, temporal interaction networks, etc.). My thesis research work (shown in Figure 1.3) boils down to developing principled learning algorithms to tackle the unique challenges associated with the trinity modules (M1, M2, M3) for rare category analysis as follows.

1.3.1 Rare Category Characterization

Collecting and annotating rare category examples are extremely expensive and time-consuming (C1, C2, C3). Therefore, directly training conventional machine learning models

in the scarcity of labels would introduce inevitable model bias and largely degrade the model performance in identifying rare category examples. Previous efforts on rare category analysis mainly rely on semi-supervised learning to characterize rare category patterns in a compact representation. However, most, if not all, of the previous works only focus on the single view, single resolution, pair-wised connectivity patterns and the static settings, which might not be optimal in real-world applications with data heterogeneity (C4).

My work on rare category characterization focuses on understanding the nature and characteristics of rare category patterns in a variety of real scenarios, where the data is represented with multiple views [27], multiple resolutions [6, 28], high-order structures [10, 16, 17], and dynamic patterns [6, 7, 10, 17, 29]. For instance, considering that the real data is often collected from multiple sources and exhibits multiple views, we have proposed an unsupervised algorithm MUVIR [27] that exploits the relationship among multiple views to estimate the overall probability of each example belonging to the rare categories; to explore the high-order rare category patterns (e.g., money laundering in the form of loop-structured transactions), we have made one of the first efforts [16] to model high-order connectivity patterns (e.g., triangle, loop, star) and have proposed a local graph clustering algorithm HOSPLOC that can identify structure-rich clusters without exploring the whole graph. Later on, we have generalized HOSPLOC to the dynamic setting and have developed a series of algorithms to compute [17] and track [10] *structure-rich* clusters in temporal networks. Compared with the previous work, the running time of our clustering algorithms only depends *polylogarithmically* on the size of the graph, which brings a useful tool for handling massive graphs in the real-world applications, such as synthetic identity detection in personally identifiable information (PII) networks [16], money laundering detection in online transaction networks [18], and emerging trend detection in time-evolving scientific collaboration networks [17].

1.3.2 Rare Category Explanation

Despite the phenomenal success of AI in recent years, many high-stake domains (e.g., financial forecasting, fraud detection, medical testing) have to rely on traditional rule-based mechanisms (e.g., decision tree), which are less effective but much more interpretable to the end-users. The main reason is that many advanced machine learning tools, especially deep learning models, often remain as the black-boxes in nature (C5), while many industries have to follow highly regulated processes - requiring prediction models to be interpretable and the output results to meet compliance. Therefore, a natural research question here is how we can make our models transparent to the end-user by identifying the right context (e.g.,

key factors, representative entities, critical timestamps).

To answer the aforementioned question, my research is mainly carried out from the following two directions: (1) data diagnosis [4, 11, 13, 15] (i.e., *how is the data distributed? which piece of information is more valuable than the others for a given task?*) (2) model diagnosis [8] (i.e., *why does the model make a certain prediction on a particular piece of information?*). For data diagnosis, we have proposed the first rare-category-oriented network embedding framework SPARC [4], which aims to learn a salient representation to characterize rare category examples. Inspired by the family of curriculum learning that simulates the cognitive mechanism of human beings, SPARC gradually selects the key network contextual information and learns the rare category oriented network representation, by shifting from the ‘easy’ concept to the ‘difficult’ concept. Our results show that (1) SPARC enables users to visualize the network layout in a salient embedding space, where the majority classes and minority classes are well separated, and (2) SPARC is able to identify valuable contextual information, which provides interpretation and guidance in the task of rare category characterization. Later on, to accommodate the data dynamics (C5), we have developed a series of representation and interpretation frameworks towards various types of data, including multi-modality time series [15], time-evolving graphs [13] and fine-grained temporal networks [11]. For model diagnosis, we have proposed the first visual analysis system RCAnalyzer (shown in Figure 1.4) for studying rare category patterns in dynamic systems, which includes: (a) a timeline view showing the overview of the given dynamic networks; (b) the matrices view showing the neighborhood contextual information of each node; (c) the example view showing the feature distribution of rare patterns; (d) the label result view showing the history prediction results as well as the model diagnosis.

1.3.3 Rare Category Generation

The ever-increasing size of data, together with the difficulty of releasing and sharing them (C6), has made the data generation a fundamental problem that is key in many high-impact applications, including fraud detection, recommendation, data security, and many more. Motivated by this, my research aims to develop deep generative models that enable scalable modeling of real data to extract key contextual information, distill knowledge, and generate plausible patterns for data augmentation in rare category analysis. In [12], we consider a practical scenario where the data is represented as a graph, and the target signals might exhibit hierarchical structures. We have developed a multi-scale graph generative model named MISC-GAN, which models the underlying distribution of the graph structures at different levels of granularity, and then ‘transfers’ such hierarchical distribution from the

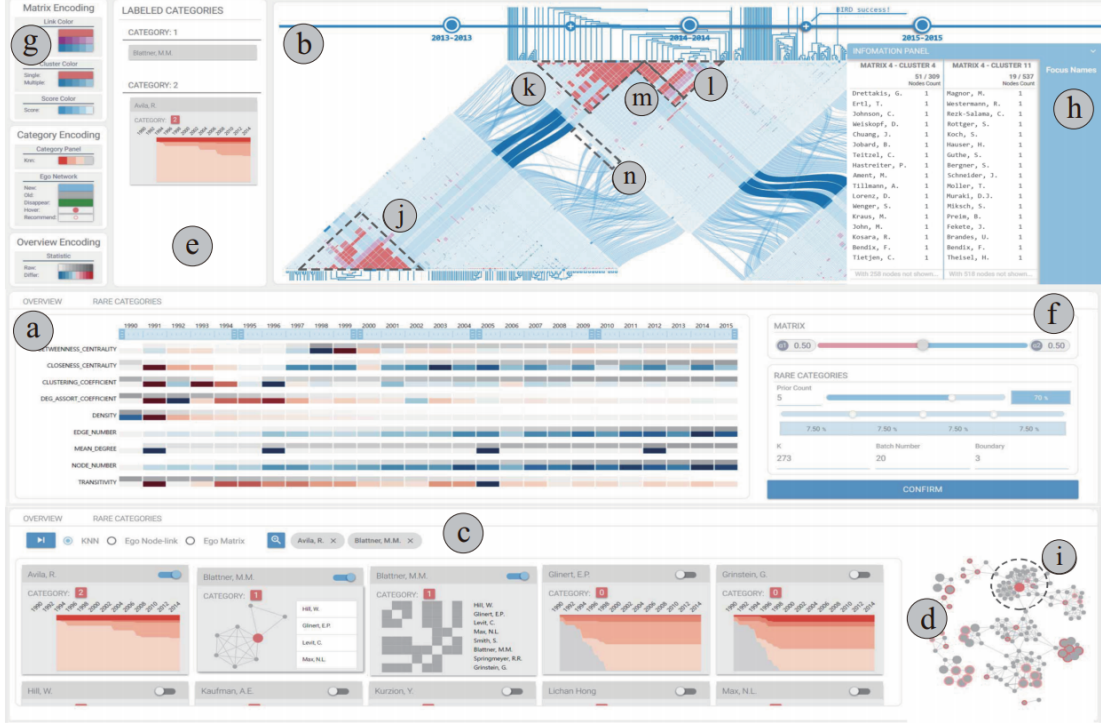


Figure 1.4: User interface of RCAnalyzer.

graphs in the domain of interest to a unique graph representation. Nonetheless, many realistic networks are intrinsically dynamic and are presented as a collection of system transactions (i.e., timestamped interactions/edges between entities). Hence, in [30], we have designed an end-to-end deep generative framework named TAGGEN that parameterizes a bi-level self-attention mechanism to jointly extract structural and temporal context information from the temporal interaction networks. Our results demonstrate that TAGGEN is able to (1) generate high-quality temporal networks, and (2) significantly boost the performance of rare category detection via data augmentation.

1.4 CONTRIBUTION AND IMPACT

Figure 1.5 summarizes my thesis work, ongoing research, and future plan in a two-dimensional conceptual space (data v.s. tasks). My ultimate goal is to build a *comprehensive and generic rare category analysis system*, which can be applied to solve a variety of tasks, ranging from rare category characterization for static data to rare category tracking for temporal data; from representing rare patterns in a salient embedding space to interpreting the prediction results and providing relevant clues for the end-users' interpretation; from mimicking the underlying distribution of rare categories to generating synthetic ones for data

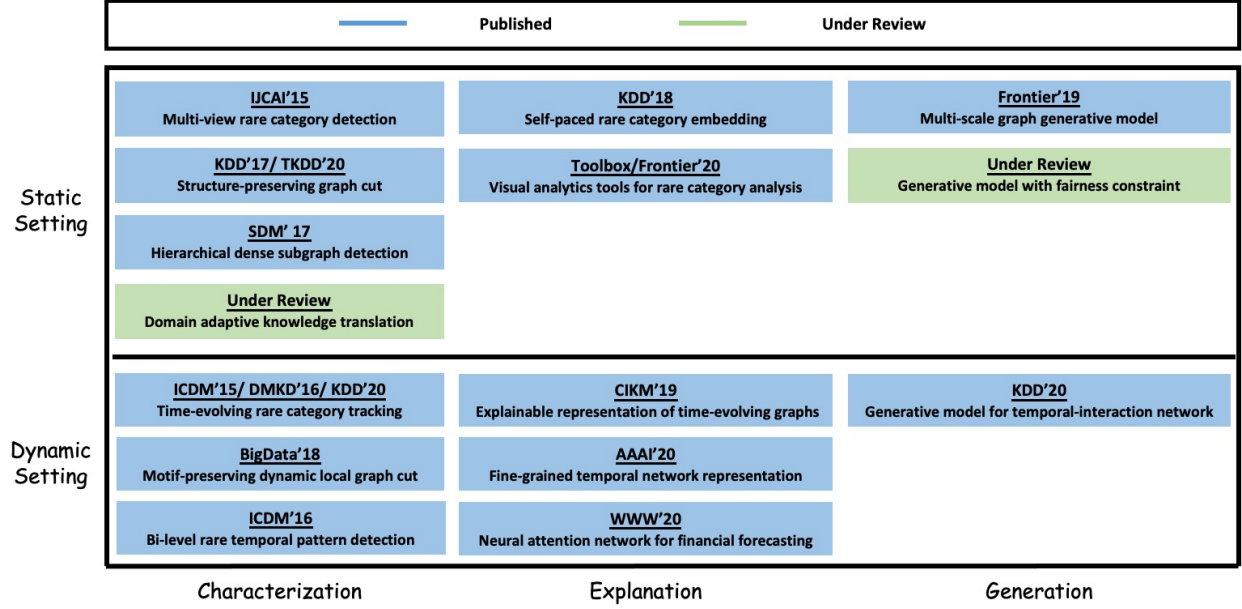


Figure 1.5: An overview of my thesis work, where the published papers are colored in blue and the under-review papers are colored in green.

augmentation. In particular, the contributions and impacts of my research are summarized as follows.

1. My research identified and analyzed the fundamental limits of rare category analysis for complex data. Motivated by such discoveries, my work aims to characterize, interpret, and augment a variety of learning tasks in the context of rare category analysis, which have been the key delivery to multiple funded projects by DARPA and DHS. Moreover, a variety of developed techniques have been transferred to the real-world systems in finance [15, 28], security [16], and healthcare [6].
2. My research on rare category analysis results in *fifteen first-authored publications* in the prestigious conferences and journals (KDD, WWW, ICDM, AAAI, IJCAI, TKDD, DMKD, etc.).
3. We have been invited by Computing Research Association (CRA) to showcase our system for analyzing complex rare categories at the *24th Annual CNSF Capitol Hill Exhibition*¹.
4. My tutorials² on complex rare category analysis attracted over *300 audience* members

¹<https://cra.org/govaffairs/blog/2018/05/2018-cnsf-exhibition/>

²<https://sites.google.com/view/dawei-zhou/talks?authuser=0>

in the top conferences of data mining and machine learning (BigData’18, SDM’19, KDD’19).

5. My work on rare category analysis has been adopted as teaching material in graduate classes at the University of Illinois at Urbana-Champaign and Arizona State University. Moreover, I was invited to give a keynote talk on graph-based rare category analysis at DGLMA’19.
6. I am the co-organizer of *Workshop on Using Alternative Data to Support Intelligent Decision for Financial Services*³ at SDM’20 and *Workshop on Deep Learning on Graphs*⁴ at AAAI’21.

1.5 ORGANIZATION

This thesis is organized into three main parts: (1) rare category characterization, (2) rare category explanation, and (3) rare category generation. We enumerate the main problems of each part in the form of questions in Table 1.1.

1.6 GENERAL NOTATION

We summarize the most common notations used in this thesis in Table 1.2. More specific notations and definitions will be covered in the corresponding chapters to explain the problems and the proposed algorithms. In general, we use regular letters to denote scalars (e.g., α), boldface lowercase letters to denote vectors (e.g., \mathbf{v}), and boldface uppercase letters to denote matrices (e.g., \mathbf{A}).

³<https://sites.google.com/view/sdm2020-finance/home>

⁴<https://deep-learning-graphs.bitbucket.io/dlg-aaai21/>

Modules	Research Problem	Chapter
I. Characterization	Multi-View Rare Category Characterization: How can we identify rare category examples when the given data exhibit multiple views?	3
	Rare Category Characterization on Time-Evolving Graphs: How can we identify rare category examples on time-evolving graphs?	4
	Bi-level Rare Temporal Pattern Characterization: How can we identify rare temporal patterns at both sequence-level and segment-level on time-series databases?	5
	High-Order Rare Category Characterization: How can we identify high-order rare categories that are represented as clusters of high-order connectivity patterns?	6
	Domain Adaptive Rare Category Characterization: How can we identify rare categories across different domains and ensure a good generalization performance of the learned predictor?	7
II. Explanation	Rare Category Representation Learning: How can we learn a salient embedding space for rare category analysis, where the minority classes are well separated from the majority classes?	8
	Visual Analytic Tool for Rare Category Explanation How can we visualize the hidden process of rare category analysis, and make it transparent to the end-user by identifying the right context (e.g., key factors, representative entities, critical timestamps)?	9
III. Generation	Temporal Interaction Network Generation: How can we model and synthesize temporal interaction networks for data augmentation in the downstream tasks (e.g., rare category analysis)?	10
	Fair Generation for Rare Categories on Graphs: How can we enforce the fairness constraints on the graph generative model so that the protected groups (e.g., rare categories) are well-preserved in the generated graphs?	11

Table 1.1: Thesis organization.

Symbol	Description
\mathcal{D}	the given dataset
\mathcal{L}	the set of labeled examples
\mathcal{U}	the set of unlabeled examples
\mathcal{Y}	the class label of \mathcal{D}
n	the size of \mathcal{D}
$\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$	the input data with n attributed examples
\mathbf{x}_i	the i^{th} example of \mathcal{X}
y_i	the class label of \mathbf{x}_i
$\mathcal{S} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(M)}\}$	the input time-series database with M time series
$\mathbf{x}^{(\mathbf{m})} = \{x_1^{(m)}, \dots, x_{n^{(m)}}^{(m)}\}$	the m^{th} time series in \mathcal{S}
$x_i^{(m)}$	the i^{th} timestamp of $\mathbf{x}^{(\mathbf{m})}$
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	the input graph
\mathcal{V}	the set of nodes in \mathcal{G}
\mathcal{E}	the set of edges in \mathcal{G}
\odot	Hadamard product
$ \cdot $	the cardinality of a set
$\ \cdot\ _p$	p -norm of a vector

Table 1.2: Symbols.

CHAPTER 2: LITERATURE REVIEW

In this chapter, we provide a literature review of the prior arts and related studies in mining rare category patterns. We organize this chapter from a data perspective, including (1) mining rare categories in attributed data, (2) mining rare categories in static graphs, (3) mining rare categories for time-series data, and (4) mining rare categories in temporal graphs. In particular, the first two sections focus on identifying rare events in the static setting, while the latter two sections aim to capture the evolving rare category patterns in the dynamic setting.

2.1 MINING RARE CATEGORIES IN ATTRIBUTED DATA.

Given an unlabeled data set with n samples $\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, and each sample comes with d -dimensional features, our goal is to identify at least one example from each class $y = 1, \dots, m$ with minimum queries. [26] is one of the first attempt for rare category exploration, which develops a mixture model to fit the data and designed a family of hint selection methods to select the rare examples with help from a human expert. Experimental results with different hint selection methods show the efficacy of the proposed rare category detection framework. [31] further studies the rare category exploration problem when the minority classes are non-separable from the majority classes. Specifically, the authors develop a nearest-neighbor-based rare category detection algorithm named NNDM, which gradually selects examples with the maximum changes in the local density on a certain scale and asks for the labeling from the oracle. Moreover, theoretical analysis shows that the methods will effectively select examples both on the boundary and in the interior of the rare categories, when the rare categories are compact, and the majority class distribution is locally smooth. Despite the promising results of NNDM with theoretical guarantees, the performance largely relies on the prior information. To alleviate the restriction of the above methods that relies on prior knowledge (e.g., the number of classes, the proportion of minority classes), [32] proposes a prior-free rare category detection algorithm named SEDER. Different from [31, 33], SEDER picks the potential rare examples with large neighborhood density changes for labeling, by performing semi-parametric density estimation. In the presence of noisy data and irrelevant features, [34] formulates the rare category exploration problem as a co-selection scheme, which recovers the relevant features and the representative examples from the rare categories. To obtain the optimal sets of relevant features and rare examples, the authors propose an effective searching procedure (i.e., PALM) based on augmented Lagrangian to

solve the optimization problem. In particular, PALM is designed in an alternative fashion to find the relevant features and the minority class examples.

2.2 MINING RARE CATEGORIES IN STATIC GRAPHS.

In many fields, graphs offer a unifying data structure for modeling structured and unstructured data. As a result, extensive researches on rare category exploration have been conducted to spot the rare category entities on graph-structured data. In particular, given an unlabeled graph $\mathcal{D} = \mathcal{G}(\mathcal{V}, \mathcal{E})$, where \mathcal{V} and \mathcal{E} denote the sets of nodes and edges in \mathcal{G} , our target is to identify initial nodes/edges from each rare categories. [33] extends the idea of [31] to the graph-structured data, by proposing a graph-based rare category detection algorithm named GRADE. They start from a global similarity matrix motivated from manifold ranking [35], which is used to get a compact representation for the examples from the minority classes. Then, a prior-oriented k -nearest-neighbor matrix is computed to capture the sharp local density changes near the boundary of minority classes and thus make it easier to capture the rare patterns. On top of GRADE, the authors develop a variation named GRADE-LI, which only requires an upper bound on the proportion of each rare category. GRADE-LI can work with the data when detailed class-membership distribution about the data is not available to the users.

Except for the plain graph, data often exhibit node-level and edge-level heterogeneity for various critical tasks in security, finance, medicine, and so on. In such data (referred to as the heterogeneous graph), each node and edge is associated with a specific type. For example, [36] proposes the notion of neighborhood formation for bipartite graphs, which computes the relevance score of all nodes to a query node v and defines the neighborhood of v as the set of nodes with higher relevance scores. Based on the neighborhood formation, the authors develop an anomaly detection algorithm to spot the abnormal nodes with low “normality” scores. [37] proposes a non-negative residual matrix factorization framework named NRMF, which aims to detect the malicious group of entities as well as provide interpretation of prediction results for data analysts. In particular, NRMF is built upon the conventional matrix factorization mechanism and imposes a residual constraint on the residual matrix in order to improve the interpretation for graph anomalies. To solve the optimization problem, the authors develop a fast optimization algorithm to incrementally compute the rank-1 approximation on the residual matrix computed from the last iteration, which runs in linear w.r.t. the size of the graph. [38] studies the problem anomaly detection in the streaming heterogeneous graphs, by proposing a clustering-based anomaly detection approach that can simultaneously address the heterogeneity and streaming nature of the

input data. In particular, the authors introduce a novel embedding mechanism that can encode the heterogeneous streaming graph into a vector representation, which will be used to perform clustering and identify the anomalous patterns. [39] proposes a GCN-based framework for predicting future events by capturing the contextual information from the raw data. The proposed framework first extracts graph representations of the events documents, then learns to predict the occurrence of future events and identify the events of interest (e.g., anomaly patterns). [40] studies the problem of video anomaly detection framework in the presence of noisy labels. Specifically, a graph convolutional network (GCN) is built to simultaneously correct noisy labels and spot abnormal actions.

2.3 MINING RARE CATEGORIES IN TIME-SERIES DATA

In the setting of time-series data, early studies of rare category exploration [41, 42, 43, 44, 45, 46, 47, 48] have a close relation to the outlier detection and disorder detection methods. They largely rely on the distanced-based mechanisms [44, 49, 50, 51] that define various similarity measurements [22] of and then identify rare patterns deviating from the normal ones. For instance, in [46], the authors propose a scalable distance-based detection algorithm for high-volume data streams, which has been demonstrated to be optimal in terms of the CPU costs; in [45], the authors study the problem of discovering rare time-series motif (i.e., repeated subsequences) from unbounded streams. To address the rarity issue of the time-series motif in a never-ending stream, the authors develop a “sticky cash” algorithm that adopts a Bloom filter to remember every incoming subsequence and efficiently detects rare motifs in the unbounded real-valued time series. Moreover, to facilitate the computation of the distance-based methods, [48] introduces a fast algorithm for time-series subsequence all-pairs-similarity-search, which shows strong implications and promising results for the task of time-series motif/discord discovery; [52] introduces a robust random cut data structure to produce a sketch or synopsis of time-series data. With that, the authors propose a scalable anomaly detection algorithm by gradually updating the time-series sketch in a continuous data stream.

A key motivation of the above methods is that the distributions of rare categories (minorities) are deviating from the normal distribution (majorities). However, there are some obvious caveats to this idea in practice. It is often the case that the identified examples are not the targets of our interest, which are drawn from noise or combinations of irrelevant features. In [53], the authors present an unsupervised deep framework for detecting insider threat in the online data streams, which outputs a ranked list of anomaly scores of individual user behaviors; in [54], the authors develop a Generative Adversarial Network (GAN) for

unsupervised multivariate anomaly detection. Different from conventional distance-based methods and supervised methods, the proposed framework detects rare temporal patterns by using the GAN trained generator and discriminator to compute the Discrimination and Reconstruction Anomaly Score (DR-Score).

2.4 MINING RARE CATEGORIES IN TEMPORAL GRAPHS

Many real-world systems are intrinsically dynamic and can be represented as temporal graphs, such as social networks, communication networks, gene interaction networks, etc. In past few years, researchers have proposed several rare category exploration models for temporal graphs [7, 29, 38, 39, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64]. Depending on the way of collection data in different application domains, the existing work can be summarized as discrete temporal graphs [65] and continuous temporal graphs [9, 11, 66].

Discrete temporal graph is often referred as time-evolving graphs, where the data \mathcal{D} is presented as a sequence of snapshots $\tilde{\mathcal{G}} = \{S^{(1)}, S^{(2)}, \dots, S^{(T)}\}$, and each snapshot $S^{(t)} = (\mathcal{V}^{(t)}, \mathcal{E}^{(t)})$, $t = 1, 2, \dots, T$. To identify rare examples on $\tilde{\mathcal{G}}$, it is natural to extend the static methods to the dynamic setting. For example, [67] proposes a parameter-free model that can monitor grouped outliers and their changes in a stream of graphs. The algorithm is designed based on Minimum Description Length (MDL), which allows the user to discover the changes in both communities as well as the points in time; [68] develops a fast incremental tensor analysis approach, which can discover both transient and periodic/repeating communities in dynamic graphs; [55] defines a commute-time distance that captures the node relationships changes and allows traditional distance-based methods to be performed on discrete temporal graphs; [60] proposes a discrete-time exponential-family random graph model to identify clusters on time-evolving graphs; [63] proposes a factorization framework that can jointly model the distribution of dynamic connections and attributes and track the evolution of evolving communities. Despite the success, the detection algorithms often suffer from the expensive computational cost, especially when extensive snapshots are given or in the online setting. To address this issue, [58] studies the problem of anomaly detection in the dynamic social networks, where both network structure and node attributes are observed over time. The proposed framework jointly models two processes, i.e., (1) normal modeling component and (2) anomaly detection component, to track the abnormal relationship between nodes' features and link generation in dynamic social networks; [62] introduces a novel community scoring metric named permanence and proposes an incremental algorithm to track the evolution of network communities in the dynamic setting. The theoretical analysis shows the updating procedure of the proposed algorithms leads to permanence maximization in the

dynamic networks.

Continuous temporal graphs are also named fine-grained temporal graphs or temporal interaction graphs, where the temporal graph is presented as a sequence of timestamped edges. Different from the discrete temporal graphs, it is intractable to directly generalize the static rare category exploration approaches to the continuous temporal graphs. For this reason, in [38], the authors, for the first time, propose to represent continuous temporal graphs with a vector representation, which is easy to compute and preserves the context information of the continuous temporal graphs. With the learned continuous temporal graph representation, the authors further develop a fast and memory-efficient detection algorithm to process any incoming nodes and edges and identify anomalies in real-time. Later on, [59] studies the problem of identifying grouped anomalies in the edge streaming setting. In particular, the data is presented as a sequence of streaming edges. The authors propose a streaming algorithm with a theoretical justification that performs graph clustering with only three integers per node and does not keep any edge in memory; [69] proposes a block-structured time series model for detecting communities on time-evolving graphs, which are designed to capture both the link persistence and community persistence over time.

CHAPTER 3: MULTI-VIEW RARE CATEGORY CHARACTERIZATION

3.1 OVERVIEW AND MOTIVATION

In contrast to the large amount of data being generated and used everyday in a variety of areas, it is usually the case that only a small percentage of the data might be of interest to us, which form the minority class. However, without initial labeled examples, the minority class might be very difficult to detect with random sampling due to the imbalance nature of the data, and the limited budget for requesting labels from a labeling oracle. Rare category detection has been proposed to address this problem, so that we are able to identify the very first examples from the minority class, by issuing a small number of label requests to the labeling oracle.

In many real-world applications, the data consists of multiple views, or features from multiple information sources. For example, in synthetic ID detection, we aim to distinguish between the true identities and the fake ones generated for the purpose of committing fraud. Each identity is associated with information from various aspects, such as demographic information, online social behaviors, banking behaviors. Another example is insider threat detection, where the goal is to detect malicious insiders in a large organization, by collecting various types of information regarding each employee’s daily behaviors. To detect the rare categories in these applications, simply concatenating all the features from multiple views may lead to sub-optimal performance in terms of increased number of label requests, as it ignores the relationship among the multiple views. Furthermore, among the multiple information sources, some may generate features irrelevant to the identification of the rare examples, thus deteriorates the performance of rare category detection.

To address this problem, in this chapter, we propose a novel framework named MUVIR for detecting the initial examples from the minority classes in the presence of multi-view data. The key idea is to integrate view-specific posterior probabilities of the example coming from the minority class given features from each view, in order to obtain the estimate of the overall posterior probability given features from all the views. In particular, the view-specific posterior probabilities can be inferred from the scores computed using a variety of existing techniques [31, 33]. Furthermore, MUVIR can be generalized to handle problems where the exact priors of the minority classes are unknown. To the best of our knowledge, this chapter is the first principled effort on rare category detection in the presence of multiple views. Compared with existing techniques, the main advantages of MUVIR can be summarized as follows.

1. Effectively leveraging the relationship among multiple views to improve the performance of rare category detection;
2. Robustness to irrelevant views;
3. Flexibility in terms of the base algorithm used for generating view-specific posterior probabilities.

The rest of this chapter is organized as follows. After a brief review of the related work in Section 3.2, we introduce the proposed framework for multi-view rare category detection in Section 3.3. In Section 3.4, we test our model on both synthetic data sets and real data sets. Finally, we conclude this chapter in Section 3.5.

3.2 RELATED WORK

Multi-view Learning. Traditional machine learning algorithms, such as kernel machines, spectral clustering and support vector machines(SVM), concatenate multiple views in one view to learn the model. However, the concatenation may cause over-fitting in the case of a very small size training data or unsupervised learning. Multi-view learning has been studied extensively in the literature. Co-training [70] is one of the fundamental multi-view algorithm. They have proved that two independent views could be used to learn the pattern based on a few labeled and many unlabeled examples. [71] refined the analysis of co-training and gave a theoretical justification that their algorithm could work on a more relax independence scenario rather than co-training. [72] proposed an independence expansion and proved that it can guarantee the success of co-training. CoMR [73] proposed a multi-view learning algorithm based on a reproducing kernel Hilbert space with a data-dependent co-regularization norm. In [74], the authors develop a kernel machine for learning in multi-view latent variable models, which also allows mixture components to be nonparametric and to learn data in an unsupervised fashion. SMVC [75] proposed a Bayesian framework for modeling multiple clusterings of data by multiple mixture distributions.

Rare Category Detection. Rare category analysis has also been studied for years. Up to now, many methods have been approached to address this problem. In this chapter, we mainly review the following two existing works on rare category detection. The first one is [31], in which algorithm NNDM is proposed standing on two assumptions: (i) *data sets have little knowledge about labels* (ii) *there is no separability or near-separability between majority and minority classes*. Both assumptions exactly meet the setting of the problem we want to figure out. The probability distribution function (pdf) of the majority class tends to

be locally smooth, while the pdf of minority class tends to be a more compact cluster. In general, the algorithm measures the changes of local density around a certain point. NNDM gives a score to each example, and the score is the maximum difference of local density between one item and all of its neighboring points. By querying the examples with the largest score, it is able to hit the region of minority class with the largest probability.

Another work about rare category detection is [33], the authors provided an upgraded algorithm GRADE based on NNDM. In this algorithm, they took the consideration of the manifold structure in minority class. For example, two examples from the same minority class on the manifold may be far away in Euclidean distance. In this case, they generate a global similarity matrix embedded all of the examples from the original feature space. The items of minority class are made to form a more compact cluster for each minority class. Based on global similarity matrix, they measure the changes of local density for each example. The changes of local density has been enlarged, and made the minority classes easier to be discovered. Furthermore, they provided an approximating algorithm to manage rare category detection with less information about priors of minority classes.

3.3 ALGORITHM

In this section, we introduce the proposed framework MUVIR for multi-view rare category detection. Notice that similar as existing techniques designed to address this problem for single-view data, we target the more challenging setting where the support regions of the majority and minority classes overlap with each other, which makes MUVIR widely applicable to a variety of real problems.

3.3.1 Notation

Suppose that we are given a set of unlabeled examples $\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, which come from m distinct classes, i.e. $y_i \in \{1, \dots, m\}$. Without loss of generality, assume that $y_i = 1$ corresponds to the majority class with prior p^1 , and the remaining classes are minority classes with prior p^c . Each example \mathbf{x}_i is described by features from V views, i.e., $\mathbf{x}_i = [(\mathbf{x}_i^1)^T, \dots, (\mathbf{x}_i^V)^T]^T$, where $\mathbf{x}_i^v \in \mathbb{R}^{d_v}$, and d_v is the dimensionality of the v^{th} view. In our method, we repeatedly select examples to be labeled by an oracle, and the goal is to discover at least one example from each minority class by requesting as few labels as possible.

3.3.2 Multi-View Fusion

In this section, for the sake of exposition, we focus on the binary case, i.e., $m = 2$, and the minority class corresponds to $y_i = 2$, although the analysis can be generalized to

multiple minority classes. As reviewed in Section 2, existing techniques for rare category detection with single-view data essentially compute the score for each example according to the change in the local density, and select the examples with the largest scores to be labeled by the oracle. Under mild conditions [31, 33], these scores reflect $P(\mathbf{x}, y = 2)$, thus are in proportion to the conditional probability $P(y = 2|\mathbf{x})$.

For data with multi-view features, running these algorithms [31, 33] on each view will generate scores in proportion to $P(y = 2|\mathbf{x}^v)$, $v = 1, \dots, V$. Next, we establish the relationship between these probabilities and the overall probability $P(y = 2|\mathbf{x})$.

Theorem 3.1. If the features from multiple views have weak dependence given the class label $y_i = 2$ [71], i.e., $P(\mathbf{x}|y = 2) \geq \alpha \prod_{v=1}^V P(\mathbf{x}^v|y = 2)$, $\alpha > 0$, then

$$P(y = 2|\mathbf{x}) \geq C \left(\prod_{v=1}^V P(y = 2|\mathbf{x}^v) \right) \times \left(\frac{\prod_{v=1}^V P(\mathbf{x}^v)}{P(\mathbf{x})} \right) \quad (3.1)$$

where $C = \frac{\alpha}{(p^2)^{V-1}}$ is a constant.

Proof.

$$\begin{aligned} P(y = 2|\mathbf{x}) &= \frac{P(y = 2)P(\mathbf{x}|y = 2)}{P(\mathbf{x})} \\ &\geq \frac{P(y = 2)\alpha \prod_{v=1}^V P(\mathbf{x}^v|y = 2)}{P(\mathbf{x})} \\ &= \alpha \frac{P(y = 2) \prod_{v=1}^V \frac{P(y=2|\mathbf{x}^v)P(\mathbf{x}^v)}{P(y=2)}}{P(\mathbf{x})} \\ &= \alpha \frac{\prod_{v=1}^V P(y = 2|\mathbf{x}^v)P(\mathbf{x}^v)}{P(\mathbf{x})(P(y = 2))^{V-1}} \\ &= \frac{\alpha}{(p^2)^{V-1}} \prod_{v=1}^V P(y = 2|\mathbf{x}^v) \frac{\prod_{v=1}^V P(\mathbf{x}^v)}{P(\mathbf{x})} \end{aligned} \quad (3.2)$$

QED.

As a special case of Theorem 3.1, when the features from multiple view are conditionally independent given the class label, i.e., $\alpha = 1$, we have the following corollary.

Corollary 3.1. If the features from multiple views are conditionally independent given the class label, then Inequality 3.1 becomes equality, and $C = \frac{1}{(p^2)^{V-1}}$.

Proof. Notice that when the features from multiple views are conditionally independent

given the class label, we have

$$P(\mathbf{x}|y = 2) = \prod_{v=1}^V P(\mathbf{x}^v|y = 2) \quad (3.3)$$

The rest of the proof follows by changing the inequality in Eq. 3.2 to equality. QED.

Based on the above analysis, in MUVIR, we propose to assign the score for each example as follows.

$$s(\mathbf{x}) = \prod_{v=1}^V s^v(\mathbf{x}^v) \left(\frac{\prod_{v=1}^V P(\mathbf{x}^v)}{P(\mathbf{x})} \right)^d \quad (3.4)$$

where $s^v(\mathbf{x}^v)$ denotes the score obtained based on the v^{th} view using existing techniques such as NNDM [31] or GRADE [33]; and $d \geq 0$ is a parameter that controls the impact of the term related to the marginal probability of the features. In particular, we would like to discuss two special cases of Eq. 3.4.

Case 1. If the features from multiple views are conditionally independent given the class label, and they are marginally independent, i.e., $P(\mathbf{x}) = \prod_{v=1}^V P(\mathbf{x}^v)$, then Corollary 3.1 indicates that $d = 0$;

Case 2. If the features from multiple views are conditionally independent given the class label, then Corollary 3.1 indicates that $d = 1$.

In Section 4, we study the impact of the parameter d on the performance of MUVIR, and show that in general, $d \in (0, 1.5]$ will lead to reasonable performance.

Notice that the proposed score in Eq. 3.4 is robust to irrelevant views in the data, i.e., the views where the examples from the majority and minority classes cannot be effectively distinguished. This is mainly due to the first part $\prod_{v=1}^V s^v(\mathbf{x}^v)$ on the right hand side of Eq. 3.4. For example, assume that view 1 is irrelevant such that the distribution of the majority class ($P(\mathbf{x}|y = 1)$) is the same as the minority class ($P(\mathbf{x}|y = 2)$). In this case, the view-specific score $s^1(\mathbf{x}^1)$, which reflects the conditional probability $P(y = 2|\mathbf{x})$, would be the same for all the examples. Therefore, when integrated with the scores from the other relevant views, view 1 will not impact the *relative* score of all the examples, thus it will not degrade the performance of the proposed framework.

3.3.3 MUVIR Algorithm

The proposed MUVIR algorithm is described in Algorithm 3.1. It takes as input the multi-view data set, the priors of all the classes (p^1, p^2, \dots, p^m), as well as some parameters, and outputs the set of selected examples together with their labels.

Algorithm 3.1: MUVIR Algorithm

Require: Unlabeled data set \mathcal{S} with features from V views, $p^1, \dots, p^m, d, \epsilon$.

Ensure: The set I of selected examples and the set L of their labels.

```
1: for  $v=1 : V$  do
2:   Compute  $s^v(\mathbf{x}_i^v)$  for all the examples using existing techniques for rare cate-
   gory detection, such as GRADE [33];
3:   Estimate  $P(\mathbf{x}_i^v)$  using kernel density estimation;
4: end for
5: Estimate  $P(\mathbf{x}_i)$  using kernel density estimation on all the features combined;
6: for  $c=2 : m$  do
7:   If class  $c$  has been discovered, continue;
8:   for  $t = 2 : n$  do
9:     for  $v = 1 : V$  do
10:      For each  $\mathbf{x}_i$  that has been labeled by the oracle, if  $\forall i, j = 1, \dots, n, i \neq j$ ,
      and  $\|\mathbf{x}_i^v, \mathbf{x}_j^v\|_2 \leq \epsilon$ , then  $s^v(\mathbf{x}_j^v) = -\infty$ ;
11:      Update the view-specific score  $s^v(\mathbf{x}_i^v)$  using existing techniques such as
      GRADE [33];
12:    end for
13:    Compute the overall score for each example  $s(\mathbf{x}_i)$  based on Eq. 3.4;
14:    Query the label of the example with the maximum  $s(\mathbf{x}_i)$ 
15:    If the label of  $\mathbf{x}_i$  is from class  $c$ , break; otherwise, mark the class of  $\mathbf{x}_i$  as
      labeled.
16:   end for
17: end for
```

MUVIR works as follows. In Step 2, we compute the view-specific score for each example, which can be done using any existing techniques for rare category detection. In Step 3, we estimate the view-specific density using kernel density estimation; whereas in Step 5, we estimate the overall density by pooling the features from all the views together. Finally, Steps 6 to 16 aim to select candidates according to $P(y = c|\mathbf{x})$. To be specific, in Step 7, we skip class c if examples from this class have already been identified in the previous iterations. Step 10 implements the feedback loop by excluding any examples close to the labeled ones from being selected in future iterations. Notice that the threshold ϵ depends on the algorithm used to obtain the view-specific scores. For example, it is set to the smallest k -nearest neighbor distance in NNDM [31], and the largest k -nearest neighbor global similarity in GRADE [33]. Step 11 updates the view-specific score for each example with enlarged neighborhood for computing the change in local density [31, 33]. In Step 13, we compute the overall score based on Eq. 3.4, and select the example with the maximum overall score to be labeled by the oracle in Step 14. In Step 15, if the labeled example is from the target class in this iteration, we proceed to the next class; otherwise, we mark the class of this examples as

labeled.

3.3.4 MUVIR with Less Information (MUVIR-LI)

In many real applications, it may be difficult to obtain the priors of all the minority classes. Therefore, In this subsection, we introduce MUVIR-LI, a modified version of Algorithm 3.1, which replaces the requirement for the exact priors with an upper bound p for all minority classes. Compared with MUVIR, MUVIR-LI is more suitable in real world applications. MUVIR-LI is described in Algorithm 3.2. It works as follows. Step 2 calculates the specific score s^v for each example. The only difference from MUVIR is that here we use upper bound p to calculate s^v , which is a less accurate measurement of changing local density than in MUVIR. The same as MUVIR, we estimate the view specific density and the overall density by applying kernel density estimation in Step 3 and Step 5. The while loop from Step 6 to Step 16 is the query processing. We calculate the overall score for each example and select the examples with the largest overall score to be labeled by oracle. We end the loop until all the classes has been discovered.

3.4 EXPERIMENTAL EVALUATION

In this section, we will present the results of our algorithm on both synthetic data sets and real data sets in multiple special scenarios, such as data sets with different number of irrelevant features, data sets with multiple classes and data sets with very rare categories, such as class proportion of 0.02%.

3.4.1 Synthetic Data Sets

Binary Class Data Sets. For binary classes, we perform experiment on 3600 synthetic data sets, and each scenario has independent 100 data sets. We consider the following three special conditions: *(i)* different number of irrelevant features, i.e. from 0 to 3 irrelevant features; *(ii)* different priors for minority class, i.e. 0.5%, 1%, 2%; *(iii)* different levels of correlation between majority class and minority class, ie. minority class stays in the center of majority class, minority class stays around the center of majority class, minority class stays at the boundary of majority class. Besides, as the distribution of majority class tends to be more scattered and the distribution of minority class is more compact, we set each data set with 5000 examples and $\sigma_{majority} : \sigma_{minority} = 40 : 1$.

Algorithm 3.2: MUVIR-LI Algorithm

Require:

Unlabeled data set \mathcal{S} with features from V views, p , d , ϵ .

Ensure:

The set I of selected examples and the set L of their labels.

```
1: for  $v = 1 : V$  do
2:   Compute  $s^v(\mathbf{x}_i^v)$  for all the examples using existing techniques for rare cate-
   gory detection, such as GRADE-LI [33];
3:   Estimate  $P(\mathbf{x}_i^v)$  using kernel density estimation;
4: end for;
5: Estimate  $P(\mathbf{x}_i)$  using kernel density estimation;
6: while not all the classes have been discovered do
7:   for  $t = 2 : n$  do
8:     for  $v = 1 : V$  do
9:       For each  $\mathbf{x}_i$  that has been labeled by the oracle, if  $\|\mathbf{x}_i^v, \mathbf{x}_j^v\|_2 \leq \epsilon$ , then
        $s^v(\mathbf{x}_j^v) = -\infty$ ;
10:      Update the view-specific score  $s^v(\mathbf{x}_i^v)$  using existing techniques such as
       GRADE-LI [33];
11:    end for;
12:    Compute the overall score for each example  $s(\mathbf{x}_i)$  based on Eq. 3.4;
13:    Query the label of the example with the maximum  $s(\mathbf{x}_i)$ 
14:    Mark the class that  $\mathbf{x}$  belongs to as discovered.
15:  end for;
16: end while
```

In the experiment, we compare MUVIR with GRADE [33] and random sampling. Figure 3.1 shows the results when the prior of minority class is 0.5%. Using random sampling, we need to label 200 examples on average to identify the minority class. In most cases, other approaches outperform random sampling. However, the learning model generated by GRADE algorithm performs worse with the increasing of irrelevant features. In contrast, MUVIR is more efficient and stable rather GRADE. The experiment with minority proportions of 1% and 2% are represented in Figure 3.2 and Figure 3.3. In these two experiment, MUVIR outperforms GRADE and random sampling in each condition with any setting of d . Comparing these three figures, we have the following observations for binary class data sets: (i) MUVIR is more reliable especially when dealing with data sets containing irrelevant features. (ii) In the case of data sets with no irrelevant features, the performance of MUVIR with different values of d are roughly the same. (iii) In the case of data sets with irrelevant features, MUVIR with $d = 1$ outperforms other methods.

Multi-classes Data Sets with Imprecise Prior. For multi-class data sets, we compare the performances among different approaches. In particular, GRADE-LI [33] and

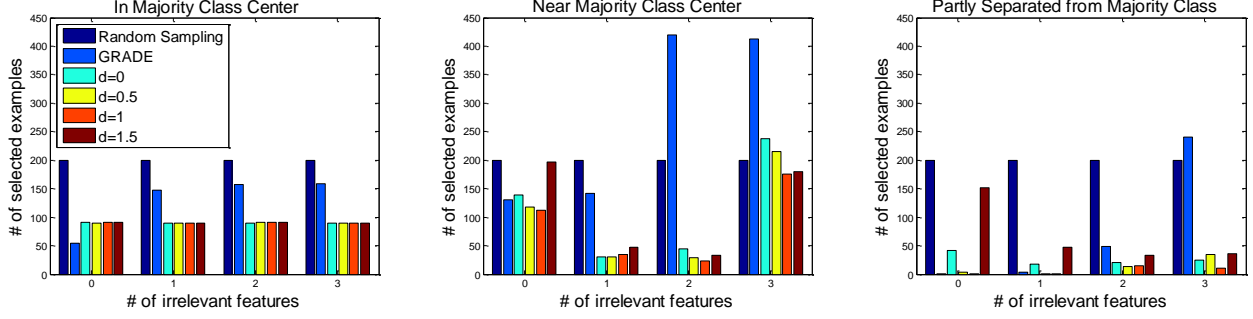


Figure 3.1: Prior of minority class is 0.5%

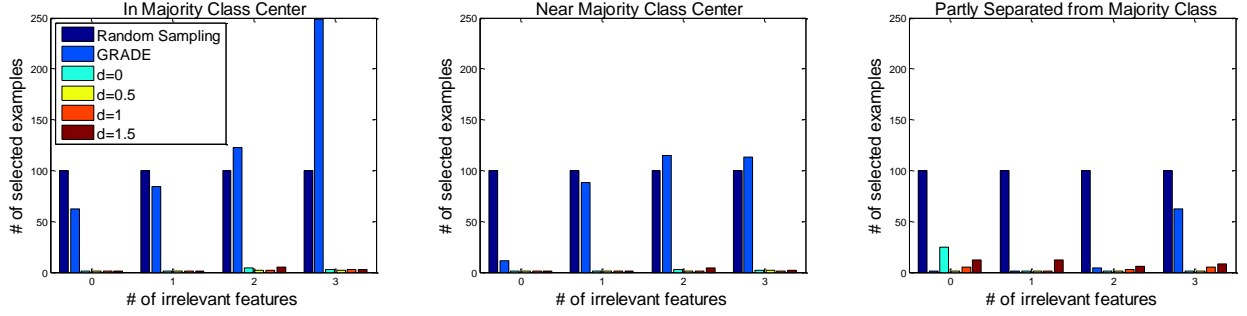


Figure 3.2: Prior of minority class is 1%

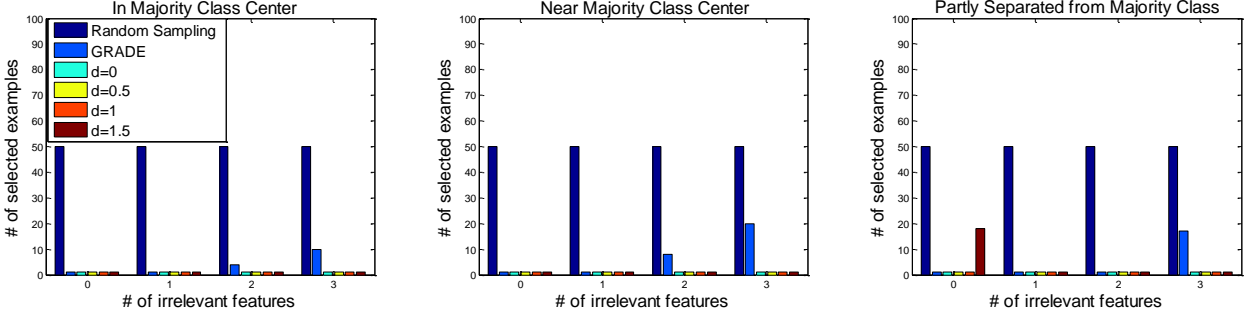


Figure 3.3: Prior of minority class is 2%

MUVIR-LI are only provided with an upper bound p on the proportion of all the minority classes. The multi-class data sets consisting of 9000 examples correspond to majority class, and the other 1000 examples correspond to 4 minority classes. The proportions of minority classes are 4%, 3%, 2%, 1%. Similar to previous experiments, we will discuss the scenario data sets contain different number of irrelevant features. Each value we represented in the figure is the median value of results from 100 same scenario data sets. From Figure 3.4, we can have the following conclusions: (i) MUVIR outperforms all other algorithms in multi-class data sets; (ii) GRADE only performs good when data sets have 1 or 0 irrelevant feature; (iii) MUVIR-LI is more reliable than GRADE-LI in all scenarios. The reason that our models have better performance is that both MUVIR and MUVIR-LI are capable to exploit the

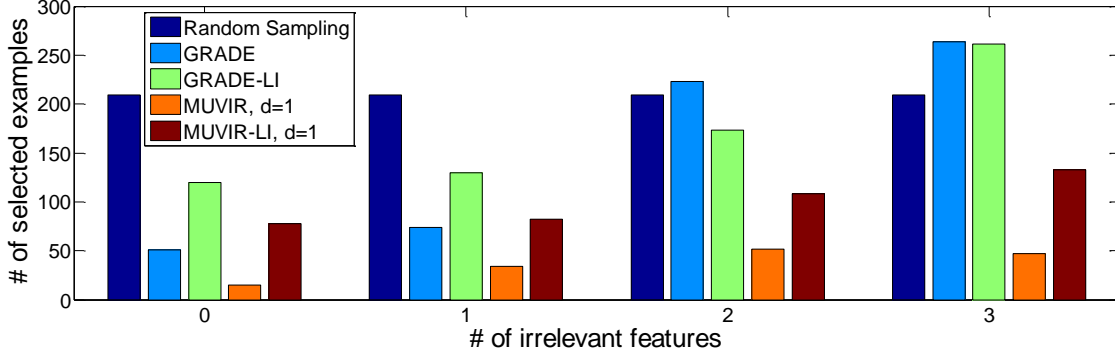


Figure 3.4: Multi-class data sets

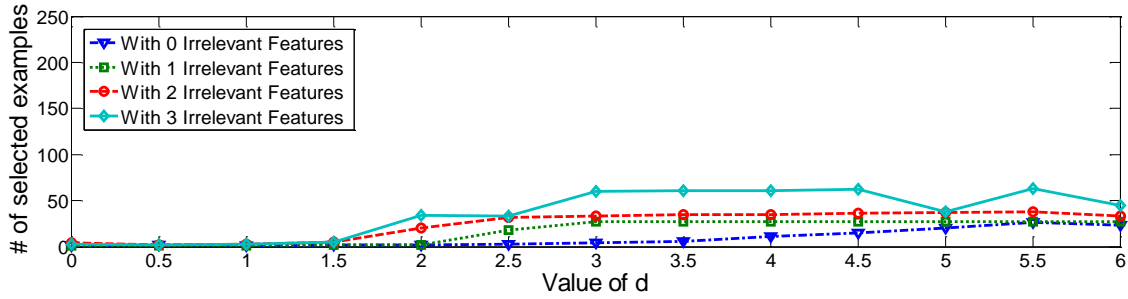


Figure 3.5: Learning curves with different degree d

relationship among multiple views and extract useful information to make predictions.

Parameter Analysis. From previous experiments, we found different parameter settings may result in different outcomes. In this experiment, we will focus on analyzing the impact from degree d and upper bound prior p . To measure the impact of these parameters, we generate 400 data sets with minority class proportion 1%. The number of irrelevant features varies from 0 to 3, and each case has 100 data sets. In Figure 3.5, the X axis represents different values of degree d , and Y axis represents the number of selected examples on average. From Figure 3.5, we can see that MUVIR performs better when $d \in (0, 1.5]$. In the following experiments, we will focus on studying the performance of our algorithm with d in this certain area.

With the same data sets, we studied the learning curves of labeling requests by applying MUVIR-LI with different upper bound p . In Figure 3.6, the X axis represents different values of upper bound proportion and Y axis represents the number of labeling requests. The red line represents the average number of labeling requests by using random sampling. When data sets without irrelevant features, MUVIR-LI works well even with upper bound p changing from 1% to 12%. When data sets with irrelevant features, MUVIR-LI can still outperforms random sampling with upper bound p changing from 1% to 8.5%. However,

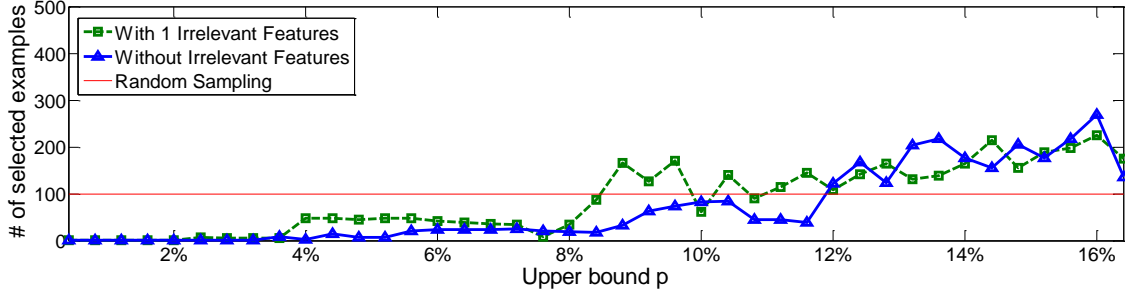


Figure 3.6: Learning curves with different prior upper bound

Views	Features
relevant view 1	education, education years, work class
relevant view 2	age, hours per week, occupation
relevant view 3	marital status, relationship, sex
relevant view 4	race, native country
irrelevant view 1	final weight
irrelevant view 2	capital loss, capital gain

Table 3.1: Relevant and irrelevant views in Adult Data set.

when the upper bound exceeds a certain level, the algorithm tends to be random sampling.

3.4.2 Real Data Sets

In this subsection, we will demonstrate our algorithm on two real data sets Statlog and Adult. Noted that, before we run our algorithms, we have preprocessed both data sets in order to keep each feature component has mean 0 and standard deviation 1. In the following experiments, we will compare MUVIR and MUVIR-LI with the following algorithms: GRADE, GRADE-LI and random sampling.

Adult data set contains 48842 instances and 14 features of each example. It is a binary classes data sets. Considering the original prior of minority class in data sets is around 24.93%. To better test the performance of our model, we keep majority class the same and down sample the minority class to 500 examples. In this way, we generate 24 data sets with minority prior of 1.3%. The details about relevant and irrelevant views are represented in Table 3.1. Figure 3.7 shows the comparison results on real data by applying 5 different approaches. In this experiment, we have not included MUVIR-LI, it is because MUVIR-LI is mainly developed for multi-class cases and Adult is a binary class data sets. By using

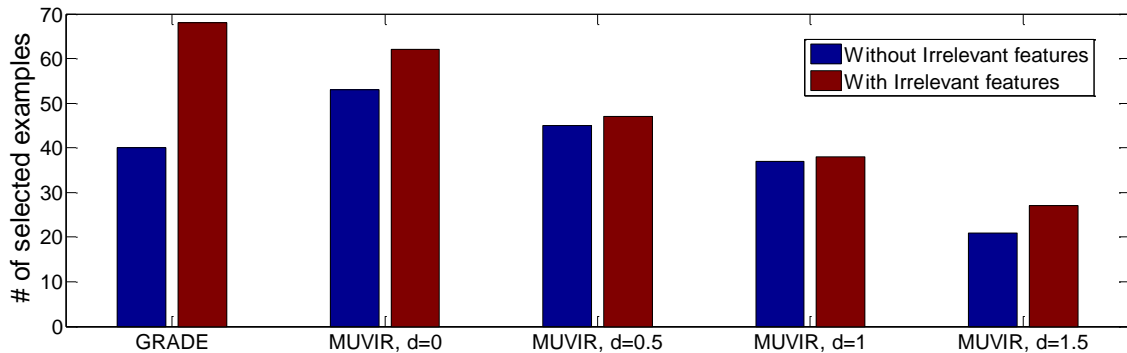


Figure 3.7: Adult

random sampling, the average number of selected examples is 76. With irrelevant views, GRADE needs 69 requests, MUVIR with $d = 0$ needs 60 requests, MUVIR with $d \neq 0$ needs around 30 to 40 requests. The results totally meet our intuition that: when dealing data sets with irrelevant views, MUVIR with $d \neq 0$ outperforms MUVIR with $d = 0$, and MUVIR with $d = 0$ outperforms GRADE. However, when dealing with data sets without irrelevant views, GRADE needs less labeling requests than MUVIR with $d = 0$, but more labeling requests than MUVIR with d around 1.

Different from Adult, Statlog contains 58000 examples and 7 classes. Among 7 classes, there are 6 minority classes, with priors varying from 0.02% to 15%. In this experiment, we compare the following 4 methods: GRADE, GRADE-LI with upper bound $p = \max_{c=2}^m p^c$, MUVIR with $d = 1$, MUVIR-LI with $d = 1$ and $p = \max_{c=2}^m p^c$. From Figure 3.8, we can see that MUVIR outperforms all other algorithms. With the same upper bound prior, GRADE-LI needs 272 labeling requests while MUVIR-LI only needs 168 labeling requests to discover all the classes. If we apply random sampling, it may needs around 5000 labeling request to only identify the smallest minority class. Compared with Adult, we have better results on Statlog. It is because the distribution of majority class and minority classes are not meshed together as in Adult. Thus, to identify the minority classes in Statlog is a much easier case.

3.5 SUMMARY

In this chapter, we have proposed a multi-view based method for rare category detection named MUVIR. Based on MUVIR, we also provided a modified version MUVIR-LI for dealing with real applications with less prior information. Different from existing methods, our methods exploit the relationship among multiple views and measure the probability belong-

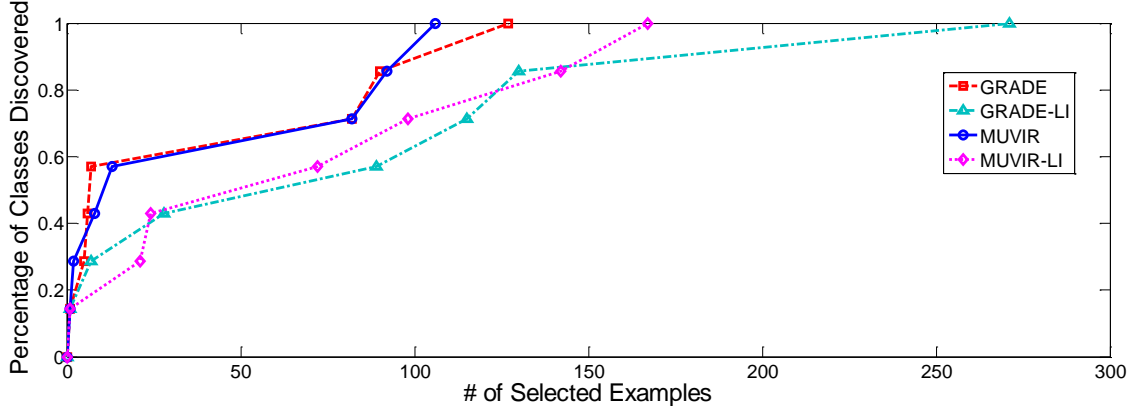


Figure 3.8: Statlog

ing to target class for all examples. Our algorithm works well with multiple special cases: data sets with irrelevant features, data sets with multiple minority class and various correlation levels between minority class and majority class. The effectiveness of our proposed methods is guaranteed by theoretical justification and extensive experiments results on both synthetic and real data sets, especially in the presence of irrelevant views.

CHAPTER 4: RARE CATEGORY CHARACTERIZATION ON TIME-EVOLVING GRAPHS

4.1 OVERVIEW AND MOTIVATION

Compared with the tremendous and rapidly changing data, the examples of interest to us only hold a very small portion. For instance, in financial synthetic identity detection [24], only a tiny proportion of identities are fraudulent, generated by mixing the identifying information from multiple sources. Such identities are created with the sole purpose of committing financial fraud. Another example is insider threat detection [76], where only a small population amongst a big organization are malicious insiders involved in treacherous behaviors, such as sabotage, espionage, etc. The small percentage of data of interest to us is called the minority class or rare category, since such examples are often self-similar. Due to the rarity of the minority classes and the limited budget on querying the labeling oracle who can provide the true label of any example at a fixed cost, it is difficult to identify examples from such classes via random sampling. To efficiently deal with this problem, rare category detection has been proposed to identify the very first example from the minority class, by requesting only a small number of labels from the oracle [26].

Most, if not all, of existing rare category detection techniques are designed for static data. However, in many real-world applications, the data is not static but evolves with time, and so are the minority classes. Examples of such scenarios are listed as follows.

1. In financial synthetic identity detection, within the transaction network, each identity could correspond to one specific node, and each transaction activity could correspond to one edge. Since each identity may keep updating his or her information, such as daily transactions and real-time online banking activities, the data is evolving over time. Our goal is to identify the identities and transactions, which have unusual characteristics and significantly differ from the majorities in the networks.
2. In insider threat detection, the insiders intentionally change their behavior patterns over time to avoid being caught. In other words, the insiders may not be abnormal all the time when compared with normal employees. Thus, how to distinguish insiders and normal employees from evolving data is a challenge.
3. In event detection in social networks, the snapshots of social networks are evolving every single second with updated vertex sets and updated edge sets, which means the event related vertex sets may shrink, expand or shift within the time-evolving social

networks. Hence, how to model, capture and track the changing target events over evolving social networks would be the main task.

Straight-forward applications of existing RCD techniques in the preceding scenarios would be very time-consuming by constructing the models from scratches at each time step. Additionally, it is critical to allocate queries among different time steps from labeling oracle, which may help detect the initial rare examples as early as possible to avoid further damage.

Addressing this issue, in this chapter, for the first time, we study the problem of incremental RCD. Specifically, we first propose two incremental algorithms, i.e., SIRD and BIRD, to detect the initial examples from the minority classes under different dynamic settings. The key idea is to efficiently update our detection model by local changes instead of reconstructing it from scratches based on the updated data at a new time step, so as to reduce the time cost of redundant and repeating computations. Furthermore, we relax the requirement of the exact priors with a soft upper bound for all the minority classes to provide a modified version - BIRD-LI. Finally, we study a unique problem of query distribution under the dynamic settings, which distributes allocated labeling budget among different time steps, and propose five query distribution strategies. This chapter is extended from our previous work [29] in terms of the detailed algorithm, theoretical justification and the comprehensive experiments on real time-evolving graph data sets.

The rest of the chapter is organized as follows. In Section 4.2, we briefly review the related work on both RCD and time-evolving graph mining. In Section 4.3, we study incremental RCD and propose three algorithms, i.e., SIRD, BIRD and BIRD-LI, to address different dynamic settings. Then, in Section 4.4, we introduce the unique problem of query distribution under the dynamic settings, and propose five strategies for allocating the labeling budget among different time steps. In Section 4.5, we demonstrate our models on both synthetic and real data sets. Finally, we conclude this chapter in Section 4.6.

4.2 RELATED WORK

4.2.1 Rare Category Analysis

Rare category detection refers to the problem of identifying the initial examples from under-represented minority classes in an imbalanced data set. Lots of techniques have been developed for solving the problem of RCD in the past decade. [26] proposed a mixture model-based algorithm, which is the first attempt in this area. In [31, 33], the authors developed an innovative method to detect rare categories via unsupervised local-density-

differential sampling strategy. [77] presented an active learning scheme via exploiting the cluster structure in data sets. In [78], the authors introduced a novel problem called rare category characterization, which not only detects but also characterizes the rare categories, and proposed an optimization framework to explore the compactness of rare categories. More recently, in [79], two prior-free methods were proposed in order to address the rare category detection problem without any prior knowledge. In [2], the authors proposed a framework named MUVIR, which could leverage existing rare category detection models on each single view and estimate the overall probability of each example belonging to the minority classes. However, all of the preceding works focus on the static data sets, and few works have been proposed to address the problem of rare category detection under dynamic settings.

4.2.2 Outlier Detection on Streaming Data

With the improvement of hardware technology on data collection, many applications require efficient mechanisms to process the outlier detection on streaming data [22]. Tons of algorithms have been proposed in the past decade. [80] presented an online discounting learning algorithm to incrementally update a probabilistic mixture model and capture outliers in data streams. In [81], the authors proposed online clustering methods, which maintained a dynamic clustering model to identify outliers under dynamic settings. Instead of only updating parameters of the prediction model, Dynamic Bayesian Network (DBN) [82], a modifiable model, was proposed to detect anomalies from environmental sensor data. Different from regular data streams, distributed data streams are collected from distributed sensors over time. [83, 84] studied the problem of outlier detection on multiple types of distributed data streams, such as air temperature sensor network data, water pollution sensor network data and wind sensor network data. Different from outlier detection, rare category detection assumes that the anomalies belong to multiple distinct classes, in the sense that the within-class similarities are much larger than the between-class similarities. In this chapter, we aim to discover these rare categories over a series of time-evolving graphs.

4.2.3 Graph Based Anomaly Detection

In the literature, there are abundant works focusing on anomaly detection in static graphs. Basically, all of the existing works study two types of static graphs: plain static graphs and attributed static graphs. Plain graph assumes the only information we have is the structure of graph. This category of anomaly detection methods aims to exploit the structure of graphs and mine the unrepresentative pattern of anomalies, e.g., global graph structure methods [85,

86]; local graph structure methods [87, 88, 89]. Attributed graph assumes both the structure and the coherence of attributes are given. [90, 91] proposed node outlier ranking methods on static attributed graphs. Yagada [92] characterized anomalies by discrediting the numerical attributes into “outlier score”. In [55], the authors proposed a fast algorithm which could detect the node relationships for localizing anomalous changes in time-evolving graphs.

More recently, an increasing number of research has been conducted under dynamic graph settings. For examples, in [93], the authors analyzed the properties of the time evolution of real graphs and proposed a “forest fire” graph-generative model; [94] studied the problem of community evolution and developed a novel method to measure the movement of individuals among communities; in [95], the authors focused on the difficulties of conversation dynamics and proposed a simple mathematical model in order to generate basic conversation structures; in [96, 97], the authors proposed several graph similarity measurements to detect the discontinuity in dynamic social networks. Besides, to reduce the time complexity, in [98], the authors proposed a fast proximity tracking method for dynamic graphs; in [99], the authors used tensor decomposition techniques to efficiently obtain the “scores” for anomalies on dynamic graphs; in [100], the authors proposed a new graph-pattern matching algorithm, which can avoid cubic-time computation; [101] raised a divide-and-conquer framework, which could find the k-nearest-neighbors efficiently on high volume of time-evolving graphs. BIRD approach [29] provided a fast updating method for the challenging problem of RCD on time-evolving graphs. In this chapter, we propose several fast-updating RCD methods which could incrementally update the models based on local changes on time-evolving graphs. This chapter extends our previous work [29] substantially by providing the detailed algorithm, theoretical justification and the comprehensive empirical evaluations on real-world time-evolving graph data sets, which are not presented in the previous version.

4.3 ALGORITHM

In this section, we introduce the proposed framework of incremental rare category detection. Our methods exploit the time-evolving nature of dynamic graphs and update the RCD model incrementally based on the local updates from time to time. To the best of our knowledge, existing rare category detection methods are all designed for static data sets, while we target a more challenging setting, in which the data is presented as time-evolving graphs. Notice that we allow the support regions of the majority and minority classes to overlap with each other in the feature space, which makes our algorithm widely applicable to a variety of real-world problems.

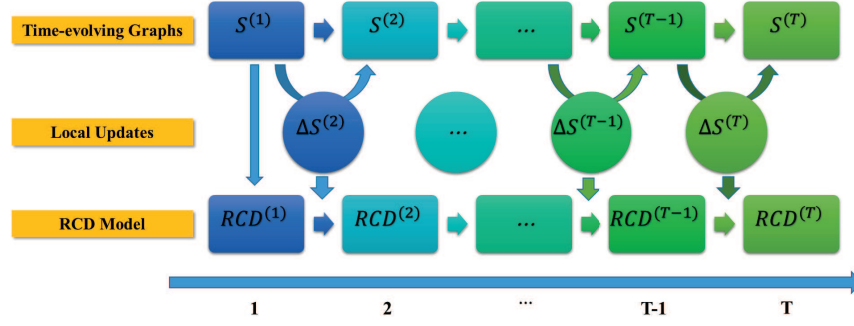


Figure 4.1: Incremental Rare Category Detection

4.3.1 Notation

Suppose we are given a series of time-evolving graphs $\tilde{\mathcal{G}} = \{S^{(1)}, \dots, S^{(T)}\}$, which are shown in Figure 4.1. For any time step $t = 1, \dots, T$, the vertices in $S^{(t)}$ are identical and only edges change over time. We assume $y_i^{(t)} = 1$ corresponds to the majority class with prior $p_1^{(t)}$, and the remaining classes are the minority classes with priors $p_c^{(t)}$ at time step t . We use $\Delta S^{(t)}$ to denote the new edges and updated weights that appear at time step t . Specifically, we have $\Delta S^{(t)} = S^{(t)} - S^{(t-1)}$.

In the following part of this chapter, we use the convention in Matlab to represent matrix elements, e.g., $S^{(t)}(i, j)$ is the element at i^{th} row and the j^{th} column of matrix $S^{(t)}$, and $S^{(t)}(:, j)$ is the j^{th} column of matrix $S^{(t)}$, etc. The main symbols we used in this chapter are listed in Table 4.1.

4.3.2 Static Rare Category Detection

In static RCD, we repeatedly select examples to be labeled by the oracle until all the minority classes in a static data set are discovered. One approach for static RCD is to make use of the manifold structure for identifying rare category examples. In [33], authors developed a graph-based RCD method named GRADE. In GRADE algorithm, they first construct a pair-wise similarity matrix W' and its corresponding diagonal matrix D , whose elements are the row sums of W' . Then, they calculate the normalized matrix by $W = D^{-1/2}W'D^{-1/2}$. Based on the normalized pair-wise similarity matrix W , they construct a global similarity matrix A as follows.

$$A = (I_{n \times n} - \alpha W)^{-1} \quad (4.1)$$

where α is a small enough positive discounting constant in the range of $(0, 1)$. By construct-

Symbol	Description
n	number of nodes
$m^{(t)}$	number of updated edges
x_i	i^{th} nodes in data set
t	time step
C	number of classes
$p_c^{(t)}$	proportion of classes c
α	constraint parameter
I	identity matrix
$S^{(t)}$	$n \times n$ original aggregated adjacency matrix at time t
$\Delta S^{(t)}$	$n \times n$ updating matrix for $S^{(t-1)}$
$M^{(t)}$	normalized $n \times n$ aggregated adjacency matrix at time t
$\Delta M^{(t)}$	$n \times n$ updating matrix for $M^{(t-1)}$
$NN^{(t)}$	$n \times n$ neighbor information matrix at time step t
$A^{(t)}$	$n \times n$ global similarity matrix at time step t

Table 4.1: Symbols.

ing the global similarity matrix, the changes of local density would become sharper near the boundary of the minority classes. Based on this intuition, GRADE could identify minority classes with much fewer queries than random sampling. However, the time complexity of calculating the global similarity matrix and finding each example’s $(K)^{th}$ nearest neighbor is $O(n^3 + K \cdot n^2)$, which is not efficient enough for time-evolving RCD applications.

4.3.3 Dynamic Rare Category Detection

In this subsection, we introduce two fast-updating incremental RCD algorithms (SIRD and BIRD) to deal with the RCD problem on time-evolving graphs. Both methods greatly reduce the computation cost for updating the global similarity matrix and finding each node’s K^{th} nearest neighbor. Similar to static rare category detection, we target the challenging case where the minority classes are not separable from the majority classes.

Single Update. We first consider the simplest case: only one self-loop edge (a, a) changes at time step t . In other words, there is only one non-zero element (a, a) in $\Delta S^{(t)}$.

Similar to [33], we use $M^{(t)}$ to denote the normalized aggregated adjacency matrix, which is defined as follows.

$$M^{(t)} = (D^{(t)})^{-1/2} S^{(t)} (D^{(t)})^{-1/2} \quad (4.2)$$

Besides, let $\Delta M^{(t)}$ denote the updating matrix for $M^{(t)}$, such as $\Delta M^{(t)} = M^{(t)} - M^{(t-1)}$. Clearly, there is also only one non-zero element existing in $\Delta M^{(t)}$. Hence, $\Delta M^{(t)}$ could be easily decomposed into the product of two column vectors uv^T , where u and v are two column vectors with only one non-zero element. To address this problem, we first introduce Theorem 4.1 to update the global similarity matrix $A^{(t)}$ more efficiently.

Theorem 4.1. The global similarity matrix $A^{(t)}$ at time step t can be exactly updated from global similarity matrix $A^{(t-1)}$ at the previous time step $t - 1$ by the following equation:

$$A^{(t)} = A^{(t-1)} + \alpha \frac{A^{(t-1)} uv^T A^{(t-1)}}{I + v^T A^{(t-1)} u} \quad (4.3)$$

where u and v^T are the two vectors decomposed from updating matrix $\Delta M^{(t)}$

Proof. Suppose there is only one edge updated at time step t , and we have $\Delta M^{(t)} = uv^T$. Thus, Eq. 4.1 could be rewritten as follows.

$$\begin{aligned} A^{(t)} &= (I - \alpha M^{(t)})^{-1} \\ &= (I - \alpha M^{(t-1)} - \alpha \Delta M^{(t)})^{-1} \\ &= (I - \alpha M^{(t-1)} - \alpha uv^T)^{-1} \end{aligned} \quad (4.4)$$

By applying the Sherman-Morrison Lemma [102], we have

$$A^{(t)} = A^{(t-1)} + \alpha \frac{A^{(t-1)} uv^T A^{(t-1)}}{I + v^T A^{(t-1)} u} \quad (4.5)$$

Hence, the global similarity matrix $A^{(t)}$ in our Algorithm 4.1 could be exactly updated at each time step. QED.

In Theorem 4.1, we can see column vectors u and v are essential for updating the global similarity matrix $A^{(t)}$. To reduce the computational complexity, in Algorithm 4.1, we use an approximate method to calculate the two column vectors u and v . The details are described as follows. We first assume that the updated edges at time step t have little impact on the row sum of adjacency matrix $S^{(t)}$ when the number of updated edges is extremely smaller

than the total number of edges. Thus, we have $D^{(t)} \cong D^{(t-1)}$. To normalize aggregated adjacency matrix of $S^{(t)}$ and $S^{(t-1)}$, we have

$$M^{(t)} = (D^{(t)})^{-1/2} S^{(t)} (D^{(t)})^{-1/2} \quad (4.6)$$

$$M^{(t-1)} = (D^{(t)})^{-1/2} S^{(t-1)} (D^{(t-1)})^{-1/2} \quad (4.7)$$

By Eq. 4.6 – Eq. 4.7, we have

$$\Delta M^{(t)} = (D^{(t-1)})^{-1/2} \Delta S^{(t)} (D^{(t-1)})^{-1/2} \quad (4.8)$$

As $\Delta M^{(t)} = uv^T$, we could easily assign $u = D(:, a)^{-1/2}$ and $v = \Delta S^{(t)}(a, b) D(:, b)^{-1/2}$.

Besides, as the time complexity of constructing a new neighbor information matrix $NN^{(t)}$ is $O(K^{(t)} \cdot n^2)$, we introduce Theorem 4.2 to efficiently update $NN^{(t)}$.

Theorem 4.2. Suppose there is only one self loop edge (a, a) being updated at time step t . If it satisfies the condition that $\frac{\alpha}{I+v^T A^{(t-1)} u} \leq \frac{\delta_i^{(t-1)}}{A_{i,a}^{(t-1)} \phi_a}$, the first $K^{(t)}$ elements in $NN^{(t)}(i, :)$ are the same as $NN^{(t-1)}(i, :)$.

Proof. Based on Theorem 4.1, we have

$$\begin{aligned} A^{(t)} &= A^{(t-1)} + \alpha \frac{A^{(t-1)} u v^T A^{(t-1)}}{I + v^T A^{(t-1)} u} \\ &= A^{(t-1)} + \alpha \frac{A^{(t-1)} \Delta M^{(t)} A^{(t-1)}}{I + v^T A^{(t-1)} u} \end{aligned} \quad (4.9)$$

Since u and v are column vectors that contain only one non-zero element, then $I + v^T A^{(t-1)} u$ is a constant value, which means it is just a scalar and will not change the order of elements in $NN^{(t)}$.

From Eq. 4.9 we also have the updating rule for each element (i, j) in $A^{(t)}$

$$A_{i,j}^{(t)} = A_{i,j}^{(t-1)} + \beta A_{i,a}^{(t-1)} A_{a,j}^{(t-1)} \quad (4.10)$$

where $\beta = \frac{\alpha}{I+v^T A^{(t-1)} u}$ is also a constant.

Let $\delta_i^{(t-1)} = \min_{j=1}^{K^{(t)}} \{NN^{(t-1)}(i, j) - NN^{(t-1)}(i, j+1)\}$ denote the smallest adjacent difference among the first $K^{(t)}$ elements in the i^{th} row of $NN^{(t-1)}$, and $\phi_a = NN^{(t-1)}(a, 1)$ denote the largest element in row a . Intuitively, as long as the largest value of $\beta A_{i,a}^{(t-1)} A_{a,j}^{(t-1)}$ is smaller than the smallest adjacent gap between any of the first $K^{(t)}$ nodes in the i^{th} row of $NN^{(t)}$, we can claim that the order of these sorted $K^{(t)}$ nodes will not change. Therefore, based

on Eq. 4.10, if the condition satisfies $\frac{\alpha}{I+v^T A^{(t-1)} u} \leq \frac{\delta_i^{(t-1)}}{A_{i,a}^{(t-1)} \phi_a}$, we can claim that the first $K^{(t)}$ elements in $NN^{(t)}(i, :)$ will not change. QED.

Algorithm 4.1: SIRD Algorithm

Require: $M^{(1)}, A^{(1)}, \Delta S^{(2)}, \dots, \Delta S^{(T)}, p_c^{(t)}, \alpha$.

Ensure: The set I of labeled nodes

- 1: Construct the $n \times n$ diagonal matrix D , where $D_{ii} = \sum_{j=1}^n S^{(1)}$, $i = 1, \dots, n$.
 - 2: Sort row i of $A^{(1)}$ decreasingly and save into $NN^{(1)}(i, :)$, where $i = 1, \dots, n$.
 - 3: **for** $t=2:T$ **do**
 - 4: Let $K^{(t)} = \max_{c=2}^C n \times p_c^{(t)}$.
 - 5: Let column vector $u = D(:, a)^{-1/2}$, and column vector $v = \Delta S^{(t)}(a, a) D(:, a)^{-1/2}$, where $\Delta S^{(t)}(a, a)$ is the non-zero element in $\Delta S^{(t)}$.
 - 6: Update the global similarity matrix $A^{(t)} = A^{(t-1)} + \alpha \frac{A^{(t-1)} u v^T A^{(t-1)}}{I+v^T A^{(t-1)} u}$.
 - 7: **for** $i=1:n$ **do**
 - 8: Based on Theorem 4.2, identify whether the first $K^{(t)}$ elements of $NN^{(t)}(i, :)$ are changed. If true, update the first $K^{(t)}$ elements in $NN^{(t)}(i, :)$; otherwise, let $NN^{(t)}(i, :) = NN^{(t-1)}(i, :)$.
 - 9: **end for**
 - 10: **end for**
 - 11: **for** $c = 2:C$ **do**
 - 12: Let $k_c = n \times p_c^{(T)}$
 - 13: Find the first k_c elements in each row of $NN^{(T)}$. Set a^c to be the largest value of them.
 - 14: Let $KNN^c(x_i, a^c) = \{x | NN^{(T)}(i, j) > a^c\}$, and $n_i^c = |KNN^c|$, where $i = 1, \dots, n$ and $j = 1, \dots, n$.
 - 15: **for** $\text{index} = 1: n$ **do**
 - 16: For each node x_i has been labeled y_i , if $A^{(T)} > a^{y_i}$, $\text{score}_j = -\infty$; else, let $\text{score}_i = \max_{A^{(T)}(i,j) > \frac{a^c}{\text{index}}} (n_i^c - n_j^c)$
 - 17: Select the nodes x with the largest score to labeling oracle.
 - 18: If the label of x is exact class c , break; else, mark the class that x belongs to as discovered.
 - 19: **end for**
 - 20: **end for**
-

Based on Theorem 4.2, we can identify the rows of $NN^{(t)}$, in which the order of the $K^{(t)}$ largest elements will not change. Thus, we only need to update the disordered rows in $NN^{(t)}$. The single-updated incremental RCD algorithm (SIRD) is shown in Algorithm 4.1. In Step 1 to Step 2, we first initialize the diagonal matrix D and neighbor information matrix $NN^{(1)}$ at time step 1. In Step 4, let $K^{(t)}$ represent the number of nodes in the largest minority class at time step t . Then, from Step 5 to Step 6, we update the global similarity matrix at each

time step. Step 7 to Step 9 updates the rows in $NN^{(t)}$, of which the $K^{(t)}$ largest elements are changed. Step 11 to 20 is the query process. First of all, we calculate the class specific a^c at Step 13, which is the largest global similarity to the $k_c^{(th)}$ nearest neighbor. Then, in Step 14, we count the number of its neighbors whose global similarity is larger than or equal to a^c , and let n_i^c denote the counts for each node x_i . In Step 16, we calculate the score of each node x_i , which represents the change of local density. At last, we select the nodes with the largest score and let them be labeled by oracle. The query process only terminates as long as all the minority classes are discovered.

The efficiency of the updating process for Algorithm 4.1 is given by the following lemma.

Lemma 4.1. The computational cost of the updating process at each time step in Algorithm 4.1 is $O(n^2 + l \cdot K^{(t)} \cdot n)$.

Proof. As described before, the computational cost for normalization and decomposition process is $O(n)$. Then, in Step 6, compared to the straightforward computation, i.e., $A^{(t-1)} = (I - \alpha M^{(t)})^{-1}$, we reduce the time complexity from $O(n^3)$ to $O(n^2)$ by avoiding the matrix inverse computation. Furthermore, from Step 7 to Step 9, we simplify the resorting process by only updating the rows, in which the top $K^{(t)}$ elements are disordered. Suppose l is the total number of rows in $NN^{(t)}$, which does not satisfy Eq. 4.3.3, then the computational cost is reduced from $O(K^{(t)} \cdot n^2)$ to $O(l \cdot K^{(t)} \cdot n)$. By leveraging each part, the computational cost of the updating process is $O(n^2 + l \cdot K^{(t)} \cdot n)$. QED.

Batch Update. In most real world applications, we always observe that a batch of edges change at the same period. Specifically, the updated aggregated adjacency matrix $\Delta M^{(t)}$ may have more than one non-zero element. Hence, $\Delta M^{(t)}$ cannot be decomposed into two column vectors, and Theorem 4.2 could not be applied in this condition. In this part, we introduce Theorem 4.3 to help us update the neighbor information matrix $NN^{(t)}$ when a batch of edges are changed.

Theorem 4.3. Suppose there are m edges $\{(a^1, b^1), \dots, (a^m, b^m)\}$ being updated at time step t . The first $K^{(t)}$ elements in $NN^{(t)}(i, :)$ are the same as $NN^{(t-1)}(i, :)$, if it satisfies the condition that $\frac{\alpha}{I + V^T A^{(t-1)} U} \leq \min_{i=1, \dots, m} \{T_i\}$, where $T_i = \min\left\{\frac{\delta_i^{(t-1)}}{A_{i, a^i}^{(t-1)} \phi_{b^i}}, \frac{\delta_i^{(t-1)}}{A_{i, b^i}^{(t-1)} \phi_{a^i}}\right\}$.

Proof. Since the aggregated adjacency matrix $M^{(t)}$ is a symmetric matrix, then, each element (a, b) , where $a \neq b$, has a symmetrical element (b, a) in $M^{(t)}$. When the two edges (a, b) and (b, a) are updated at time step t , we have $\Delta M^{(t)} = \Delta M_1^{(t)} + \Delta M_2^{(t)}$, where $\Delta M_1^{(t)}$ has only one non-zero element (a, b) , and $\Delta M_2^{(t)}$ has only one non-zero element (b, a) . Similar

to Eq. 4.9, we have an approximate updating rule as follows.

$$\begin{aligned} A^{(t)} &\cong A^{(t-1)} + \alpha \frac{A^{(t-1)} \Delta M_1^{(t)} A^{(t-1)}}{I + (v^{(1)})^T A^{(t-1)} u^{(1)}} \\ &+ \alpha \frac{A^{(t-1)} \Delta M_2^{(t)} A^{(t-1)}}{I + (u^{(1)})^T A^{(t-1)} v^{(1)}} \end{aligned} \quad (4.11)$$

where $\Delta M_1^{(t)} = u^{(1)}(v^{(1)})^T$, $\Delta M_2^{(t)} = v^{(1)}(u^{(1)})^T$ and $u^{(1)}$, $v^{(1)}$ are two column vectors.

Besides, we also have

$$A^{(t)} = A^{(t-1)} + \beta(A^{(t-1)} \Delta M_1^{(t)} A^{(t-1)} + A^{(t-1)} \Delta M_2^{(t)} A^{(t-1)}) \quad (4.12)$$

where $\beta = \frac{\alpha}{I + (v^{(1)})^T A^{(t-1)} u^{(1)}}$, and β is a constant. Therefore, $A_{i,j}^{(t)} = A_{i,j}^{(t-1)} + \beta A_{i,a}^{(t-1)} A_{b,j}^{(t-1)} + \beta A_{i,b}^{(t-1)} A_{a,j}^{(t-1)}$.

Based on Theorem 4.2, we can claim that the largest $K^{(t)}$ elements in $NN^{(t)}(i, :)$ will not change, if it satisfies

$$\frac{\alpha}{I + V^T A^{(t-1)} U} \leq T_1 \quad (4.13)$$

where $T_1 = \min\{\frac{\delta_i^{(t-1)}}{A_{i,a^1}^{(t-1)} \phi_{b^1}}, \frac{\delta_i^{(t-1)}}{A_{i,b^1}^{(t-1)} \phi_{a^1}}\}$.

Similarly, when there are $m^{(t)}$ pairs of edges being updated at time step t , we can claim that if it satisfies

$$\frac{\alpha}{I + V^T A^{(t-1)} U} \leq \min_{m=1}^{m^{(t)}} \{T_m\} \quad (4.14)$$

where $T_m = \min\{\frac{\delta_i^{(t-1)}}{A_{i,a^c}^{(t-1)} \phi_{b^c}}, \frac{\delta_i^{(t-1)}}{A_{i,b^c}^{(t-1)} \phi_{a^c}}\}$, then the first $K^{(t)}$ elements in $NN^{(t)}(i, :)$ will not change. QED.

The Batch-update Incremental Rare Category Detection (BIRD) algorithm is shown in Algorithm 4.2. Step 1 and Step 2 are the initialization step. Step 3 to Step 12 updates the global similarity matrix $A^{(t)}$ and the neighbor information matrix $NN^{(t)}$. Different from Algorithm 4.1, Step 5 to Step 8 iteratively updates the global similarity matrix $A^{(t)}$ based on $m^{(t)}$ changed edges. Another difference is that, in Step 10, T is the minimum value of the thresholds calculated from $m^{(t)}$ updated edges. At last, Step 13 to Step 20 is the query process, which is the same as what we have described in Algorithm 4.1.

The efficiency of batch-edges updating in Algorithm 4.2 is proved by the following lemma.

Lemma 4.2. In Algorithm 4.2, the computational cost of the updating process at each time step is $O(m^{(t)}n^2 + l \cdot K^{(t)} \cdot n)$.

Proof. Different from Algorithm 4.1, in Algorithm 4.2, we have $m^{(t)}$ updated edges at time step t . We need to update the global similarity matrix $A^{(t)}$ for $m^{(t)}$ times. Thus, the computation cost of updating the global similarity matrix is $O(m^{(t)}n^2)$. Let l be the number of rows in $NN^{(t)}$, which do not satisfy equation 4.14. For updating these rows in $NN^{(t)}$, the computational complexity is $O(l \cdot K^{(t)} \cdot n)$. Thus, in total, the computation cost of updating process at each time step is $O(m^{(t)}n^2 + l \cdot K^{(t)} \cdot n)$. QED.

Algorithm 4.2: BIRD Algorithm

Require: $M^{(1)}, A^{(1)}, \Delta S^{(2)}, \dots, \Delta S^{(T)}, p_c^{(t)}, \alpha$.

Ensure: The set I of labeled nodes

- 1: Construct the $n \times n$ diagonal matrix D , where $D_{ii} = \sum_{j=1}^n S^{(1)}_{ij}$, $i = 1, \dots, n$.
 - 2: Sort row i of $A^{(1)}$ decreasingly and save into $NN^{(1)}(i, :)$, where $i = 1, \dots, n$.
 - 3: **for** $t=2:T$ **do**
 - 4: Let $K^{(t)} = \max_{l=c}^C n \times p_c^{(t)}$.
 - 5: **for** $m = 1: m^{(t)}$ **do**
 - 6: Let column vector $u = D(:, a^m)^{-1/2}$, and column vector $v = \Delta S^{(t)}(a^m, b^m)D(:, b^m)^{-1/2}$, where $\Delta S^{(t)}(a^m, b^m)$ is the non-zero element in $\Delta S^{(t)}$.
 - 7: Update the global similarity matrix, i.e., $A^{(t)} = A^{(t-1)} + \alpha \frac{A^{(t-1)}uv^T A^{(t-1)}}{I + v^T A^{(t-1)}u}$.
 - 8: **end for**
 - 9: **for** $i=1:n$ **do**
 - 10: Based on Theorem 4.3, identify whether the first $K^{(t)}$ elements of $NN^{(t)}$ ($i, :$) are changed. If true, update the first $K^{(t)}$ elements in $NN^{(t)}(i, :)$; otherwise, let $NN^{(t)}(i, :) = NN^{(t-1)}(i, :)$.
 - 11: **end for**
 - 12: **end for**
 - 13: **while** not all the classes have been discovered **do**
 - 14: Calculate n_i for each node, where $i = 1, \dots, n$.
 - 15: **for** $\text{index} = 1: n$ **do**
 - 16: For each node x_i has been labeled y_i , if $A^{(T)} > a$, $\text{score}_j = -\infty$; else, let $\text{score}_i = \max_{A^{(T)}(i,j) > \frac{a}{\text{index}}} (n_i - n_j)$
 - 17: Select the nodes x with the largest score to labeling oracle.
 - 18: Mark the class that x belongs to as discovered.
 - 19: **end for**
 - 20: **end while**
-

4.3.4 BIRD with Less Information

In many applications, it may be difficult to obtain the exact priors of all the minority classes. In this subsection, we introduce BIRD-LI, a modified version of BIRD, which

requires only an upper bound prior $p^{(t)}$ for all the minority classes existing at time step t . To be specific, BIRD-LI calculates $NN^{(1)}$ and diagonal matrix D at the first time step, which is the same as BIRD. Then, the global similarity matrix $A^{(t)}$ and the neighbor information matrix $NN^{(t)}$ could be updated from the first time step to the time step T . The only difference between BIRD and BIRD-LI is that the size of the minority class $K^{(t)}$ is calculated based on an estimated upper bound prior instead of the exact ones for all the minority classes. After the updating process, BIRD-LI calculates an overall score for the minority classes and selects the nodes with the largest overall score to be labeled by the oracle.

BIRD-LI is described in Algorithm 4.3. It works as follows: Step 1 to Step 2 is the initial process for calculating $NN^{(1)}$ and the diagonal matrix D at the first time step. Step 3 to Step 12 aims to update the global similarity matrix $A^{(T)}$ and the neighbor information matrix $NN^{(T)}$ from time step 1 to time step T , which is the same as BIRD. The while loop from Step 13 to Step 20 is the query process. We calculate an overall score for the minority classes and select the nodes with the largest overall score to be labeled by the oracle. BIRD-LI only terminates the loop until all the classes are discovered.

4.4 QUERY DYNAMICS

In the previous section, we introduce two incremental RCD methods, i.e., BIRD and SIRD, which are used for identifying rare categories on time-evolving graphs. Taking the advantage of BIRD and SIRD, we can efficiently update the initial RCD model at time step 0 to any future time step T . However, in many real word applications, we may not want to make queries to oracle at each time step or we may only be allowed with a limited number of queries. In these two cases, we introduce the following two open problems: (1) query locating (QL): how to find the optimal time step T to discover rare categories; (2) query distribution (QD): how to allocate limited number of queries into different time steps.

4.4.1 Query Locating

First of all, we introduce the query locating problem. In real world applications, it could be the case that we are given a series of unlabeled time-evolving graphs $S^{(1)}, S^{(2)}, \dots, S^{(T)}$, and we need to select an optimal time step T_{opt} , so that we can identify the minority classes with as less queries as possible (ALAP) and as early as possible (AEAP).

Before presenting our methods, let us introduce the two main factors that may affect the required number of queries in rare category detection. The first factor is $P(y = 2|x_i)$, which is the probability that example x_i belongs to the minority class given the features of x_i . A considerable number of works have already studied it before, such as MUVIR [2],

Algorithm 4.3: BIRD-LI Algorithm

Require: $M^{(1)}, A^{(1)}, \Delta S^{(2)}, \dots, \Delta S^{(T)}, p^{(t)}, \alpha$.

Ensure: The set I of labeled nodes and the L of their labels

- 1: Construct the $n \times n$ diagonal matrix D , where $D_{ii} = \sum_{j=1}^n S^{(1)}, i = 1, \dots, n$.
 - 2: Sort row i of $A^{(1)}$ decreasingly and save into $NN^{(t)}(i, :)$, where $i = 1, \dots, n$.
 - 3: **for** $t=2:T$ **do**
 - 4: Let $K^{(t)} = n \times p^{(t)}$.
 - 5: **for** $m = 1: m^{(t)}$ **do**
 - 6: Let column vector $u = D(:, a^m)^{-1/2}$, and column vector $v = \Delta S^{(t)}(a^m, b^m)D(:, b^m)^{-1/2}$, where $\Delta S^{(t)}(a^m, b^m)$ is a non-zero element in $\Delta S^{(t)}$.
 - 7: Update the global similarity matrix, i.e., $A^{(t)} = A^{(t-1)} + \alpha \frac{A^{(t-1)}_{uv} A^{(t-1)}_{vu}}{I + v^T A^{(t-1)} u}$, where u and v^T are the two vectors decomposed from normalized updating matrix $\Delta M^{(t)}$.
 - 8: **end for**
 - 9: **for** $i=1:n$ **do**
 - 10: Based on Eq. 4.14, identify whether the first $K^{(t)}$ elements of $NN^{(t)}(i, :)$ are changed. If true, update the first $K^{(t)}$ elements in $NN^{(t)}(i, :)$; otherwise, let $NN^{(t)}(i, :) = NN^{(t-1)}(i, :)$.
 - 11: **end for**
 - 12: **end for**
 - 13: **while** not all the classes have been discovered **do**
 - 14: Calculate n_i for each node, where $i = 1, \dots, n$
 - 15: **for** $\text{index} = 1: n$ **do**
 - 16: For each node x_i has been labeled y_i , if $A^{(T)} > a$, $\text{score}_j = -\infty$; else, let $\text{score}_i = \max_{A^{(T)}(i,j) > \frac{a}{\text{index}}} (n_i - n_j)$
 - 17: Select the nodes x with the largest score to labeling oracle.
 - 18: Mark the class that x belongs to as discovered.
 - 19: **end for**
 - 20: **end while**
-

GRADE [33] and NNDM [31]. Another factor is the density D_i at x_i , the definition of which is introduced in Theorem 4.4. When the density D_i at example x_i is high, it means there are many other examples close or similar to example x_i . Suppose there are two nodes x_i and x_j in graph G , where $P(y = 2|x_i) = P(y = 2|x_j)$ and $D_i > D_j$. Since the density at node x_i is larger than the density at node x_j , there is a higher probability that multiple classes are overlapped in the neighborhood of x_i . To some extent, higher density D_i implies higher probability of mis-classifying x_i . Thus, the value of $P(y = 2|x_i)$ is negatively correlated with the number of required queries, and the value of density D_i is positively correlated with the number of required labels. For the second factor, we introduce the following theorem to estimate local density based on the global similarity matrix constructed before.

Theorem 4.4. For each example x_i , the density of x_i is positively correlated with $D_i^{(t)}$ at time step t , where $D_i^{(t)} = \sum_{j=1}^n A_{i,j}^{(t)}$, $i = 1, \dots, n$.

Proof. Notice that $A^{(t)}(i, j)$ represents the global similarity between x_i and x_j . Thus, $D_i^{(t)} = \sum_{j=1}^n A_{i,j}^{(t)}$ is the aggregated global similarity between example x_i and all the existing examples on graph. If the density of example x_i is high, then it is always true that there are lots of examples which are similar or close to x_i . In other words, the density $D_i^{(t)}$ should be large. Similarly, when the density of x_i is low, the value of $D_i^{(t)}$ should be small. In conclusion, for any existing example x_i in the graph, its density is positively correlated with $D_i^{(t)}$. QED.

We let $score^{(t)} = P(y = 2|x_i^{(t)})$, which could be obtained using existing techniques such as MUVIR [2] or GRADE [33]. Under this circumstance, we propose to assign the hardness of identifying the minority classes at time step t as follows.

$$I^{(t)} = \left\{ k_c \max_{i=1, \dots, k_c} \frac{score_i^{(t)}}{D_i^{(t)}} \right\}^{-1} \quad (4.15)$$

where k_c is the number of examples in the minority class c . In Figure 4.2, the left figure shows the exact number of queries needed to identify rare categories from a series of time-evolving graphs. The right figure shows the value of $I^{(t)}$ calculated by Eq. 4.15. We can see these two curves are highly correlated.

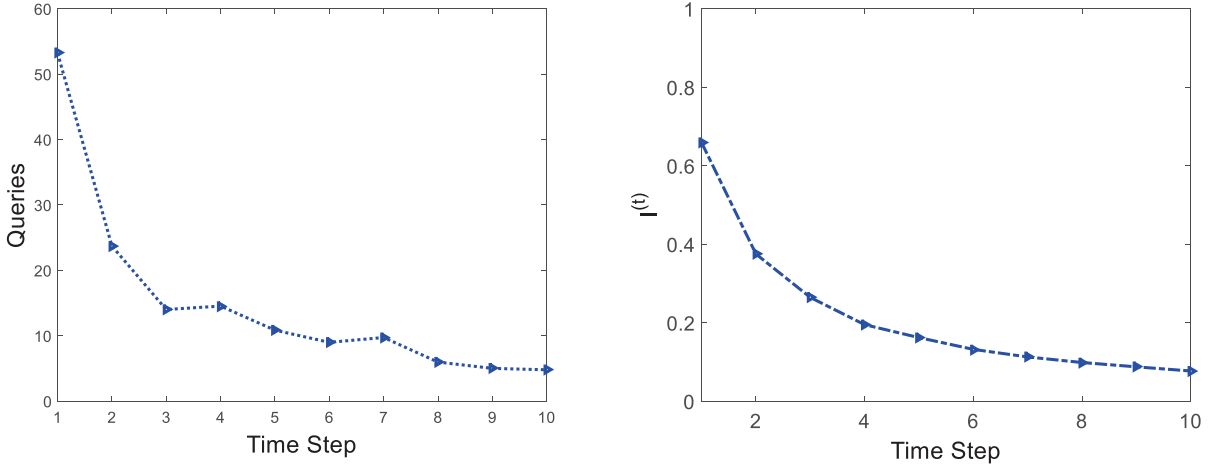


Figure 4.2: Correlation

Let $RS^{(t)}$ denote the number of required queries by random sampling at time step t . Simultaneously, let $C = \frac{RS^{(1)} - RS^{(T)}}{T}$. Intuitively, we could achieve optimal solution T_{opt} , when the difference between the “exact” saved number of queries and the estimated saved

Algorithm 4.4: Query Distribution Algorithm

Require: Strategy $S, M^{(1)}, A^{(1)}, NN^{(1)}, \Delta S^{(2)}, \dots, \Delta S^{(T)}, p^{(t)}, \alpha$.

Ensure: The set I of labeled nodes and the L of their labels.

```
1: for  $t = 1:T$  do
2:   Let  $K^{(t)} = \max_{l=c}^C n \times p_l^{(t)}$ .
3:   Calculate  $B^{(t)}$  as given Strategy  $S$ .
4:   Calculate  $NN^{(t)}$  as described in Algorithm 4.2.
5:   while not all the classes have been discovered do
6:     Find the  $(K^{(t)})^{th}$  element in each row of  $NN^{(t)}$ . Set  $a^c$  to be the largest
       value of them.
7:     Let  $KNN^c(x_i, a^c) = \{x | NN^{(T)}(i, j) > a^c\}$ , and  $n_i^c = |KNN^c|$ , where
        $i = 1, \dots, n$  and  $j = 1, \dots, n$ .
8:     for  $index = 1: B^{(t)}$  do
9:       For each node  $x_i$  has been labeled  $y_i$ , if  $A^{(T)} > a^{y_i}$ ,  $score_j = -\infty$ ; else,
       let  $score_i = \max_{A^{(T)}(i,j) > \frac{a^c}{index}} (n_i^c - n_j^c)$ 
10:      Select the nodes  $x$  with the largest score to labeling oracle.
11:      If the label of  $x$  is exact class  $c$ , break; else, mark the class that  $x$  belongs
       to as discovered.
12:    end for
13:  end while
14:  If all the minority classes are discovered, break.
15: end for
```

number of queries, i.e., $C * T_{opt}$, is maximized. The formulation is shown as follows.

$$\max_{t=1, \dots, T} \frac{I^{(1)} - I^{(t)}}{I^{(1)} - I^{(T)}} \cdot (RS^{(1)} - RS^{(T)}) - C \cdot t \quad (4.16)$$

4.4.2 Query Distribution

In this subsection, we discuss a more general problem: Query Distribution. In real-world applications, it could be the case that the updated graphs come as streams, and we need to allocate our query budget among multiple time steps. Hence, we need a method to allocate the queries properly among different time steps and enable us to find the minority class examples with the minimum query budget and time.

To further explore this problem, we generate a synthetic data set containing two classes, in which the initial proportion of the minority class equals to 0.1%. We increase the proportion of the minority class by 1% in each time step. In Figure 4.3, each point $(Q; T)$ represents the minimum required budget Q for identifying the minority class by time step T , and the

budget is evenly allocated from time step 1 to time step T . From this figure, we have 3 observations: (i) if we need to finish the task by time step 1, then the largest number of queries is required; (ii) if we only need to finish the task by the last time step, the smallest number of queries is required. (iii) the point at time step 3 is likely to hold a good trade-off, which has a relatively low querying number and early detection time.

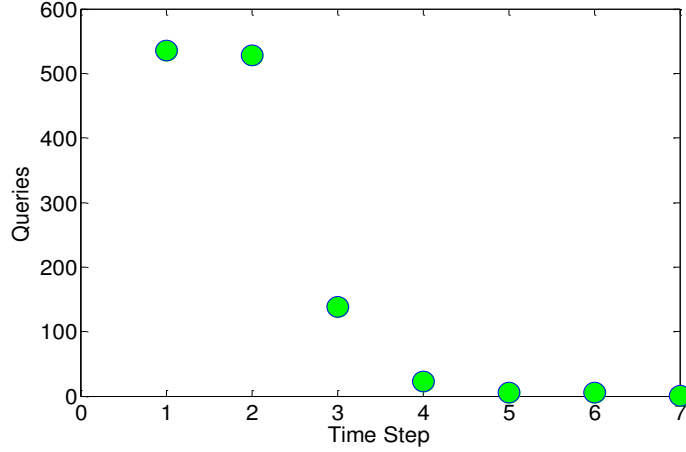


Figure 4.3: Query Allocation

To further study the query dynamics problem, we propose 5 potential strategies for the query distribution problem:

- $S1$. Allocate all the budget at the first time step.
- $S2$. Allocate all the budget at the last time step.
- $S3$. Allocate all the budget into time step T_{opt} .
- $S4$. Allocate the query budget evenly among different time steps.
- $S5$. Allocate the query budget into different time steps following exponential distribution, such as $e^{-\alpha t}$, where $\alpha > 0$.

For query distribution problem, we propose Algorithm 4.4. Different from the query process of Algorithm 4.2, in Step 3, we need to apply a strategy to calculate the certain budget $B^{(t)}$ for time step t . If we have not found the minority class within $B^{(t)}$ at time step t , then we go to the next time step. The overall algorithm stops either when the minority class is discovered or when there is no budget to use.

We compare the performance of these five strategies with both synthetic data sets and real data sets in Section 4.5.

4.5 EXPERIMENTAL EVALUATION

The analysis in Section 3 and Section 4 shows the advantage of our model in rare category detection on time-evolving graphs. In this section, we aim to empirically verify the effectiveness and the efficiency of the proposed algorithms on both synthetic data sets and real data sets.

4.5.1 Data Sets and Setup

Six time-evolving graph data sets are used for testing our proposed algorithms. Among these 6 data sets, there is 1 synthetic data set, 3 semi-real data sets and 2 real data sets. In Table 4.2, we list several statistics of each data set.

Name	Instance	Time Steps	Number of Classes
Synthetic Data	5,000	6	2
Abalone	4,177	6	20
Adult	48,842	6	2
Statlog	58,000	6	6
Epinion	5,665	16	24
Twitter	16,149	5	6

Table 4.2: Statistics of Different Data Sets.

The synthetic data set contains 5,000 instances, and we assume the proportion of the minority class is increasing over time. Hence, to generate the time-evolving graphs in later time steps, we let the proportion of a certain minority class increase by 1% and simultaneously let the proportion of the majority class decrease by 1% at each time step. Meanwhile, we generate additional 6 time-evolving graphs for 6 more time steps.

The Abalone data set comes from a biology study. In this data set, we need to predict the age of abalone based on multiple features. The age varies from 1 to 29, which roughly forms a normal distribution. Specifically, there are very few examples lying in the two extreme sides of the distribution. We separate the Abalone data set into 5 classes, i.e., one majority class and 4 minority classes. The proportion of the majority class is 56.93%, and the proportion of the smallest class is 0.4%. Besides, we choose the minority class with the smallest prior to evolve over time.

The Adult data set comes from a demographic census, which aims to predict whether the income of people exceeds \$50K per year or not. In Adult data set, there are 48,842 examples

containing one majority class and one minority class. The majority class is the population of income below \$50K, and the minority class is the population of income above \$50K. In this data set, around 24% of examples belong to the minority class. Since we stand on the problem of the rare category detection, we keep the majority class the same and only take 500 examples from the minority class. In this way, we can generate 24 data sets with the minority priors of 1.3%. Notice that all the experimental results for the Adult data set are calculated from the average values of the 24 sub-data sets.

The Statlog data set comes from a shuttle schedule database, which consists of 58,000 examples and 7 classes. However, we only include the 6 largest classes in our evaluation, because the smallest class only contains 13 examples. After this modification, the priors of the 5 minority classes vary from 0.04% to 15%. Same as before, we incrementally increase the proportion of the smallest minority class by 1% in each time step.

The Epinion data set is a collection of about 5,665 instances and 10,382 features over 16 time steps crawled from Epinion.com. Epinions is a product review site, where users can share their reviews about products. Users themselves can also build trust networks to seek advice from others. In this data set, each product is an instance, and the features for each product are formed by the bag-of-words model upon its reviews. In addition, the smallest class in Epinion only consists 0.03% vertices while the proportion of the largest class is 17.56%.

The Twitter data set is crawled from Twitter streaming API based on a set of terrorism related keywords, such as shoot, kidnap, blast and etc.. We include 16,149 English-speaking twitter users from 6 countries and around 10 millions tweets from 4/25/2015 to 5/5/2015. Then, we extract 209 features based on users' profiles, sentiments analysis, topic model analysis and users' ego-network analysis. In this data set, there are 56% of users from Turkey, 0.09% from Syria, 0.3% from Iraq, 1.3% from Iran, 36% from Saudi Arabia and 5.8% from Yemen. We separate the users into 6 classes based on their nationalities and generate a time-evolving graph in each 2-day interval.

4.5.2 Performance Evaluation

First of all, we demonstrate the effectiveness upon 1,000 synthetic data sets and 3 semi-synthetic data sets. We generate 1,000 synthetic data sets, and each of them contains 5,000 examples belonging to two classes. Besides, we initialize the priors of the minority classes as 1% and increase these priors by 1% at each time step. We also make use of 3 real data sets which meet the scenario of rare category detection. The details of these 3 real data sets are summarized in Table 4.2. Then, we synthesize additional 6 time-evolving graphs from

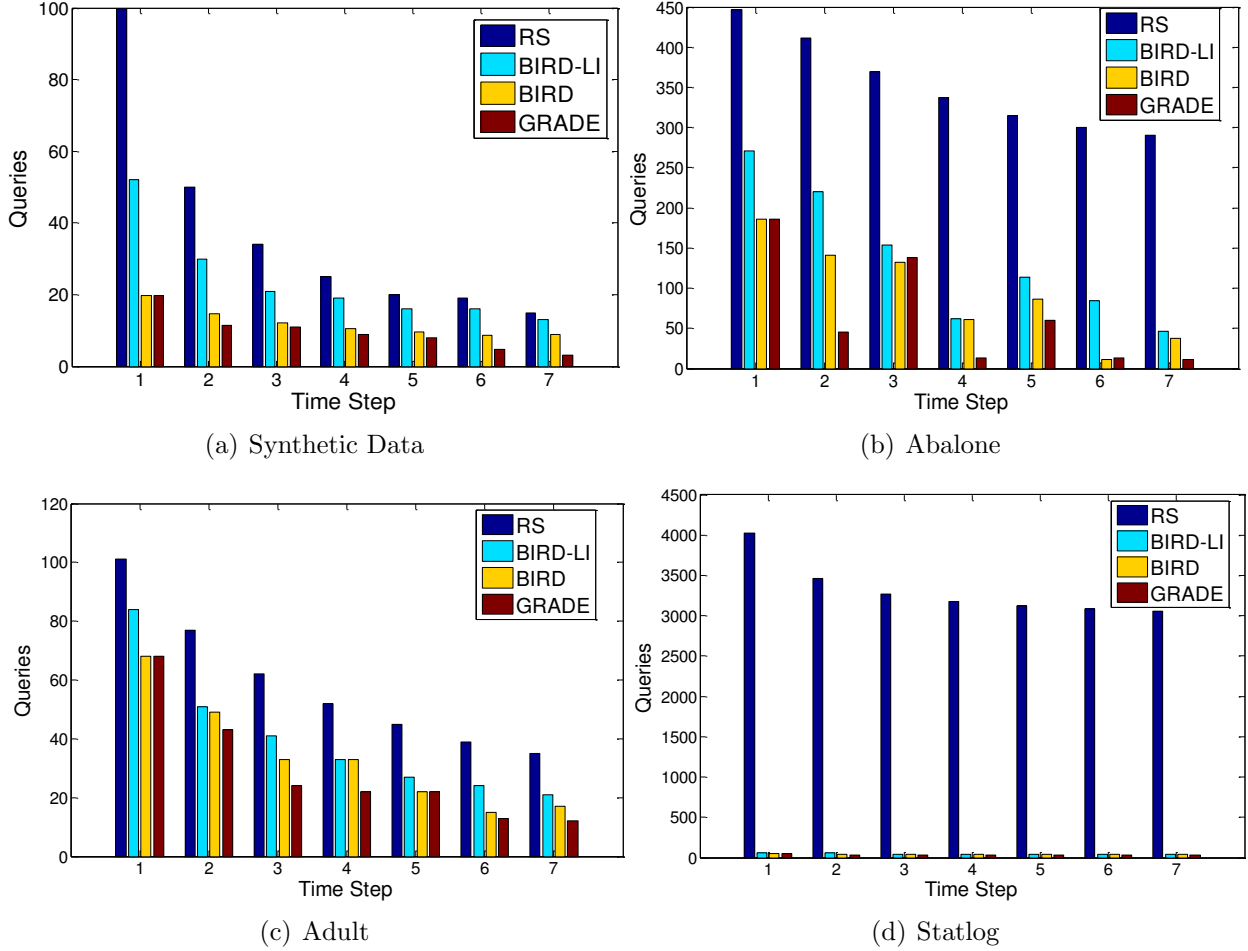


Figure 4.4: Performance on Synthetic and Semi-synthetic Data Sets

time step 2 to time step 7. For these time-evolving graphs, we let the proportion of a certain minority class increase by 1% and simultaneously let the proportion of the majority class decrease by 1% at each time step. Figure 4.4(a) shows the comparison results of 4 different methods: random sampling (RS), BIRD, BIRD-LI and GRADE. Notice that BIRD and BIRD-LI perform the query process upon the approximate aggregated adjacency matrix, while GRADE is performed on the exact adjacency matrix at each time step. Besides, we provide BIRD-LI with a much looser prior upper bound, e.g., we input 5% as the upper bound instead of using the exact prior of 1%. Then, we perform the same comparison experiments on 3 semi-synthetic data sets, which are shown in Figure 4.4(b), Figure 4.4(c) and Figure 4.4(d). At last, we evaluate our algorithms on two real data sets in Figure 4.5 and Figure 4.6. Different from the previous cases, the proportions of the minority classes vary randomly instead of increasing over time. In general, we have the following observations: (i) both BIRD and BIRD-Li outperform random sampling in any conditions; (ii) all of these 4

methods perform better when the prior of minority class is getting larger; (iii) BIRD gives a comparable performance as GRADE does; (iv) BIRD-LI is quite robust and requires only a few more queries than BIRD does in most cases.

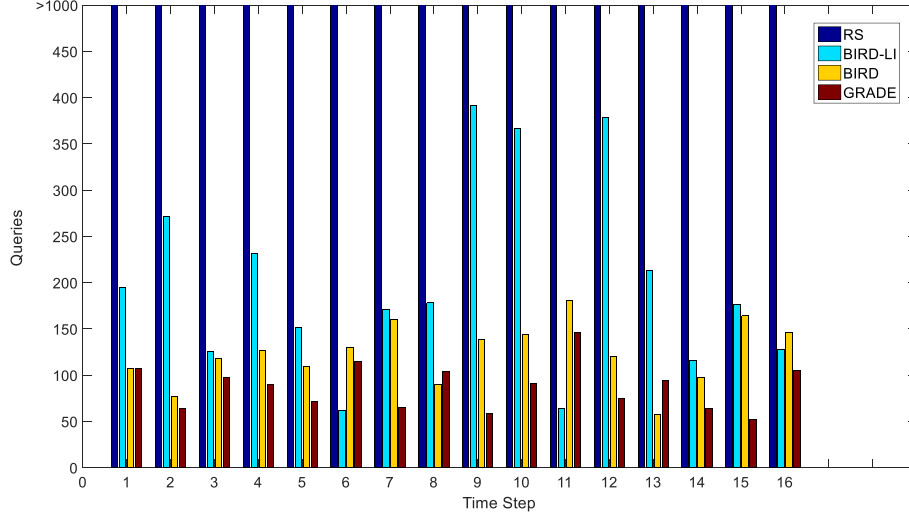


Figure 4.5: Performance on Epinion Data Set

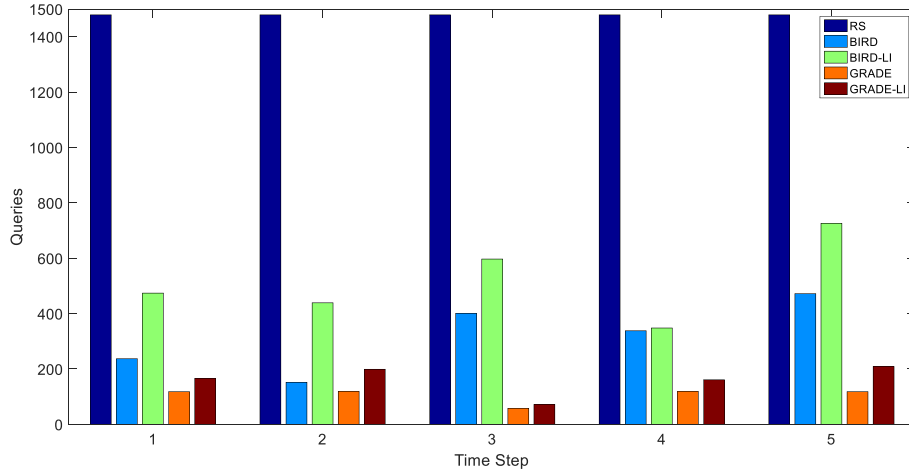


Figure 4.6: Performance on Twitter Data Set

4.5.3 Efficiency of Batch Update

We run the experiments with Matlab 2014a on a workstation with CPU 3.5 GHz 4 processors, 256 GB memory and 2 T disk space. For both BIRD and GRADE, the most time-consuming step is updating the global similarity matrix $A^{(t)}$ and neighbor information matrix $NN^{(t)}$ at each time step. In this subsection, we report the running time of updating

$A^{(t)}$ and $NN^{(t)}$ from an initial time step to the second time step. To better visualize the performance, we run the experiment on an increasing size of graph, i.e., from 500 examples in graph to 1,000 examples in graph. For each certain size, we have 100 identical-setting data sets. Each point in Figure 4.7 is computed based on the average value of the 100 data sets under identical settings. As we mentioned before, the computation cost of GRADE is $O(n^3)$, and our method only costs $O(n^2)$. From Figure 4.7, we can see the difference of running time is gradually increasing over time. The difference is limited when the number of examples is 500. However, when the size of graph goes to 10,000, the running time of BIRD is 6.227 seconds, while the running time of GRADE is 41.41 seconds, which is 7 times of BIRD. Moreover, the difference would be even more significant when we run algorithms on a series of time steps.

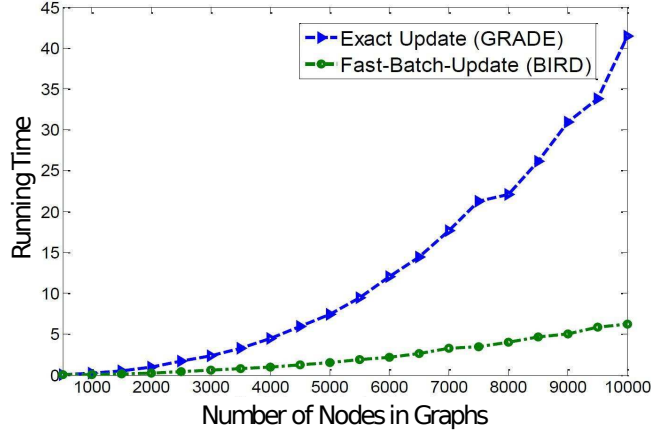


Figure 4.7: Efficiency

4.5.4 Query Dynamics

In this subsection, we show the performance of query locating and query distribution. In Figure 4.8, we apply the query locating methods on 3 real data sets. As the proportion is increasing over time, the labeling request is decreasing in general. Besides, we also observe that T_{opt} is always located at the left bottom of each graph, which meets our ALAP and AEAP intuitions.

Furthermore, by applying Algorithm 4.4, we perform the results of 5 different strategies on one binary-class synthetic data set and one binary-class real data set, i.e., Adult. In both Figure 4.9(a) and F.g 4.9(b), we observe that Strategy $S1$ is always located at the left top of the figure, which holds the time optimal; Strategy $S2$ is always located at the right bottom of the figure, which holds the budget optimal; Strategy $S3$ is always located at the

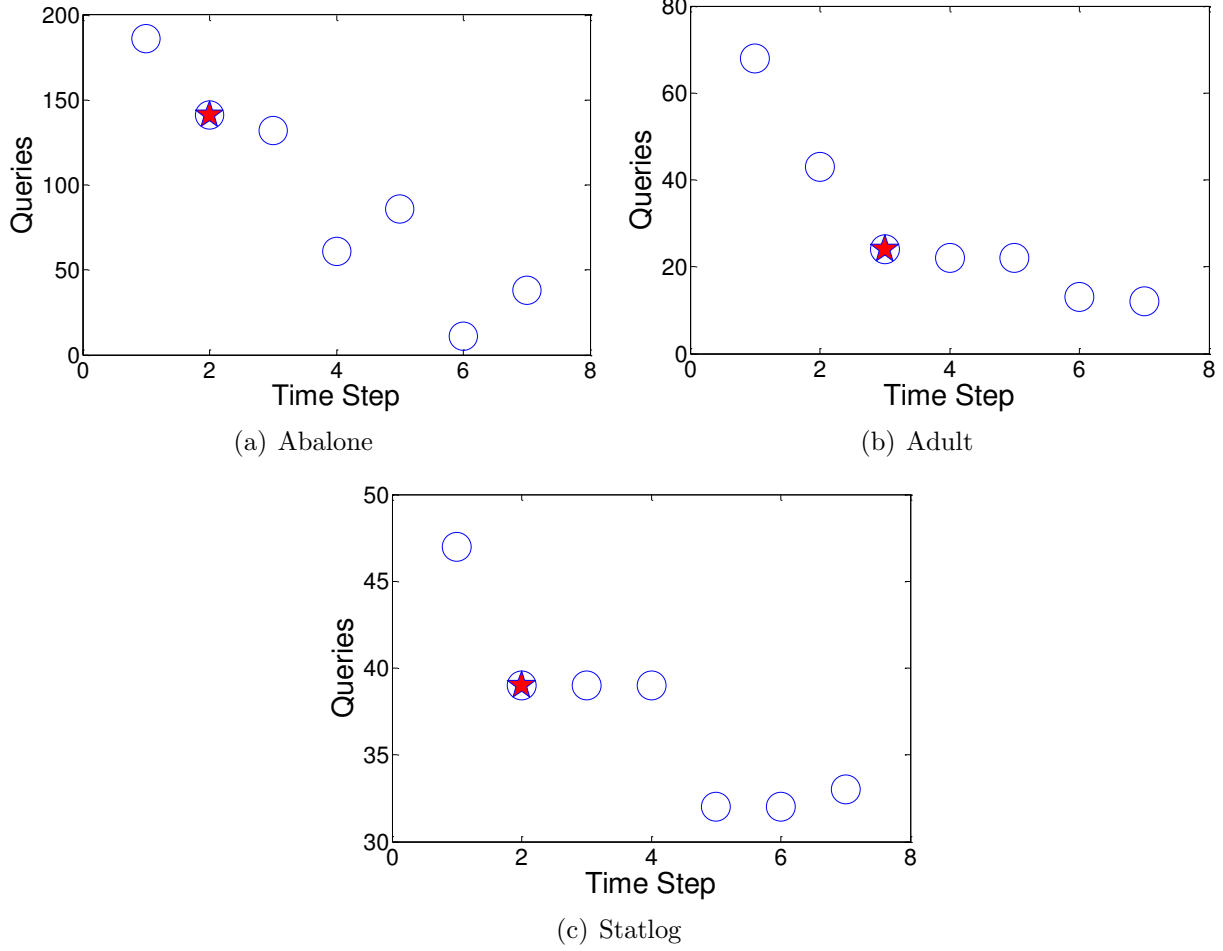


Figure 4.8: Query Locating

left bottom of the figure, which considers both the time and the budget factors. All of these 3 observations are consistent with our intuitions.

Besides, we also find two interesting observations. The first one is that, in Figure 4.9(a), Strategy $S4$ performs slightly better than Strategy $S5$, while Strategy $S5$ outperforms Strategy $S4$ in Figure 4.9(b). The reason is as follows. Strategy $S5$ always allocates most of the budget at the earliest few time steps, which is why Strategy $S5$ could identify minority class examples at time step 1 in Figure 4.9(b). But, if Strategy $S5$ cannot discover the minority class at the beginning, it will finish the task later than Strategy $S4$, which is why $S5$ performs worse than $S4$ in Figure 4.9(a). Strategy $S4$ allocates its budget evenly among each time steps. However, when the task is relatively tough at the beginning few time steps and relatively easy at the end, Strategy $S4$ may fail. This is what is happening in Figure 4.9(b). Another interesting observation is that, in Figure 4.9(b), Strategy $S3$ only queries 27 examples at time step 3 for discovering the minority class, while Strategy $S4$ needs

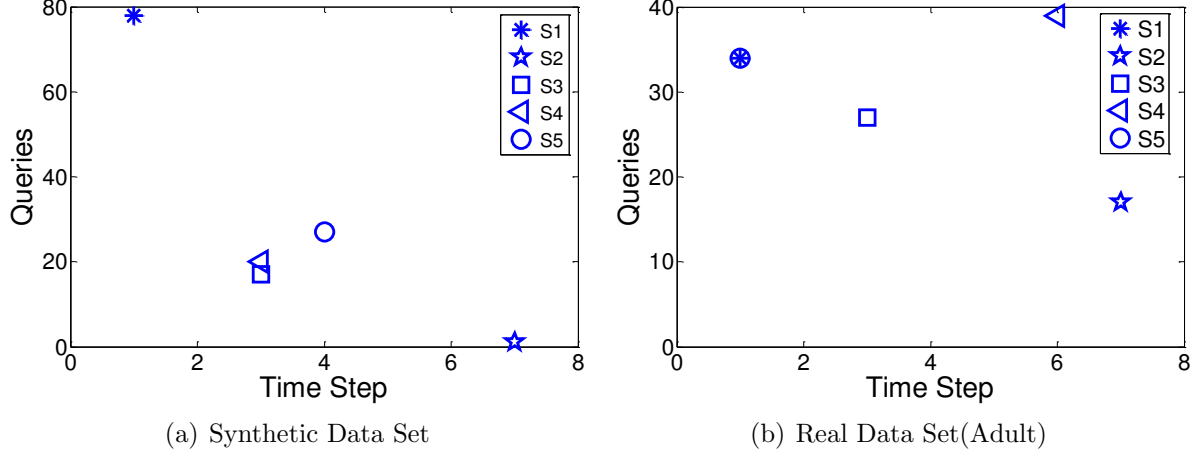


Figure 4.9: Query Distribution

39 labeling requests. Since the graph is evolving over time, Strategy $S4$ may automatically skip some minority-class examples when these examples are pretty similar to the previous labeled examples, which is the reason why Strategy $S4$ requires more queries.

4.6 SUMMARY

In this chapter, we mainly focus on the problem of efficiently and incrementally identifying under-represented rare category examples from time-evolving graphs. We propose two fast incremental updating algorithms, i.e., BIRD and SIRD, as well as a generalized version of BIRD named BIRD-LI to handle the problems where the exact priors of the minority classes are unknown. Furthermore, based on our algorithms, we introduce five strategies to deal with the novel problem – query distribution. To the best of our knowledge, this is the first attempt in rare category detection under these dynamic settings. The comparison experiments with state-of-the-art methods demonstrate the effectiveness and the efficiency of the proposed algorithms.

CHAPTER 5: BI-LEVEL RARE TEMPORAL PATTERN CHARACTERIZATION

5.1 OVERVIEW AND MOTIVATION

In the era of big data, we are exposed to large amount of temporal data, such as biomedical signals [103], financial transaction records [104], and network traffic [105]. Besides the large volume of data, we are also facing the following challenges: (1) the class membership is often highly skewed in the sense that the minority classes (rare temporal patterns) are overwhelmed by the majority classes (normal temporal patterns); (2) it is usually the case that identifying the minority classes is more important than identifying the majority classes in the temporal data; (3) the minority classes are often non-separable from the majority classes. For example, most of the ECG signals collected by wearable devices are normal, generated by healthy people, and only a small number of them are abnormal, generated by people with certain heart diseases such as arrhythmia. Without domain specific knowledge, it can be very difficult to distinguish between abnormal ECG signals and normal ones. In malicious insider identification, the daily activities of most employees are normal, and only a small number of employees are malicious insiders with abnormal activities. Since these guileful insiders usually try to camouflage as normal employees, these abnormal activities may be very similar to the normal ones. Furthermore, within the abnormal temporal sequences, there may only be a few time segments exhibiting similar abnormal patterns, forming a rare category of temporal patterns. For instance, the ECG signal of an individual with arrhythmia may only show irregular heartbeats in a few time segments; the malicious insiders may behave abnormally every now and then. Figure 5.1 illustrates such bi-level structure of the temporal data, where abnormal sequences contain at least one abnormal time segment, and normal sequences only contain normal time segments. In this chapter, we aim to detect abnormal sequences and abnormal segments simultaneously, which correspond to the bi-level rare temporal pattern detection.

To the best of our knowledge, such bi-level structure (sequence level vs. segment level) is not exploited in existing works on outlier detection for temporal data, which focus on either the sequence level, or the segment level. Furthermore, they fail to explore the similarity among the abnormal time segments, treating them as isolated outliers. On the other hand, existing works on rare category analysis are mainly focused on static data, which are not readily applicable to temporal data with rare categories of abnormal patterns.

To bridge this gap, in this chapter, we study the problem of rare temporal pattern detection by exploiting the bi-level structure in the data. Our proposed model is based on

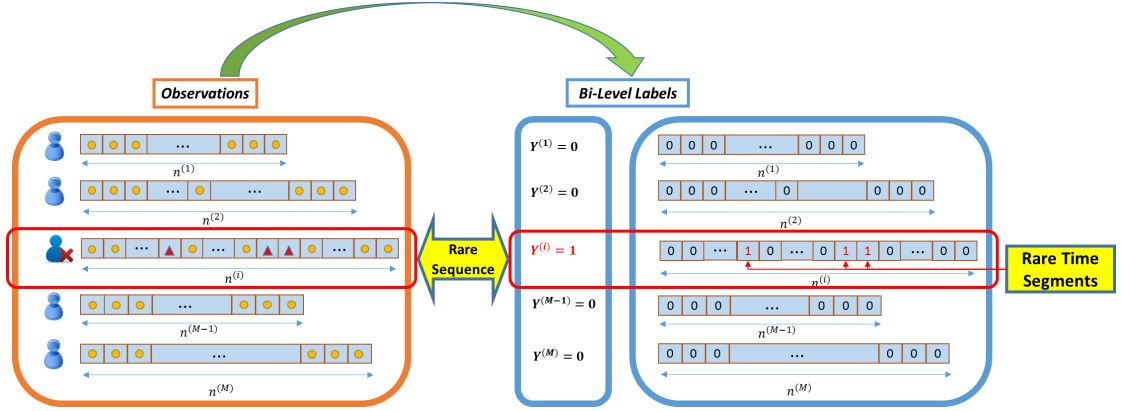


Figure 5.1: Illustration of the Bi-Level Structure in the Temporal Data.

an optimization framework that maximizes the likelihood of observing the data on both the sequence level and the segment level. Furthermore, it uses sequence-specific simple hidden Markov models to generate segment-level labels, and leverages the similarity among the abnormal time segments to estimate the model parameters. To solve the optimization problem, we propose an unsupervised algorithm for detecting rare temporal patterns named BIRAD and its semi-supervised version named BIRAD-K. Both algorithms are based on Block Coordinate Update, which repeatedly update the sequence-level labels, segment-level labels, and the model parameters. We analyze these algorithms in terms of convergence and time complexity, and empirically evaluate their performance on both synthetic and real data sets.

The major contributions of this chapter can be summarized as follows:

- 1 A novel problem setting of bi-level rare temporal pattern detection;
- 2 An optimization framework maximizing the likelihood of observing the data on both the sequence level and the segment level;
- 3 BIRAD and BIRAD-K algorithms for solving the optimization framework;
- 4 Analysis of the proposed algorithms in terms of convergence and efficiency;
- 5 Experimental results on both synthetic and real data sets demonstrating the performance of the proposed algorithms from various aspects.

The rest of this chapter is organized as follows. After a brief review of the related work in Section 5.2, we introduce the bi-level model, the optimization framework, and the proposed algorithms with performance analysis in Section 5.3. In Section 5.4, we present the experimental results on both synthetic and real data sets, which demonstrate the effectiveness and efficiency of the proposed framework. Finally, we conclude this chapter in Section 5.5.

5.2 RELATED WORK

In this section, we briefly review the related work on rare category analysis, outlier detection for temporal data, and multi-instance learning.

5.2.1 Rare Category Analysis

Rare category detection is the problem of identifying minority classes from the under-represented feature spaces, while minimizing the number of labeling requests. Up until now, several techniques have been developed for rare category detection in different scenarios. [26] introduced the problem setting of rare category detection and experimented with different hint selection strategies to detect useful anomalies. In [31, 33], the authors presented two active learning schemes to detect rare categories via unsupervised local-density-differential sampling strategy. More recently, in [2], the authors studied the problem of rare category detection on multi-view data and proposed a Bayesian framework named MUVIR, which exploited the relationship between multiple views and estimated the overall probability of each example belonging to the minority class. In [29], the authors proposed a fast method for rare category detection on time-evolving graphs, which incrementally updated the detection models based on local updates. In this chapter, we further study the problem of rare category detection on temporal data and aim to exploit the bi-level structure of abnormal temporal sequences / time segments.

5.2.2 Outlier Detection for Temporal Data

Outlier detection, also called anomaly detection or novelty detection, refers to the problem of finding instances that do not conform to the expected behavior in the data. This problem has been studied in various domains, such as heterogenous networks [105, 106, 107], crowdsourcing [108, 109] and spatiotemporal channels [110, 111]. Prior works mainly focused on two categories of temporal outliers: outliers in time series databases and outliers within the given time series [22]. For the first category of outliers, the previous methods aim to identify a few time series as outliers, such as clustering methods [112], parametric methods [113], window-based methods [114]. For the second category of outliers, the methods aim to find particular elements or subsequences on the given time series. For example, in [115], the authors presented an autoregressive data-driven model to identify outliers in environmental data streams; in [45], the authors studied a more challenging problem that outlier detection faced with a never-ending data stream. Different from existing works on

outlier detection for temporal data, our work focuses on the more challenging case where the abnormal temporal patterns are non-separable from the normal ones, and we propose to leverage the relationship between abnormal temporal sequences and abnormal time segments for the sake of improving the detection accuracy.

5.2.3 Multi-Instance Learning

Multi-instance learning is a variation of supervised learning, where examples are considered as bags consisting of multiple individual instances. [116] is the earliest literature that introduced and showed the importance of multi-instance learning. In the past decades, various techniques were proposed targeting multi-instance learning. In [117, 118], diverse density based frameworks are proposed for solving the multi-instance learning problem, by measuring the intersection of the positive bags minus the union of the negative bags. [119] presented an extended version of support vector machine on multi-instance learning, and developed a heuristic method to solve the mixed integer quadratic programs. [120] is the first study on the problem of multi-instance learning under the condition that examples are not independent and identically distributed (i.i.d) by constructing an undirected graph of each bag and designing a graph kernel to classify the positive and negative examples. Similar to multi-instance learning, in our model, the segment-level labels collectively determine the corresponding sequence-level label. However, here we assume that the relationship among adjacent time segments is governed by segment-specific simple hidden Markov models, and many existing works on multi-instance learning can be seen as special cases of our proposed model.

5.3 ALGORITHM

In this section, we propose a bi-level model for detecting the rare temporal patterns. We start with notation and problem definition. Then we present the model formulation, followed by the optimization techniques. Finally, we introduce both the unsupervised algorithm BIRAD for detecting the rare temporal patterns and its semi-supervised version BIRAD-K.

5.3.1 Notation and Problem Definition

Suppose that we are given a set of M temporal sequences $S = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(M)}\}$, and, in temporal sequence $\mathbf{x}^{(m)}$ where $m = 1, \dots, M$, there are $n^{(m)}$ temporal segments, i.e., $\mathbf{x}^{(m)} = \{x_1^{(m)}, \dots, x_{n^{(m)}}^{(m)}\}$. Let $\mathbf{y}^{(m)} = \{y_1^{(m)}, \dots, y_{n^{(m)}}^{(m)}\} \in \{0, 1\}^{1 \times n^{(m)}}$ denote the segment-level labels, or hidden states of temporal segments in $\mathbf{x}^{(m)}$, and $Y^{(m)} \in \{0, 1\}$ denote the

sequence-level label, or hidden state of $\mathbf{x}^{(m)}$. Without loss of generality, we assume that: (1) $y_i^{(m)} = 1$ corresponds to abnormal segments, and $y_i^{(m)} = 0$ corresponds to normal segments; (2) $Y^{(m)} = 1$ corresponds to abnormal temporal sequences, and $Y^{(m)} = 0$ corresponds to normal sequences. As the bi-level structure illustrated in Figure 5.1, only a small portion of temporal sequences in S correspond to abnormal sequences, in which only a small portion of temporal segments are abnormal segments. Therefore, the abnormal segments would be extremely rare when considering the whole data set S . Our goal is to identify anomalies in the sequence level as well as the segment level. For the sake of clarity, we also introduce the following indicator function $I(\mathbf{y}^{(m)}) = \max_{i=1}^{n^{(m)}} y_i^{(m)}$.

5.3.2 Model Formulation

Our model lies in inference about the bi-level hidden state process given the observations S , which involves calculating the following posterior distribution.

$$\begin{aligned} Pr(\mathbf{y}^{(m)}|\mathbf{x}^{(m)}) &\propto Pr(\mathbf{y}^{(m)}, \mathbf{x}^{(m)}) \\ &= Pr(\mathbf{x}^{(m)})Pr(\mathbf{y}^{(m)}|\mathbf{x}^{(m)}) \end{aligned} \quad (5.1)$$

Thus, we propose the objective of our model as follows.

$$\operatorname{argmax}_{\mathbf{y}^{(1:M)}} \sum_{m=1}^M \ln Pr(\mathbf{y}^{(m)}, \mathbf{x}^{(m)}) \quad (5.2)$$

As the data could be categorized into normal and abnormal temporal sequences, we can rewrite Eq. 5.2 as follows.

$$\begin{aligned} &\operatorname{argmax}_{\mathbf{y}^{(1:M)}} \sum_{Y^{(m)}=1} \ln Pr(\mathbf{y}^{(m)}, \mathbf{x}^{(m)}) \\ &+ \sum_{Y^{(m)}=0} \ln Pr(\mathbf{y}^{(m)}, \mathbf{x}^{(m)}) \end{aligned} \quad (5.3)$$

By introducing sequence-level label $Y^{(m)}$ to the preceding equation, we have

$$\begin{aligned} &\operatorname{argmax}_{\mathbf{y}^{(1:M)}} \sum_{m=1}^M \ln [Pr(\mathbf{y}^{(m)}, \mathbf{x}^{(m)}|Y^{(m)} = 1) \\ &\times Pr(Y^{(m)} = 1)]^{Y^{(m)}} + \ln [Pr(\mathbf{y}^{(m)}, \mathbf{x}^{(m)}|Y^{(m)} = 0) \\ &\times Pr(Y^{(m)} = 0)]^{(1-Y^{(m)})} \\ &\text{s.t. } Y^{(m)} = \max_i y_i^{(m)} \\ &m = 1, \dots, M, i = 1, \dots, n^{(t)} \end{aligned} \quad (5.4)$$

Let L_0 denote $\ln Pr(\mathbf{y}^{(m)}, \mathbf{x}^{(m)}|Y^{(m)} = 0)$, and L_1 denote $\ln Pr(\mathbf{y}^{(m)}, \mathbf{x}^{(m)}|Y^{(m)} = 1)$. We can rewrite Eq. 5.4 as follows.

$$\begin{aligned}
& \underset{\mathbf{y}^{(1:M)}}{\operatorname{argmax}} \sum_{m=1}^M (1 - Y^{(m)}) [L_0(\mathbf{x}^{(m)}) + \ln \Pr(Y^{(m)} = 0)] \\
& + Y^{(m)} [L_1(\mathbf{x}^{(m)}) + \ln \Pr(Y^{(m)} = 1)] \\
& \text{s.t. } Y^{(m)} = \max_i y_i^{(m)} \\
& m = 1, \dots, M, i = 1, \dots, n^{(t)}
\end{aligned} \tag{5.5}$$

In order to model the joint probability of the segment-level labels $\mathbf{y}^{(m)}$ and the temporal data $\mathbf{x}^{(m)}$, we propose to use simple Hidden Markov Models (HMM) [121]. In particular, we have the following three assumptions: (1) the Markov assumption, i.e., the next state is dependent only upon the current state, where the state corresponds to the segment-level label $y_i^{(m)}$; (2) the stationarity assumption, i.e., state transition probabilities are independent of the actual time at which the transitions take place; (3) the output independence assumption, i.e., current output (observation) is statistically independent of the previous outputs (observations). Next we elaborate on modeling normal and abnormal temporal sequences.

Modeling Normal Temporal Sequences: For the sake of exposition, we first model the normal temporal sequence, i.e., $Y^{(m)} = 0$. The log likelihood of any normal temporal sequence $\mathbf{x}^{(m)}$ is defined by

$$\begin{aligned}
L_0 &= \ln \Pr(\mathbf{y}^{(m)}, \mathbf{x}^{(m)} | Y^{(m)} = 0) \\
&= \ln[\Pr(\mathbf{x}^{(m)} | \mathbf{y}^{(m)}, Y^{(m)} = 0) \times \Pr(\mathbf{y}^{(m)} | Y^{(m)} = 0)]
\end{aligned} \tag{5.6}$$

Based on the Markov assumption and output independence assumption, we have

$$\Pr(\mathbf{y}^{(m)} | Y^{(m)} = 0) = \Pr(y_1^{(m)} | Y^{(m)} = 0) \times \prod_{j=2}^{n^{(m)}} \Pr(y_j^{(m)} | y_{j-1}^{(m)}, Y^{(m)} = 0) \tag{5.7}$$

By applying the stationarity assumption, we have

$$\Pr(\mathbf{x}^{(m)} | \mathbf{y}^{(m)}, Y^{(m)} = 0) = \prod_{i=1}^{n^{(m)}} \Pr(x_i^{(m)} | y_i^{(m)}, Y^{(m)} = 0) \tag{5.8}$$

Plugging Eq. 5.7 and Eq. 5.8 into Eq. 5.6, we have

$$L_0 = \ln \left[\prod_{i=1}^{n^{(m)}} \Pr(x_i^{(m)} | y_i^{(m)}, Y^{(m)} = 0) \times \Pr(y_1^{(m)} | Y^{(m)} = 0) \times \prod_{j=2}^{n^{(m)}} \Pr(y_j^{(m)} | y_{j-1}^{(m)}, Y^{(m)} = 0) \right] \tag{5.9}$$

On the other hand, we assume that any normal temporal segment $x_i^{(m)}$ is drawn from an unknown Gaussian distribution, although the proposed model can be generalized to other parametric distributions: $\Pr(x_i^{(m)} | y_i^{(m)}, Y^{(m)} = 0) \sim \mathcal{N}(x_i^{(m)}, \mu_0, \sigma_0)$, where mean μ_0 and variance σ_0 are not given. Hence, $\mathcal{N}(x_i^{(m)}, \mu_0, \sigma_0)$ could be interpreted as the emission

probability of $x_i^{(m)}$ given hidden state 0.

Then, Eq. 5.9 can be rewritten as follows.

$$L_0 = \sum_{i=1}^{n^{(m)}} \ln \mathcal{N}(x_i^{(m)}, \mu_0, \sigma_0) + \ln \Pr(y_1^{(m)} | Y^{(m)} = 0) \\ + \sum_{j=2}^{n^{(m)}} \ln \Pr(y_j^{(m)} | y_{j-1}^{(m)}, Y^{(m)} = 0) \quad (5.10)$$

For any normal temporal sequences $\mathbf{x}^{(m)}$, there is no segment-level state transition, i.e., all temporal segments are normal. Therefore, L_0 could be simplified as follows.

$$L_0 = \sum_{i=1}^{n^{(m)}} \ln \mathcal{N}(x_i^{(m)}, \mu_0, \sigma_0) \quad (5.11)$$

Modeling Abnormal Temporal Sequences: As we mentioned before, if temporal sequence $\mathbf{x}^{(m)}$ contains at least one abnormal segment $x_i^{(m)}$ with $y_i^{(m)} = 1$, then we claim that temporal sequence $\mathbf{x}^{(m)}$ is abnormal, i.e., $Y^{(m)} = 1$. Similar as before, we have the following log likelihood.

$$L_1 = \ln \Pr(\mathbf{y}^{(m)}, \mathbf{x}^{(m)} | Y^{(m)} = 1) \quad (5.12)$$

In our model, we assume that the features from abnormal time segments are generated from the same compact distribution across all abnormal temporal sequences. Similar to Eq. 5.9, by taking advantage of the HMM assumptions and Bayes' Rule, we can rewrite Eq. 5.12 as follows.

$$L_1 = \ln \left[\prod_{(i=1)}^{n^{(m)}} \Pr(x_i^{(m)} | y_i^{(m)}, Y^{(m)} = 1) \right. \\ \left. \times \Pr(y_1^{(m)} | Y^{(m)} = 1) \times \prod_{j=2}^{n^{(m)}} \Pr(y_j^{(m)} | y_{j-1}^{(m)}, Y^{(m)} = 1) \right] \quad (5.13)$$

For any abnormal temporal sequence $\mathbf{x}^{(m)}$, we define the corresponding Hidden Markov Model [121] $\lambda = (N, O, \mathbf{A}, \mathbf{B}, \pi)$ as follows.

1. N , the number of hidden states in the model. In this chapter, $N = 2$, i.e., normal and abnormal states.
2. O , the number of distinct observations. In our model, for any temporal sequence $\mathbf{x}^{(m)}$, the number of observations is the length of $\mathbf{x}^{(m)}$.
3. \mathbf{A} , $N \times N$, the state transition probability distribution. \mathbf{A} is an $N \times N$ matrix. In

this chapter, we have: $\mathbf{A} = \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix}$, where a_{ij} denotes the transition probability

from state i to state j . And we have $a_{ij} \in [0, 1]$ and $a_{i0} + a_{i1} = 1$, $i \in \{0, 1\}$.

4. \mathbf{B} , the observation emission probability distribution, which is an $N \times O$ matrix. We assume that normal time segments meet distribution $\mathcal{N}(\mu_0, \sigma_0)$, while abnormal time segments meet distribution $\mathcal{N}(\mu_1, \sigma_1)$.
5. π , the initial state probability distribution, of which the length is N . In our model, for any temporal sequence $\mathbf{x}^{(m)}$, we define $a_0^{(m)}$ as the probability that the initial temporal segment $x_1^{(m)}$ is abnormal. Then, we can write the initial state probability distribution of sequence $\mathbf{x}^{(m)}$ as $[1 - a_0^{(m)}, a_0^{(m)}]$.

Based on the HMM model $\lambda = (N, O, \mathbf{A}, \mathbf{B}, \pi)$, Eq. 5.13 can be rewritten as follows.

$$\begin{aligned}
L_1 = & \sum_{i=1}^{n^{(m)}} [y_i^{(m)} \ln \mathcal{N}(x_i^{(m)}, \mu_1, \sigma_1) + (1 - y_i^{(m)}) \ln \mathcal{N}(x_i^{(m)}, \mu_0, \sigma_0)] \\
& + \sum_{j=2}^{n^{(m)}} [y_{j-1}^{(m)} y_j^{(m)} \ln a_{11} + y_{j-1}^{(m)} (1 - y_j^{(m)}) \ln(1 - a_{11}) + (1 - y_{j-1}^{(m)}) y_j^{(m)} \ln a_{01} \\
& + (1 - y_{j-1}^{(m)}) (1 - y_j^{(m)}) \ln(1 - a_{01})] + [y_1^{(m)} \ln a_0 + (1 - y_1^{(m)}) \ln(1 - a_0)]
\end{aligned} \tag{5.14}$$

Overall Objective Function: Plugging Eq. 5.11 and Eq. 5.14 into the objective function in Eq. 5.5, we have

$$\begin{aligned}
& \underset{\mathbf{y}^{(1:M)}, a_0, a_{11}, \mu_1, \sigma_1, \mu_0, \sigma_0}{\operatorname{argmax}} \sum_{m=1}^M Y^{(m)} \left\{ \sum_{i=1}^{n^{(m)}} [y_i^{(m)} \ln \mathcal{N}(x_i^{(m)}, \mu_1, \sigma_1) + (1 - y_i^{(m)}) \ln \mathcal{N}(x_i^{(m)}, \mu_0, \sigma_0)] \right. \\
& + [y_1^{(m)} \ln a_0 + (1 - y_1^{(m)}) \ln(1 - a_0)] + \sum_{j=2}^{n^{(m)}} [y_{j-1}^{(m)} y_j^{(m)} \ln a_{11} + y_{j-1}^{(m)} (1 - y_j^{(m)}) \ln(1 - a_{11}) \\
& + (1 - y_{j-1}^{(m)}) y_j^{(m)} \ln a_{01} + (1 - y_{j-1}^{(m)}) (1 - y_j^{(m)}) \ln(1 - a_{01})] + \ln Pr(Y^{(m)} = 1) \Big\} \\
& + (1 - Y^{(m)}) \left\{ \sum_{i=1}^{n^{(m)}} \ln \mathcal{N}(x_i^{(m)}, \mu_0, \sigma_0) + \ln Pr(Y^{(m)} = 0) \right\} \\
& \text{s.t.} \quad a_0, a_{11}, a_{01} \in [0, 1] \\
& \quad Y^{(m)} = \max_i y_i^{(m)} \\
& \quad m = 1, \dots, M, i = 1, \dots, n^{(t)}
\end{aligned} \tag{5.15}$$

5.3.3 Optimization

Given any finite observation sequence, it is challenging to maximize the posterior probability by adjusting the HMM model parameters $(\mathbf{A}, \mathbf{B}, \pi)$. In fact, there is not a practical

method to exactly solve this problem. However, a number of iterative procedures, such as EM based methods [122] and gradient based methods [122], have been proposed to obtain a local maximum of this problem. In the following two subsections, we will introduce two simple and fast algorithms, i.e., BIRAD and BIRAD-K, targeting the novel setting of bi-level rare temporal pattern detection. Both of these two algorithms are built upon Block Coordinate Update (BCU) method [123, 124, 125], which divides all the variables into multiple blocks and iteratively updates them. To be specific,

Updating Initial State Probability Distribution: By taking the partial derivative of Eq. 5.15 with respect to a_0 , and letting it equal to zero, we have the following closed form update rule.

$$a_0^{(m)} = y_1^{(m)} \quad (5.16)$$

Updating State Transition Probability Distribution: By taking the partial derivative of Eq. 5.15 with respect to a_{11} and a_{01} , and letting them equal to zero, we have the following closed form update rules.

$$a_{11} = \frac{\sum_{m=1}^M \sum_{j=2}^{n^{(m)}} Y^{(m)} y_{j-1}^{(m)} y_j^{(m)}}{\sum_{m=1}^M \sum_{j=2}^{n^{(m)}} Y^{(m)} y_{j-1}^{(m)}} \quad (5.17)$$

$$a_{01} = \frac{\sum_{m=1}^M \sum_{j=2}^{n^{(m)}} Y^{(m)} (1 - y_{j-1}^{(m)}) y_j^{(m)}}{\sum_{m=1}^M \sum_{j=2}^{n^{(m)}} Y^{(m)} (1 - y_{j-1}^{(m)})} \quad (5.18)$$

Updating Observation Emission Probability Distribution: By taking the partial derivation of Eq 5.15 with respect to μ_1 , σ_1 , μ_0 , σ_0 , and letting them equal to zero, we have the following closed form update rules.

$$\mu_1 = \frac{\sum_{m=1}^M \sum_{t=1}^{n^{(m)}} y_t^{(m)} x_t^{(m)}}{\sum_{m=1}^M \sum_{t=1}^{n^{(m)}} y_t^{(m)}} \quad (5.19)$$

$$\sigma_1 = \frac{\sum_{m=1}^M \sum_{t=1}^{n^{(m)}} y_t^{(m)} |x_t^{(m)} - \mu_1|^2}{\sum_{m=1}^M \sum_{t=1}^{n^{(m)}} y_t^{(m)} - 1} \quad (5.20)$$

$$\mu_0 = \frac{\sum_{m=1}^M \sum_{t=1}^{n^{(m)}} (1 - y_t^{(m)}) x_t^{(m)}}{\sum_{m=1}^M \sum_{t=1}^{n^{(m)}} (1 - y_t^{(m)})} \quad (5.21)$$

$$\sigma_0 = \frac{\sum_{m=1}^M \sum_{t=1}^{n^{(m)}} (1 - y_t^{(m)}) |x_t^{(m)} - \mu_0|_2^2}{\sum_{m=1}^M \sum_{t=1}^{n^{(m)}} (1 - y_t^{(m)}) - 1} \quad (5.22)$$

Updating Bi-level Labels: In this part, we give an easy and fast update strategy for updating bi-level labels. For updating the sequence-level labels, we first score each temporal sequence by comparing the log likelihood of the sequence being labeled as abnormal vs. normal in each iteration. Then, the sequences with higher scores would be labeled as abnormal and the rests will be labeled as normal. The details will be illustrated in BIRAD and BIRAD-K. For updating the segment-level labels, there are the following two cases. When the sequence-level label $Y^{(m)} = 0$, we can directly label each segment in $\mathbf{y}^{(m)}$ as $0_{1 \times n^{(m)}}$. When the sequence-level label $Y^{(m)} = 1$, we apply Viterbi algorithm [126] to iteratively update the most likely hidden states, or segment-level labels, $\mathbf{y}^{(\mathbf{m})}$, which maximizes the objective function in Eq. 5.15.

5.3.4 BIRAD Algorithm

Based on the update rules introduced in the previous subsection, we first introduce the unsupervised method — Bi-level Rare Temporal Anomaly Detection (BIRAD) algorithm. It is given an unlabeled temporal sequence data set S and the proportion of abnormal sequences P as inputs. It outputs the hidden states of all temporal sequences and temporal segments in S . The algorithm iteratively updates the HMM parameters $\lambda = (\mathbf{A}, \mathbf{B}, \pi)$ and the bi-level hidden states until convergence, or a certain stopping criterion is satisfied. The details of BIRAD are presented in Algorithm 5.1.

BIRAD works as follows. First, Step 1 initializes the sequence-level/ segment-level labels. Specifically, one potential way to initialize the bi-level hidden states is to randomly select $M \times P$ temporal sequences and label them as 1, while the rest are labeled as 0. Then, we can initialize any hidden states of temporal segments to be identical as the hidden state of the corresponding temporal sequence. Next, Step 2 to Step 18 applies the BCU optimization process. From Step 3 to Step 5, BIRAD updates the initial probability vector π , transition probability distribution \mathbf{A} and emission probability distribution \mathbf{B} based on the updated labels $\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(M)}$. In Step 7 to Step 10, BIRAD updates the segment-level hidden states of $\mathbf{x}^{(\mathbf{m})}$ and calculates the scores for each temporal sequence $\mathbf{x}^{(\mathbf{m})}$, which estimate the probability of a sequence being abnormal rather than normal. Step 12 updates the sequence-level/ segment-level labels based on $score^{(m)}$. Step 13 to Step 17 checks if there is any inconsistency between $\mathbf{y}^{(\mathbf{m})}$ and $Y^{(m)}$. If any inconsistency exists, these temporal sequences are labeled as

Algorithm 5.1: Bi-level Rare Temporal Anomaly Detection (BIRAD)

Require:

Temporal sequence data set $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(M)}$
Proportion of abnormal sequences P .

Ensure:

$Y^{(1)}, \dots, Y^{(M)}; \mathbf{y}^{(1)}, \dots, \mathbf{y}^{(M)}.$

- 1: Initialize sequence-level and segment-level labels.
 - 2: **while** stopping criterion is not satisfied **do**
 - 3: Update initial state probability distribution π by Eq. 5.16.
 - 4: Update transition probability distribution \mathbf{A} by Eq. 5.17 to Eq. 5.18.
 - 5: Update emission probability distribution \mathbf{B} by Eq. 5.19 to Eq. 5.22.
 - 6: **for** $m = 1 : M$ **do**
 - 7: Update hidden states $\mathbf{y}^{(m)}$ of $\mathbf{x}^{(m)}$ by Viterbi Algorithm.
 - 8: Compute $L_1(\mathbf{x}^{(m)})$ in Eq. 5.14 based on updated $\mathbf{y}^{(m)}$.
 - 9: Compute $L_0(\mathbf{x}^{(m)})$ in Eq. 5.11 based on updated $\mathbf{y}^{(m)}$.
 - 10: Compute $score^{(m)} = L_1(\mathbf{x}^{(m)}) + \ln P - L_0(\mathbf{x}^{(m)}) - \ln(1 - P)$
 - 11: **end for**
 - 12: Label the temporal sequences with positive scores as abnormal, i.e., $Y^{(m)} = 1$, and keep the updated prediction labels $\mathbf{y}^{(m)}$. Label the remaining temporal sequences as normal, i.e., $Y^{(m)} = 0$, and label the segments in these sequences as normal, i.e., $\mathbf{y}^{(m)} = 0_{1 \times n^{(m)}}$.
 - 13: **for** $m = 1 : M$ **do**
 - 14: **if** $I(\mathbf{y}^{(m)}) \neq Y^{(m)}$ **then**
 - 15: Let $Y^{(m)} = 0$ and $\mathbf{y}^{(m)} = 0_{1 \times n^{(m)}}$.
 - 16: **end if**
 - 17: **end for**
 - 18: **end while**
 - 19: **return** $Y^{(1)}, \dots, Y^{(M)}; \mathbf{y}^{(1)}, \dots, \mathbf{y}^{(M)}.$
-

normal. At last, in Step 19, BIRAD returns the predicted bi-level labels.

Next, we analyze the convergence of the proposed BIRAD algorithm. We first derive Lemma 5.1 and Lemma 5.2, which show that the update rules in Algorithm 5.1 are upper-bounded and non-decreasing. Lemma 5.1 and Lemma 5.2 lead to Theorem 5.1, which shows the convergence of BIRAD.

Lemma 5.1 (Upper-bounded). The overall objective function in Eq. 5.15 is upper-bounded.

Proof Sketch. Suppose there exists Bayes error $e_0^{(m)}$ in labeling normal temporal sequence $\mathbf{x}^{(m)}$, and $e_1^{(t)}$ in labeling abnormal temporal sequence $\mathbf{x}^{(t)}$, where $m = 1, \dots, M$ and $e_0^{(m)}, e_1^{(t)} \in [0, 1]$.

We can rewrite the overall objective function as follows.

$$\begin{aligned}
L &= \operatorname{argmax}_{\mathbf{y}^{(i)}} \sum_{m=1}^M \ln Pr(\mathbf{y}^{(m)}, \mathbf{x}^{(m)}) \\
&= \operatorname{argmax}_{\mathbf{y}^{(i)}} \sum_{m=1}^M \ln [Pr(\mathbf{y}^{(m)}, \mathbf{x}^{(m)} | Y^{(m)} = 1) \times Pr(Y^{(m)} = 1)]^{Y^{(m)}} \\
&\quad + \ln [Pr(\mathbf{y}^{(m)}, \mathbf{x}^{(m)} | Y^{(m)} = 0) \times Pr(Y^{(m)} = 0)]^{(1-Y^{(m)})} \\
&\leq \operatorname{argmax}_{\mathbf{y}^{(i)}} \sum_{m=1}^M (1 - Y^{(m)}) \ln \{(1 - e_0^{(m)}) \times (1 - P)\} \\
&\quad + Y^{(m)} \ln \{(1 - e_1^{(m)}) \times P\}
\end{aligned} \tag{5.23}$$

Due to properties of the parametric distributions of normal and abnormal time segments, as well as the transition probabilities, it is easy to see that Eq. 5.15 is upper-bounded. QED.

Lemma 5.2 (Non-decreasing). The objective function in Eq. 5.15 is non-decreasing in general under the update rules in Algorithm 5.1.

Proof. By separately taking second-order derivatives of Eq. 5.15 with respect to the variables of initial probability π , transition probability distribution \mathbf{A} and emission probability distribution \mathbf{B} , it is easy to see that the three Hessian matrices we obtain are negative semi-definite. Thus, when all but one block are fixed, Eq. 5.15 is a concave function with respect to the free block. In other words, the overall objective function Eq. 5.15 is non-decreasing when we only update the blocks of the initial probability, the transition probability and the emission probability.

The same conclusion could also be reached when we update the segment-level labels with other blocks fixed, as the Viterbi algorithm always returns the optimal labels $\mathbf{y}^{(m)}$ for any input sequence $\mathbf{x}^{(m)}$. On the sequence-level, the BIRAD algorithm firstly scores each temporal sequence by comparing the log likelihood of the sequence being labeled as abnormal vs. normal. Then all the temporal sequences with positive scores are labeled $Y^{(m)} = 1$, and the ones with negative scores are labeled $Y^{(m)} = 0$. At last, BIRAD algorithm corrects the inconsistency between sequence-level and segment-level labels for the following two cases: (1) $Y^{(m)} = 1$ and $\mathbf{y}^{(m)} = 0^{1 \times n^{(m)}}$; (2) $Y^{(m)} = 0$ and $\mathbf{y}^{(m)}$ contains at least one segment-level label as 1. For case 1, it is easy to see Eq. 5.15 increases by $\ln Pr(Y^{(m)} = 0) - \ln Pr(Y^{(m)} = 1)$ after correction of $Y^{(m)}$, where $Pr(Y^{(m)} = 0) \gg Pr(Y^{(m)} = 1)$. For case 2, the overall objective function in Eq. 5.15 keeps the same value after correction of $\mathbf{y}^{(m)}$. In this way, the objective function value with the resulting sequence-level and the associated segment-level labels is no smaller than any alternative label assignments. Therefore, the objective function in Eq. 5.15 is non-decreasing under the update rules of Algorithm 5.1. QED.

Theorem 5.1 (Local Optimum). The proposed BIRAD algorithm converges to the local optimal.

Proof. According to Lemma 5.1 and Lemma 5.2, the objective function is non-decreasing and upper-bounded based on the update rules in Algorithm 5.1. Therefore, the proposed BIRAD algorithm converges to a local optimal. QED.

We also analyze the computational complexity of the BIRAD algorithm in the following theorem.

Theorem 5.2 (Time Complexity). The time complexity of Algorithm 5.1 (with Viterbi algorithm) is $O(LMO)$.

Proof. Let L be the required number of iterations for Algorithm 5.1 to converge. The time complexity of Viterbi Algorithm is $O(N^2O)$, where N is the number of hidden states, and O is the length of a given temporal sequence. In each iteration of Algorithm 5.1, we call Viterbi Algorithm M times. Thus, we have the time complexity of Algorithm 5.1 as $O(LMO)$. QED.

5.3.5 BIRAD-K Algorithm

In some cases, we may be able to start with a few labeled examples, i.e., labeled segments. To accommodate these cases, we introduce a modified semi-supervised version of Algorithm 5.1 named BIRAD-K in Algorithm 5.2.

To be specific, BIRAD-K is given a temporal sequence data set S with only one labeled abnormal segment $X_{AG}^{(AQ)}$, where AQ is the sequence-level index and AG is the segment-level index of $X_{AG}^{(AQ)}$, and prior P as input. Compared with BIRAD, BIRAD-K works better with noisy data, e.g., data with outliers or changing points. The details of BIRAD-K are described in Algorithm 5.2. Step 1 initializes the bi-level hidden states. Step 2 calculates K , which is the number of abnormal temporal sequences in the data set. Step 3 to Step 9 is the BCU process. Identical to BIRAD, we first update the initial probability vector π , transition probability distribution \mathbf{A} and emission probability distribution \mathbf{B} based on the updated labels from the last iteration. Next, we calculate the scores for identifying abnormal temporal sequences in Step 5. Different from BIRAD in Step 6, we label the temporal sequences with the top K scores as abnormal and the rest as normal. Step 7 ensures $Y^{(AQ)}$ and $y_{AG}^{(AQ)}$ are always labeled as 1. Step 8 checks if there is any inconsistency between sequence-level labels and segment-level labels. Finally, in Step 10, BIRAD-K returns all the consistent prediction labels upon convergence.

Algorithm 5.2: Bi-level Rare Temporal Anomaly Detection with K Segments Selected (BIRAD-K)

Require:

Temporal sequence data set $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(M)}$ with only one labeled abnormal segment $x_{AG}^{(AQ)}$
Proportion of abnormal sequences P .

Ensure:

$Y^{(1)}, \dots, Y^{(M)}; \mathbf{y}^{(1)}, \dots, \mathbf{y}^{(M)}$.

- 1: Initialize sequence-level and segment-level labels.
 - 2: Compute $K = m \times P$
 - 3: **while** stopping criterion is not satisfied **do**
 - 4: Update HMM model $\lambda = (\pi, A, B)$ as Step 3 to Step 5 in Algorithm 5.1.
 - 5: Update the segment-level hidden states and anomaly score $score^{(m)}$ for each temporal sequence $\mathbf{x}^{(m)}$ as Step 6 to Step 11 in Algorithm 5.1.
 - 6: Label the temporal sequences with the top K scores as abnormal, i.e., $Y^{(m)} = 1$, and keep the updated prediction labels $y^{(m)}$. Label the remaining temporal sequences as normal, i.e., $Y^{(m)} = 0$, and label the segments in these temporal sequences as normal, i.e., $\mathbf{y}^{(m)} = 0_{1 \times n^{(m)}}$.
 - 7: Correct $Y^{(AQ)}$ or $y_{AG}^{(AQ)}$, if either of them are updated as 0.
 - 8: Check and fix the inconsistency between sequence-level labels and segment-level labels as Step 13 to Step 17 in Algorithm 5.1.
 - 9: **end while**
 - 10: **return** $Y^{(1)}, \dots, Y^{(M)}; \mathbf{y}^{(1)}, \dots, \mathbf{y}^{(M)}$.
-

5.4 EXPERIMENTAL EVALUATION

In this section, we evaluate the performance of our proposed algorithms, i.e., BIRAD and BIRAD-K, on both synthetic and real data sets in comparison with four state-of-the-art unsupervised methods, i.e., NNDB [31], GRADE [33], DPCA- T^2 [127], DPCA- Q [127], and one semi-supervised method, i.e., Semi-DTW-D [128]. The RCD methods, i.e., NNDB and GRADE, require the exact proportion of abnormal time segments in the entire data set. This is the reason why the RCD algorithms produce the same precision and recall rate in the results shown in Figure 5.2. For the two PCA methods, the principal components are associated with 95% of the total variance explanation. Semi-DTW-D is a semi-supervised learning method for time series classification. In the comparison experiments, BIRAD-K and Semi-DTW-D are given a single labeled abnormal segment as training data.

5.4.1 Data Set Description

The synthetic data set is generated from auto-regression model with 3 different coefficients a_1 , a_2 and a_3 . It contains 95 normal temporal sequences and 5 abnormal sequences, and

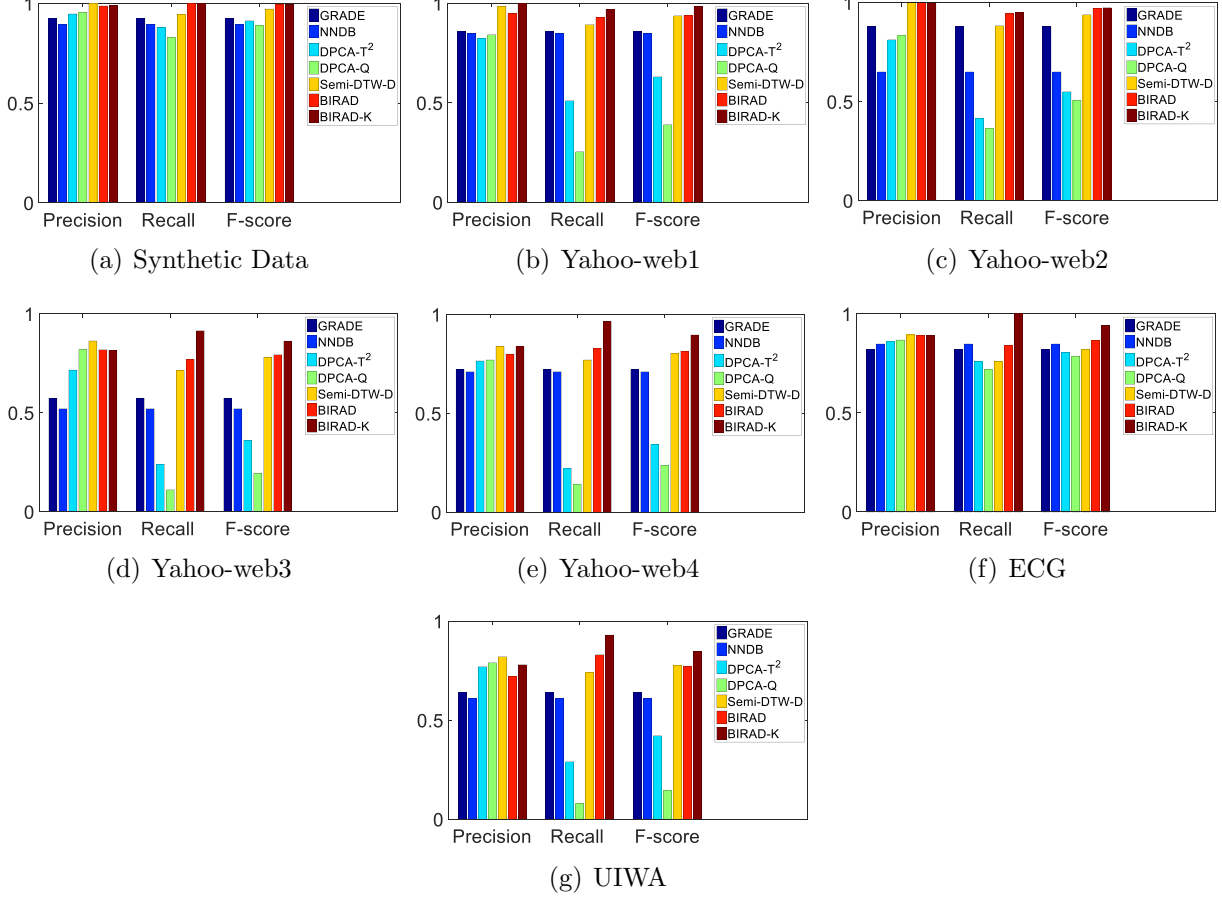


Figure 5.2: Performance Evaluation on Real Data

each temporal sequence consists of 1,000 observations. In normal sequences, all data points fit the model with coefficients a_1 . In abnormal sequences, there are 980 normal data points that fit the model with coefficients a_2 , and 20 abnormal data points that fit the model with coefficients a_3 .

In our experiments, we include 4 temporal data sets from Yahoo! Webscope program ¹. Each data set contains around 80 temporal sequences, and each sequence contains around 1,500 observations. The first data set contains regular anomaly points. The second and third data sets contain periodic outliers. The third and fourth data sets include anomaly points as well as changing points. To match the scenario of our studying problem, each data set is modified as containing 95% synthetic normal sequences and 5% abnormal sequences.

ECG data set is a collection of 100 ECG signal records, which is extracted from a public ECG database ². Each record consists of ~ 300 segments, where each segment corresponds

¹<https://webscope.sandbox.yahoo.com/catalog.php?datatype=s&did=70>

²<https://www.physionet.org/physiobank/database/ptbdb/>

to one certain heart beat pulse. In this data set, 10% signal records are abnormal temporal sequences. Meanwhile, there are around 2% abnormal segments in these abnormal sequences. Our goal is to detect noisy and unstable heart beat pulses, which may be produced due to movements or changes of the environment conditions.

At last, ADL data set [129] comprises information regarding the sensor logs of users’ daily activities during a 35-day interval. The data set is labeled with 10 different daily behaviors, i.e., “*Leaving*”, “*Toileting*”, “*Showering*”, “*Sleeping*”, “*Breakfast*”, “*Lunch*”, “*Dinner*”, “*Snack*”, “*Spare – Time*”, “*Grooming*”. In this experiment, we consider “*Snack*” as the abnormal behavior, which only comprises around 5% of data, and the rest as the normal behaviors. In the end, we aim to identify all the time intervals of “*Snack*” for each user.

5.4.2 Effectiveness Analysis

In this subsection, we evaluate the effectiveness of BIRAD and BIRAD-K over 1 synthetic data set and 6 real data sets based on precision, recall and F-score (defined as $F\text{-score} = 2 \cdot \text{Recall} \cdot \text{Precision} / (\text{Recall} + \text{Precision})$). Notice that, in these experiments, we are able to identify all the abnormal temporal sequences, and the following results are respect to $\mathbf{y}^{(m)}$, $m = 1, \dots, M$.

First, the proposed algorithms are evaluated on the synthetic data set and 4 Yahoo-web data sets, all of which are temporal data sets with anomalies. From Figure 5.2(a) to Figure 5.2(e), we can discover the significant advantages of our proposed methods. The PCA methods always produce very low recall rate, which indicates that the PCA methods may not be suitable for capturing anomalies in the subspace with maximized variance. For NNDB and GRADE, they are very stable for both precision and recall rates, but perform unsatisfied when facing more complex conditions, such as changing points. In Figure 5.2(d), both NNDB and GRADE achieve very low precision and recall rates. This is because they are built upon static methods, thus not effective in handling the temporal variations. Compared with BIRAD and BIRAD-K, we find that Semi-DTW-D always achieves good precision scores, while the recall rates are lower. This is because Semi-DTW-D is designed for time series classification, which only measures the distance between temporal segments, but has not considered the hidden state transition between the adjacent temporal segments. It can be seen that our proposed methods always outperform the other methods, especially in the sense of recall rate and F-score rate. Comparing BIRAD and BIRAD-K, it is shown that BIRAD-K performs slightly better than BIRAD, especially in Figure 5.2(d) and Figure 5.2(e). This implies that BIRAD-K algorithm may be more suited for applications with outliers or changing points.

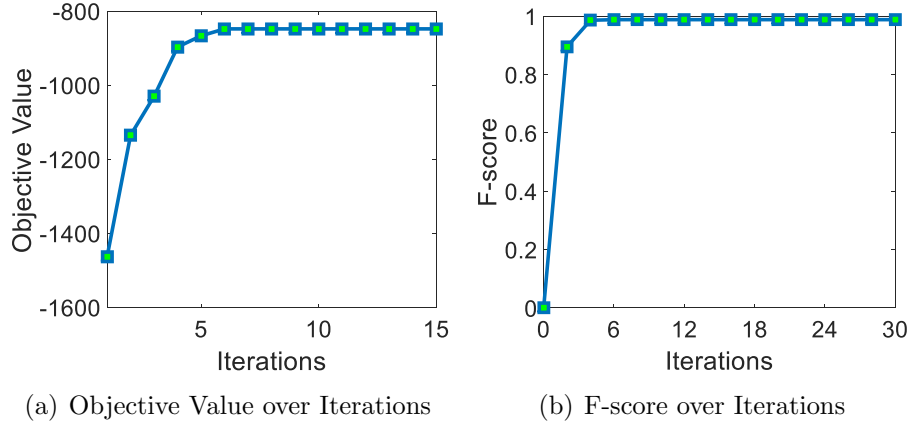


Figure 5.3: Convergence Analysis

Next, two challenging real world problems are considered for anomaly detection. In Figure 5.2(f), we study the problem of anomaly pattern detection on ECG signals. It reveals that all the methods perform very well on this data set, however our BIRAD and BIRAD-K algorithms still outperform the others. In addition, in Figure 5.2(g), we apply our algorithms on wireless sensor networks data so as to detect all the abnormal behaviors. Due to the unremovable randomness in human’s daily behaviors, this problem is more challenging than the previous 5 data sets. In this experiment, lack of the ability to extract temporal information is the main reason why GRADE and NNDB get much lower precision than the others. Compared with BIRAD and BIRAD-K, the PCA methods and Semi-DTW-D get a lower recall rates because it may not be able to precisely catch the rules of state transition, especially in the occurrence of randomness. In general, we have the following observations about our proposed algorithms from these 6 experiments: (1) Both BIRAD and BIRAD-K outperform our 3 baseline algorithms in most cases; (2) BIRAD produces comparable results as BIRAD-K in most cases; (3) BIRAD-K performs modestly better than BIRAD especially in the presence of outliers and changing points.

5.4.3 Convergence and Efficiency Analysis

In this subsection, we first examine the convergence of BIRAD algorithm on the synthetic data set. Figure 5.3(a) illustrates the non-decreasing and upper-bounded characteristics of the objective function when applying BIRAD. In Figure 5.3(b), we present the changes of F-score among different iterations. It is shown that the F-score monotonically increases with objective values and then saturates, implying that the performance improves with increasing objective values.

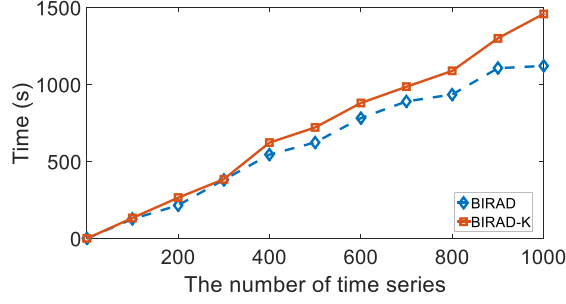


Figure 5.4: Efficiency Analysis on Increasing Number of Temporal Sequences

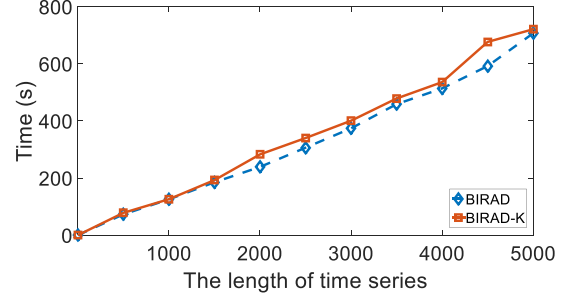


Figure 5.5: Efficiency Analysis on Increasing Length of Temporal Sequences

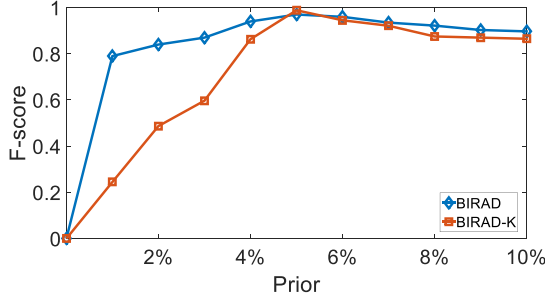


Figure 5.6: Parameter Analysis

Then, we examine the running time and parameter sensitivity of BIRAD and BIRAD-K algorithms. First, we perform our algorithms on a series of synthetic data sets with increasing number of temporal sequences. Let the prior be 5% and the length of each temporal sequence be 1,000, we generate a series of synthetic data sets with increasing number of temporal sequences, from 100 to 1,000. The results are shown in Figure 5.4. After that, we test our algorithms on a series of data sets with increasing sequence length. Different from the experiments in Figure 5.4, we let each data set contain 100 temporal sequences and the prior be 5%, and we generate the series of synthetic data sets with increasing sequence length, from 500 to 5,000. The results are shown in Figure 5.5. From the preceding two experiments, we have the following observations: (1) BIRAD is slightly faster than BIRAD-K; (2) the running time of both algorithms increases linearly in general for both cases, i.e., increasing the sequence length and increasing the number of temporal sequences. we run the experiments with Matlab 2014a on a workstation with four 3.5 GHz CPUs, 256 GB memory and 2 TB disk space.

5.4.4 Parameter Analysis

In this subsection, we empirically study the parameter sensitivity of BIRAD and BIRAD-K algorithms on the synthetic data set. Figure 5.6 shows our analysis results. Notice that

the exact proportion of abnormal temporal sequences is 5% in the data set. For BIRAD-K algorithm, we can see the F-score increases sharply as the prior changes from 1% to 5%. This is because BIRAD-K discovers more abnormal sequences with the increase of input prior ($P < 5\%$). As the prior goes beyond 5%, the F-score of BIRAD-K slightly diminishes but stabilizes near 0.89. The reason is that several normal temporal sequences are included in the group of abnormal sequences, as the input prior exceeds the exact prior. Thus, the input prior would introduce a bias especially when we update the transition probability distribution A and emission probability distribution B . Different from the previous case, the experiments show that the precision rate reduces slightly and the recall rate is kept stable when the input prior ($P > 5\%$) increases. Compared with BIRAD-K, we can see the F-score rates of BIRAD are more stable. This implies that BIRAD is more reliable than BIRAD-K in the cases with unprecise priors.

5.5 SUMMARY

In this chapter, we introduce a novel data mining problem - bi-level rare temporal pattern detection, which aims to fill the gap in the literature by conducting rare category analysis on temporal data. Specifically, we address the challenging case where the labels of the temporal data are highly skewed on both the sequence-level and the segment-level. We formulate the problem as an optimization problem, which maximizes the likelihood of observing the data on both the sequence-level and the segment-level. To solve the optimization problem, we propose an unsupervised algorithm BIRAD and its semi-supervised version BIRAD-K, which iteratively update the model parameters based on the block coordinate update method and return the bi-level labels that are consistent on the sequence-level and the segment-level. The comparison experiments with state-of-the-art techniques demonstrate the effectiveness of our proposed algorithms. In our future work, we will extend the proposed framework to the cases when multiple types of rare temporal patterns exist such that the number of hidden states $N > 2$.

CHAPTER 6: HIGH-ORDER RARE CATEGORY CHARACTERIZATION

6.1 OVERVIEW AND MOTIVATION

Graph analysis has gained in popularity in the past decade, due to the increasing prominence of network data in a variety of real-world applications, from social networks to collaboration networks, from biological systems to e-commerce systems. Graph clustering algorithms represent an important family of tools for studying the underlying structure of networks. While most existing graph clustering algorithms are inherently limited to lower-order connectivity patterns [130, 131, 132], i.e., vertices and edges. They fail to explore the higher-order network structures, which are of key importance in many high impact domains. For example, triangles have been proven to play the fundamental roles in understanding community structures [133]; a multi-hop loop structure may indicate the existence of money laundering activities in financial networks [134]; a star-shaped structure may correspond to a set of synthetic identities in personally identifiable information (PII) networks of bank customers [135].

Despite its importance, a key challenge associated with finding structure-rich subgraphs is the prohibitive computational cost. Many existing works on high-order graph clustering are either based on spectral graph theory [136, 137], or estimating the frequency of the high-order connectivity patterns [138, 139]. These methods may not be scalable to large-scale networks especially when modeling various complex network structures, such as loop-shaped structures, star-shaped structures and cliques. In this chapter, we aim to answer the following open questions. First (**Q1. Model**), it is not clear that how to model various types of high-order connectivity patterns (e.g., triangles, loops and stars) that exist in the given graphs. Some motif-based graph clustering algorithms [136, 140, 141] have been proposed recently, while they are mainly designed for the 3^{rd} -order network structures (e.g., triangle). Second (**Q2. Algorithm**), how should we design a fast graph clustering algorithm that produces *structure-preserving* graph partitions in the massive real-world networks? This question has been largely overlooked in the previous studies. Third (**Q3. Generalization**), how can we generalize our algorithm to solve real-world problems on various types of graphs such as signed graphs, bipartite graphs, and multi-partite graphs?

To address above challenges, we propose a novel high-order *structure-preserving* graph clustering framework named HOSGRAP, which partitions the graph into *structure-rich* clusters in polylogarithmic time with respect to the number of edges in the graph. In particular, we start with a generic definition of high-order conductance, and define the high-

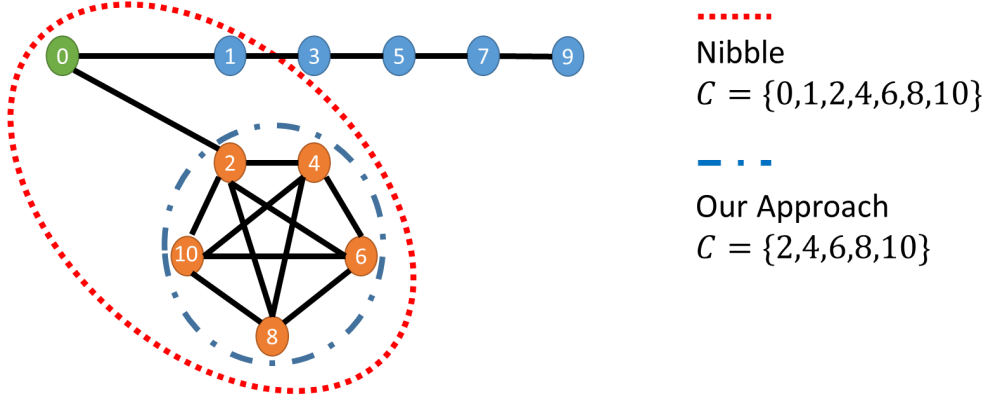


Figure 6.1: A synthetic network where vertex 0 is connected with two kinds of network structures: clique and line. The local clusters found by our approach (within the blue dash-dot line) and the Nibble algorithm [142] (within the red dotted line) with the same initial vertex, i.e., vertex 0, where algorithm is conducted on the basis of 3-node line (illustrated in Table 6.1).

order diffusion core, which is based on a high-order random walk induced by *user-specified high-order* network structure. Then, inspired by the family of local graph clustering algorithms [142, 143, 144] for efficiently identifying a low-conductance cut without exploring the entire graph, we generalize the key idea to high-order network structures and propose our fast high-order graph clustering framework HOSGRAP, which runs in polylogarithmic time with respect to the number of edges in the graph. It starts with a seed vertex and iteratively conducts high-order random walks [140, 145] to explore its neighborhood until a subgraph with a small high-order conductance is found. Our algorithm operates on the tensor representation of graph data which allows the users to specify what kind of network structures to be preserved in the returned cluster. In addition, we provide analyses regarding the effectiveness and efficiency of the proposed algorithm. Furthermore, we generalize our proposed HOSGRAP algorithm to the scenarios when the given networks are signed networks, bipartite networks and multi-partite networks. At last, we perform extensive experiments to demonstrate the effectiveness and the efficiency of the proposed methods. Figure 6.1 compares the clusters returned by our method and the Nibble algorithm [142], which shows that our method is better at partitioning a subgraph with the rich *user-specified high-order* network structure.

The main contributions of the chapter are summarized below.

1. **Problem.** We formally define the problems of *Structure-Preserving Local Graph Cut* as well as *Structure-Preserving Graph Clustering*, and identify their unique challenges arising from real applications.

2. **Algorithms and Analysis.** We propose a family of algorithms, i.e., HOSPLOC and HOSGRAP, to effectively identify structure-rich clusters with a *polylogarithmic* time complexity. Theoretical analyses show that our proposed algorithms can capture near-optimal structure-rich clusters under mild conditions.
3. **Generalization and Application.** We generalize our proposed HOSPLOC and HOSGRAP algorithms from binary graphs to signed networks, bipartite networks, and multi-partite networks in real applications.
4. **Evaluation.** Extensive experimental results on synthetic and real networks demonstrate the performance of the proposed HOSPLOC and HOSGRAP algorithms in terms of effectiveness, scalability, and parameter sensitivity.

The rest of our chapter is organized as follows. A brief overview of related literature is presented in Section 6.2, followed by the introduction of notation and preliminaries in Section 6.3. In Section 6.4, we present the proposed HOSGRAP algorithm as well as the analyses regarding its effectiveness and efficiency. Then, we introduce its generalizations and applications in Section 6.5. Experimental results are presented in Section 6.6 before we conclude the chapter in Section 6.7.

6.2 RELATED WORK

6.2.1 Local Spectral Clustering on Graphs

Nowadays, large-scale networks data appear in a broad spectrum of disciplines, from social networks [11, 146] to collaborative networks [13, 147], from rare category detection [2, 4, 6, 7, 23, 29, 148] to community detection [149, 150, 151, 152, 153], from data augmentation [12, 30, 154] to crowd-sourcing [155, 156]. Local spectral clustering techniques provide a simple, efficient time alternative to recursively identify a local sparse cut C with an upper-bounded conductance. In [142], the authors introduce an almost-linear Laplacian linear solver and a local clustering algorithm, i.e., Nibble, which conducts cuts that can be combined with balanced partitions. In [143, 144], the authors extend Nibble algorithm [142] by using personalized PageRank vector to produce cuts with less running time on undirected and directed graphs. More recently, [157] proposes a local graph clustering algorithm with the same guarantee as the Cheeger inequalities, of which time complexity is slightly super linear in the size of the partition. In [158], the authors introduce randomized local partitioning algorithms that find sparse cuts by simulating the volume-biased evolving set process. To

model the high-order connectivity patterns, [16] proposes a local graph clustering algorithm named HOSPLOC that identifies the structure-rich clusters by exploring the high-order structures in the neighborhood of the initial vertex in the given graph. Meanwhile, [159] develops a motif-based local graph clustering algorithm that approximately finds clusters with the minimal motif conductance, a generalization of the conductance metric for network motifs. Later on, in [160], the authors approach the problem the problem of discovering user-guided clustering in heterogeneous information networks, by transcribe the high-order interaction signals (i.e., network motifs) based on a non-negative tensor factorization methods. More recently, researchers aim to generalize HOSPLOC to the dynamic setting and develop a series of algorithms to compute [17] and track [10] “structure-rich” clusters in temporal networks. However, to my best of knowledge, this chapter is the first local clustering framework that focuses on modeling high-order network structures and partitions the graph into structure-rich clusters in polylogarithmic time with respect to the number of edges in the graph.

6.2.2 High-order Markov Chain Models

The o^{th} order Markov chain S describes a stochastic process that satisfies [145]

$$\begin{aligned} &Pr(S_{t+1} = i_1 | S_t = i_2, \dots, S_{t-o+1} = i_{o+1}, \dots, S_1 = i_{t+1}) \\ &= Pr(S_{t+1} = i_1 | S_t = i_2, \dots, S_{t-o+1} = i_{o+1}) \end{aligned} \quad (6.1)$$

where i_1, \dots, i_{t+1} denote the set of states associated with different time stamps. Specifically, this means the future state only depends on the past o states. There are many cases that one would like to model observed data as a high-order Markov chain in different real-world problems, such as airport travel flows [161], web browsing behavior [162] and wind turbine design [163]. To solve these problems, many previous works [163, 164, 165] approximate the limiting probability distribution of high-order Markov chain as a linear combination of transition probability matrix.

More recently, in [140], the authors introduce a rank-1 approximation of high-order Markov chain limiting distribution and propose a recursive algorithm to compute it. Later on, [145] introduces a computationally tractable approximation of the high-order PageRank named multi-linear PageRank, where the underlying stochastic process is a vertex-reinforced random walk. In [166], the authors introduce a novel stochastic process, i.e., spacey random walk, whose stationary distribution is given by the tensor eigenvector, and show the convergence properties of these dynamics. In [136, 141], the authors propose the similar spectral

clustering frameworks that allow for modeling third-order network structures and conduct partition while preserving such structures on the given graph. Followed by [136], [137] proposes a tensor spectral co-clustering method by modeling higher-order data with a novel variant of a higher-order Markov chain, i.e., the super-spacey random walk. Compared to the existing high-order Markov chain models, we propose a novel scalable local clustering algorithm that can identify clusters with a small conductance and also preserve the *user-specified high-order* network structures in a *polylogarithmic* time complexity. Moreover, we also provide provable theoretical bounds on the effectiveness and efficiency of the proposed high-order graph clustering framework.

6.3 PRELIMINARIES

In this section, we formally define the structure-preserving graph cut and the structure-preserving graph clustering problems. Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} consists of n vertices, and \mathcal{E} consists of m edges, we let $A \in \mathbb{R}^{n \times n}$ denote the adjacency matrix of graph \mathcal{G} , $D \in \mathbb{R}^{n \times n}$ denote the diagonal matrix of vertex degrees, and $d(v) = D(v, v)$ denote the degree of vertex $v \in \mathcal{V}$. The transition matrix of a random walk on graph \mathcal{G} is

$$M = (A^T D^{-1} + I)/2 \quad (6.2)$$

where $I \in \mathbb{R}^{n \times n}$ is an identity matrix. For convenience, we define the indicator vector $\chi_C \in \{0, 1\}^n$ as follows.

$$\chi_C(v) = \begin{cases} 1 & v \in C \\ 0 & \text{Otherwise} \end{cases} \quad (6.3)$$

In particular, the initial distribution of a random walk starting from vertex v could be denoted as χ_v .

The volume of a subset $C \subseteq \mathcal{V}$ is defined as the summation of vertex degrees in C , i.e., $\mu(C) = \sum_{v \in C} d(v)$. We let \bar{C} be the complementary set of C , i.e., $\bar{C} = \{v \in \mathcal{V} | v \notin C\}$. The conductance [167] of subset $C \subseteq \mathcal{V}$ is therefore defined as

$$\Phi(C) = \frac{|\mathcal{E}(C, \bar{C})|}{\min(\mu(C), \mu(\bar{C}))} \quad (6.4)$$

where $\mathcal{E}(C, \bar{C}) = \{(u, v) | u \in C, v \in \bar{C}\}$, and $|\mathcal{E}(C, \bar{C})|$ denotes the number of edges in $\mathcal{E}(C, \bar{C})$. Besides, we represent the elements in a matrix or a tensor using the convention similar to Matlab, e.g., $M(i, j)$ is the element at the i^{th} row and j^{th} column of the matrix

M , and $M(i, :)$ is the i^{th} row of M , etc.

We let \mathbb{N} denote the k^{th} -order user-defined structure. Table 6.1 summarizes the examples of network structures \mathbb{N} of different orders and the corresponding Markov chain. Notice that the order of the network structure is different from the order of the Markov chain (or random walk). For example, the edges in \mathcal{E} are considered as the 2nd-order network structures, and they correspond to the 1st-order Markov Chain (random walk) due to the matrix representation of \mathcal{E} . We use k to denote the order of the network structure \mathbb{N} . As what will be explained next, the k^{th} -order network structures correspond to the $(k - 1)^{\text{th}}$ -order Markov chain (random walk).





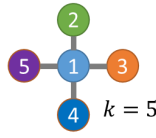
\mathbb{N}	Illustration	Order of \mathbb{N}	Markov Chain	Random Walks	Graph Clustering Algorithms
Vertex		1 st -order	0 th -order	N/A	N/A
Edge		2 nd -order	1 st -order	1 st -order	1 st -order
3-node Line		3 rd -order	2 nd -order	2 nd -order	2 nd -order
Triangle					
k -node Star		k^{th} -order	$(k - 1)^{\text{th}}$ -order	$(k - 1)^{\text{th}}$ -order	$(k - 1)^{\text{th}}$ -order

Table 6.1: Network Structures \mathbb{N} and Markov Chains.

With the above notion, our problems can be formally defined as follows:

Problem 6.1. Structure-Preserving Local Graph Cut

Input: (i) an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, (ii) a user-defined network structure \mathbb{N} , (iii) the initial vertex v .

Output: a local cluster C that largely preserves the user-defined structures \mathbb{N} .

Problem 6.2. Structure-Preserving Graph Clustering

Input: (i) an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, (ii) a user-defined network structure \mathbb{N} , (iii) the number of clusters.

Output: a graph partition D that largely preserves the user-defined structures \mathbb{N} in the returned clusters.

6.4 ALGORITHM

In the previous section, we introduced the notations and problem definitions. Now, we generalize the idea of truncated local clustering to produce clusters that preserve the *user-specified high-order* network structures. We start by reviewing the basics of the Nibble algorithm for local clustering on graphs [142], which pave the way for discussion of the proposed structure-preserving graph clustering algorithm. Then, we introduce the adjacency tensor and the associated transition tensor based on the *user-specified high-order* network structures, followed by the discussion on the stationary distribution of high-order random walk. After that, we introduce the definitions of high-order conductance and high-order diffusion core. Finally, we present our proposed HOSPLOC and HOSGRAP algorithms with theoretical analyses in terms of the effectiveness and efficiency.

6.4.1 Background: Truncated Local Graph Clustering Algorithm

Given an undirected graph \mathcal{G} and a parameter $\phi > 0$, to find a cluster C from \mathcal{G} such that $\Phi(C) \leq \phi$ or to determine no such C exists is an NP-complete problem [168]. Nibble algorithm [142] is one of the earliest attempts to partition a graph with a bounded conductance in polylogarithmic time. Starting from a given vertex, Nibble provably finds a local cluster in time $(O(2^b \log^6 m)/\phi^4))$, where b is a constant which controls the lower bound of the output volume. This is proportional to the size of the output cluster. The key idea behind Nibble is to conduct truncated random walks by using the following truncation operator

$$[q]_\epsilon(u) = \begin{cases} q(u) & \text{if } q(u) \geq d(u)\epsilon \\ 0 & \text{Otherwise} \end{cases} \quad (6.5)$$

where $q \in \mathbb{R}^n$ is the distribution vector over all the vertices in the graph, and ϵ is the truncation threshold that can be computed as follows [142]

$$\epsilon = \frac{1}{(1800 \cdot (l+2)t_{last}2^b)} \quad (6.6)$$

where l can be computed as $l = \lceil \log_2(\mu(\mathcal{V})/2) \rceil$, and t_{last} can be computed as

$$t_{last} = (l+1) \left\lceil \frac{2}{\phi^2} \ln \left(c_1(l+2)\sqrt{\mu(\mathcal{V})/2} \right) \right\rceil \quad (6.7)$$

Then, Nibble applies the vector-based partition method [142, 169, 170] that sorts the

probable nodes based on the ratio of function I_x to produce a low conductance cut. To introduce function I_x mathematically, we first define $S_j(q)$ to be the set of top j vertices u that maximizes $q(u)/d(u)$. That is $S_j(q) = \{\pi(1), \dots, \pi(j)\}$, where π is the permutation that follows $\frac{q(\pi(i))}{d(\pi(i))} \geq \frac{q(\pi(i+1))}{d(\pi(i+1))}$. In addition, we let $\lambda_j(q) = \sum_{u \in S_j(q)} d(u)$ denote the volume of the set $S_j(q)$. Finally, the function I_x is defined as follows

$$I_x(q, \lambda_j(q)) = \frac{q(\pi(j))}{d(\pi(j))}. \quad (6.8)$$

6.4.2 Adjacency Tensor and Transition Tensor

For an undirected graph \mathcal{G} , the corresponding adjacency matrix A could be considered as a matrix representation of the existing edges on \mathcal{G} . If each vertex in graph \mathcal{G} corresponds to a distinct state, we can interpret the transition matrix M as the transition matrix of the 1st-order Markov chain. Specifically, the transition probability between vertex i and vertex j is given by $M(i, j) = Pr(S_{t+1} = i | S_t = j)$. Moreover, if M is stochastic, irreducible and aperiodic [171], we can compute a positive and unique vector $\bar{x} = M\bar{x}$, where $\bar{x} \in \mathbb{R}^n$ is the limiting or stationary probability distribution of the random walk.

However, in many real applications, we may want to explore and capture more complex and high-order network structures. To model the *user-specified* network structure \mathbb{N} , we introduce the definition of adjacency tensor \mathbf{T} and the transition tensor \mathbf{P} to represent the high-order random walk induced by the high-order network structures \mathbb{N} .

Definition 6.1 (Adjacency Tensor). Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the k^{th} -order network structure \mathbb{N} on \mathcal{G} could be represented in a k -dimensional adjacency tensor \mathbf{T} as follows

$$T(i_1, i_2, \dots, i_k) = \begin{cases} 1 & \{i_1, i_2, \dots, i_k\} \subseteq \mathcal{V} \text{ and form } \mathbb{N}. \\ 0 & \text{Otherwise.} \end{cases} \quad (6.9)$$

Definition 6.2 (Transition Tensor). Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and the adjacency tensor \mathbf{T} for the k^{th} -order network structure \mathbb{N} , the corresponding transition tensor \mathbf{P} could be computed as

$$P(i_1, i_2, \dots, i_k) = \frac{T(i_1, i_2, \dots, i_k)}{\sum_{i_1=1}^n T(i_1, i_2, \dots, i_k)} \quad (6.10)$$

By the above definition, we have $\sum_{i_1} P(i_1, \dots, i_k) = 1$. Therefore, if each vertex in \mathcal{G} is a distinguishable state, we can interpret the k^{th} -order transition tensor \mathbf{P} as a $(k-1)^{\text{th}}$ -order Markov chain (random walk), i.e., $Pr(S_{t+1} = i_1 | S_t = i_2, \dots, S_{t-k+2} = i_k) = P(i_1, \dots, i_k)$.

Intuitively, if $i_1 \neq i'_1$, and they both form \mathbb{N} together with i_2, \dots, i_k , then the probabilities of the next state being i_1 and being i'_1 are the same given $S_t = i_2, \dots, S_{t-k+2} = i_k$. Notice that the transition matrix M of a lazy random walk defined in Subsection 4.1 can be considered as a special case of Definition 6.2 with the 2nd-order network structure \mathbb{N} , if we allow self-loops.

6.4.3 Stationary Distribution

For the k^{th} -order network structure \mathbb{N} and the corresponding $(k-1)^{\text{th}}$ -order random walk with transition tensor \mathbf{P} , if the stationary distribution \mathbf{X} exists, where \mathbf{X} is a $(k-1)$ -dimensional tensor, then it satisfies [145]

$$X(i_1, i_2, \dots, i_{k-1}) = \sum_{i_k} P(i_1, i_2, \dots, i_k) X(i_2, \dots, i_k). \quad (6.11)$$

where $X(i_1, \dots, i_{k-1})$ denotes the probability of being at states i_1, \dots, i_{k-1} in consecutive time steps upon convergence of the random walk, and $\sum_{i_1, \dots, i_{k-1}} X(i_1, \dots, i_{k-1}) = 1$.

However, for this system, storing the stationary distribution requires $O(n^{(k-1)})$ space complexity. For the sake of computational scalability, in high-order random walks, a commonly held assumption is ‘rank-one approximation’ [136, 140], i.e.,

$$X(i_2, \dots, i_k) = q(i_2) \dots q(i_k) \quad (6.12)$$

where $q \in \mathbb{R}_+^{n \times 1}$ with $\sum_i q(i) = 1$. Then, we have $\sum_{i_2, \dots, i_k} P(i_1, \dots, i_k) q(i_2) \dots q(i_k) = q(i_1)$. In this way, the space complexity of the stationary distribution of high-order random walk is reduced to $O(n)$. Although q is an approximation of the true stationary distribution of the high-order random walk, [140] theoretically demonstrates the convergence and effectiveness of the nonnegative vector q if \mathbf{P} satisfies certain properties.

Following [136, 140], in this chapter, we also adopt ‘rank-one approximation’ and assume the stationary distribution of the high-order random walk satisfies Eq. 6.12. To further simplify the notation, we let \bar{P} denote the $(k-2)$ -mode unfolding matrix of the k -dimensional transition tensor P . Thus, the $(k-1)^{\text{th}}$ -order random walk satisfies:

$$q = \bar{P}(q \otimes \dots \otimes q) \quad (6.13)$$

where \otimes denotes the Kronecker product symbol. For example, for the third-order network structure \mathbb{N} (e.g., triangle), the transition tensor $\mathbf{P} \in \mathbb{R}^{n \times n \times n}$ can be constructed based on Definition 6.2. Then, the 1-mode unfolding matrix \bar{P} of \mathbf{P} can be written as $\bar{P} = [P($

, :, 1), P(:, :, 2), ..., P(:, :, n)]. where $\bar{P} \in \mathbb{R}^{n \times n^2}$. In this way, the associated second-order random walk with respect to the triangle network structure satisfies $q = \bar{P}(q \otimes q)$.

6.4.4 High-Order Conductance

Given a high-order network structure \mathbb{N} , it is usually the case that the user would like to find a local cluster C on the graph \mathcal{G} such that: (1) C contains a rich set of network structures \mathbb{N} ; (2) by partitioning all the vertices into C and \bar{C} , we do not break many such network structures. For example, in financial fraud detection, directed loops may refer to money laundering activities. In this case, we would like to ensure the partition preserves rich directed loops inside the cluster and breaks such structure as less as possible. It is easy to see that the traditional definition of the conductance $\Phi(C)$ introduced in Subsection 4.1 does not serve this purpose. Therefore, we introduce the following generalized definition of conductance to preserve user-defined high-order network structure \mathbb{N} .

Definition 6.3 (k^{th} -order Conductance). For any cluster C in graph \mathcal{G} and the k^{th} -order network structure \mathbb{N} , the k^{th} -order conductance $\Phi(C, \mathbb{N})$ is defined as

$$\Phi(C, \mathbb{N}) = \frac{\text{cut}(C, \mathbb{N})}{\min\{\mu(C, \mathbb{N}), \mu(\bar{C}, \mathbb{N})\}} \quad (6.14)$$

where $\text{cut}(C, \mathbb{N})$ denotes the number of network structures broken due to the partition of \mathcal{G} into C and \bar{C} , i.e.,

$$\begin{aligned} \text{cut}(C, \mathbb{N}) = & \sum_{i_1, \dots, i_k \in \mathcal{V}} T(i_1, \dots, i_k) - \sum_{i_1, i_2, \dots, i_k \in C} T(i_1, \dots, i_k) \\ & - \sum_{i_1, \dots, i_k \in \bar{C}} T(i_1, \dots, i_k) \end{aligned} \quad (6.15)$$

and $\mu(C, \mathbb{N})$ ($\mu(\bar{C}, \mathbb{N})$) denotes the total number of network structures \mathbb{N} incident to the vertices within C (\bar{C}), i.e.,

$$\begin{aligned} \mu(C, \mathbb{N}) &= \sum_{i_1 \in C; i_2, \dots, i_k \in \mathcal{V}} T(i_1, i_2, \dots, i_k) \\ \mu(\bar{C}, \mathbb{N}) &= \sum_{i_1 \in \bar{C}; i_2, \dots, i_k \in \mathcal{V}} T(i_1, i_2, \dots, i_k). \end{aligned} \quad (6.16)$$

Claim 6.1. Definition 6.3 provides a generic definition of network conductance with respect to any network structure, and it subsumes existing measures of network conductance. In

particular.

- When \mathbb{N} represents edges, $\Phi(C, \mathbb{N})$ is twice the traditional conductance $\Phi(C)$ introduced in Subsection 4.1.
- When \mathbb{N} represents triangles, $\Phi(C, \mathbb{N})$ is the same as the ‘high-order conductance’ ϕ_3 introduced in [136].

6.4.5 High-Order Diffusion Core

Similar to the Nibble algorithm, we are given a seed vertex v , and our goal is to find a cluster C containing or near v without looking at the whole graph. The main advantage of our proposed work is that, given the *user-specified high-order* network structure \mathbb{N} , we are able to produce a local cluster that preserves such structure within the cluster C and does not break many such structures by partitioning the graph into C and \bar{C} .

To this end, we perform high-order random walk with transition tensor \mathbf{P} defined in Definition 6.2, starting from the seed vertex v . Let $q^{(t)}$ denote the distribution vector over all the vertices after the t^{th} iteration of the high-order random walk. Ideally, a seed vertex chosen within a cluster C with low conductance should lead to the discovery of this cluster. However, as pointed out in [142], for the 2nd-order network structure and the associated 1st-order random walk, if the vertices within the cluster are more strongly attached to vertices outside the cluster than inside it, they may not be good candidates for the seed, as the random walk will have a relatively high chance of escaping the cluster after a few iterations. Therefore, they propose the definition of the diffusion core to characterize the subset of vertices within the cluster, such that the random walks starting from such vertices stay inside the cluster for a long time. Here, we generalize the definition of a diffusion core to high-order network structures as follows.

Definition 6.4 (k^{th} -Order ξ -Diffusion Core). For any cluster C , we define $C^{k,\xi} \in C$ to be the k^{th} -order ξ -diffusion core of C , such that

$$\chi_{\bar{C}^{k,\xi}}^T q^{(t)} \leq \xi \frac{\text{cut}(C, \mathbb{N})}{\mu(C, \mathbb{N})} \quad (6.17)$$

where $q^{(t)}$ denotes the diffusion distribution of t -step high-order random walks, and ξ is a positive constant that controls the compactness of the diffusion core.

Note that the left hand side of Eq. 6.17, $\chi_{\bar{C}^{k,\xi}}^T q^{(t)}$, represents the probability that a high-order random walk terminates outside the cluster C after t steps, which is also called the

escaping probability of the cluster C . On the right hand side of Eq. 6.17, the numerator could be considered as the total number of the k^{th} -order random walk paths to escape cluster C , while the denominator could be regarded as the total number of the k^{th} -order random walk paths starting from C . It is easy to see that $\chi_{\bar{C}^{k,\xi}}^T q^{(t)}$ is positively correlated with $\frac{\text{cut}(C, \mathbb{N})}{\mu(C, \mathbb{N})}$. Since, for a given C , $\chi_{\bar{C}^{k,\xi}}^T q^{(t)}$ is a computable constant, we consider Eq. 6.17 as the compactness constraint for the k^{th} -order ξ -diffusion core $C^{k,\xi} \in C$.

Proposition 6.1. For any cluster C and the k^{th} -Order ξ -diffusion core $C^{k,\xi} \in C$, we have

$$\chi_{\bar{C}^{k,\xi}}^T q^{(t)} \leq \xi \Phi(C, \mathbb{N}). \quad (6.18)$$

Proof. Given a cluster $C \in \mathcal{V}$ and a k^{th} -order network structure \mathbb{N} , the corresponding k^{th} -order conductance can be computed as $\Phi(C, \mathbb{N}) = \frac{\text{cut}(C, \mathbb{N})}{\min\{\mu(C, \mathbb{N}), \mu(\bar{C}, \mathbb{N})\}}$. Obviously, we can divide the proof into the following two cases.

Case 1 : when $\mu(C, \mathbb{N}) \geq \mu(\bar{C}, \mathbb{N})$, $\Phi(C, \mathbb{N}) = \frac{\text{cut}(C, \mathbb{N})}{\mu(\bar{C}, \mathbb{N})} \geq \frac{\text{cut}(C, \mathbb{N})}{\mu(C, \mathbb{N})}$.

Case 2 : when $\mu(C, \mathbb{N}) < \mu(\bar{C}, \mathbb{N})$, $\Phi(C, \mathbb{N}) = \frac{\text{cut}(C, \mathbb{N})}{\mu(C, \mathbb{N})}$.

Thus, we have $\Phi(C, \mathbb{N}) \geq \frac{\text{cut}(C, \mathbb{N})}{\mu(C, \mathbb{N})}$. Meanwhile, by Definition 6.4, it turns out that $\chi_{\bar{C}^{k,\xi}}^T q^{(t)} \leq \xi \frac{\text{cut}(C, \mathbb{N})}{\mu(C, \mathbb{N})} \leq \xi \Phi(C, \mathbb{N})$. QED.

6.4.6 High-Order Structure-Preserving Graph Cut

Basically, the proposed HOSPLOC could be decomposed into three main steps: (1) approximately compute the distribution of high-order random walk starting at any vertex from which the walk does not mix rapidly; (2) truncate all small entries in $q^{(t)}$ to 0, thus we can limit the computation to the neighborhood of the seed; (3) apply the vector-based graph partition method [142, 169, 170] to search for a *structure-rich* cut with a small conductance.

Now, we are ready to present our proposed HOSPLOC algorithm. The given inputs are the transition tensor \mathbf{P} , the transition matrix M , the seed vertex v , the conductance upper-bound ϕ , the maximum iteration number t_{\max} , and the constants b , c_1 , ξ . Note that constant b controls the volume lower bound of the returned set C , i.e., $2^b \leq \mu(C)$, and c_1 is a constant which guarantees that the elements in C have a large probability of staying within C . Step 1 to Step 4 are the initialization process. Step 1 constructs unfolding matrix \bar{P} of the transition tensor \mathbf{P} . Step 2 to Step 4 compute the truncation constant ϵ and the truncated initial distributions vectors $r^{(m)}$, $m = 1, \dots, k-1$. The iterative process between Step 5 and Step 16 aims to identify the proper high-order local cluster C : Step 6 calculates the updated distribution over all the vertices in current iteration; Step 7 calculates the truncated local distribution $r^{(t)}$; the iterative process stops when it finds a proper cluster which satisfies the

Algorithm 6.1: High-Order Structure-Preserved Local Cut (HOSPLOC)

Require:

- (1) k^{th} -order transition tensor \mathbf{P} ,
- (2) Transition matrix M ,
- (3) Initial vertex v ,
- (4) Conductance upper bound ϕ ,
- (5) Maximum iteration number t_{\max} ,
- (6) Parameters b, c_1, ξ .

Ensure:

Local cluster C .

- 1: Construct the unfolding matrix \bar{P} of the transition tensor \mathbf{P} .
 - 2: Compute constant ϵ based on Eq. 6.6.
 - 3: Set initial distribution vectors $q^{(t)} = M^{(t-1)}\chi_v$, where $t = 1, \dots, k-1$.
 - 4: Compute truncated initial local distribution vectors $r^{(t)} = [q^{(t)}]_\epsilon$, $t = 1, \dots, k-1$.
 - 5: **for** $t = k : t_{\max}$ **do**
 - 6: Update distribution vector $q^{(t)} = \bar{P}(r^{(t-1)} \otimes \dots \otimes r^{(t-k+1)})$.
 - 7: Update truncated distribution vectors $r^{(t)} = [q^{(t)}]_\epsilon$.
 - 8: **if** there exists a j such that:
 - 9: (a) $\Phi(S_j(q^{(t)})) \leq \phi$,
 - 10: (b) $2^b \leq \lambda_j(q^{(t)})$,
 - 11: (c) $I_x(q^{(t)}, 2^b) \geq \frac{\xi}{c_1(l+2)2^b}$. **then**
 - 12: return $C = S_j(q^{(t)})$ and quit.
 - 13: **else**
 - 14: Return $C = \emptyset$.
 - 15: **end if**
 - 16: **end for**
-

three constraints in Step 9 to Step 11, where condition (a) guarantees that the conductance of C is upper-bounded by ϕ , condition (b) ensures that the volume of C is lower-bounded by 2^b , and condition (c) enforces that elements in C have a large probability mass.

Next, we analyze the proposed HOSPLOC algorithm in terms of effectiveness and efficiency. Regarding the effectiveness, we will show that for any cluster C , if the seed vertex comes from the k^{th} -order ξ -diffusion core, i.e., $v \in C^{k,\xi}$, then the non-empty set C' returned by HOSPLOC has a large overlap with C . To be specific, we have the following theorem for the effectiveness of HOSPLOC.

Theorem 6.1 (Effectiveness of HOSPLOC). Let C be a cluster on graph \mathcal{G} such that $\Phi(C, \mathbb{N}) \leq \frac{1}{c_2(l+2)}$, where $2c_1 \leq c_2$. If HOSPLOC runs with starting vertex $v \in C^{k,\xi}$ and returns a non-empty set C' , then we have $\mu(C' \cap C) \geq 2^{b-1}$.

Proof. Let $q^{(t)}$, $t \leq t_{\max}$, be the distribution of t -step high-order random walk when the set

$C' = S_j(q^{(t)})$ is obtained. Then, based on Proposition 6.1, we have the following inequality

$$\chi_{\bar{C}}^T q^{(t)} \leq \chi_{\bar{C}^k, \xi}^T q^{(t)} \leq \xi \Phi(C, \mathbb{N}) \leq \frac{\xi}{c_2(l+2)}. \quad (6.19)$$

In Step 11 of Algorithm 6.1, condition (c) guarantees that

$$I_x(u) = \frac{q^{(t)}(u)}{d(u)} \geq \frac{\xi}{c_1(l+2)2^b} \quad (6.20)$$

where $u \in S_j(q^{(t)})$. Since $d(u) \geq 0$ and $c_1(l+2)2^b \geq 0$, we can infer the following inequality from Eq. 6.20

$$d(u) \leq \frac{1}{\xi} c_1(l+2)2^b q^{(t)}(u). \quad (6.21)$$

Let j' be the smallest integer such that $\lambda_{j'}(q^{(t)}) \geq 2^b$. In Step 10 of Algorithm 6.1, condition (b) guarantees that $j' \leq j$. By Eq. 6.19 and Eq. 6.21, we have

$$\begin{aligned} & \mu(S_{j'}(q^{(t)}) \cap \bar{C}) \\ &= \sum_{u \in S_{j'}(q^{(t)}) \cap \bar{C}} d(u) \\ &\leq \sum_{u \in S_{j'}(q^{(t)}) \cap \bar{C}} \frac{1}{\xi} c_1(l+2)2^b q^{(t)}(u) \\ &\leq \frac{1}{\xi} c_1(l+2)2^b (\chi_{\bar{C}}^T q^{(t)}) \\ &\leq \frac{\xi c_1(l+2)2^b}{\xi c_2(l+2)} \leq 2^{b-1}. \end{aligned} \quad (6.22)$$

Due to $2^b \leq \lambda_{j'}(q^{(t)})$, it turns out that $\mu(S_{j'}(q^{(t)}) \cap C) \geq 2^{b-1}$. Since $j \geq j'$, we have the final conclusion

$$\mu(S_j(q^{(t)}) \cap C) \geq \mu(S_{j'}(q^{(t)}) \cap C) \geq 2^{b-1}. \quad (6.23)$$

QED.

Regarding the efficiency of HOSPLOC, we provide the following lemma to show the *polylogarithmic* time complexity of HOSPLOC with respect to the number of edges in the graph.

Lemma 6.1 (Efficiency of HOSPLOC). Given graph \mathcal{G} and the k^{th} -order network structure \mathbb{N} , $k \geq 3$, the time complexity of HOSPLOC is bounded by $O\left(t_{\max} \frac{2^{bk}}{\phi^{2k}} \log^{3k} m\right)$.

Proof. To bound the running time of HOSPLOC, we first show that each iteration in Algo-

rithm 6.1 takes time $O(\frac{1}{\epsilon^k})$. Instead of conducting dense vector multiplication or Kronecker product, we track the nonzeros in both matrixes and vectors. Here, we let \mathcal{V}^t denote the set of vertices such that $\{u \in \mathcal{V}^{(t)} | r^{(t)}(u) > 0\}$, and $\mathcal{V}^{(\hat{t})}$ be the set with the maximum number of nonzero elements in $\{\mathcal{V}^{(t)} | 1 \leq t \leq t_{max}\}$. In Step 6, the Kronecker product chain $r^{(t-1)} \otimes \dots \otimes r^{(t-k+1)}$ can be performed in time proportion to $|\mathcal{V}^{(t-1)}| \dots |\mathcal{V}^{(t-k+1)}| \leq |\mathcal{V}^{(\hat{t})}|^{(k-1)} \leq \mu(\mathcal{V}^{(\hat{t})})^{(k-1)}$. Also, [142] shows that $\mu(\mathcal{V}^{(t)}) \leq 1/\epsilon$ for all t . Therefore, to compute $r^{(t-1)} \otimes \dots \otimes r^{(t-k+1)}$ takes $O(\mu(\mathcal{V}^{(\hat{t})})^{(k-1)}) \leq O(1/\epsilon^{(k-1)})$ time. After that, the matrix vector product can be computed in $O(\mu(\mathcal{V}^{(t)}, \mathbb{N})) \leq O(\mu(\mathcal{V}^{(\hat{t})}, \mathbb{N})) \leq O(\mu(\mathcal{V}^{(\hat{t})}))^k \leq O(\frac{1}{\epsilon^k})$. The truncation in Step 7 can be computed in time $O(|\mathcal{V}^{(\hat{t})}|)$. Step 8 to Step 15 require sorting the vertices in $|\mathcal{V}^t|$ according to $r^{(t)}$, which takes time $O(|\mathcal{V}^{(\hat{t})}| \log |\mathcal{V}^{(\hat{t})}|)$. In sum, the time complexity of each iteration in HOSPLOC is $O(\frac{1}{\epsilon^k})$.

Since the algorithm runs at most t_{max} iterations, the overall time complexity of HOSPLOC is $O(\frac{t_{max}}{\epsilon^k})$. By Eq. 6.6, we can expand $O(\frac{t_{max}}{\epsilon^k})$ as follows

$$\begin{aligned} O\left(\frac{t_{max}}{\epsilon^k}\right) &= O\left(t_{max} \left(\frac{2^b \log^3 \mu(\mathcal{V})}{\phi^2}\right)^k\right) \\ &= O\left(t_{max} \frac{2^{bk}}{\phi^{2k}} \log^{3k} m\right). \end{aligned} \tag{6.24}$$

QED.

Remark 1: The major computation overhead of Algorithm 6.1 comes from Step 6. Note that $O\left(t_{max} \frac{2^{bk}}{\phi^{2k}} \log^{3k} m\right)$ is a strict upper-bound for considering extreme cases. While, due to the power law distribution in real networks, we may usually have $|\mathcal{V}^{(t)}| \leq \sqrt{\mu(\mathcal{V}^{(\hat{t})})}$. Then, the complexity of Algorithm 6.1 can be reduced to $O(t_{max}/\epsilon^{k/2}) = O(t_{max}(2^b/\phi^2)^{k/2} \log^{3k/2} m)$.

Remark 2: Suppose the maximum iteration number of Nibble and HOSPLOC are both upper-bounded by t_{max} , then the time complexity of Nibble is $O\left(\frac{t_{max} 2^b \log^4 m}{\phi^2}\right)$. Considering the $k = 3$ case, the time complexity of HOSPLOC is $O\left(t_{max} \frac{2^{3b}}{\phi^6} \log^9 m\right)$. Without considering the impact from the other constants, we can see that similar to Nibble, HOSPLOC also runs in *polylogarithmic* time complexity with respect to the number of edges in the graph.

6.4.7 High-Order Structure-Preserving Graph Clustering

We now present the high-order structure-preserving graph clustering algorithm named HOSGRAP in Algorithm 6.2, that perform structure-preserving graph partitioning by routinely calling HOSPLOC. The inputs of Algorithm 6.2 are mostly the same as Algorithm 6.1,

the only differences are that Algorithm 6.2 requires cluster number c and the vertices distribution $\psi_{\mathcal{V}}$ for sampling initial vertices in order to call HOSPLOC. Step 1 is the initialization step, while Step 2 to Step 8 are the main loop of HOSPLOC that aims to partition the graph into c structure-rich subgraphs. Specifically, Step 3 to Step 4 construct the subgraph $\mathcal{G}^{(j)}$ and its corresponding transition tensor $\mathbf{P}^{(j)}$ by indexing \mathcal{G} and \mathbf{P} ; Step 5 samples the initial vertex from $\mathcal{G}^{(j)}$ according to $\psi_{\mathcal{V}}$, while Step 6 computes the value of b that controls the minimum volume of the returned cluster; in the end, Step 7 calls HOSPLOC to conduct graph cut by using the above computed parameters. If the returned cluster C in Step 6 is nonempty, we will update the partition $D = D \cup \{C, \bar{C}\}$, otherwise, we will return the current graph partition D . The algorithm stops when the graph is partitioned into c structure-rich subgraphs.

Algorithm 6.2: High-Order Structure-Preserved Graph Partitioning (HOSGRAP)

Require:

- (1) k^{th} -order transition tensor \mathbf{P} and transition matrix M ,
- (2) Vertex distribution $\psi_{\mathcal{V}}$,
- (3) Conductance upper bound ϕ ,
- (4) Maximum iteration number t_{\max} ,
- (5) Parameters c_1, ξ ,
- (6) Partition number c .

Ensure:

- Graph Partitioning $D = D_1 \cup \dots D_j$.
- 1: Set $\mathcal{G}^{(1)} = \mathcal{G}$, $\mathbf{P}^{(1)} = \mathbf{P}$, $M^{(1)} = M$ and $j = 1$.
 - 2: **while** $j < c$ **do**
 - 3: Construct the subgraph $\mathcal{G}^{(j)} = (\mathcal{V}^{(j)}, \mathcal{E}^{(j)})$ regarding the largest component in D .
 - 4: Compute the transition tensor $\mathbf{P}^{(j)}$ and the transition matrix $M^{(j)}$ of subgraph $\mathcal{G}^{(j)}$.
 - 5: Randomly sample a initial vertex in $\mathcal{G}^{(j)}$ according to $\psi_{\mathcal{V}}$.
 - 6: Choose a b in $1, \dots, \lceil \log m \rceil$ according to
$$Pr(b = i) = \frac{2^{-ki}}{1 - 2^{-k\lceil \log m \rceil}}.$$
 - 7: Partitioning $\mathcal{G}^{(j)}$ into C and \bar{C} via HOSPLOC algorithm. If C is nonempty, let $D = D \cup \{C, \bar{C}\}$, and $j = j + 1$; otherwise, return the current graph partitioning D .
 - 8: **end while**
-

As HOSGRAP calls HOSPLOC via a subroutine, we analyze the complexity of Algorithm 6.2 based on the results from Lemma 6.1. Lemma 6.2 shows that the expected

running time of HOSGRAP algorithm is bounded by $O\left(L \frac{t_{max} \log^{3k+1} m}{\phi^{2k}}\right)$, where L denotes the iteration number of Algorithm 6.2.

Lemma 6.2 (Efficiency of HOSGRAP). Given graph \mathcal{G} and the k^{th} -order network structure \mathbb{N} , $k \geq 3$, the time complexity of HOSGRAP is bounded by $O\left(L \frac{t_{max} \log^{3k+1} m}{\phi^{2k}}\right)$.

Proof. Based on Lemma 6.1, the expected running time of inner loop (Step 3 to Step 7) in Algorithm 6.2 can be bounded by

$$\begin{aligned} & O\left(\sum_{i=1}^{\lceil \log m^{(j)} \rceil} \frac{2^{-ik}}{1 - 2^{-\lceil \log m^{(j)} \rceil k}} t_{max} \frac{2^{ik}}{\phi^{2k}} \log^{3k} m^{(j)}\right) \\ & \leq O\left(\sum_{i=1}^{\lceil \log m \rceil} \frac{1}{1 - 2^{-\lceil \log m \rceil k}} \frac{t_{max} \log^{3k} m}{\phi^{2k}}\right) \\ & \leq O\left(t_{max} \frac{\log^{3k+1} m}{\phi^{2k}}\right) \end{aligned} \tag{6.25}$$

where $m^{(j)}$ is the number of edges in the subgraph $\mathcal{G}^{(j)}$. Suppose the overall iterations of HOSGRAP is L , then the expected running time of HOSGRAP is upper bounded by $O\left(L \frac{t_{max} \log^{3k+1} m}{\phi^{2k}}\right)$. Note that the iteration number L is naturally larger than the number of clusters c . When $L=c$, it indicates the fact that HOSGRAP successfully identifies a cluster in each iteration before stopping the algorithm. QED.

6.5 GENERALIZATIONS AND APPLICATIONS

In this section, we introduce several generalizations and applications of our proposed HOSPLOC algorithm on signed networks, bipartite networks and multi-partite networks.

6.5.1 Community Detection on Signed Networks

First, we extend our proposed framework, i.e., HOSPLOC, to solve problems on signed graphs. In many real applications, the high-order network structures of interest to us are presented with signed edges. For instance, Figure 6.2 presents an unstable 3-node network structure and a stable 3-node network structure based on social status theorem [172]. In community detection [173], we may want to ensure (1) the stable configurations to be rich

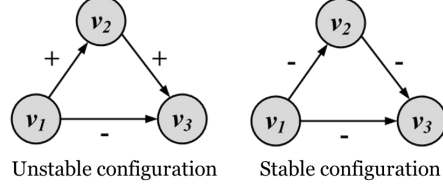


Figure 6.2: Social Status Theory Example: (Left) A directed “+” edge from node v_1 to node v_2 shows that v_2 has a higher status than v_1 . (Right) A directed “-” edge from node v_1 to node v_2 shows vice versa.

within communities and sparse in-between different communities; (2) the unstable configurations to be sparse within communities and rich in-between different communities. For this purpose, the adjacency tensor can be constructed as follows

$$T(i_1, i_2, \dots, i_k) = \begin{cases} 1 & \{i_1, i_2, \dots, i_k\} \text{ is stable structure} \\ 0 & \{i_1, i_2, \dots, i_k\} \text{ is unstable structure} \\ \alpha & \text{Otherwise} \end{cases} \quad (6.26)$$

where $\{i_1, i_2, \dots, i_k\} \in \mathcal{V}$ and constant $0 < \alpha < 1$. By this way, we can ensure: (1) the returned cluster of HOSPLOC contains rich stable structures; (2) the partition would most likely break unstable structures.

6.5.2 User Behavior Modeling on Bipartite Networks

We now turn our attention to the problem of user behavior modeling on the advertisement networks. Given an advertisement network $B = (\mathcal{V}_B, \mathcal{E}_B)$, the bipartite graph B contains two types of nodes, i.e., user nodes \mathcal{V}_U and advertiser campaign nodes \mathcal{V}_A , i.e., $\mathcal{V}_B = \{\mathcal{V}_U, \mathcal{V}_A\}$. The edges \mathcal{E}_B only exist between user nodes \mathcal{V}_U and advertiser campaign nodes \mathcal{V}_A . Intuitively, the customers with similar activities on the advertisement network should be included in the same cluster. For this reason, we choose 4-node loop as the base network structure for HOSPLOC algorithm. Specifically, suppose both user nodes u_1, u_2 have user-campaign interactions with the advertiser campaign nodes a_1 and a_2 , then we have a 4-node loop, which is shown in Figure 6.3. In this problem, we consider the advertisement network as an undirected graph, and the adjacency tensor can be constructed as follows

$$T(i_1, i_2, i_3, i_4) = \begin{cases} 1 & \{i_1, i_2, i_3, i_4\} \text{ form a 4-nodes loop} \\ 0 & \text{Otherwise} \end{cases} \quad (6.27)$$

where $\{i_1, i_2, i_3, i_4\} \in \mathcal{V}_B$. Starting from an initial vertex, the returned cluster C_B by HOSPLOC would represent a local user-campaign community, which consists of both similar users and the users' favorite advertiser campaigns.

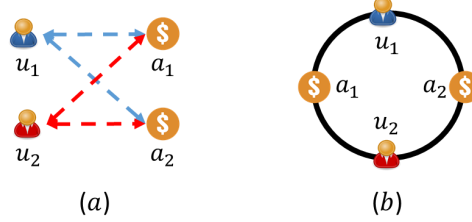


Figure 6.3: The illustration of user-advertisement interaction. (a) An example of two users both participate in two advertisement campaigns. (b) An four-node loop induced from (a).

6.5.3 Synthetic ID Detection on Multi-partite Networks

Here, we explain how to detect synthetic IDs on the PII network by using our proposed HOSPLOC algorithm. The PII network is a typical multi-partite network, where each partite set of nodes represents a particular type of PII, such as users' names, users' accounts, and email addresses, and the edges only exist between different partite sets of nodes. In synthetic ID fraud [135], criminals often use modified identity attributes, such as phone number, home address and email address, to combine with real users' information and create synthetic IDs to do malicious activities. Hence, for the synthetic IDs, there is a high possibility that their PIIs would be shared by multiple identities, which may compose rich star-shaped structures. In this case, the adjacency tensor can be constructed as

$$T(i_1, i_2, \dots, i_k) = \begin{cases} 1 & \{i_1, i_2, \dots, i_k\} \text{ form a } k\text{-node star} \\ 0 & \text{Otherwise} \end{cases} \quad (6.28)$$

where $\{i_1, i_2, \dots, i_k\} \in \mathcal{V}_B$. Note that the returned partition may consist of various types of nodes. However, it is viable to trace back from the extracted PII nodes and discover the set of synthetic identities.

6.6 EXPERIMENTAL EVALUATION

In this section, we present the experimental evaluations. The experiments are designed to answer the following questions: In particular, we aim to answer the following questions:

- *Effectiveness*: How effective is the proposed HOSPLOC algorithm for conducting a local cut with preserving high-order network structures, and how effective is the proposed HOSGRAP algorithm for performing structure-preserving graph clustering?
- *Scalability*: How fast and scalable is the proposed HOSPLOC and HOSGRAP algorithms?
- *Parameter Sensitivity*: How robust is the proposed algorithms with changing parameters?
- *Case Study*: What’s the performance of the proposed algorithms when we are solving problems on bipartite graph and multi-partite graph.

6.6.1 Experiment Setup

Category	Network	Type	Nodes	Edges
Citation	Author	Undirected	61,843	402,074
	Paper	Undirected	62,602	10,904
Infrastructure	Airline	Undirected	2,833	15,204
	Oregon	Undirected	7,352	15,665
	Power	Undirected	4,941	13,188
Social	Epinion	Undirected	75,879	508,837
Review	Rating	Bipartite	8,724	90,962
Financial	PII	Multi-partite	375	519

Table 6.2: Statistics of the Networks.

Data sets: We evaluate our proposed algorithm on both synthetic and real-world network graphs. The statistics of all real data sets are summarized in Table 6.2.

- **Collaboration Network:** We use two collaboration networks from Aminer¹. In network (Author), the nodes are authors, and an edge only exists when two authors have a co-authored paper. In network (Paper), the nodes are distinct papers, and an edge only exists when one paper cites another paper.

¹<https://aminer.org/data>

- **Infrastructure Network:** In network (Airline)², the nodes represent 2,833 airports, and the edges represent the U.S. flights in a one-month interval. Network (Oregon) [174] is a network of routers in Autonomous Systems inferred from Oregon route-views between March 31, 2001, and May 26, 2001. Network (Power)³ contains the information of the power grid of the western states of U.S. A node represents a generator, a transformer or a substation, and an edge represents a power supply line.
- **Social Network:** Network (Epinion) [174] is a who-trust-whom online social network. Each node represents a user, and one edge exists if and only if when one user trusts another user.
- **Review Network:** Network (Rating) [175] is a bipartite graph, where one side of nodes represent 643 users, and another side of nodes represent 7,483 movies. Edges refer to the positive ratings, i.e., rating score larger than 2.5, on MovieLens website. Note that this network is a subgraph from the original one, due to storing the 4th-order transition tensor of the original graph, i.e., 100s K vertices and millions edges, requires too much memory.
- **Financial Network:** Network (PII) is a multi-partite graph, which consists of five types of vertices, i.e., 112 bank accounts, 71 names, 80 emails, 35 addresses, and 77 phone numbers. Edges only exist between account vertices and PII vertices.

Comparison Methods: In our experiments, we compare our methods with both local and global graph clustering methods. Specifically, the comparison algorithm includes three local algorithms, i.e., (1) Nibble [142]; (2) NPR [143]; (3) LS-OQC [176], and two global clustering algorithms, i.e., (1) NMF [177]; (2) TSC [136]. Among these five baseline algorithms, TSC algorithm is designed based on high-order Markov chain, which can model high-order network structures, i.e., triangle.

6.6.2 Effectiveness Analysis

The effectiveness comparison results conducted on six real undirected graphs by the following three evaluation metrics. Among them, (1) **Conductance** [167] in Eq. 6.4 measures the general quality of a cut on graph, which quantitatively indicates the compactness of a cut; (2) **The 3rd-Order Conductance** could be computed based on Eq. 6.14 by treating

²<http://www.levmuchnik.net/Content/Networks/NetworkData.html>

³<http://konect.uni-koblenz.de/networks/opsahl-powergrid>

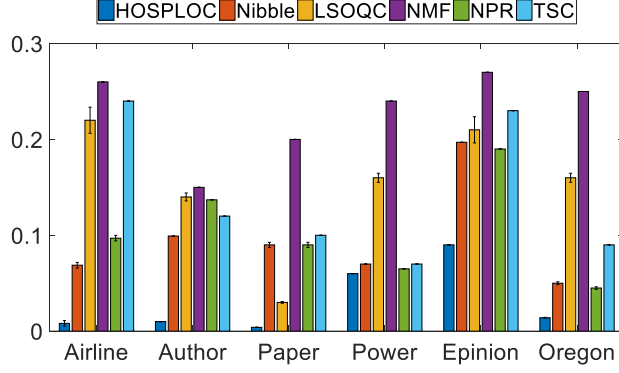


Figure 6.4: The average conductance of the returned graph cut. Lower is better.

triangle as the network structure \mathbb{N} , which estimates how well the network structure \mathbb{N} is preserved in the returned cut from being broken by the partitions; (3) **Triangle Density** [167] is defined as $\tau(C) = 3t(C)/w(C)$, where $t(\mathcal{G})$ is the number of triangles in C and $w(C)$ is the number of wedges in C . Conventionally, we have $t(C) = 0$ if there is no wedge in the given C . Here we use Triangle Density to measure the ratio of how rich the triangle is included in the returned cluster C .

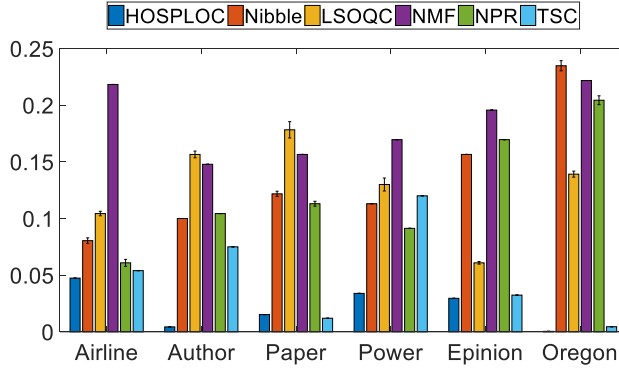


Figure 6.5: The average 3rd-order conductance of the returned graph cut. Lower is better.

A. Quantitative Evaluations for Problem 1. The comparison results for the structure-preserving local graph cut problem are shown from Figure 6.4 to Figure 6.6. Moreover, to evaluate the convergence of local algorithms, we randomly select 30 vertices from one cluster on each testing graph and run all the local algorithms multiple times by treating each of these nodes as an initial vertex. In particular, the heights of bars indicate the average value of evaluation metrics, and the error bars (only for local algorithms) represent the standard deviation of evaluation metrics in multiple runs. We have the following observations: (1)

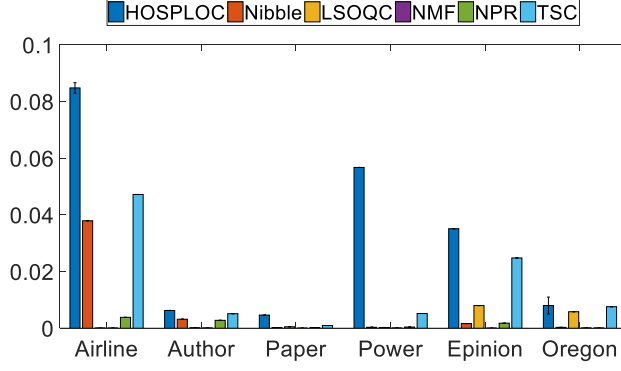


Figure 6.6: The average triangle density of the returned graph cut. Higher is better.

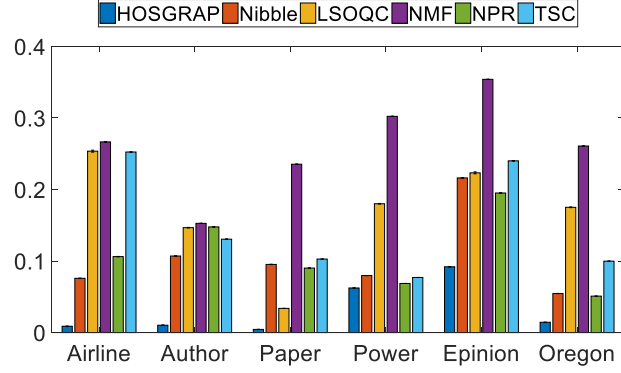


Figure 6.7: The average conductance over the partitioned subgraphs. Lower is better.

In general, local algorithms perform better than the global algorithm, and our HOSPLOC algorithm consistently outperforms the others on all the evaluation metrics. For example, compared to the best competitor, i.e., TSC, on network (Airline), HOSPLOC algorithm is 97% smaller on conductance, 12.2% smaller on the 3rd-order conductance, 80% larger on triangle density. (2) High-order Markov chain models, i.e., HOSPLOC and TSC, could better preserve triangles in the returned cluster. For example, on network (Epinion), both HOSPLOC and TSC return a cluster with much higher triangle density and much lower the 3rd-order conductance. (3) HOSPLOC algorithm shows a more robust convergence property than the other local clustering algorithm by comparing the size of error bars. For example, among the three local algorithms, only HOSPLOC algorithm returns the identical cluster on network (Paper) with different initial vertexes.

B. Quantitative Evaluations for Problem 2. The comparison results for the Structure-Preserving Graph Partition problem are presented in Figure 6.7 to Figure 6.9. For conducting

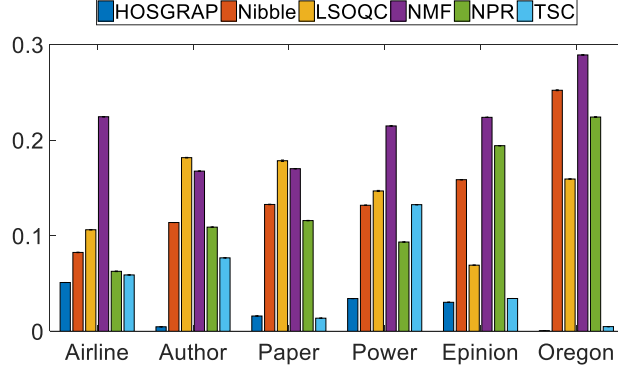


Figure 6.8: The average 3rd-order conductance over the partitioned subgraphs. Lower is better.

HOSGRAP, we manually set the vertex distribution ψ_V following the degree distribution, and the partition number $c = 5$. In Figure 6.7 to Figure 6.9, the height of the bars indicate the averaged value of the metrics of the partitioned subgraphs, and the error bars represent the standard deviation of evaluation metrics in 30-times runs. In general, we observe that our proposed HOSGRAP algorithm outperforms the baseline methods in all the six data sets across all the metrics.

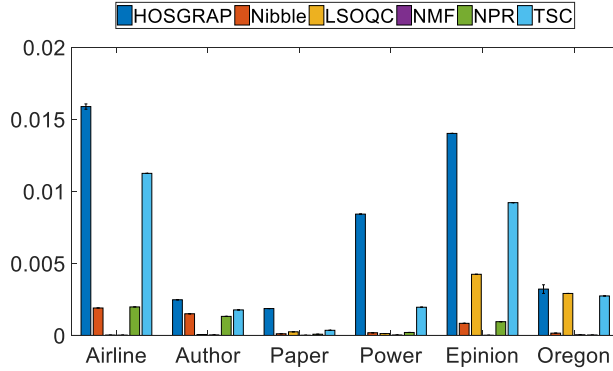


Figure 6.9: The average triangle density over the partitioned subgraphs. Higher is better.

6.6.3 Scalability Analysis

In this subsection, we study the scalability of our proposed Framework. We let triangle as the user-defined network structure and the partition number $c = 3$. Since our method is built on higher order of random walk than Nibble, we consider Nibble as the running time

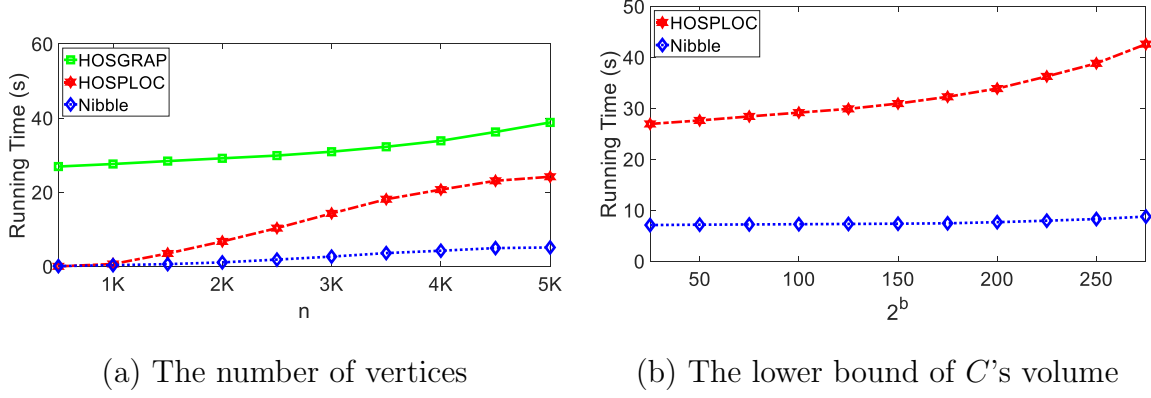


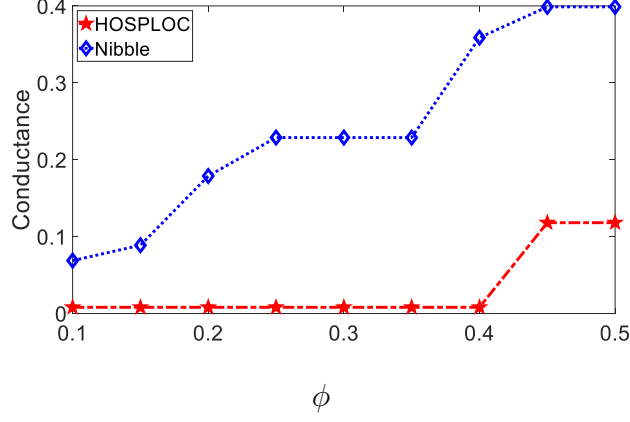
Figure 6.10: Scalability analysis w.r.t. the number of nodes n and the the lower bound volume of the returned clusters 2^b .

lower bound of HOSPLOC algorithm. Notice that all the results in Figure 6.10 are the average values of multiple runs by using 30 different initial vertexes on the same graph. In Figure 6.10 (a), we show the running time of HOSGRAP, HOSPLOC and Nibble on a series of synthetic graphs with increasing number of vertices but fixed edge density of 0.5%. We observe that although HOSGRAP and HOSPLOC require more time than Nibble in each run, the running time of HOSGRAP and HOSPLOC increases *polylogarithmically* with the size of the graph $|\mathcal{V}|$, which demonstrate our scalability analysis in Lemma 6.1 and Lemma 6.2. In Figure 6.10 (b), we show the running time of HOSPLOC and Nibble versus the lower bound of output volume on the synthetic graph with 5000 vertices and 0.5% edge density, by keeping the other parameters fixed. Note that HOSGRAP is not included in Figure 6.10 (b), since the the lower bound of output volume 2^b is not an input variable of HOSGRAP algorithm. We can see that the running time of HOSPLOC is polynomial with respect to 2^b , which is consistent with our time complexity analysis.

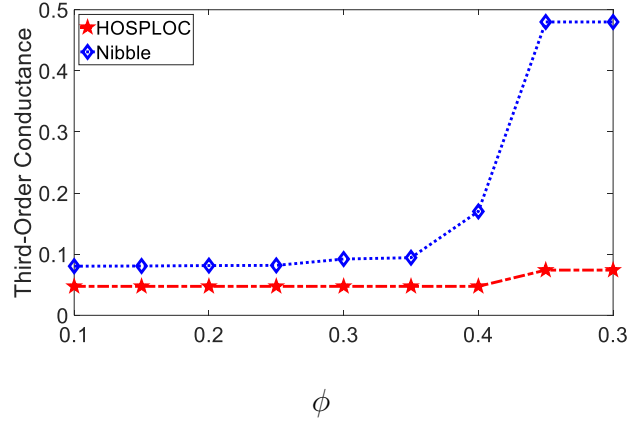
6.6.4 Parameter Analysis

In this subsection, we analyze the parameter sensitivity of our proposed HOSPLOC algorithm with triangle as the specified network structure, by comparing with Nibble algorithm on the synthetic graph with 5000 vertices and 0.5% edge density. Here, we mainly focused on HOSPLOC algorithm, as HOSGRAP could be considered as multiple runs of HOSPLOC algorithm with different initialization. In the experiments, we evaluate the conductance and the 3^{rd} -order conductance of the returned cut with different values of input parameter ϕ . In Figure 6.11, we have the following observations: (1) HOSPLOC returns the optimal

cut even with a very loose conductance upper bound ϕ . In Figure 6.11 (a), we can see the output conductance of HOSPLOC converges to the minimum value when $\phi = 0.4$, while the output conductance of Nibble converges to its minimum value until $\phi = 0.1$. (2) Both the conductance and the 3rd-order conductance of HOSPLOC's cut are always smaller than Nibble's cut with different ϕ .



(a) Conductance



(b) The 3rd-order conductance

Figure 6.11: Parameter analysis w.r.t. conductance upper-bound ϕ . Lower is better.

6.6.5 Case Study

In this subsection, we will consider more complex network structures and perform our proposed HOSPLOC algorithm on bipartite and multi-partite networks.

Case Study on Bipartite Graph. We conduct a case study on the network (Rating) to find a local community consisting of similar taste users and their favorite movies. In this

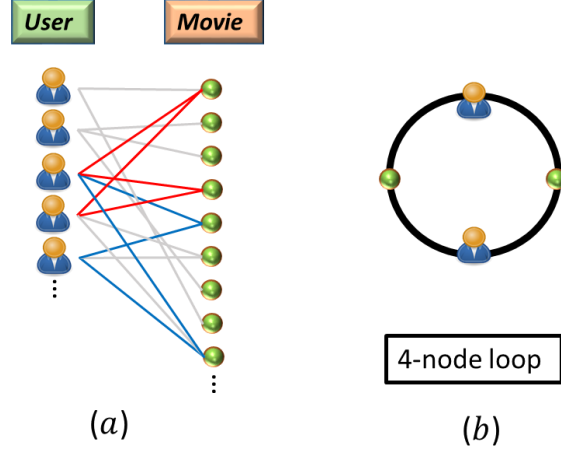


Figure 6.12: Case study on bipartite network Rating. (a) An example of detected community by HOSPLOC on Rating. (b) An example of 4-node loop on Rating.

case study, we construct the transition tensor on the basis of 4-node loop based on Eq. 6.27. Figure 6.12 (a) presents a miniature of the cluster identified by our proposed HOSPLOC algorithm regarding 4-node loop that illustrated in Figure 6.12 (b). For example, in Figure 6.12, the highlighted red loop shows that both of the third and the fourth users like the first and the fourth movies, while the highlighted blue loop represents that both of the third and the fifth users like the fifth and the last movies. It seems the fifth user does not like the first movie due to no direct connection between them. While the interesting part is the first, the fifth and the last movies are from the same series, i.e., Karate Kid I, II, III. Moreover, the fourth movie, i.e., Back to School, and Karate Kid I, II, III, are all from the category of comedy. It turns out that our HOSPLOC algorithm returns a community of comedy movies and their fans.

Case Study on Multi-partite Graph. Here, we conduct a case study on the network (PII) to identify suspicious systemic IDs. In this case, we treat 5-node star as the underlying network structure, and the corresponding transition tensor could be generated by Eq. 6.28. Figure 6.13 (a) presents a subgraph of the returned cut by our proposed HOSPLOC algorithm regarding 5-node star that illustrated in Figure 6.13 (b). We can see that many PIIs are highly shared by different accounts. For example, the account connected with blue lines shares the home address and email address with the account connected with purple lines, while the account connected with red lines shares the holder’s name and phone number with the account connected with blue lines. Comparing with the regular dense subgraph detection methods, our method can better identify the IDs who share their PIIs with others, by exploring the nature structure of PII, i.e., 5-node star, on the given graph.

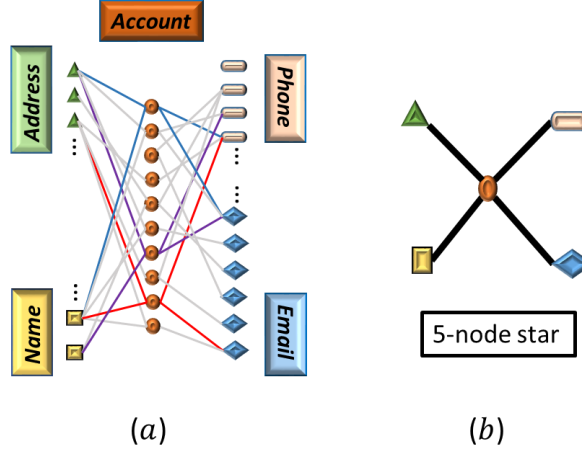


Figure 6.13: Case study on multi-partite network PII. (a) An example of detected community by HOSPLOC on PII. (b) An example of 5-node star on PII.

6.7 SUMMARY

In this chapter, we propose a structure-preserving graph cut algorithm, i.e., HOSPLOC, that gives users the flexibility to model any high-order network structures and returns a small high-order conductance cluster which largely preserves the *user-specified* network structures. Base on HOSPLOC, we further develop a partitioning algorithm named HOSGRAP that largely preserves the user-defined structures in the returned clusters. Besides, we analyze its performance in terms of the optimality of the obtained cluster and the *polylogarithmic* time complexity on massive graphs. Furthermore, we generalize the proposed algorithms to solve real problems on signed networks, bipartite networks and multi-partite networks, by exploring the useful high-order network connectivity patterns, such as loops and stars. Finally, the extensive empirical evaluations on a diverse set of networks demonstrate the effectiveness and scalability of our proposed HOSPLOC and HOSGRAP algorithms.

CHAPTER 7: DOMAIN ADAPTIVE RARE CATEGORY CHARACTERIZATION

7.1 OVERVIEW AND MOTIVATION

In the age of big data, graph presents a robust data structure for modeling relational data from various domains, ranging from physics to biology, from neuroscience to social science. Graph neural networks (GNNs) [178, 179, 180, 181] provide a powerful tool to distill knowledge and learn expressive representations from graph structured data. While significant achievements have been made, most successful GNNs are trained in a supervised manner that requires abundant task-specific labels. Nevertheless, in many high-impact domains [182, 183, 184] (e.g., brain networks constructed by fMRI), collecting high-quality labels is quite resource-intensive and time-consuming, which largely restricts the potential of GNNs in real-world applications.

Inspired by the profound success of the pre-trained models in the vision [185] and language [186] domains, the recent advances [187, 188, 189] have focused on pre-training GNNs by directly learning from proxy signals that are extracted from graphs. The hope is that the extracted proxy signals are informative and task-invariant, and as such, the learned graph representation can be generalized to novel tasks that have not been observed before. Nevertheless, the current GNNs pre-training strategies are still at the early stage and suffer from multiple limitations. Most prominently, the performance of the existing pre-training strategies largely relies on the quality of proxy signals. It has shown that noisy and irrelevant proxy signals often lead to negative transfer [190] and marginal improvement in many application scenarios [191]. For example, one may want to predict the chemical properties [183, 192] of a family of novel molecules (e.g., the emerging COVID-19 variants), while the data available (e.g., the known coronavirus) for pre-training are mostly homologous but with diverse structures and disparate feature spaces. In this case, how can we eliminate the risk of negative transfer and thus guarantee the generalization performance? What is worse, the issue of negative transfer can be exacerbated when encountering the heterogeneity of graph signals, where the graph signals are shown in different types (e.g., class-memberships, attributes, centrality scores) at different granularities (e.g., nodes, edges, subgraphs) within graphs. To integrate such heterogeneous proxy signals, the current GNNs pre-training strategies are conventionally hand-engineered or ad-hoc by using some hyperparameters. Consequently, the inductive bias from humans might be injected into the pre-trained model and deteriorate the generalization performance in the downstream tasks.

We identify the following challenges. First (*Cross-Graph Heterogeneity*), how can we

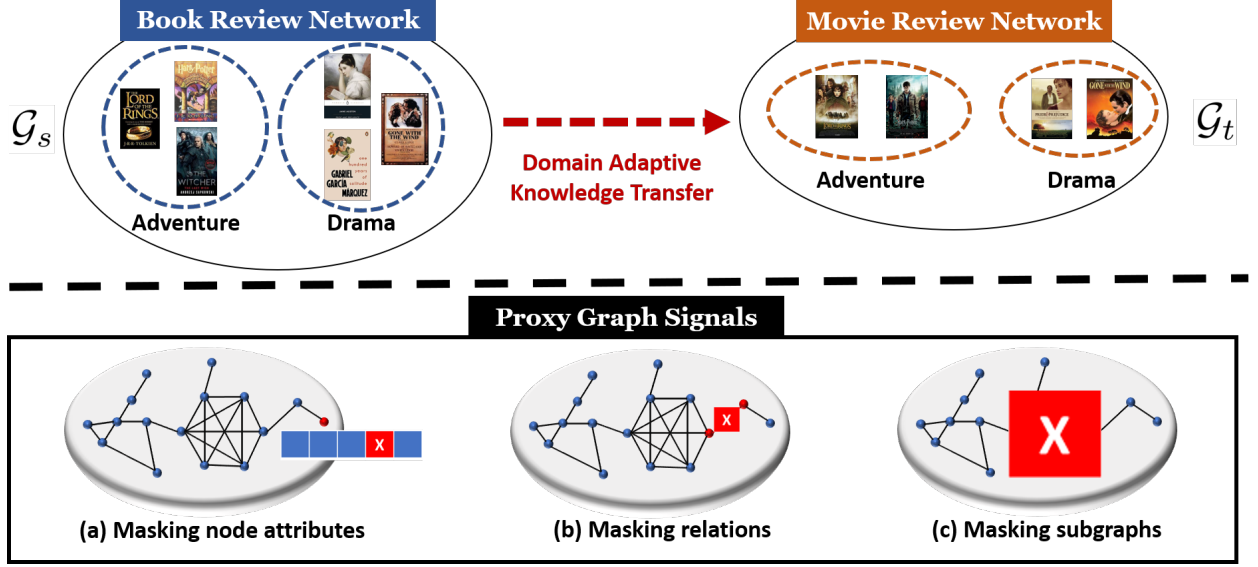


Figure 7.1: An illustrative example of domain adaptive graph pre-training. (Top) Domain-adaptive knowledge transfer between the book review network and the movie review network. (Bottom) Proxy graph signals (red masks) at the granularity of nodes, edges, and subgraphs.

eliminate the negative transfer and effectively translate the knowledge learned in the source graph to solve novel tasks on the target graph? Second (*Graph-Signal Heterogeneity*), how can we efficiently comprehend the contribution of complex graph signals and further enhance generalization performance in the target domain? To address these challenges, it is crucial to learn a domain-adaptive representation to enable knowledge transfer from the source domain to the target domain and carefully select proxy signals relevant to the downstream tasks.

To this end, we propose an end-to-end framework, namely MENTORGNN, which seamlessly embraces both of the aforementioned objectives for pre-training GNNs. In particular, to address cross-graph heterogeneity, we propose an encoder-decoder architecture that summarizes contextual information at different granularities of graphs and learns a composition of nonlinear mappings across different domains. Moreover, to address graph-signal heterogeneity, we develop a curriculum learning paradigm [193], where a teacher model (i.e., graph signal re-weighting scheme) gradually generates a domain adaptive curriculum to guide the pre-training of the student model (i.e., GNNs) to enhance the generalization performance in the tasks of interest. We summarize our contributions as follows.

- **Problem.** We formalize the *domain-adaptive graph pre-training* problem and identify unique challenges motivated from the real applications.
- **Algorithm.** We propose a novel method named MENTORGNN, which 1) automatically learns a knowledge transfer function and 2) generates a domain-specific curricu-

lum for pre-training GNNs across diverse domains. We also present a natural and interpretable generalization bound for domain-adaptive graph pre-training.

- **Evaluation.** We systematically evaluate the performance of MENTORGNN under two experimental settings: 1) single-graph transfer and 2) multi-graph transfer. Extensive results prove the superior performance of MENTORGNN under both settings. We find that MENTORGNN largely alleviates the negative transfer issue and leads up to 17.17% accuracy improvement over the non-pre-trained models.

The rest of the chapter is organized as follows. We review the related literature in Section 7.2. In Section 7.3, we introduce the preliminary and problem definition, followed by the details of our proposed framework MENTORGNN in Section 7.4. Experimental results are reported in Section 7.5. Finally, we conclude this chapter in Section 7.6.

7.2 RELATED WORK

In this section, we briefly review the recent advances in the context of pre-training strategies and domain adaption for graphs.

7.2.1 Pre-Training Strategies for Graphs.

To tackle the label scarcity in the graph representation learning, pre-training strategies for graph neural network models are proposed [187, 188, 189, 194, 195]. The core idea is to capture the generic graph information across different tasks and transfer it to the target task to save the amount of labeled data and domain-specific features. Some recent methods effectively extract knowledge at both the level of individual nodes as well as the entire graphs via various techniques, including contrastive predictive coding [196], context prediction [187, 188, 194], and mutual information maximization [187, 197]. To be specific, in [188], to learn the transferable structural information, the authors design three topology-based tasks for pre-training GNNs, i.e., link reconstruction, centrality ranking, and cluster preserving. Also, the authors in [189] propose a pre-training framework for GNNs by combining the self-supervised pre-training strategy on the node level (i.e., neighborhood prediction and attributes masking) and the supervised pre-training strategy on the graph level (i.e., graph property prediction). However, the current pre-training strategies often lack the capability of domain adaptation thus limiting their ability to leverage valuable information from other data sources. In this chapter, for the first time, we propose a novel method named MENTORGNN that enables pre-training GNNs across graphs and domains.

7.2.2 Domain Adaptation.

To ensure that the learned knowledge is transferable between the source domains (or tasks) and the target domains (or tasks), many efforts tend to learn the domain-invariant or task-invariant representations, such as [198, 199]. After the domain adaptation meets the graph-structured data [200], and many GNNs are proposed [178, 179, 181], the transferability of GNNs has been theoretically analyzed in terms of convolution operations [201]. To learn the domain-invariant representations with GNNs, UDA-GCN is proposed [202] in the unsupervised learning setting to learn invariant representations via a dual graph convolutional network component. Especially, targeting to efficiently label the nodes on a target graph to reduce the annotation cost of training, GPA [191] is proposed to learn a transferable policy with reinforcement learning techniques among full-labeled source graphs, which can be directly generalized to unlabeled target graphs. Despite the success of the domain adaptation on graphs, little effort has been contributed to the generalization performance of the deep learning models on relational data (e.g., graphs). Here we propose a new generalization bound for domain-adaptive pre-training on graph-structured data.

7.3 PRELIMINARIES

We first introduce the notations used throughout this chapter. Given that, we briefly review the current pre-training strategies for GNNs and a population-based theoretical model for domain adaptation.

Notations. We use regular letters to denote scalars (e.g., α), boldface lowercase letters to denote vectors (e.g., \mathbf{v}), and boldface uppercase letters to denote matrices (e.g., \mathbf{A}). In this chapter, we denote the source graph and the target graph using $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s, \mathbf{X}_s)$ and $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t, \mathbf{X}_t)$, where \mathcal{V}_s (\mathcal{V}_t) is the set of nodes, \mathcal{E}_s (\mathcal{E}_t) is the set of edges, and \mathbf{X}_s (\mathbf{X}_t) is the node attributes in \mathcal{G}_s (\mathcal{G}_t).

Pre-taining strategies for GNNs. Previous studies [187, 188, 189, 194, 203, 204, 205] have been proposed to use easily-accessible graph signals to capture domain-specific knowledge and pre-train GNNs. These attempts are mostly designed to parameterize and optimize the GNNs by predicting the masked graph signals (e.g., node attributes [203, 204], edges [206], subgraphs [189], network proximity [188, 194], etc.) from the visible ones in the given graph \mathcal{G} . In Figure 7.1, we instantiate the masked proxy signals at the level of nodes, edges, and subgraphs. To predict any proxy graph signal $\mathbf{s} \in \mathcal{G}$, we use $f : \mathbf{s} \rightarrow [0, 1]$ to denote the true labeling function and $h : \mathbf{s} \rightarrow [0, 1]$ to denote the learned hypothesis by GNNs. The risk of the hypothesis $h(\cdot)$ can be computed as $\epsilon(h, f) := \mathbb{E}_{\mathbf{s} \in \mathcal{G}}[|h(\mathbf{s}) - f(\mathbf{s})|]$. In

general, the overall learning objective of pre-training models for GNNs can be formulated as follows.

$$\operatorname{argmax}_{\theta} \log h(\mathcal{G}|\hat{\mathcal{G}}, \theta) = \sum_{\mathbf{s} \in \mathcal{G}} \beta_{\mathbf{s}} \log h(\mathbf{s}|\hat{\mathcal{G}}, \theta) \quad (7.1)$$

where $\hat{\mathcal{G}}$ is the corrupted graph with some masked graph signals, \mathbf{s} is the masked graph signals sampled from the original graph \mathcal{G} , $\beta_{\mathbf{s}}$ is a hyperparameter that balances the weight of the graph signal \mathbf{s} , and θ is the hidden parameters of the GNNs $h(\cdot)$. By maximizing Eq. 7.1, we can encode the contextual information of the selected proxy graph signals into the pre-trained GNNs $h(\cdot)$. With that, the learned GNNs can be fine-tuned and performed in the downstream tasks of interest.

Domain adaptation. Following the conventional notations, we let $f_s(\cdot)$ and $f_t(\cdot)$ be the true labeling functions in the source domain and the target domain. Given $f_s(\cdot)$ and $f_t(\cdot)$, $\epsilon_s(h) = \epsilon_s(h, f_s)$ and $\epsilon_t(h) = \epsilon_t(h, f_t)$ denote the corresponding risks in terms of a hypothesis $h(\cdot)$. With that, Ben-David et al. proved a domain adaptive generalization bound in terms of domain discrepancy and empirical risks. To approximate the empirical risks, one common approach [207, 208] is to assume the data points are sampled i.i.d. from both the source and the target domains. However, due to the relational nature of graphs, samples (e.g., two connected nodes) are often connected and non-i.i.d. in the given graphs. Thus, the generalization bound with empirical risks may not hold in graph-structured data. To get rid of the i.i.d. assumption, an alternative approach is to define the generalization bound based on the true data distribution [198, 207]. For instance, Theorem 7.1 [198] provides a population result, which does not rely on the empirical risks and can be naturally deployed on graph-structured data.

Theorem 7.1. [198] Let $\langle \mathcal{D}_s, f_s \rangle$ and $\langle \mathcal{D}_t, f_t \rangle$ be the source and the target domains, for any function class $\mathcal{H} \subseteq [0, 1]^{\mathcal{X}}$, and $\forall h \in \mathcal{H}$, the following inequality holds:

$$\epsilon_t(h) \leq \epsilon_s(h) + d_{\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_t) + \min\{\mathbb{E}_{\mathcal{D}_s}[|f_s - f_t|], \mathbb{E}_{\mathcal{D}_t}[|f_s - f_t|]\} \quad (7.2)$$

As shown in Figure 7.1, our goal is to learn a knowledge transfer function denoted as $g(\cdot)$, such that the knowledge obtained by a GNN model in the source graph \mathcal{G}_s can be transferred to the target graph \mathcal{G}_t and pre-train the GNNs in \mathcal{G}_t . Given the above notations, we formally define the *domain-adaptive graph pre-training* problem as follows.

Problem 7.1. Domain Adaptive Graph Pre-Training

Given: (i) source graph $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s, \mathbf{X}_s)$, (ii) target graph $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t, \mathbf{X}_t)$, (iii) user-

defined graph neural network architecture for pre-training.

Find: (i) the knowledge transfer function $g(\cdot)$, (ii) the pre-trained model $h(\cdot)$ that leverages the knowledge obtained from both \mathcal{G}_s and \mathcal{G}_t .

7.4 ALGORITHM

In this section, we introduce our proposed framework MENTORGNN (shown in Figure 7.2) to address Problem 7.1. The key challenges of Problem 7.1 lie in a dual-level heterogeneity, namely *cross-graph heterogeneity* and *graph-signal heterogeneity*. Next, we dive into two major modules of MENTORGNN, i.e., cross-graph adaptation (colored in blue in Figure 7.2) and curriculum learning (colored in orange in Figure 7.2), that are designed specifically for addressing the aforementioned two challenges.

7.4.1 Cross-Graph Adaptation via Multi-Scale Encoder-Decoder

The core obstruction of cross-graph adaptation lies in how to effectively translate the knowledge learned from \mathcal{G}_s to \mathcal{G}_t without any supervision of cross-graph association (e.g., partial network alignments). Specifically, given \mathcal{G}_s and \mathcal{G}_t , we aim to learn a transformation function that leverages both network structures and node attributes over the entire graph, i.e., $(\mathcal{V}_t, \mathcal{E}_t, \mathbf{X}_t) \simeq g((\mathcal{V}_s, \mathcal{E}_s, \mathbf{X}_s))$. This may require a large parameter space and be computationally intractable, especially when both \mathcal{G}_s and \mathcal{G}_t are large. In order to alleviate the computational complexity, we propose a multi-scale cross-graph adaption scheme (the blue region in Figure 7.2) that learns the translation function $g(\cdot)$ at a coarser resolution instead of directly translating knowledge between \mathcal{G}_s and \mathcal{G}_t . The intuition is that many real graphs from different domains may share similar high-level organizations. For instance, in Figure 7.1 (Top), the book review network (\mathcal{G}_s) and the movie review network (\mathcal{G}_t) come from two diverse domains, but may share similar high-level organizations (e.g., alignments between book genres and movie genres). That is to say, the communities on \mathcal{G}_s have related semantic meanings to the communities on \mathcal{G}_t .

Motivated from this observation, we develop an encoder-decoder architecture that explores the cluster-within-cluster hierarchies to better characterize the graph signals at multiple granularities. As shown in Figure 7.2, the encoder \mathcal{P} learns the multi-scale representation of \mathcal{G}_s by pooling the source graph from the fine-grained representation \mathbf{X}_s to the coarse-grained representation $\mathbf{X}_s^{(L)}$, while the decoder \mathcal{U} aims to reconstruct the target graph from the coarse-grained representation $\mathbf{X}_t^{(L)}$ to the fine-grained representation \mathbf{X}_t . Here we let

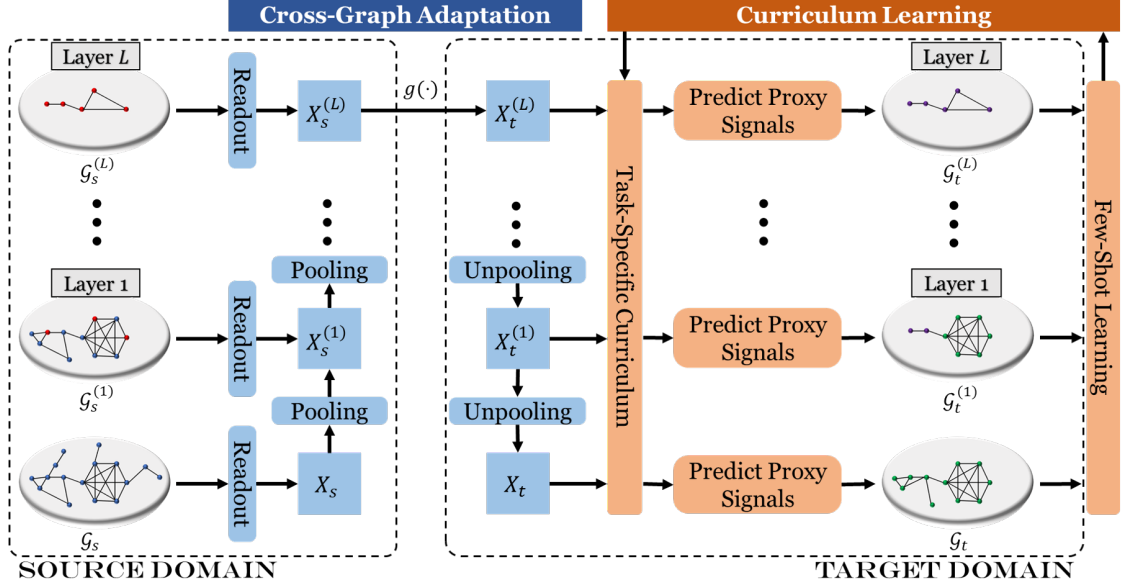


Figure 7.2: An overview of the MENTORGNN, which is composed of two modules, namely cross-graph adaptation (colored in blue) and curriculum learning (colored in orange).

the number of layers L in the encoder and the decoder be the same in order to make \mathcal{G}_s and \mathcal{G}_t comparable with each other at different scales.

Encoder: The encoder is defined as a set of pooling matrices $\mathcal{P} = \{\mathbf{P}^{(1)}, \dots, \mathbf{P}^{(L)}\}$, where $\mathbf{P}^{(l)} \in \mathbb{R}^{n_s^{(l)} \times n_s^{(l+1)}}$, $n_s^{(l)}$ and $n_s^{(l+1)}$ are the number of super nodes at layer l and layer $l+1$. Specifically, following [209], the differentiable pooling matrix $\mathbf{P}^{(l)}$ at layer l is defined as follows.

$$\mathbf{P}^{(l)} = \text{softmax}(\text{GNN}_{l,\text{pool}}(\mathbf{A}^{(l)}, \mathbf{X}_s^{(l)})) \quad (7.3)$$

where each entry $\mathbf{P}^{(l)}(i, j)$ indicates the assignment coefficient from the node i at layer l to the corresponding supernode j at layer $l+1$, and $\text{GNN}_{l,\text{pool}}$ is the corresponding surrogate GNN to generate the assignment matrix $\mathbf{P}^{(l)}$. In our implementation, we consider the output dimension of $\text{GNN}_{l,\text{pool}}$ as a hyper-parameter which is the maximum number of supernodes in the layer $l+1$. With that, the l^{th} -layer coarse-grained representation $\mathbf{X}_s^{(l)}$ of \mathcal{G}_f can be approximated by

$$\mathbf{X}_s^{(l)} = \mathbf{P}^{(l-1)'} \dots \mathbf{P}^{(1)'} \mathbf{X}_s \quad (7.4)$$

Decoder: The decoder is composed of a translation function $g(\cdot)$ and a set of differentiable unpooling matrices $\mathcal{U} = \{\mathbf{U}^{(1)}, \dots, \mathbf{U}^{(L)}\}$. To be specific, the translation function $g(\cdot)$ learns

a non-linear mapping between \mathcal{G}_s and \mathcal{G}_t at the coarsest level L . In this chapter, we define $g(\cdot)$ as a multilayer perceptron (MLP) with ReLU, and the differentiable pooling matrix $\mathbf{U}^{(l)} \in \mathbb{R}^{n_t^{(l+1)} \times n_t^{(l)}}$ at the layer l is defined as follows.

$$\mathbf{U}^{(l)} = \text{softmax}(\text{GNN}_{l,\text{unpool}}(\mathbf{A}^{(l)}, \mathbf{X}^{(l)})) \quad (7.5)$$

In contrast to $\text{GNN}_{l,\text{pool}}$, $\text{GNN}_{l,\text{unpool}}$ is an opposite operation that aims to reconstruct $\mathbf{X}^{(l-1)}$ from its coarse representation $\mathbf{X}^{(l)}$. With the learned translation function $g(\cdot)$ and the differentiable unpooling matrices \mathcal{U} , the hidden representation of \mathcal{G}_t can be computed as follows.

$$\hat{\mathbf{X}}_t = \mathbf{U}^{(1)'} \dots \mathbf{U}^{(L)'} \hat{\mathbf{X}}_t^{(L)} \quad (7.6)$$

where $\hat{\mathbf{X}}_t^{(L)} = g(\mathbf{X}_s^{(L)})$ is the translated embedding from \mathcal{G}_s to \mathcal{G}_t at the L^{th} layer.

7.4.2 Graph Signal Comprehension via Curriculum Learning

In the presence of various proxy signals (e.g., node attributes, edges, and subgraph embeddings), the current GNNs pre-training methods mostly formulate the problem as a weighted combination of multiple proxy signal encoders by incorporating some hyper-parameters to balance their contributions. Here we propose to automatically learn a graph signal re-weighting scheme to capture the distribution of the real contribution of each proxy signal towards the downstream tasks. The learned sample weighting scheme specifies a curriculum under which the GNNs will be pre-trained gradually from the easy concepts to the hard concepts. In our problem setting, the curriculum is presented as a sequence of proxy signals that are extracted from the given graph.

Consider a GNN-based pre-training problem with K types of graph signals extracted from graphs at L levels of resolutions, we formulate the learning objective as follows.

$$\mathcal{L} = \underset{\Theta, \theta}{\text{argmin}} \sum_{l=1}^L \sum_{k=1}^K \sum_{\mathbf{s}_i^{(l,k)} \in \mathcal{G}_t^{(l)}} g_m(\mathbf{s}_i^{(l,k)}; \Theta) \mathcal{J}(\mathbf{Y}_t, h(\hat{\mathbf{X}}_t^{(l)}, \mathbf{s}_i^{(l,k)}, \theta)) + G(g_m(\mathbf{s}_i^{(l,k)}, \Theta), \lambda) \quad (7.7)$$

where $\mathbf{s}_i^{(l,k)}$ denotes the i^{th} sample of the type- k graph signals extracted from the l^{th} -layer of \mathcal{G}_t , $h(\hat{\mathbf{X}}_t^{(l)}, \mathbf{s}_i^{(l,k)}, \theta)$ denotes the pre-trained GNNs for predicting graph signal $\mathbf{s}_i^{(l,k)}$ with the input of graph representation $\hat{\mathbf{X}}_t^{(l)}$ at the l^{th} -layer, $g_m(\mathbf{s}_i^{(l,k)}; \Theta)$ denotes the time-varying weights for each training proxy signal $\mathbf{s}_i^{(l,k)}$, $G(g_m(\mathbf{s}_i^{(l,k)}, \Theta), \lambda)$ is the curriculum regularizer

parameterized by the learning threshold λ , and $\mathcal{J}(\mathbf{Y}_t, h(\hat{\mathbf{X}}_t^{(l)}, \mathbf{s}_i^{(l,k)}, \theta))$ denotes the prediction loss of a downstream task over a handful labeled examples \mathbf{Y}_t . Note that such labeled examples can be any types of graph signals, including the class-memberships of nodes, edges, and even subgraphs.

The objective in Eq. 7.7 can be interpreted as a teacher-student training paradigm [210, 211, 212]. In particular, the labeling function $h(\hat{\mathbf{X}}_t^{(l)}, \mathbf{s}_i^{(l,k)}, \theta)$ parametered by θ serves as the student model, which aims to pre-train GNNs by predicting the graph signal $\mathbf{s}_i^{(l,k)}$; the teacher model $g_m(\mathbf{s}_i^{(l,k)}; \Theta)$ parameterized by Θ aims to learn the time-varying weights to measure the importance of each graph signal $\mathbf{s}_i^{(l,k)}$, and provides guidance to pre-train GNNs on the target graph \mathcal{G}_t . To regularize the learning curriculum, one prevalent choice is to employ some predefined curriculums [187, 188, 189, 194, 203, 204, 205], which have been extensively explored in the existing literatures [210, 211, 212, 213]. Here we consider a curriculum regularizer derived from the robust non-convex penalties [213] as follows.

$$G(g_m(\mathbf{s}_i^{(l,k)}; \Theta), \lambda_1, \lambda_2) = \frac{1}{2} \lambda_2 g_m^2(\mathbf{s}_i^{(l,k)}; \Theta) - (\lambda_1 + \lambda_2) g_m(\mathbf{s}_i^{(l,k)}; \Theta) \quad (7.8)$$

where $\lambda = \{\lambda_1, \lambda_2\}$ are both positive hyperparameters. Since $G(g_m(\mathbf{s}_i^{(l,k)}; \Theta), \lambda_1, \lambda_2)$ is a convex function in terms of $g_m(\mathbf{s}_i^{(l,k)}; \Theta)$, the closed-form solution of our learning curriculum can be easily obtained as follows.

$$g_m(\mathbf{s}_i^{(l,k)}; \Theta^*) = \begin{cases} 1(\mathcal{J}(\mathbf{Y}_t, h(\hat{\mathbf{X}}_t^{(l)}, \mathbf{s}_i^{(l,k)}, \theta)) \leq \lambda_1) & \lambda_2 = 0 \\ \min(\max(0, 1 - \frac{\mathcal{J}(\mathbf{Y}_t, h(\hat{\mathbf{X}}_t^{(l)}, \mathbf{s}_i^{(l,k)}, \theta)) - \lambda_1}{\lambda_2}), 1) & \text{Otherwise} \end{cases} \quad (7.9)$$

The learning threshold $\lambda = \{\lambda_1, \lambda_2\}$ plays a key role in controlling the learning pace of MENTORGNN. When $\lambda_2 = 0$, the algorithm will only select the “easy” graph signals of $\mathcal{J}(\mathbf{Y}_t, h(\hat{\mathbf{X}}_t^{(l)}, \mathbf{s}_i^{(l,k)}, \theta)) \leq \lambda_1$ in training labeling function $h(\cdot)$, which is close to the binary scheme in self-paced learning [210, 213]. When $\lambda_2 \neq 0$, $g_m(\mathbf{s}_i^{(l,k)}; \Theta^*)$ will return continuous values in $[0, 1]$, and the graph signals with the loss of $\mathcal{J}(\mathbf{Y}_t, h(\hat{\mathbf{X}}_t^{(l)}, \mathbf{s}_i^{(l,k)}, \theta)) \leq \lambda_1 + \lambda_2$ will not be selected in pre-training. In practice, we gradually augment the value of $\lambda_1 + \lambda_2$ to enforce MENTORGNN learning from the “easy” graph signals to the “hard” graph signals by mimicking the cognitive process of human and animals [214]. Within each iteration with fixed λ_1 and λ_2 , MENTORGNN is optimized in an alternative fashion [215]. Specifically, when we train the student models $h(\hat{\mathbf{X}}_t^{(l)}, \mathbf{s}_i^{(l,k)}, \theta)$, we keep Θ fixed and minimize the prediction loss $\mathcal{J}(\mathbf{Y}_t, h(\hat{\mathbf{X}}_t^{(l)}, \mathbf{s}_i^{(l,k)}, \theta))$; when we train the teacher model $g_m(\mathbf{s}_i^{(l,k)}; \Theta)$, then we keep θ fixed and update the learning curriculum that will be used to guide the student models in the next iteration.

7.4.3 Generalization Bound for Domain Adaptive Graph Pre-training

Given a source graph \mathcal{G}_s and a target graph \mathcal{G}_t , how much can we guarantee the generalization performance of MENTORGNN in pre-training GNNs? Here, we present two approaches towards the generalization bound of MENTORGNN in the presence of multiple types of graph signals at different granularities: one by a union bound argument and the other relying on the graph signal learning curriculum (discussed in Subsection 7.4.2). Let $f_s(\cdot)$ be the true labeling function in \mathcal{G}_s , and $f_t^{(l,k)}(\cdot)$ be the true labeling function for the k^{th} type graph signals $\mathbf{s}^{(l,k)}$ at the l^{th} level of \mathcal{G}_t . One straightforward idea is to leverage the generalization error between $f_s(\cdot)$ in the source graph and each $f_t^{(l,k)}(\cdot)$ in the target graph by applying Theorem 7.1 multiple (i.e., $l \times k$) times. Following this idea, we can obtain the following worst-case generalization bound of MENTORGNN, which largely depends on the largest generalization error between $f_s(\cdot)$ and $f_t^{(l,k)}(\cdot)$.

Corollary 7.1. (Worst Case) Given \mathcal{G}_s and \mathcal{G}_t , $f_s(\cdot)$ is the labeling function in \mathcal{G}_s , and $f_t^{(l,k)}(\cdot)$ is the labeling function for the k^{th} type graph signals $\mathbf{s}^{(l,k)}$ at the l^{th} level of granularity in \mathcal{G}_t . Then, for any function class $\mathcal{H} \subseteq [0, 1]^\mathcal{X}$, and $\forall h \in \mathcal{H}$, the following inequality holds:

$$\epsilon_t(h) \leq \epsilon_s(h) + d_{\mathcal{H}}(\mathcal{G}_s, \mathcal{G}_t) + \max_{l,k} \{\min\{\mathbb{E}_{\mathcal{G}_s}[|f_s - f_t^{(l,k)}|], \mathbb{E}_{\mathcal{G}_t}[|f_s - f_t^{(l,k)}|]\}\} \quad (7.10)$$

The worst-case generalization bound shown in Corollary 7.1 could be pessimistic in practice, especially when the graphs are large and noisy. However, extensive work [187, 188, 189, 194, 205] has empirically shown that leveraging multiple types of graph signals often leads to the improved performance in many application domains, even in the presence of noisy data and irrelevant features. The key observation is that the information from multiple types of graph signals is often redundant and complementary. That is to say, when the majority of graph signals are related, a few irrelevant graph signals may not hurt the overall generalization performance too much. Hence, the natural question is: *can we obtain a better generalization bound than the one shown in Corollary 7.1?* To answer this question, we present a re-weighting case of the generalization bound for MENTORGNN, that is developed based on the obtained learning curriculum $g_m(\mathbf{s}^{(l,k)}, \Theta)$ as follows.

Corollary 7.2. (Re-weighting Case) Given \mathcal{G}_s and \mathcal{G}_t , $f_s(\cdot)$ is the labeling function in \mathcal{G}_s , and $f_t^{(l,k)}(\cdot)$ is the labeling function for the k^{th} type graph signals $\mathbf{s}^{(l,k)}$ at the l^{th} level of granularity in \mathcal{G}_t . Then, for any function class $\mathcal{H} \subseteq [0, 1]^\mathcal{X}$, $\forall h \in \mathcal{H}$, and $\sum_{l,k,i} g_m(\mathbf{s}_i^{(l,k)}, \Theta) = 1$, the following inequality holds:

$$\epsilon_t(h) \leq \epsilon_s(h) + d_{\mathcal{H}}(\mathcal{G}_s, \mathcal{G}_t) + \sum_{l,k,i} g_m(\mathbf{s}_i^{(l,k)}, \Theta) \min\{\mathbb{E}_{\mathcal{G}_s}[|f_s - f_t^{(l,k)}|], \mathbb{E}_{\mathcal{G}_t}[|f_s - f_t^{(l,k)}|]\} \quad (7.11)$$

Remark 1: Corollary 7.1 and Corollary 7.2 are all based on the true data distribution, and as such, the derived generalization bound might slightly deviate from the empirical results in practice. However, we argue that the state-of-the-art GNNs [178, 179, 181, 216] are reliable and accurate, of which the performance is very close to the true labeling function. With that being said, our theoretical results can well approximate the generalization bound of MENTORGNN in practice.

Remark 2: Corollary 7.1 reduces the multi-type multi-granularity of graph signals into an aggregated version with a linear combination using the time-varying weights $g_m(\mathbf{s}_i^{(l,k)}, \Theta)$. In fact, the worst-case generalization bound can be considered as a special case of the generalization bound shown in Corollary 7.2. Based on the following inequality, it is easy to prove that Corollary 7.2 provides a much tighter generalization bound than Corollary 7.1.

$$\begin{aligned} & \sum_{l,k,i} g_m(\mathbf{s}_i^{(l,k)}, \Theta) \min\{\mathbb{E}_{\mathcal{G}_s}[|f_s - f_t^{(l,k)}|], \mathbb{E}_{\mathcal{G}_t}[|f_s - f_t^{(l,k)}|]\} \\ & \leq \max_{l,k} \{\min\{\mathbb{E}_{\mathcal{G}_s}[|f_s - f_t^{(l,k)}|], \mathbb{E}_{\mathcal{G}_t}[|f_s - f_t^{(l,k)}|]\}\}. \end{aligned} \quad (7.12)$$

7.5 EXPERIMENTAL EVALUATION

We evaluate MENTORGNN by comparing it with the state-of-the-art methods in both the single-source graph transfer and the multi-source graph transfer settings. Moreover, we conduct a case study to study the generalization performance of MENTORGNN in the dynamic setting.

7.5.1 Experimental Setup

Data sets: Cora [217] data set is a citation network consisting of 2,708 scientific publications and 5,429 edges. Each edge in the graph represents the citation from one chapter to another. CiteSeer [217] data set consists of 3,327 scientific publications, which could be categorized into six classes, and this citation network has 9,228 edges. PubMed [218] is a diabetes data set, which consists of 19,717 scientific publications in three classes and 88,651 edges. The Reddit [219] data set was extracted from Reddit posts in September 2014. After pre-processing, three Reddit graphs are extracted and denoted as Reddit1, Reddit2, and Reddit3, respectively. Reddit1 consists of 4,584 nodes and 19,460 edges; Reddit2 consists of 3,765 nodes and 30,494 edges; Reddit3 consists of 4,329 nodes and 35,191 edges.

Baselines: We compare our method with the following baselines: (1) GCN [178]: graph

convolutional network, which is directly trained and tested on the target graph; (2) GAT [179]: graph attention network, which is directly trained and tested on the target graph; (3) DGI [205]: deep graph informax, which is directly trained and tested on the target graph; (4) GPA [191]: a policy network for transfer learning across graphs. Since GPA is designed for zero-shot setting, we fine-tune the pre-trained model with 100 iterations in the downstream tasks, and then make the final prediction; (5) MENTORGNN-V: one variant of MENTORGNN, which only considers node attributes as graph signal; (6) MENTORGNN-C: one variant of MENTORGNN, which does not utilize curriculum learning.

Implementation details: In the implementation of MENTORGNN, we consider two types ($K = 2$) of graph signals, i.e., node attributes and edges, at $L = 3$ levels of granularity. The output dimension of the first level of granularity is 500, and the output dimension of the second level of granularity is 100. We use Adam [220] as the optimizer with a learning rate of 0.005 and a two-layer GAT [179] with a hidden layer size of 50 as our backbone structure. MENTORGNN and its variants are trained for a maximum of 2000 episodes. The experiments are performed on a Windows machine with eight 3.8GHz Intel Cores and a single 16GB RTX 5000 GPU. The data and code are available in the anonymous link¹.

7.5.2 Single-Graph Transfer

In this subsection, we first consider the single-graph transfer setting, where we are given a single source graph (e.g., Cora [217], CiteSeer [217], and PubMed [218]) and a single target graph (e.g., Reddit1, Reddit 2, Reddit 3 [219]). Our goal is to pre-train GNNs across two graphs and then fine-tune the model to perform node classification on the target graph. We split the data set into training, validation, and test sets with the fixed ratio of 4%:16%:80%, respectively. Each experiment is repeated five times, and we report the average accuracy and the standard deviation of all methods across different data sets in Table 7.1. Results reveal that our proposed method and its variants outperform all baselines. Specifically, we have the following observations: (1) compared with the traditional GNNs, e.g., GCN, GAT and DGI, our method and its variants can further boost the performance by utilizing the knowledge learned from both the source graph and the target graph; (2) the performance of GPA is worse than GCN, GAT and DGI. A simple guess is that the performance of GPA largely suffers from the label scarcity issue in our setting. In particular, the results of GPA in [191] requires 67% samples for training on Reddit1, while we are only given 4% samples of Reddit1 for training. (3) compared with MENTORGNN-V which discards

¹<https://drive.google.com/drive/folders/107n6e0CjpDLZt0s6KAsEvDeMvo1Fuq0p?usp=sharing>

the link signal, MENTORGNN could further improve the performance by up to 3.31% by utilizing the structural information of the graph; (4) compared with MENTORGNN-C which does not utilize the curriculum learning, MENTORGNN boosts the performance by up to 1.88% in the setting of CiteSeer \rightarrow Reddit2. Overall, the comparison experiments verify the fact that MENTORGNN largely alleviates the impact of negative transfer and improves the generalization performance across all data sets.

Method	Cora \rightarrow Reddit1	Cora \rightarrow Reddit2	Cora \rightarrow Reddit3	CiteSeer \rightarrow Reddit2	PubMed \rightarrow Reddit3
GCN	0.8736 ± 0.0151	0.8996 ± 0.0158	0.8816 ± 0.0050	0.8996 ± 0.0158	0.8816 ± 0.0050
GAT	0.9420 ± 0.0154	0.9241 ± 0.0094	0.8985 ± 0.0088	0.9241 ± 0.0094	0.8985 ± 0.0088
DGI	0.7845 ± 0.0208	0.9062 ± 0.0071	0.8388 ± 0.0098	0.9062 ± 0.0071	0.8388 ± 0.0098
GPA	0.7011 ± 0.0116	0.7157 ± 0.0053	0.7271 ± 0.0063	0.7162 ± 0.0055	0.7210 ± 0.0103
MENTORGNN-V	0.9448 ± 0.0131	0.9584 ± 0.0045	0.9454 ± 0.0071	0.9637 ± 0.0072	0.9454 ± 0.0071
MENTORGNN-C	0.9508 ± 0.0097	0.9640 ± 0.0060	0.9655 ± 0.0072	0.9646 ± 0.0064	0.9734 ± 0.0104
MENTORGNN	0.9562 ± 0.0059	0.9815 ± 0.0053	0.9741 ± 0.0046	0.9834 ± 0.0048	0.9785 ± 0.0050

Table 7.1: Accuracy of node classification in the setting of single-graph transfer.

7.5.3 Multi-Graph Transfer

In this subsection, we evaluate our proposed model MENTORGNN in the multi-graph transfer setting, where multiple source graphs are given for pre-training GNNs. In our experiments, we use three source citation networks (i.e., Cora, CiteSeer, and PubMed) as the source graphs, and our goal is to transfer the knowledge extracted from multiple source graphs to improve the performance of the node classification task on a single target graph (e.g., Reddit1, Reddit 2, Reddit 3). Similar to the single-graph transfer setting, we use the same data split scheme to generate the training set, the validation set, and the testing set. The full results in terms of prediction accuracy and standard deviation over five runs are reported in Table 7.2. In general, our proposed method and its variants outperform all the baselines. Moreover, by combining the results (Table 7.1) in the single source-graph setting, it is interesting to see that the performances of GPA and MENTORGNN are both slightly improved when we leverage multiple source graphs simultaneously. For instance, when we consider the Reddit1 as the target graph, the accuracy scores of GPA and our method improve by 0.42% and 0.72% respectively compared with the single-graph transfer setting (i.e., Cora \rightarrow Reddit1).

7.5.4 Case Study: Metabolic Pattern Modeling on Protein-Protein Interaction Graph

Protein analysis is of great significance in many biological applications [221]. In this case study, we aim to apply MENTORGNN to study and capture the metabolic patterns of

molecules in the dynamic setting. To be specific, given a dynamic protein-protein interaction (PPI) graph named Breitkreutz [222] that consists of five snapshots $\mathcal{G} = \{\mathcal{G}^1, \dots, \mathcal{G}^5\}$. Each node represents a protein, and each timestamped edge stands for the interaction between two proteins. In order to train MENTORGNN, we manipulate the past snapshot as the source graph and the future snapshot as the target graph. The five snapshots naturally form four pairs of [source graph, target graph] in the chronological order, i.e., $\langle \mathcal{G}^1, \mathcal{G}^2 \rangle$, $\langle \mathcal{G}^2, \mathcal{G}^3 \rangle$, $\langle \mathcal{G}^3, \mathcal{G}^4 \rangle$, $\langle \mathcal{G}^4, \mathcal{G}^5 \rangle$. In our implementation, we treat the first three pairs as the training set and use the remaining one for testing. After obtaining the knowledge transfer function $g(\cdot)$ via MENTORGNN, we aim to reconstruct the last snapshot \mathcal{G}^5 by adapting from \mathcal{G}^4 . Figure 7.3 shows the synthetic \mathcal{G}^5 generated by MENTORGNN using different portions of the training set as well as the ground-truth \mathcal{G}^5 . In detail, Figure 7.3(a)-(c) show the generated \mathcal{G}^5 by learning from $\{\langle \mathcal{G}^1, \mathcal{G}^2 \rangle\}$, $\{\langle \mathcal{G}^1, \mathcal{G}^2 \rangle, \langle \mathcal{G}^2, \mathcal{G}^3 \rangle\}$, and $\{\langle \mathcal{G}^1, \mathcal{G}^2 \rangle, \langle \mathcal{G}^2, \mathcal{G}^3 \rangle, \langle \mathcal{G}^3, \mathcal{G}^4 \rangle\}$, respectively; and Figure 7.3(d) shows the ground-truth \mathcal{G}^5 by using t-SNE [223]. Each dot is the projected node in the embedding space, and the different colors correspond to three substructures of the PPI network. In general, we observe that Figure 7.3(c) is most similar to the ground-truth in Figure 7.3(d), Figure 7.3(b) is the next, and Figure 7.3(a) is the least similar one. Through this comparison, we know that our MENTORGNN indeed captures the evolution pattern, and the generated molecule graphs are more trustworthy given more temporal information.

Method	Reddit1	Reddit2	Reddit3
GCN	0.8736 ± 0.0151	0.8996 ± 0.0158	0.8816 ± 0.0050
GAT	0.9420 ± 0.0154	0.9241 ± 0.0094	0.8985 ± 0.0088
DGI	0.7845 ± 0.0208	0.9062 ± 0.0071	0.8388 ± 0.0098
GPA	0.7053 ± 0.0091	0.7193 ± 0.0027	0.7308 ± 0.0033
MENTORGNN-V	0.9586 ± 0.0054	0.9848 ± 0.0033	0.9801 ± 0.0024
MENTORGNN-C	0.9634 ± 0.0055	0.9844 ± 0.0045	0.9795 ± 0.0051
MENTORGNN	0.9621 ± 0.0015	0.9857 ± 0.0020	0.9811 ± 0.0025

Table 7.2: Accuracy of node classification in the setting of multi-graph transfer.

7.5.5 Parameter Sensitivity

In this subsection, we study the impact of the learning thresholds λ_1 and λ_2 on MENTORGNN. We conduct the case study in the single-graph transfer setting of Cora \rightarrow Reddit2.

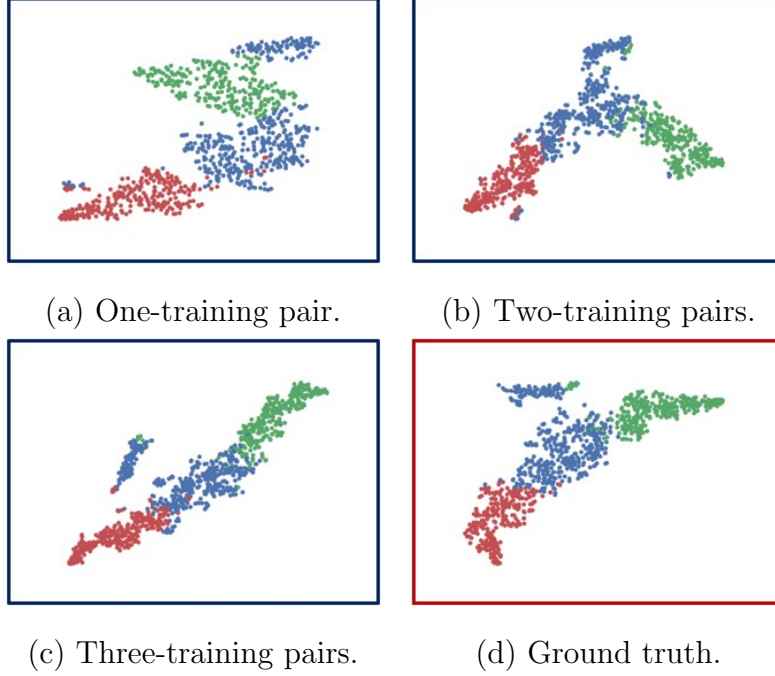
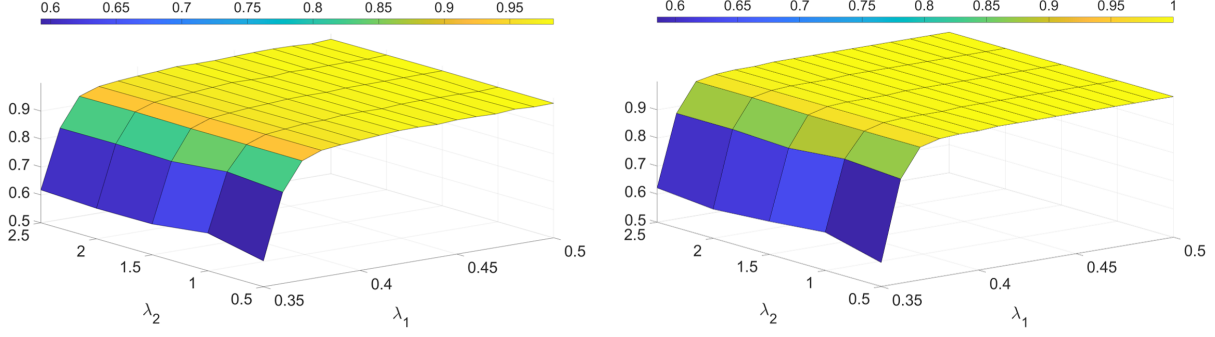


Figure 7.3: Metabolic pattern modeling on Breitkreutz graph. (a)-(c) show the network layouts of the generated \mathcal{G}^5 by learning from $\{< \mathcal{G}^1, \mathcal{G}^2 >\}$, $\{< \mathcal{G}^1, \mathcal{G}^2 >, < \mathcal{G}^2, \mathcal{G}^3 >\}$, and $\{< \mathcal{G}^1, \mathcal{G}^2 >, < \mathcal{G}^2, \mathcal{G}^3 >, < \mathcal{G}^3, \mathcal{G}^4 >\}$, respectively; and (d) shows the network layout of the ground-truth \mathcal{G}^5 .

In Figure 7.4, we report the training accuracy and the testing accuracy of our model with a diverse range of λ_1 and λ_2 . In general, we observe that (1) the model often achieves the best training accuracy and testing accuracy when $\lambda_2 = 1$; (2) given a fixed λ_2 , both of the training accuracy and testing accuracy are improved with the increasing of λ_1 . It is because the learned teacher model enforces the pre-trained GNNs learning from the “easy concepts” (i.e., small λ_1) to the “hard concepts” (i.e., large λ_1). In such way, the pre-trained GNNs encodes more and more graph signals in the learned graph representation and thus achieves better performance.

7.5.6 Time Complexity of Pre-Training

In this subsection, we show the time complexity of the proposed method by reporting the running time of MENTORGNN on a series of synthetic graphs with the increasing number of nodes. To control the number of nodes, we generate the synthetic graphs via ER algorithm [224] and use Gaussian noise to generate feature matrix with the dimension of 300. In Figure 7.5, we gradually increase the number of nodes from 500 to 4000, and MENTORGNN is trained for 1000 episodes. Based on the results in Figure 7.5, we observe



(a) Training accuracy on Cora \rightarrow Reddit2 (b) Testing accuracy on Cora \rightarrow Reddit2

Figure 7.4: Parameter analysis w.r.t. the learning thresholds λ_1 and λ_2

that the complexity of the proposed method is quadratic to the number of nodes.

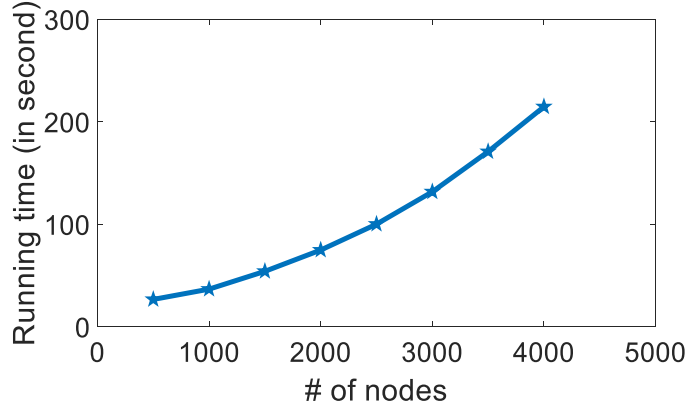


Figure 7.5: Time complexity

7.6 SUMMARY

Pre-training deep learning models is of key importance in many graph mining tasks. In this chapter, we study a novel problem named domain-adaptive graph pre-training, which aims to pre-train GNNs from diverse graph signals across disparate graphs. To address this problem, we present an end-to-end framework named MENTORGNN, which generates a learning curriculum to guide the pre-training of GNNs and thus transfers the knowledge from the source graph to the target graph. We also propose a new generalization bound for MENTORGNN in domain-adaptive pre-training. Extensive empirical results show that the pre-trained GNNs with fine-tuning over a few labels achieve significant improvements in the settings of single-graph transfer and multi-graph transfer.

CHAPTER 8: RARE CATEGORY REPRESENTATION LEARNING

8.1 OVERVIEW AND MOTIVATION

In many real-world applications, it is usually the case that the rare categories play an essential role despite their extreme scarcity. For example, in transaction networks, the vast majority of online transactions are legitimate, and only a small number may be fraudulent; in social networks, the majority users could be loss of sight to the underlying emerging trends, which could potentially turn into a burst in the near future; in computer networks, the percentage of network intrusion among the huge volumes of routine network traffic is small, but the loss might be significant.

One key challenge for analyzing the rare categories is the non-separable nature, i.e., the support regions of majority and minority in networks are usually non-separable. For example, in the financial fraud detection, the fraudulent people often try to camouflage their synthetic identities within the normal ones in order to bypass the fraud detection systems [225]; in the spam detection, the junk mails are deliberately made like the normal ones [226]. In addition, due to the highly skewness and non-separable nature of rare categories, labeling rare category examples is extremely expensive. In the extreme case, we may need to train the rare category analysis model from very few or only one labeled example. That said, it is therefore a very important and challenging task to identify such minority classes given that they are (1) highly skewed, (2) non-separable and (3) sparsely labeled. To be more specific, in this chapter, we want to answer the following two open questions: First (*T1. Embedding*), how to learn a salient rare category oriented embedding representation in order to better characterize them when the minority classes are non-separable from the majority classes? Second (*T2. Characterization*) how to accurately characterize the rare examples in the scarcity of label information?

Recently developed network embedding techniques [227, 228, 229], that encode graph structural information into a low dimensional representation, have received much success in boosting the performance of various network interface capabilities such as entity classification [230], author identification [231] and community detection [232]. However, these network embedding models are usually trained by uniformly drawing graph context without considering the scenario that the networks may exhibit imbalanced class distribution. Thus, the context information of rare categories may not be well preserved in the extracted training context pairs by existing context sampling methods [227, 228, 229, 230], which could be a key issue in the follow-up rare category characterization.

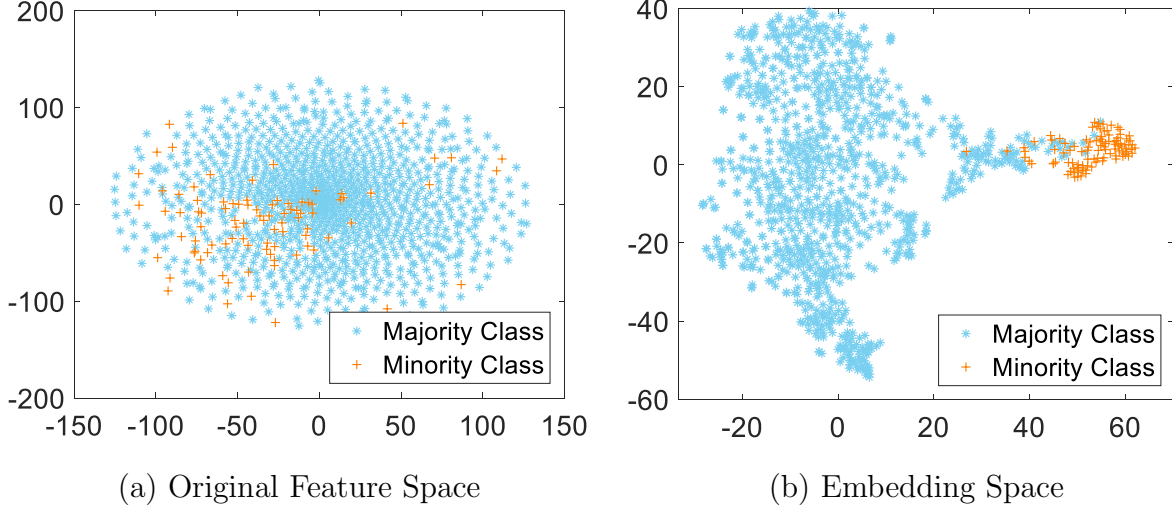


Figure 8.1: Rare category oriented network representation: the majority and minority classes are not separable in the original feature space, but become well separated in the embedding space induced by SPARC.

To counter the negative effects from learning in an imbalanced data set, extensive deep models [233, 234] have been proposed based on the re-sampling strategy [235], the cost sensitive learning [236] or adapting learning [237]. However, in the rare category characterization setting, training the aforementioned deep models in the scarcity of labeled rare category examples often suffers from the inevitable errors during label propagation. Thus, how to maintain a ‘safe and secure’ label propagation is of the key importance in learning the underlying distribution of rare categories.

To address the above challenges, in this chapter, we propose a generic rare category analysis framework named SPARC, that jointly predict the rare category examples and the neighborhood context in the graph. Our proposed SPARC is designed to jointly address two tasks, namely *T1. Embedding* and *T2. Characterization*, in a mutually beneficial way. In order to alleviate the influence of ambiguous data during model training, we integrate the self-paced learning paradigm into our framework to jointly select the rare category oriented graph contexts and maintain a reliable label propagation for training our proposed SPARC model.

The main contributions of this chapter are summarized below.

1. **Problem.** We formalize the problems of rare category oriented network representation and characterization learning in attributed networks, and identify their unique challenges from the nature of rare categories.

2. **Algorithms.** We propose a generic rare category analysis framework named SPARC, which is able to jointly predict the rare category examples and the neighborhood context in the attributed network.
3. **Evaluations.** Extensive experimental results on real networks demonstrate the performance of the proposed SPARC algorithm.

The rest of the chapter is organized as follows. Related works are reviewed in Section 8.2, followed by the notation and problem definition in Section 8.3. In Section 8.4, we present our proposed framework SPARC. Experimental results are reported in Section 8.5 before we conclude the chapter in Section 8.6.

8.2 RELATED WORK

In this section, we briefly review the related works regarding rare category analysis, network representation and curriculum learning.

8.2.1 Rare Category Analysis

Different from outlier detection [238, 239, 240] that targets to find abnormal patterns that do not conform to the expectation, and imbalanced classification [235] that aims to increase the overall accuracy, rare category analysis explores the compactness of the minorities and characterizes them from the highly skewed data sets. Rare category analysis (RCA) is first introduced by Pelleg and Moore [26], where the rare categories are defined as the minority clusters that exhibit a compact property in an imbalanced data distribution. The unique challenges of RCA come from the highly skewed data distribution, together with the non-separability nature of the rare categories from the majority classes. Up until now, researchers have proposed various methods for the RCA problem, such as sampling-based methods [2, 33, 235], ensemble-based methods [241], algorithm-adaptation-based methods [242], and maximum-margin-based methods [243]. Recently, [244] presented a deep representation model for the imbalanced data by enforcing the deep model to explore and maintain the inter-cluster and inter-class margins. [16] proposed a local graph clustering algorithm that identifies the structure-rich clusters by exploring the high-order structures in the neighborhood of the initial vertex in the given graph. However, very little work (if any) is devoted to learning a rare category oriented graph representation in the class-imbalanced networks. In this chapter, we propose a rare category oriented network embedding approach,

which jointly leverages the neighbored context information and the label information of rare examples, in order to better characterize the rare categories in the embedding space.

8.2.2 Network Representation

The pioneer works of graph representation can be traced back to the early 2000s, when many methods [245, 246, 247, 248] were developed for learning a low-dimensional graph representation with a minimized reconstruction error. While the network interface abilities of these methods may suffer from overfitting or poor scalability in real applications [231, 232]. Recently, a surge of research interests on network embedding by employing Skipgram model [249] has been observed in the network science. Among them, DeepWalk [227] firstly generalizes the Skipgram model to embed the graph context in a low-dimensional representation, where the graph context is extracted based on a truncated random walk; LINE [228] further extends the model by introducing an optimized objective function that incorporates the first-order and the second-order proximities to learn network representation; node2vec [229] preserves both homophily and structural equivalence relationships by generating the graph context with a biased random walk. In spite of the general-purposed network embedding approaches, a diversity of researches have been conducted to learn network representations for solving specific tasks with training examples or prior knowledge, such as author identification [231], entity classification [230] and community detection [232]. Despite the success of these methods, embedding representation of class-imbalanced networks has heretofore received little attention. In this chapter, we aim to learn a salient rare category oriented embedding representation, such that the minority classes are well separated from the majority classes, which facilitates the follow-up rare category analysis tasks such as detection [7, 29, 33], prediction [250], clustering [3, 16] and classification [6, 243, 251].

8.2.3 Curriculum Learning

Inspired by the cognitive process of humans, Bengio’s group proposes the curriculum learning (CL) paradigm, in which the underlying model is gradually trained from easy aspects of a task to the complex ones based on the predetermined ‘curriculum’ [214, 252]. This theory has been successfully applied to various applications, such as geometrical shape classification [214], teaching a robot of the concept of ‘graspability’ [253], grammar induction [254], etc. However, the heuristical curriculum design in CL turns out onerous or conceptually difficult in many real problems [210]. To eliminate this issue, Kumar et al. [210] propose a new learning paradigm named self-paced learning (SPL), which automatically learns a

‘curriculum’ by minimizing the loss function with a self-paced regularizer. In particular, SPL jointly updates the model parameters \mathbf{w} and the ‘curriculum’ indicator variable \mathbf{v} by optimizing the following objective:

$$\min_{\mathbf{w}, \mathbf{v}} \sum_i v_i \mathbb{L}(y_i, f(\mathbf{x}_i, \mathbf{w})) - \lambda \sum_i v_i, \quad s.t. \mathbf{v} \in [0, 1]^n, \quad (8.1)$$

where $\mathbb{L}(y_i, f(\mathbf{x}_i, \mathbf{w}))$ denotes the loss function, and λ is the self-paced parameter for controlling the learning pace. BCU [125] (Block-Coordinate Update) is usually adopted to solve the above bi-convex optimization problem by dividing the variables into disjoint blocks and alternatively optimizing one block while keeping the rest fixed. More recently, in [213], the authors develop a unified framework that improves CL and SPL by considering both the prior knowledge and the learning progress during training; in [255], the authors propose a self-paced co-training algorithm, which is proved to guarantee the theoretical effectiveness under the ϵ -expansion assumption. In this chapter, we advance the SPL scheme to the scenario of rare category analysis in the scarcity of labeled example, in order to gradually learn the rare category oriented network representation and the characterization model in a mutually beneficial way.

8.3 PRELIMINARIES

Throughout the chapter, we use lowercase letters to denote scalars (e.g., α), boldface lowercase letters to denote vectors (e.g., \mathbf{v}), and boldface uppercase letters to denote matrices (e.g., \mathbf{A}). Following the convention in Matlab, we represent the i^{th} row of matrix \mathbf{A} as $\mathbf{A}(\mathbf{i}, :)$, the j^{th} column of matrix \mathbf{A} as $\mathbf{A}(:, \mathbf{j})$, the entry of the i^{th} row and the j^{th} column in matrix \mathbf{A} as $\mathbf{A}(\mathbf{i}, \mathbf{j})$, and the transpose of matrix \mathbf{A} as \mathbf{A}^T . Given an attributed network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, where \mathcal{V} consists of n vertices, \mathcal{E} consists of m edges, and $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\} \in \mathbb{R}^{n \times r}$ denotes the set of nodes’ attributes, we use \mathbf{A} to represent the adjacency matrix of \mathcal{G} . Let $\mathbf{x}_1, \dots, \mathbf{x}_L \in \mathbb{R}^r$ denote the L labeled examples, where we assume there is at least one from each minority class; let $\mathbf{x}_{L+1}, \dots, \mathbf{x}_{L+U} \in \mathbb{R}^r$, where $n = L + U$, denote the U unlabeled examples, which either come from the majority class, i.e., $y_i \in \{0\}$, or the $c \geq 1$ minority classes, i.e., $y_i \in \{1, \dots, c\}$. With the above notation, our problem can be formally defined as follows:

Problem 8.1. Rare Category Embedding Representation (RCE)

Input: (i) an attributed network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, (ii) one-shot or few-shot labeled examples $\mathbf{x}_1, \dots, \mathbf{x}_L$, and (iii) the desired embedding dimension d .

Output: a d -dimensional embedding representation $\mathcal{E} \in \mathbb{R}^{n \times d}$ that preserves the underlying structure and context information, especially for the rare categories.

The output of Problem 8.1 is a low-dimensional matrix \mathcal{E} , where the i^{th} row (i.e., a d -dimensional vector \mathbf{e}_i) encodes the discriminative attributes and topology context information of node i that are beneficial for characterizing rare categories. The premise of network embedding models is to preserve different types of proximities between vertices and their neighborhood in a semi-supervised, e.g., [230], or unsupervised manner, e.g., [227, 228, 229]. However, the existing methods are not best suited for characterizing the rare categories, which are (1) under-represented in the given network, (2) non-separable from the majority classes, and (3) provided with scarce labeled examples in a massive attributed network. Here, we aim to learn a rare category oriented embedding representation that can incorporate the label and context information to better characterize the minority classes.

Problem 8.2. Rare Category Characterization (RCC)

Input: (i) an attributed network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, and (ii) one-shot or few-shot labeled examples $\mathbf{x}_1, \dots, \mathbf{x}_L$.

Output: a list of predicted rare category examples.

The main challenges of Problem 8.2 come from the highly skewed class membership and the scarce training data. Due to these issues, the existing imbalanced classification algorithms and semi-supervised learning techniques may suffer from overfitting and inevitable errors in label propagation. Notice that Problem 8.1 and Problem 8.2 are related with one another, and may be mutually beneficial if jointly solved in the sense that (1) incorporating the rare category oriented graph context information that is preserved in RCE is crucial for characterizing the rare examples in Problem 8.2, and (2) the trained RCC model could serve as a ‘supervisor’ to determine the rare category oriented graph context for learning the network representation in Problem 8.1. Due to these reasons, we present a generic rare category analysis framework in the following section, which is capable to learn from a handful or even one-shot training example and maintain a ‘safe and secure’ label propagation process in order to jointly address Problem 8.1 and Problem 8.2.

8.4 ALGORITHM

In this section, we present our rare category analysis framework SPARC, which simultaneously learns the graph embedding and predicts the rare category examples in a mutually

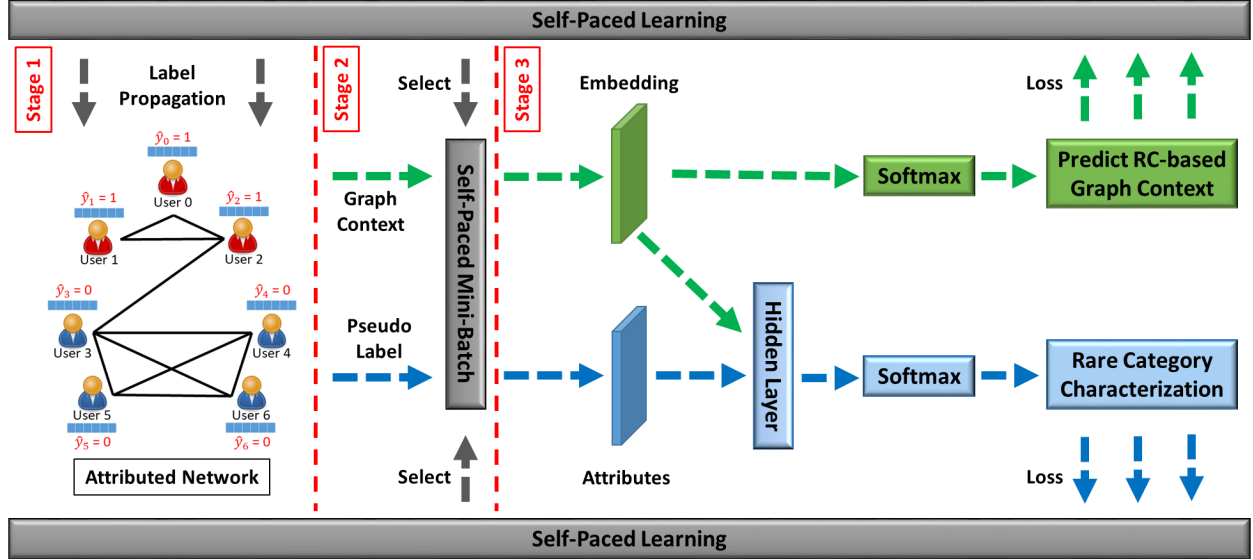


Figure 8.2: Illustration of the proposed SPARC framework. The minority class examples and the majority class examples are represented by the red and blue icons, respectively. In the given networks, only one minority class example (i.e., User 0) is labeled, while the remaining nodes are iteratively assigned with pseudo labels \hat{y} by the rare category characterization model, in order to learn the underlying distribution of the rare category.

beneficial way. We first formulate it as a generic optimization problem, and then present the details on how to jointly learn a rare category oriented embedding and characterize rare category examples within a self-paced learning paradigm.

8.4.1 A Generic Joint Learning Framework

To address the proposed RCE and RCC problems, our joint learning framework should take into consideration the following key aspects. First (*skewed distribution*), in order to detect and characterize the rare categories, our joint learning framework should have the capability to model the imbalanced class memberships in the given networks. Second (*non-separability*), the minority classes and majority classes are often non-separable in both the network topology space (i.e., \mathbf{A}) and the feature space (i.e., \mathbf{X}). Therefore, rare category oriented representation should result in the minority examples being largely separated from the majority classes in the embedding space. Third (*label scarcity*), due to the hardness and expensive cost of labeling rare category examples, our proposed framework should be capable to learn from few shot or even only one labeled rare category example.

We start by illustrating our framework in the binary case with only one majority class and one minority class in the given network. The extension to multi-class RCC problem

will be discussed later in Subsection 4.2. With these design objectives in mind, we propose a generic rare category analysis framework as an optimization problem with the following objective function:

$$\begin{aligned}
\mathcal{L}_b &= \mathcal{L}_s + \mathcal{L}_{rc} + \mathcal{L}_{tc} + \mathcal{L}_{sp} + \mathcal{L}_{co} \tag{8.2} \\
&= \underbrace{\sum_{i=1}^L c_{y_i, \hat{y}_i} \log \Pr(\hat{y}_i = 1 - y_i | \mathbf{x}_i, \mathbf{e}_i)}_{\mathcal{L}_s: \text{ cost sensitive learning}} \\
&\quad - \underbrace{\sum_{i=1}^{L+U} v_i^{(1)} \log \Pr(\hat{y}_i = 1 | \mathbf{x}_i, \mathbf{e}_i)}_{\mathcal{L}_{rc}: \text{ predict rare category examples}} \tag{8.3} \\
&\quad - \underbrace{\sum_{i=1}^{L+U} v_i^{(0)} \mathbb{E}_{(i, c, \gamma)} \log \sigma(\gamma \theta_{\mathbf{c}}^T \mathbf{e}_i)}_{\mathcal{L}_{tc}: \text{ predict graph context}} \\
&\quad - \underbrace{\sum_{i=1}^{L+U} \lambda^{(1)} v_i^{(1)} + \lambda^{(0)} v_i^{(0)}}_{\mathcal{L}_{sp}: \text{ self-paced regularizer}} - \underbrace{\alpha \sum_{i=1}^{L+U} v_i^{(1)} v_i^{(0)}}_{\mathcal{L}_{co}: \text{ consensus regularizer}}
\end{aligned}$$

where the objective function consists of five terms. The first term \mathcal{L}_s is the cost sensitive loss over the labeled data, in which c_{y_i, \hat{y}_i} denotes the misclassification cost of labeling node i belonging to class y_i into a different class $\hat{y}_i \neq y_i$. In particular, we let $c_{1,0} > c_{0,1} \geq 1$ in order to further penalize the errors of classifying the minority class examples into the majority class. The second term \mathcal{L}_{rc} corresponds to the characterization step, which learns the underlying distribution of the target rare category from both labeled and unlabeled data. The third term \mathcal{L}_{tc} corresponds to the embedding step, which minimizes the prediction loss regarding the sampled graph context pairs. The fourth term is the self-paced regularizer \mathcal{L}_{sp} , which globally maintains the learning pace of the embedding step (\mathcal{L}_{tc}) and the characterization step (\mathcal{L}_{rc}) by utilizing self-paced vectors, i.e., $\mathbf{v}^{(0)}, \mathbf{v}^{(1)} \in [0, 1]^n$, respectively. The last term is the consensus regularizer \mathcal{L}_{co} , where α is a positive constant to balance the impact of this term on the overall objective function.

Based on Eq. 8.2, we propose the overall SPARC framework as shown in Figure 8.2, where the RCE and RCC models are gradually trained in a mutually beneficial way via multiple self-paced cycles to maintain a ‘safe and secure’ label propagation. In particular, within each training cycle, our proposed framework SPARC can be decomposed into three stages. In the first stage, SPARC assigns the pseudo labels to the potential rare category examples

based on the current prediction model. The second stage is the key step of our proposed SPARC model, which jointly selects the rare category oriented graph contexts and reliable predictions for training RCE and RCC models. The third stage involves the construction of two deep neural networks (DNN), including the RCE DNN (upper level) and the RCC DNN (lower level). By using the sampled graph context in Stage 2, the RCE DNN is trained to learn a salient embedding space for the RCC problem. Given the input feature vector \mathbf{x}_i and the learned embedding vector \mathbf{e}_i , the RCC DNN is updated by learning from both the labeled and unlabeled data. In particular, the posterior probability $Pr(y_i|\mathbf{x}_i, \mathbf{e}_i)$ in Eq. 8.2 is written as $Pr(y_i|\mathbf{x}_i, \mathbf{e}_i) = \frac{\exp[\mathbf{h}^k(\mathbf{x}_i)^T, \mathbf{h}^l(\mathbf{e}_i)^T]\theta_y}{\sum_{y'} \exp[\mathbf{h}^k(\mathbf{x}_i)^T, \mathbf{h}^l(\mathbf{e}_i)^T]\theta_{y'}}$, where \mathbf{h}^k denotes the k^{th} hidden layer, and $[\cdot, \cdot]$ denotes the concatenation operator of two vectors. In the next cycle, the learned RCC DNN will be used for label propagation in Stage 1, and the learned RCE will be fed into the RCC DNN in Stage 3. To further show how SPARC works, we focus on the following three aspects.

Impact of the Self-Paced Learning: In the case of non-separable rare categories with scarce training data, deep discriminative models often suffer from the errors during label propagation. To address this issue, our framework exploits the SPL scheme to gradually learn from the labeled and unlabeled data, which has demonstrated its robustness in the semi-supervised setting [253, 256]. For jointly modeling the RCE and RCC problems, we design our SPARC framework via dual-level SPL, by leveraging the idea of co-training [70, 255]. In particular, the overall objective of SPARC in Eq. 8.2 can be interpreted as the sum of a self-paced RCE model \mathcal{L}_{RCE} , a self-paced RCC model \mathcal{L}_{RCC} and a consensus regularizer \mathcal{L}_{co} as follows:

$$\mathcal{L}_b = \mathcal{L}_{RCC} + \mathcal{L}_{RCE} + \mathcal{L}_{co} \quad (8.4)$$

where

$$\mathcal{L}_{RCC} = \mathcal{L}_s - \sum_{i=1}^{L+U} v_i^{(1)} \log Pr(\hat{y}_i = 1|\mathbf{x}_i, \mathbf{e}_i) - \sum_{i=1}^{L+U} \lambda^{(1)} v_i^{(1)} \quad (8.5)$$

$$\mathcal{L}_{RCE} = - \sum_{i=1}^{L+U} v_i^{(0)} \mathbb{E}_{(i,c,\gamma)} \log \sigma(\gamma \theta_{\mathbf{c}}^T \mathbf{e}_i) - \sum_{i=1}^{L+U} \lambda^{(0)} v_i^{(0)} \quad (8.6)$$

In other words, \mathcal{L}_{RCC} is mainly used in RCC DNN to address Problem 2, whereas \mathcal{L}_{RCE} is mainly used in RCE DNN to address Problem 1. In addition, the consensus regularizer \mathcal{L}_{co} is imposed on both \mathcal{L}_{RCC} and \mathcal{L}_{RCE} to ensure the ‘learning curriculum’ generated by SPARC emphasizes on learning the underlying distribution of rare categories.

We adopt BCU [125] to update the dual-level SPL in an alternative way. When we

update the self-paced vector $\mathbf{v}^{(1)}$, the partial derivative of Eq. 8.2 with respect to $v_i^{(1)}$ (the i^{th} element of $\mathbf{v}^{(1)}$), $i = 1, \dots, n$, can be derived as:

$$\frac{\partial \mathcal{L}_b}{\partial v_i^{(1)}} = -\log \Pr(\hat{y}_i = 1 | \mathbf{x}_i, \mathbf{e}_i) - \lambda^{(1)} - \alpha v_i^{(0)} \quad (8.7)$$

Thus, the closed-form solution to update $v_i^{(1)}$ is

$$v_i^{(1)} = \begin{cases} 1 & -\log \Pr(\hat{y}_i = 1 | \mathbf{x}_i, \mathbf{e}_i) < \lambda^{(1)} + \alpha v_i^{(0)} \\ 0 & \text{Otherwise} \end{cases} \quad (8.8)$$

By updating self-paced vector $\mathbf{v}^{(1)}$, we can identify the reliable predictions in order to learn the underlying distribution of rare category in RCC DNN. To be specific, given the self-paced parameter $\lambda^{(1)}$, examples with a higher confidence to belong to the minority class, i.e., $\log \Pr(\hat{y}_i = 1 | \mathbf{x}_i, \mathbf{e}_i) > -\lambda^{(1)} - \alpha v_i^{(0)}$, are assigned with $v_i^{(1)} = 1$; otherwise, $v_i^{(1)} = 0$.

When we update the $\mathbf{v}^{(0)}$, the similar closed-form solution can be derived as follows.

$$v_i^{(0)} = \begin{cases} 1 & -\log \sigma(\gamma \mathbf{w}_c^T \mathbf{e}_i) < \lambda^{(0)} + \alpha v_i^{(1)} \\ 0 & \text{Otherwise} \end{cases} \quad (8.9)$$

The goal of this step is to formally define which graph context pairs (i, c, γ) will be fed into the training pool for learning the network embedding \mathcal{E} . In each iteration, the graph context pairs (i, c, γ) whose prediction losses are smaller than a certain threshold, i.e., $-\log \sigma(\gamma \mathbf{w}_c^T \mathbf{e}_i) < \lambda^{(0)} + \alpha v_i^{(1)}$, are selected ($v_i^{(0)} = 1$) to be fed into the following RCE DNN.

Furthermore, the consensus regularizer \mathcal{L}_{co} is imposed on the self-paced vectors $\mathbf{v}^{(0)}$ and $\mathbf{v}^{(1)}$ to ensure the selected graph context pairs (i, c, γ) are rare category oriented and within the user-defined level of learning difficulty. The constant α is used to balance the two learning principals, i.e., learning from rare category related graph context ($v_i^{(1)} = 1$) or learning graph context with less difficulty ($v_i^{(0)} = 1$). To be more specific, when α is larger, $\mathbf{v}^{(0)}$ will be closer to $\mathbf{v}^{(1)}$ such that more rare category related graph context will be selected to train RCE DNN; when α is smaller, $\mathbf{v}^{(0)}$ will select more vertices with ‘easy’ graph context.

RCE in the Scarcity of Labeled Minority Classes Examples: To learn the graph embedding that preserves the similarities among rare category examples while maximally separating these examples from the majority class examples, we follow the negative-sampling-based graph embedding models [228, 229], which minimize the cross entropy loss of predicting graph context pairs (i, c) to positive labels ($\gamma = 1$) or negative labels ($\gamma = -1$) as follows.

Algorithm 8.1: Rare Category Oriented Context Sampling

Require:

Graph \mathcal{G} , indicator vector \mathbf{I} and parameters μ , r and s_{neg} .

Ensure:

Rare category oriented graph context pairs.

- 1: Draw a number $random \sim Unif(0, 1)$.
 - 2: **if** $random < r$ **then**
 - 3: Uniformly sample a random walk \mathbb{W} of length μ and generate one positive graph context pair $(i, c, +1)$ and s_{neg} negative graph context pairs $(i, c, -1)$ by existing methods [227, 228].
 - 4: **else**
 - 5: Shuffle an initial vertex v_i from the nonzero elements in \mathbf{I} and conduct a random walk \mathbb{W} of length μ .
 - 6: Uniformly sample a positive graph context pair $(i, c, +1)$ with $I(i) = I(c)$ and s_{neg} negative graph context pairs $(i, c, -1)$ with $I(i) \neq I(c)$.
 - 7: **end if**
-

$$\min -\mathbb{E}_{(i,c,\gamma)} \log \sigma(\gamma \theta_c^T \mathbf{e}_i) \quad (8.10)$$

where $\sigma(x)$ is the sigmoid function, i.e., $\sigma(x) = 1/(1 + e^{-x})$. Recently, [230] further developed a label informed graph embedding method that injects the label information into the sampled positive graph context pairs and demonstrated its effectiveness in the semi-supervised learning setting. However, in our problem setting, the above methods may fail due to the following reasons: (1) the learned embeddings (e.g., [228, 229, 230]) are not sensitive to the minority class examples since the sampled graph context pairs using the above methods may mostly come from the majority classes; (2) the scarcity of the labeled minority class examples imposes severe limitation on sampling rare category oriented graph context pairs. In the extreme case, when there is only one labeled minority example, the existing method [230] cannot generate the label informed positive context pairs $(i, c, +1)$ as there is no way to find a pair of nodes (i, c) from the same minority class within the labeled set.

To address the above deficiencies, we develop a rare category oriented context sampling strategy in Algorithm 8.1. The given input of Algorithm 8.1 is the graph \mathcal{G} , an indicator vector \mathbf{I} , and some constant parameters including the length of the performed random walks μ , the probability r and the number of negative samples s_{neg} . In particular, the indicator vector \mathbf{I} can be generated by any offline RCC models, while our proposed SPARC model utilizes the self-paced vector $\mathbf{v}^{(1)}$ to serve as the indicator vector \mathbf{I} that determines the potential rare category examples based on the current RCC DNN. Algorithm 8.1 samples

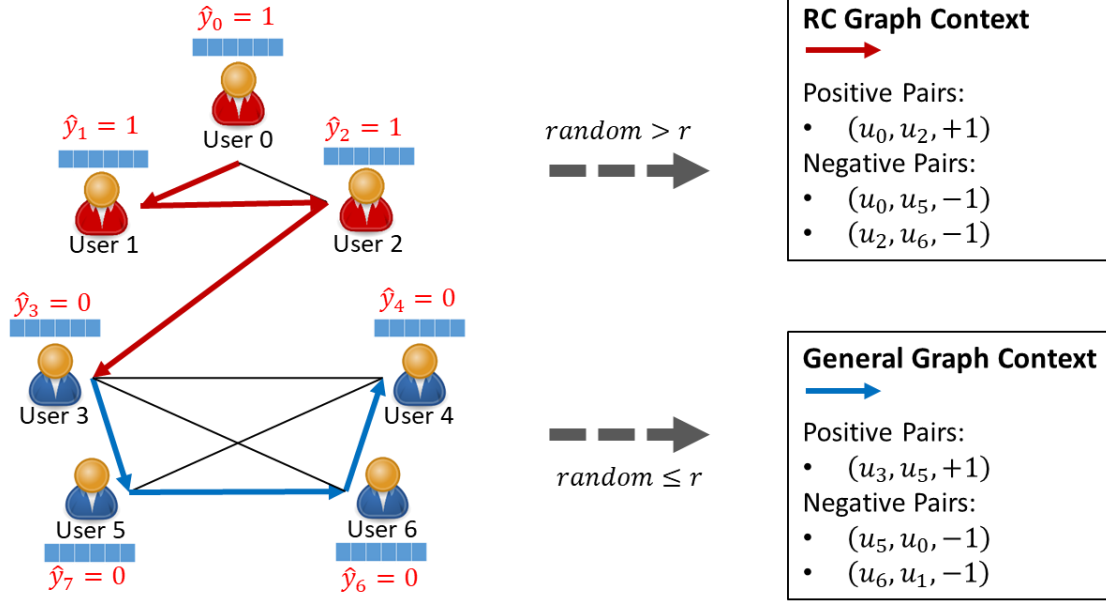


Figure 8.3: An example of sampling dual graph context by Algorithm 6.1, when the window size $d = 2$, $\mu = 3$ and $s_{neg} = 2$. In particular, if $random > r$, we sample the rare category related context pairs based on the random walk (e.g., red path) starting from the labeled rare category example (e.g., User0); otherwise, we extract the graph context by uniformly sampling a random walk (e.g., blue path) from the given network.

two types of graph contexts, i.e., the general graph context and the rare category related graph context, where the first one preserves the general graph structure, while the second one focuses on learning the local context of the rare category examples. An example of sampling graph context is shown in Figure 8.3. With probability r , the general graph contexts are extracted by the existing methods [227, 228]. With probability $(1 - r)$, we sample rare category related context pairs (i, c, γ) . In particular, when $\gamma = +1$, node i and node c are believed to belong to the same minority class, i.e., $I(i) = I(c)$; when $\gamma = -1$, node i and node c are believed to belong to the different classes, i.e., $I(i) \neq I(c)$.

Remarks: We would like to emphasize that Algorithm 8.1 is designed for the class-imbalanced networks. More specifically, (1) to counter the skewed distribution when sampling graph pairs, Algorithm 8.1 uses a probability r to balance the proportion of general graph context pairs and the rare category graph context pairs; (2) in scarcity of labeled rare category examples, our method generates rare category oriented graph context pairs (i, c, γ) based on the pseudo labels (i.e., indicator vector I) instead of using real labels to alleviate the limitation of insufficient labeled examples.

RCC with Respect to Labeled Majority and Minority Class Examples: Here,

we show the underlying training process of the RCC DNN regarding the labeled majority class examples and the labeled minority class examples. For each labeled minority class example i , the hidden layers of RCC DNN are updated by minimizing the following objective.

$$\begin{aligned}\mathcal{L}_{min} &= c_{1,0} \log Pr(\hat{y}_i = 0|\mathbf{x}_i, \mathbf{e}_i) - v_i^{(1)} \log Pr(\hat{y}_i = 1|\mathbf{x}_i, \mathbf{e}_i) \\ &= c_{1,0} \log(1 - Pr(\hat{y}_i = 1|\mathbf{x}_i, \mathbf{e}_i)) - v_i^{(1)} \log Pr(\hat{y}_i = 1|\mathbf{x}_i, \mathbf{e}_i)\end{aligned}\quad (8.11)$$

To further simplify the above objective, we let $a = 1 - Pr(\hat{y}_i = 1|\mathbf{x}_i, \mathbf{e}_i)$ and $b = Pr(\hat{y}_i = 1|\mathbf{x}_i, \mathbf{e}_i)$. Since $(a + b)^2 \geq 4ab$, we have $2\log(a + b) \geq \log 4 + \log a + \log b$, which could be written in terms of $Pr(\hat{y}_i = 1|\mathbf{x}_i, \mathbf{e}_i)$ as follows:

$$\log(1 - Pr(\hat{y}_i = 1|\mathbf{x}_i, \mathbf{e}_i)) \leq -\log Pr(\hat{y}_i = 1|\mathbf{x}_i, \mathbf{e}_i) - \log 4 \quad (8.12)$$

By substituting Eq. 8.12 back into Eq. 8.11, we have: $\mathcal{L}_{min} \leq -(c_{1,0} + v_i^{(1)}) \log Pr(\hat{y}_i = 1|\mathbf{x}_i, \mathbf{e}_i) - c_{1,0} \log 4$.

Similar as above, for each labeled majority class example j , the RCC DNN aims to minimize the following objective:

$$\mathcal{L}_{maj} = (c_{0,1} - v_j^{(1)}) \log Pr(\hat{y}_j = 1|\mathbf{x}_j, \mathbf{e}_j) \quad (8.13)$$

Remarks: Based on the above derived objectives regarding labeled majority class examples and labeled minority class examples, we have the following observations: (1) \mathcal{L}_{min} is monotonically decreasing over $Pr(\hat{y}_i = 1|\mathbf{x}_i, \mathbf{e}_i)$ as $c_{1,0} > 1$ and $v_i^{(1)} \in \{0, 1\}$. That is, the probability of the labeled minority class examples ($y = 1$) belonging to the minority class $Pr(\hat{y}_i = 1|\mathbf{x}_i, \mathbf{e}_i)$ is maximized along with minimizing \mathcal{L}_{min} . (2) \mathcal{L}_{maj} is monotonically nondecreasing over $Pr(\hat{y}_j = 1|\mathbf{x}_j, \mathbf{e}_j)$ as $c_{0,1} - v_j^{(1)} \geq 0$. That is, the probability of the labeled majority class examples ($y = 0$) belonging to the minority class $Pr(\hat{y}_i = 1|\mathbf{x}_j, \mathbf{e}_j)$ is minimized along with optimizing \mathcal{L}_{min} . (3) The overall objective of SPARC emphasizes on learning the underlying distribution of the minority class as $c_{1,0} + v_i^{(1)} > c_{0,1} - v_j^{(1)}$. For a special case, when $c_{0,1} - v_j^{(1)} = 0$, i.e., $c_{0,1} = v_j^{(1)} = 1$, the labeled majority class examples with $v_j^{(1)} = 1$ are not taken into consideration. The intuition is that our proposed framework SPARC is designed to be tolerant of the majority class example j that may not be separable from the minority class examples, i.e., $\log Pr(\hat{y}_j = 1|\mathbf{x}_j, \mathbf{e}_j) > -\lambda^{(1)} - \alpha v_i^{(0)}$.

8.4.2 Optimization Algorithm

To optimize the overall objective function in Eq. 8.2, we adopt stochastic gradient descent (SGD) [257] to train our model in an alternative way. The optimization algorithm is sum-

marized in Algorithm 8.2. The given input is the attributed network \mathcal{G} , labels of training data $\mathbf{Y} = \{y_1, \dots, y_L\}$ and some parameters including batch iterations T_1 and T_2 , batch size N_1 and N_2 , self-paced parameters $\lambda^{(0)}$ and $\lambda^{(1)}$ and α . Within each iteration, we first sample N_1 labeled examples and update RCC DNN by taking a gradient step of $\mathcal{L}_s + \mathcal{L}_{rc}$. Note that, in the first iteration, $\mathcal{L}_{rc} = 0$ as $\mathbf{v}^{(0)}$ and $\mathbf{v}^{(1)}$ are initialized to all-zero vectors. We then optimize the RCE DNN over N_2 sampled graph context pairs (i, c, γ) . The above procedures are repeated with T_1 and T_2 times respectively. Step 9 updates the self-paced vectors $\mathbf{v}^{(0)}$ and $\mathbf{v}^{(1)}$, and Step 10 augments the self-paced parameters $\lambda^{(0)}$, $\lambda^{(1)}$ in order to learn the more ‘difficult’ concept in the next iterations. The algorithm stops when the user-defined stopping criterions are satisfied. Algorithm 8.2 can be extended to solve the multi-class RCE and RCC problems by optimizing the following objective function.

$$\mathcal{L}_m = \sum_{i=1}^L \sum_{c=1}^C c_{y_i, \hat{y}_i} \log Pr(\hat{y}_i = c | \mathbf{x}_i, \mathbf{e}_i) \quad (8.14)$$

$$\begin{aligned} & - \sum_{i=1}^{L+U} \sum_{c=1}^C v_i^{(c)} \log Pr(\hat{y}_i = c | \mathbf{x}_i, \mathbf{e}_i) + v_i^{(0)} \log \mathbb{E}_{(i, c, \gamma)} \log \sigma(\gamma \theta_{\mathbf{c}}^T \mathbf{e}_i) \\ & - \sum_{i=1}^{L+U} [\sum_{c=1}^C \lambda^{(c)} v_i^{(c)} + \lambda^{(0)} v_i^{(0)}] \\ & - \alpha \sum_{i=1}^{L+U} \sum_{c=1}^C v_i^{(c)} v_i^{(0)} \end{aligned} \quad (8.15)$$

where $\mathbf{v}^{(c)} \in [0, 1]^n$ denotes the self-paced vector of class c , and $\lambda^{(c)}$ is the self-paced parameter that controls the learning pace.

Compared with the objective function in Eq. 8.2 for the binary case, the only difference is that each term in Eq. 8.14 is defined by cumulating the prediction loss over multiple classes instead of only two. Following Algorithm 8.2, our proposed framework SPARC is iteratively trained based on the extracted graph context pairs and label propagated examples that come from different classes. In the end, SPARC returns the rare category oriented network representation and the predication labels of each vertex in the given network.

8.5 EXPERIMENTAL EVALUATION

In this section, we demonstrate the performance of our proposed SPARC algorithm in the sense of the saliency of the RCE representation and the accuracy of the RCC classifier for rare category analysis. Moreover, we also present a case study to illustrate the impacts and

Algorithm 8.2: SPARC: Joint Learning Framework for RCC and RCE

Require:

Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, labels $\mathbf{Y} = \{y_1, \dots, y_L\}$, and parameters $T_1, T_2, N_1, N_2, \lambda^{(0)}, \lambda^{(1)}, \alpha$.

Ensure:

- (1) Rare category oriented embedding $\mathcal{E} \in \mathcal{R}^{|\mathcal{V}| \times d}$;
 - (2) A list of predicted rare category examples.
 - 1: **while** Stopping criterion is not satisfied **do**
 - 2: **for** $t = 1 : T_1$ **do**
 - 3: Sample N_1 labeled instances and update hidden layers' parameters θ by taking a gradient step for $\mathcal{L}_s + \mathcal{L}_{rc}$.
 - 4: **end for**
 - 5: **for** $t = 1 : T_2$ **do**
 - 6: Sample N_2 graph context pairs by Algorithm 8.1 with indicator vector $\mathbf{v}^{(0)}$.
 - 7: Update the rare category oriented embedding \mathcal{E} by taking a gradient step for \mathcal{L}_{tc}
 - 8: **end for**
 - 9: Update $\mathbf{v}^{(0)}$ and $\mathbf{v}^{(1)}$ separately based on Eq. 8.9 and Eq. 8.8, and make sure all the labeled rare examples are selected.
 - 10: Augment $\lambda^{(0)}, \lambda^{(1)}$.
 - 11: **end while**
-

the underlying procedures of the self-paced learning in our proposed SPARC framework.

8.5.1 Experiment Setup

Data sets: The statistics of all real data sets used in our experiments are summarized in Table 8.1.

- **Collaboration Networks:** DBLP¹ data set provides the bibliographic information of the publications in IEEE Visualization Conference during 1990 ~ 2015. Each vertex represents a paper, and an edge exists if and only if when one paper cites another paper. The class membership is defined based on 20 research topics in the data visualization area. SO² data set is collected from Stack Overflow, where each node represents a Stack Overflow user and each edge indicates one comment from one user to another. The class memberships are defined based on the users' reputation score, i.e., the majority of the users have regular scores (< 3000) while only a few users have considerably high scores (> 3000).

¹<http://www.vispubdata.org/site/vispubdata/>

²<https://archive.org/details/stackexchange>

Category	Network	Classes	Smallest Class	Nodes	Edges
Collaboration	DBLP	20	1.91%	2,309	7,913
	SO	2	1.29%	3,262	19,926
NLP	Citeseer	6	3.42%	3,327	4,732
	Cora	7	1.14%	2,708	5,429
	Pubmed	3	4.05%	19,717	44,318
Social	Epinion	19	1.38%	75,879	508,837

Table 8.1: Statistics of the network data sets.

- **NLP Networks:** Citeseer, Cora and Pubmed are three text classification data sets³, where each node represents a document and each edge indicates the citation link between the documents. The bag-of-words representation is adopted as the node attributes in these three data sets. NELL [230] is an entity classification data set, where the entities and the relations between entities are extracted from the NELL knowledge database, and the attributes of each entity are obtained by the bag-of-words representation of the associated description text.
- **Social Network:** Epinion [174] data set is a who-trust-whom social network, where each node represents a user, and an edge exists if and only if two users both give positive reviews (rating ≥ 2.5 out of 5) to the same item. The class membership of each user is defined based on the most frequently reviewed item category.

Comparison Methods: We compare SPARC with the recent network embedding and rare category analysis models. DeepWalk [227] and LINE [228] are unsupervised network embedding algorithms, which learn embedding based on word2vec model and use logistic regression as the classifier. PLANETOID [230] is a semi-supervised framework for attributed networks, which learns an embedding based on both topology context and label context to better infer the class memberships of unlabeled examples. GRADE [33] is a graph based rare category detection algorithm that takes the input of adjacency matrix \mathbf{A} , while RACH [243] is a rare category characterization algorithm that takes the feature vectors \mathbf{X} as input.

Implementation details: The experiments are performed on a Windows machine with eight 3.8GHz Intel Cores and a single 16GB RTX 5000 GPU. All the data and code are public available at ⁴.

³<http://lincs.umiacs.umd.edu/projects//projects/lbc/>

⁴<http://publish.illinois.edu/daweizhou/files/2019/10/SPARC.zip>

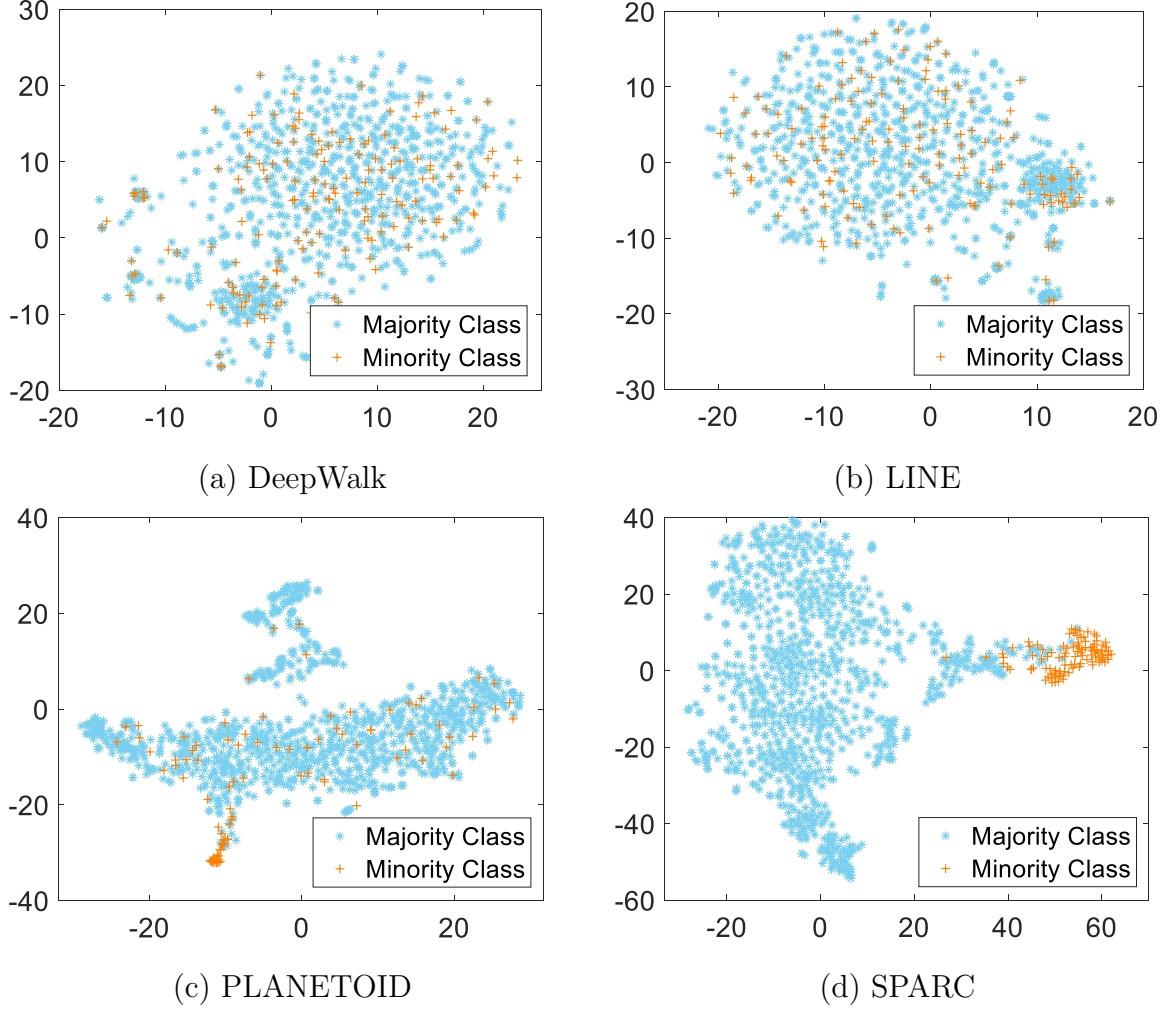


Figure 8.4: 2-D t-SNE visualization of network embedding.

8.5.2 Network Layout

A simple but useful way to evaluate the network representation approaches is to visualize the network layout in the embedding space, and we take the NLP network that extracted from the Pubmed data set for an example. We separate the network into binary classes by letting the smallest class be the minority class and the residual be the majority class. Laying out this NLP network is very challenging as the data is noisy and the classes, i.e., categories of documents, always overlap with one another. We compare our proposed SPARC algorithm with three state-of-the-art network embedding algorithms including two unsupervised methods, i.e., DeepWalk and LINE, and one semi-supervised method, i.e., PLANETOID. Note that, the unsupervised embedding methods only take as input the graph \mathcal{G} , while the semi-supervised methods, i.e., PLANETOID and our proposed SPARC, are further pro-

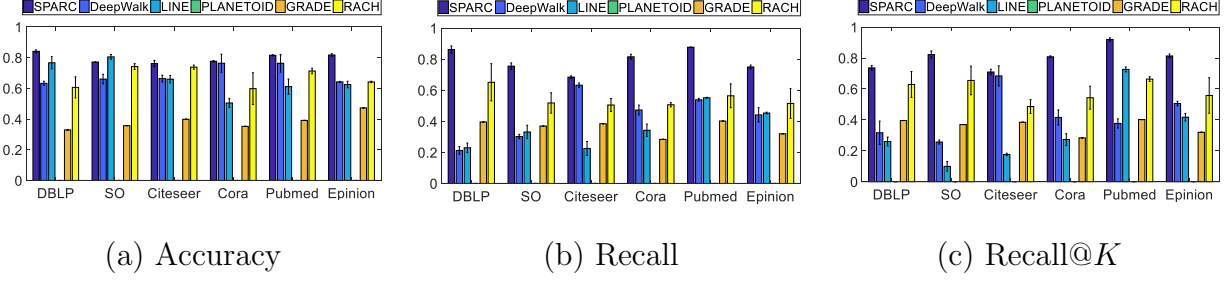


Figure 8.5: Effectiveness analysis with one labeled minority class example.

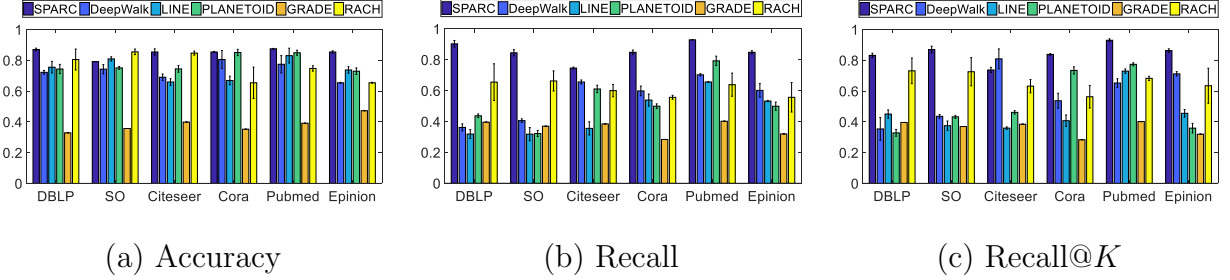


Figure 8.6: Effectiveness analysis with 5% labeled minority class examples.

vided with the training data consisting of labeled examples from both the majority and the minority classes. In particular, we first map the given network into a 129-dimensional space with different embedding methods, and then we employ the nonlinear dimensionality reduction method, i.e., t-SNE [258], to a 2-D space for the better visualization, which is shown in Figure 8.4. We can clearly observe that (1) the semi-supervised embedding methods perform better than the unsupervised methods as the classes are better separated; (2) with the same amount of training data, the rare examples are better clustered by using SPARC than PLANETOID. One explanation is that PLANETOID samples the graph context without considering that the class membership is imbalanced, which results in the neighborhood context of rare examples not well preserved in the embedding space.

8.5.3 Effectiveness Analysis

The comparison results in terms of effectiveness across a diverse set of networks by using 1, 5% and 10% labeled rare category examples are shown from Figure 8.5 to Figure 8.7, where the height of the bars indicates the average value of evaluation metrics, and the error bars represent the standard deviation of evaluation metrics in multiple runs by randomly shuffling the initial training examples. Note that PLANETOID can not be trained with

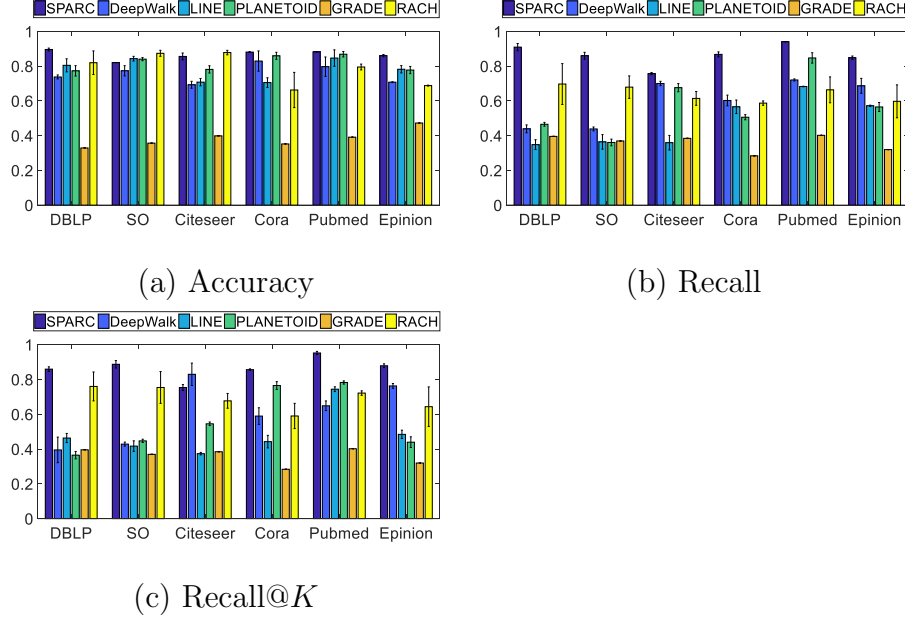


Figure 8.7: Effectiveness analysis with 10% labeled minority class examples.

only one labeled rare category example, thus the corresponding results are not reported in Figure 8.5. By considering the smallest class in each data set as the rare category, we adopt the following three commonly used metrics for the rare category analysis [243]: (1) accuracy, which measures the rate of the correctly classified majority and minority class examples; (2) recall, which measures the percentage of the discovered rare category examples; (3) recall@ K , which shows the ratio of true rare examples being retrieved in the returned top K examples, where K equals the number of rare category examples in the given network. In general, we observe that: (1) Our proposed SPARC algorithm outperforms the comparison methods across all the data sets and evaluation metrics in most cases. For example, on DBLP network with only one labeled minority class example, compared with the best competitor RACH, SPARC is 39% higher on Accuracy, 32% higher on Recall and 17% higher on Recall@K. (2) Our proposed SPARC algorithm is more robust (i.e., smaller error bar) than the comparison methods with different initial training examples. One intuitive explanation might be that the training ‘curriculum’ generated by SPARC guides the learning process towards a better local optimum in the parameter space.

8.5.4 Case Study: Impact of Self-Paced Learning

Dual Graph Context Selection: To illustrate the impact of SPL on RCE, we conduct a case study on Pubmed to show how the rare category oriented graph contexts are extracted

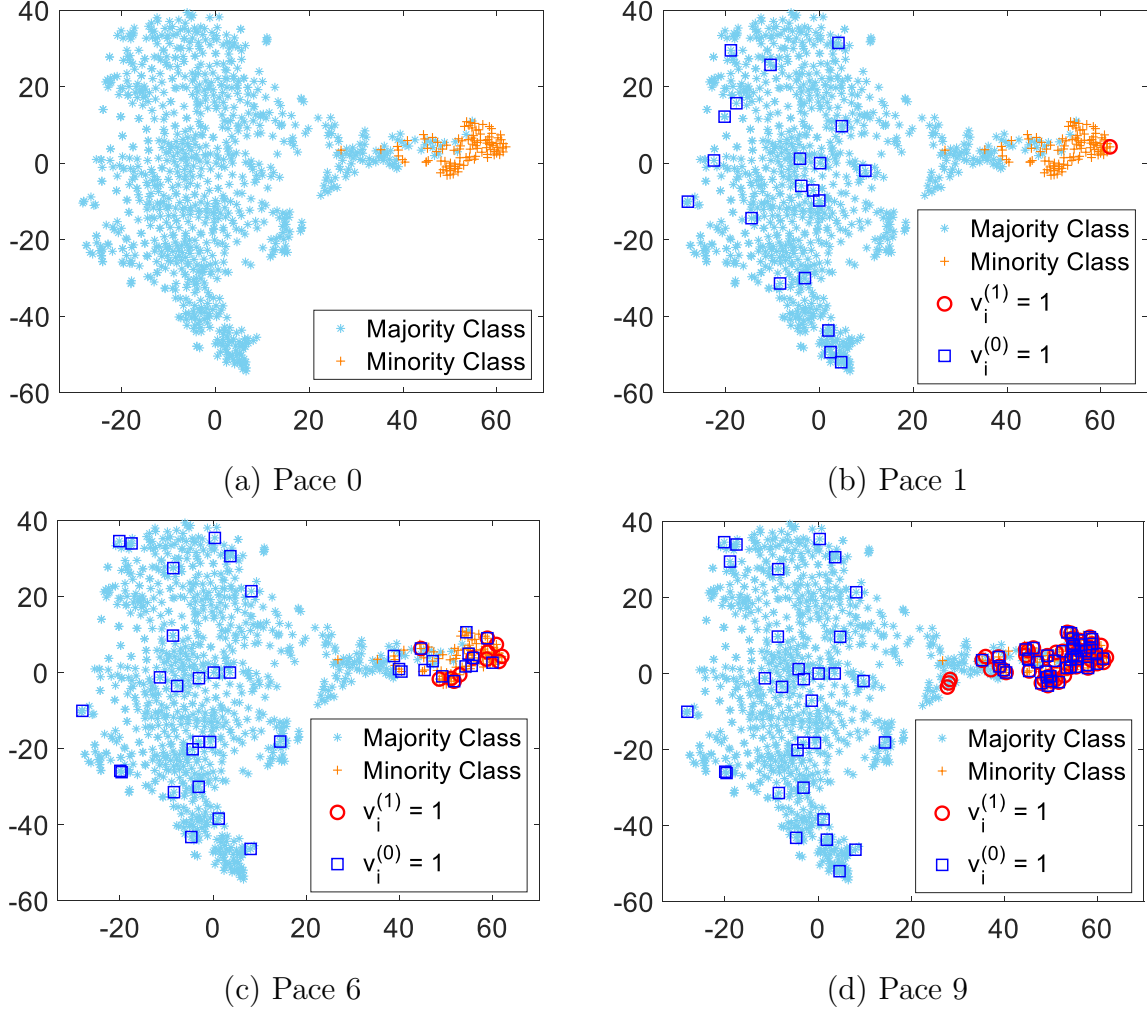


Figure 8.8: Self-paced dual graph context selection.

over paces. In particular, we show the vertices that were selected by the self-paced vectors $\mathbf{v}^{(0)}$ and $\mathbf{v}^{(1)}$ on the final embedding space of SPARC. Remember the self-paced vectors are updated over iterations (i.e., paces) by shifting from the ‘easy’ concept to the target ‘difficult’ one. In Figure 8.8, we observe that: (1) In the initial iteration (i.e., Pace 0), no vertices are selected by $\mathbf{v}^{(0)}$ and $\mathbf{v}^{(1)}$. (2) After that, $\mathbf{v}^{(1)}$ mainly selects the examples in the region of the minority class, while $\mathbf{v}^{(0)}$ selects examples across the whole network. (3) From Pace 1 to Pace 9, the overlap between the selected examples in $\mathbf{v}^{(0)}$ and $\mathbf{v}^{(1)}$ is increasing, which indicates that the RCE emphasizes on learning the context information of the minority class.

Parameter Sensitivity: We study the sensitivity of self-paced parameters $\lambda^{(0)}$ and $\lambda^{(1)}$ on Pubmed. Recall that $\lambda^{(0)}$ and $\lambda^{(1)}$ control the paces of learning from graph context and the underlying distribution of rare examples. In Figure 8.9, we report the recall rates of SPARC

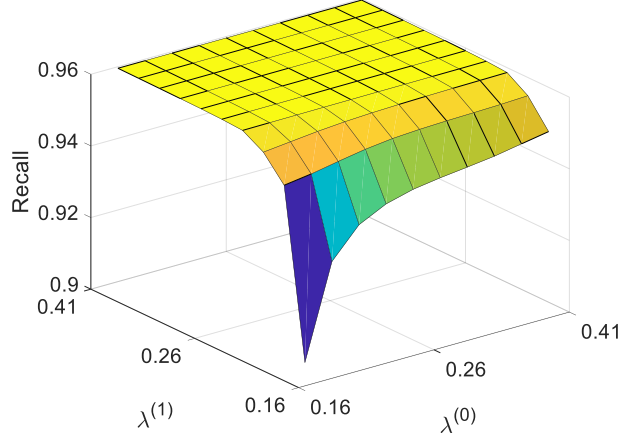


Figure 8.9: Parameter sensitivity analysis.

by iteratively augmenting the values of $\lambda^{(0)}$ and $\lambda^{(1)}$. We have the following observations: (1) Recall is generally increasing with the values of $\lambda^{(0)}$ and $\lambda^{(1)}$ over paces. An intuitive explanation is that when $\lambda^{(0)}$ and $\lambda^{(1)}$ are augmented, the richer context information of rare examples is extracted for training SPARC according to Eq. 8.8 and Eq. 8.9, which leads to a better prediction model. (2) In the early stage (i.e., $\lambda^{(0)} = \lambda^{(1)} = 0.16$), recall increases faster (slower) with respect to $\lambda^{(1)}$ ($\lambda^{(0)}$). In other words, learning from rare examples with propagated labels is more important than learning from the graph context for the RCC task in the initial iterations.

8.6 SUMMARY

In this chapter, we focus on analyzing the rare categories in class-imbalanced networks. We start by formally defining the RCE and RCC problems related to the rare categories, and then identify their unique challenges due to the nature of rare categories in the attributed networks, i.e., highly skewness, non-separability and label scarcity. To address these challenges, we propose a generic rare category analysis framework named SPARC, which jointly learns the network representation and rare category characterization model in a mutually beneficial way by shifting from the ‘easy’ concept to the target ‘difficult’ one, in order to facilitate more reliable label propagation to the large number of unlabeled examples. The empirical evaluations on real-world data sets demonstrate the effectiveness of our proposed framework SPARC from multiple perspectives.

CHAPTER 9: VISUAL ANALYTIC TOOL FOR RARE CATEGORY EXPLANATION

9.1 OVERVIEW AND MOTIVATION

In many cases, relations among objects can be modeled as time-evolving networks, such as the collaborations among researchers, transactions among traders, and communications in social networks. These relations reflect how individuals act in a network over time and reflect the goals of their activities [259]. Most individuals in a network behave normally, while a minority may act differently from the others, indicating anomalous situations. Anomalies could be positive, such as superstars in a collaboration network and recipients or benefactors in a financial network, or negative enough to damage the development of the entire graph, such as frauds in a trading network and criminals or spies in a communication network. In either case, finding these anomalous changing behaviors of network structures is valuable.

Most of the existing anomaly detection algorithms are automatic, and do not take human insights into account. In contrast, active learning is a special case of machine learning that improves automatic algorithms' performance with human knowledge. Following an active learning procedure, many rare category detection (RCD) methods are thus developed following [26, 31, 32, 260, 261], i.e., candidates that are most likely to represent rare categories are detected and shown to be labeled by users. Rare category detection methods are one set of anomaly detection algorithms which recognize abnormal individuals as rare categories because their number is usually very small. Once labeled, the algorithm will propagate the label to the nearby instances which are similar to the labeled one in a feature space. Those representative candidates are usually centers of rare categories. This procedure has one major limitation, i.e., it is still difficult for users to make a correct judgment (i.e., whether or not the candidate represents a rare category) by only showing one single data instance to them with the entire context information missing. This is particularly difficult for detecting rare categories from a dynamic graph as both the temporal and structural information need to be considered while labeling a candidate. Therefore, visualization could be helpful in terms of supporting the interactive data exploration and providing a rich context representation.

However, challenges exist in designing such a visualization system to support the process of rare category detection in a dynamic network. First, although capturing the temporal dynamics of a changing structure itself is a problem that has been extensively studied [262], none of the existing techniques is developed to support the visualization of rare categories. Second, capturing the changing structures of rare categories in the context of a big dynamic graph is challenging as the rare categories are usually very small and their evolutions could

be very likely to be ignored. Third, to better support the decision-making process, the visualization should be able to differentiate different structures in detail, and this is not easy to achieve.

To address the above challenges, in this chapter, we propose a novel visualization system called RCanalyzer. RCanalyzer represents a large dynamic network in the form of a series of connected triangular matrices with each matrix representing a snapshot (Figure 9.1). A hierarchical clustering algorithm and a tree cut algorithm are developed to produce an adaptive focus+context view that aggregates the graph structure into a hierarchy so that a large graph can be fully displayed while showing the detailed structures of potential rare categories. The proposed matrix based visualization facilitates an in-context visual comparison of substructures in a dynamic graph, which improves the efficiency of rare category detection. In particular, this chapter has the following contributions:

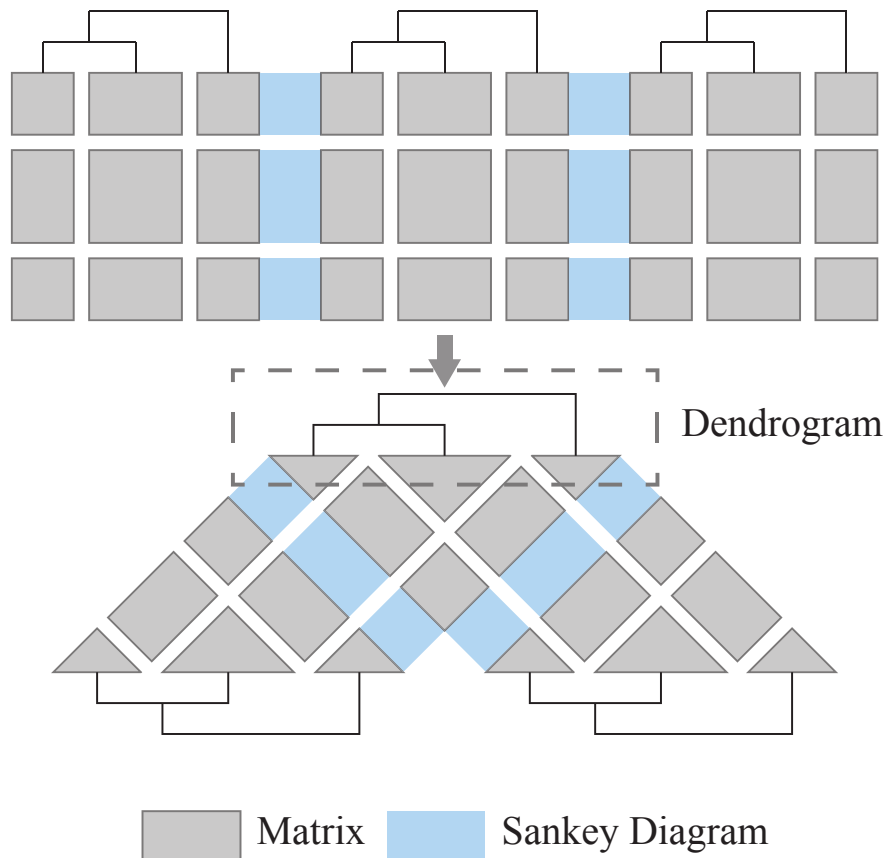


Figure 9.1: The basic design of the matrices view is a combination of matrix, Sankey diagram and dendrogram. Compared to a square matrix, triangles are more space efficient.

- A novel tree cut algorithm that produces a multi-focus view to illustrate the substructure details of multiple rare categories in the context of a big dynamic graph.

- A novel dynamic network visualization design in the form of a series of connected triangular matrices that highlights the detected rare categories in both the temporal and topological context, facilitating the substructure comparison.
- An integrated visual analysis system that supports the detection of rare categories and facilitates rare category labeling.

The chapter is organized as follows. Related work is discussed in section 9.2. The BIRD algorithm and analytical tasks are introduced in section 9.3. In section 9.4 we introduce the design of our system. System evaluations are introduced in section 9.5. We discuss our work in section 9.6 and conclude the chapter in section 9.7.

9.2 RELATED WORK

9.2.1 Dynamic network anomaly detection

Anomaly detection in dynamic networks refers to the detection of anomalous nodes, edges, subgraphs, and time-evolving changes. Several existing surveys have reviewed the most popular anomaly detection methods used in dynamic networks [56, 263]. Ranshous et al. categorized the existing methods into 5 types [56]: community-based, compression-based, decomposition-based, distance-based, and probabilistic-model-based. For example, based on compression based methods, a graph stream can be divided into multiple segmentations using the minimum description length (MDL) principle. Anomaly changes can be then detected at the time points when a new segment begins [67]. Probabilistic-model-based methods usually construct a "normal" model and use it to detect anomalies that deviate from the "normal" model. For example, when the number of communications deviates from the expected number generated by conjugate Bayesian models, the time point would be considered as an anomaly [264].

As we mentioned in Section 9.1, these anomaly detection works do not capture user's intention. In contrast, rare category detection refers to a series of active learning methods which incorporate human knowledge. Many RCD methods requires prior information to detect the minority classes [2, 26, 29, 31, 33, 243, 265]. However, many data sets don't have any prior information. To avoid this limitation, Huang et al. [260, 261], He et al. [32] presented a series of prior-free methods. Compactness-assumption-based methods [29, 31, 33, 265] assume that the distribution of the major categories is smooth and compact and compactness-isolation-assumption-based methods [260, 266] require the rare categories to be isolated from the major category. Lin et al. present RCLens [267], a visual analytics

system supporting user-guided rare category exploration and identification. RCLens is able to support users identify rare categories in a high dimensional dataset. However, it is not designed for rare category identification in dynamic networks.

9.2.2 Visualization of anomaly

Many visualization techniques have been developed to help the detection and analysis of anomalies [268, 269, 270, 271]. Dimension reduction methods, such as principal component analysis (PCA) [272], and multidimensional visualization techniques, such as parallel coordinate plots [273] and DICON [274], are commonly used to visualize the data distribution and show outliers with abnormal distribution. In ViDX [275], an extended Marey’s graph is used to show outliers in the manufacturing procedure. Anomalies in network traffic data [276, 277, 278] and social media data [279, 280, 281] have also drawn a lot of attention. Fluxflow [280] detects the diffusion of anomalous information in social media and TargetVue [281] uses glyph-based designs to show the anomalous behaviors in online communication systems based on an unsupervised learning model. Wang et al. [282] presented SentiView to visualize the sentiment in internet topics and enables analysts to monitor abnormal events on the internet. Fan et al. [283] presented an interactive visual analytics approach which combines active learning and visual interaction to detect anomalies.

Compared to the existing methods, our method focuses on detecting the rare categories in dynamic networks based on RCDs. To the best of our knowledge, there isn’t an existing visualization system that supports labeling users in analyzing and labelling anomalies based on RCDs. Moreover, we developed a series of interactions which enable users to compare rare categories within entire dynamic networks.

9.2.3 Visualization of dynamic networks

Visualization of dynamic networks has had a lot of study over the years. A fine survey by Beck et al. [262] has reported the state of art of dynamic network visualization. Beck et al. classify the visualization techniques of dynamic networks into animated diagrams [284, 285] and timelines of a series of static charts, such as node-link diagrams or adjacency matrices. Timelines with matrix-based and flow-based representation methods are most relevant to our work. Archambault et al. [286] found that small multiple-based techniques have better performance than animation-based techniques.

Matrix-based techniques can be classified into two categories. The first category embeds a timeline into each cell of the matrix. Gestaltlines [287], fingerprint glyphs [288], and the

horizon graph [289] are used to show the evolution of dyadic relations in a matrix. However, this category of methods often does not fit well with large data sets. The second category lays a sequence of adjacency matrices in a certain order [290, 291, 292]. Van den Elzen et al. [293] reduce the matrices into points and lay the points by production methods. NodeTrix [294] and Dendrogramix [295] both visualize a static graph by combining several visualization representation. However, they are not designed for visualizing dynamic networks and thus cannot show the change of networks properly.

Flow-based techniques use flow metaphors to represent the evolution of communities in networks [296, 297]. Sankey diagram [298] and ThemeRiver [299] are the most common methods used. For example, Vehlow et al. [296] use Sankey diagrams to show the changes of community structures. Flow-based techniques aggregate networks by group information, and thus often lack details of the local areas of the network.

In this chapter, we combine adjacency matrices, Sankey diagrams, and tree structures based on a multi-focus tree cut algorithm and visualize focused areas with fine-grained detail and unfocused areas with coarse-grained detail within a sequence of matrices.

9.3 PRELIMINARIES

Rare category detection (RCD) algorithms aim to find an initial example of rare classes in the data [26]. To best of our knowledge, Batch-update Incremental RCD (BIRD) [29] is the first (and the only) work designed for detecting rare categories in dynamic networks. It takes snapshots of dynamic network topology at two different time steps as input and iteratively detects *rare category candidates*, which potentially belong to a rare category. In this section, we first introduce related concepts of BIRD, and then introduce the analytical tasks users should complete based on RCAnalyzer to detect rare categories in dynamic networks.

9.3.1 Batch-update incremental RCD (BIRD)

Here, we review the key ideas of the incremental rare category detection algorithm - BIRD [29, 265], which pave the way for our forthcoming introduction of the rare category visual analytic system.

The Batch-update incremental Rare Category Detection (BIRD) algorithm aims to detect rare categories in dynamic networks. According to BIRD, a pair of nodes is closely connected if their transition probability is high. Therefore, the BIRD algorithm believes the transition probability of nodes in one rare category should have a lower bound and the transition probability of nodes in different rare categories should have an upper bound [33]. Therefore, a

rare category is a group of connected nodes that possess the following two features: (1) These nodes form a compact structure, which means they are closely connected. The transition probabilities among these nodes are relatively high and larger than the lower bound. (2) The compact structure should have a clear border. The transition probabilities among the nodes in this structure (rare category) and the other rare categories are relatively low and smaller than the upper bound. There are two visual examples showing these two features intuitively in Figure 9.2.

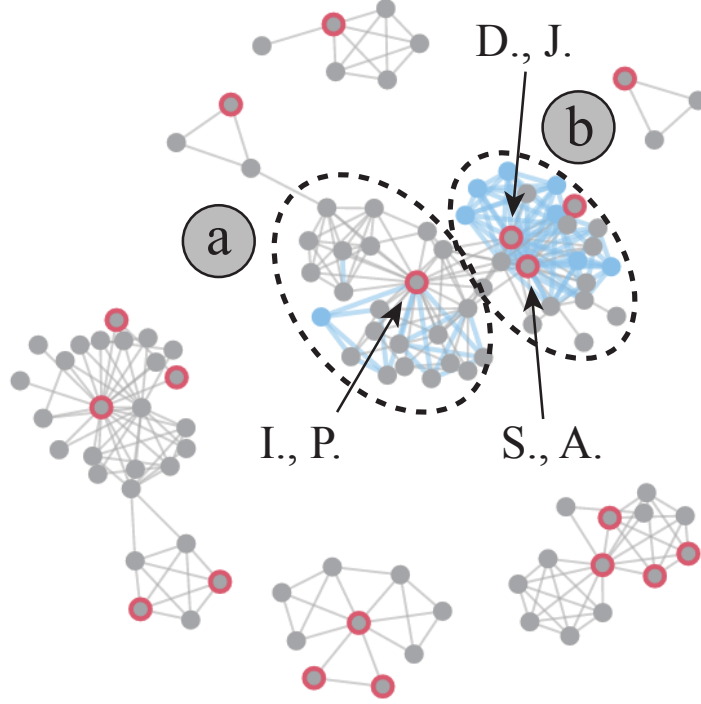


Figure 9.2: The compact neighborhood structures of D., J. and S., A. (A) and I., P. (B).

BIRD is an iterative algorithm. In each iteration, it detects a node whose neighborhood density changes significantly between two given adjacent time steps in a dynamic network. This node is potentially a representative node of a rare category.

Similar to the existing graph-based RCD algorithms [2, 31, 33], the BIRD algorithm can be mainly separated into the following two parts:

1. Compute the global similarity matrix A ,

$$A = (I - \alpha W)^{-1} \quad (9.1)$$

where I is an identity matrix, W denotes the transition probability matrix of the given graph G , and α is a positive discounting constant in the range of $(0, 1)$. Note that

the global similarity matrix A helps sharpen the changes of the local density near the boundaries of each class. This considerably reduces the workload of identifying rare categories in the query process.

2. Update the query score iteratively based on the labeling information from users and return the example with the largest query score to users for inspection. In general, the query process selects the examples from regions where local density changes the most, and thus the queried examples tend to have a high probability of hitting the regions of rare categories.

Before algorithm BIRD [29, 265], previous studies [26, 31, 33, 243] were all built for static graphs. For this reason, BIRD extends the problem to the dynamic setting and efficiently updates the RCD model by using the local changes to avoid reconstructing it from scratch. To be specific, the BIRD algorithm (1) efficiently updates the global similarity matrix $A^{(t)}$ at each time step t based on the global similarity matrix $A^{(t-1)}$ at previous time step $t - 1$ and the updated edges in current time step t ; (2) locally updates the query scores of the examples which may be infected by the changes in current time step t .

The original BIRD algorithm outputs the rare category candidate with the highest query score and waits for users to label the candidate. The query process might repeat many times. Thus, we slightly modify the BIRD algorithm by making the algorithm output candidates with top k query scores, where k is a manually set parameter.

The workflow of analyzing rare categories in dynamic networks with BIRD contains three stages. First, users set parameters and select two adjacent snapshots to initialize BIRD. Second, users analyze and identify rare categories based on the candidates detected by BIRD. Third, users label the candidates. The label result is returned to BIRD. When users think that all rare categories between the two snapshots are found, they can select other time steps and repeat the workflow to analyze other rare categories.

9.3.2 Analytical tasks

According to the analysis workflow, we summarize what analytical tasks should be completed by users based on these data as follows:

- T1. Set parameters to initialize BIRD. Users need to set a series of parameters before BIRD can detect rare category candidates.
- T2. Identify new rare categories from the examples detected by BIRD. After BIRD is initialized, it will iteratively output detected rare category candidates. Users first

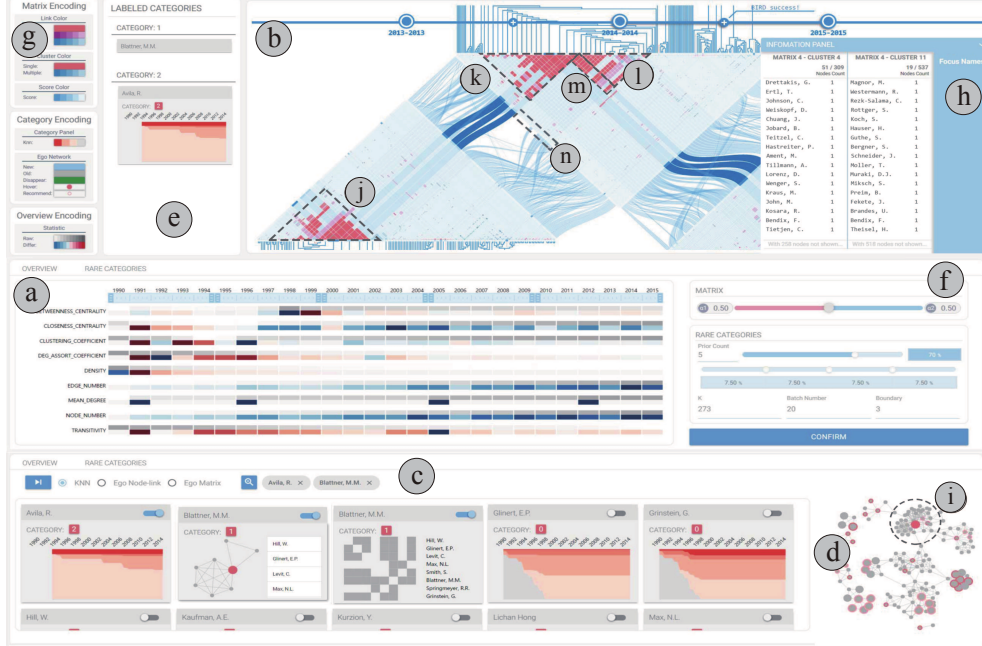


Figure 9.3: User interface of RCanalyzer. (a) the timeline view; (b) the matrices view; (c) the instance view; (d) the sub-network view; (e) the label result view; (f) the parameter panel; (g) the encoding panel; and (h) the information panel. BIRD detects W. D., X. W., and H. L. between 2014 and 2015. (i) the compact neighborhood structures formed by them and their surrounding area in the sub-network view; (j) the small community constituted by them and their surrounding area in 2013; (k) the same area as (j) in 2014; (l) a dense structure appeared beside (k); (m) two nodes in (k) have a lot of connections to nodes in (l); (n) the Sankey diagram shows 8 nodes in (l) are nodes in 2014. (l) indicates the existence of a paper with lots of coauthors, which might be a result of multilateral cooperation. The abnormal change of the surrounding areas of W. D., X. W., and H. L. make them a rare category.

identify candidates that truly belong to rare categories by analyzing their neighborhood structure. Then users compare the detected rare category with labeled rare categories to determine whether it is a new rare category.

- T3. Label the examples based on analysis results. After analyzing rare category candidates, users label each candidate by a specific number. Labels are then returned to BIRD.

9.4 SYSTEM DESIGN

In this section, we first introduce the design requirements of the RCanalyzer for completing the analytical tasks, and then we introduce the design of the RCanalyzer in detail.

9.4.1 Design requirements

We identify the following design requirements that the RCanalyzer should fulfill based on the analytical tasks.

For setting parameters to initialize BIRD (T9.3.2), we identify the following design requirements:

- **R1. Provide an overview of dynamic networks.** Users need to first explore the entire dynamic networks and understand the overall change of dynamic networks. With an overview, users can decide on which time periods they would focus on.

To identify examples belonging to rare categories among all detected examples (T9.3.2), we identify the following design requirements:

- **R2. Capture the changing structures of rare categories in the context of dynamic networks.** It is necessary to show the evolution of candidates in the background of the entire network. This helps users to identify the differences between the instance and the majority class.
- **R3. Reveal the features of detected examples.** It is essential to show the features of the surrounding area of candidates to identify rare categories. The features include the ego network of the instance and the similar nodes detected by BIRD.
- **R4. Reserve the context of labeled rare categories.** The system should remind users what kind of rare categories are detected and support the comparison between new candidates and labeled categories.

To label the examples based on analysis results (T9.3.2), we identify the following design requirements:

- **R5. Enable users to set and reset the labels of candidates.** The system should enable users to label rare categories and change labels of rare categories when they make mistakes.

9.4.2 System pipeline

From the design requirements, we designed the user interface of the RCanalyzer (see Figure 9.3). It consists of a) a timeline view, which shows a high-level overview of dynamic networks (R9.4.1); b) the matrices view, which shows the aggregated adjacency matrix of dynamic networks at each time segment initially (R9.4.1) and shows the details of the

neighborhood of multiple vertices after node selection (R9.4.1); c) the example view, which shows the feature of candidates (R9.4.1) and the query history of BIRD (R9.4.1); d) the label result view, which shows the historical label result and enables users to reset labels of labeled categories (R9.4.1 and R9.4.1).

Based on the analytical tasks, we design the architecture of the RCanalyzer, as shown in Figure 9.4.

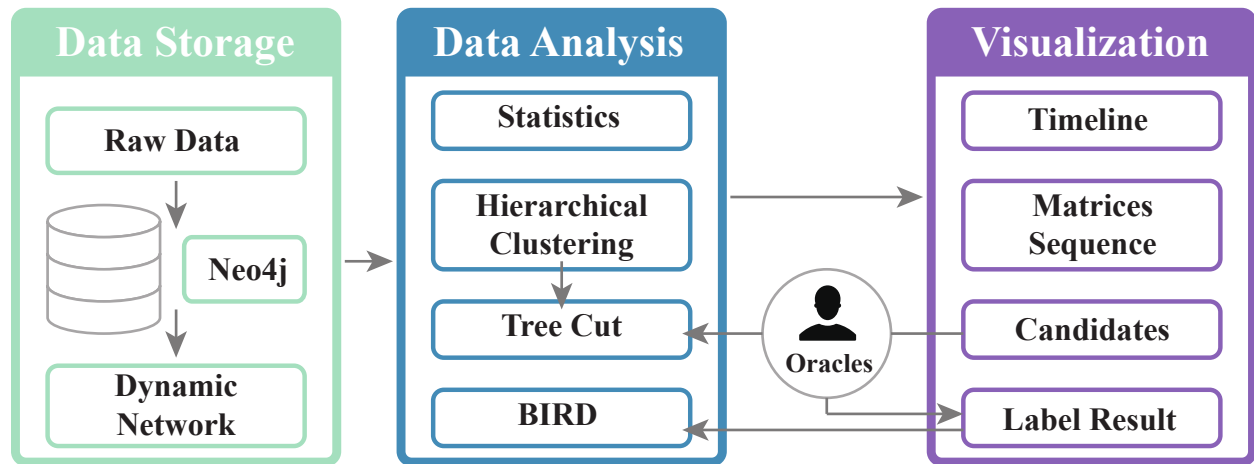


Figure 9.4: System pipeline.

The RCanalyzer consists of three major modules, a data storage module, a data analysis module, and a frontend visualization module. We use neo4j to store the dynamic networks in the data storage module. The data analysis module contains four components, consisting of BIRD, a hierarchical clustering algorithm, a tree cut algorithm and a bundle of statistics metrics. BIRD detects candidates of the rare categories iteratively. The hierarchical clustering algorithm extracts a tree structure from the network topology and the tree cut algorithm groups nodes to clusters based on the tree structure and the network topology. The statistics metrics measure the macro condition of the dynamic network.

The visualization module contains four major views: a timeline view, which shows the variation of network statistics and assist users select, merge, and filter time steps; a matrices view, which visualizes the network dynamics based on the tree cut result; an instance view, which displays the features of the rare category candidates detected by BIRD; a label result view, which reminds users what rare categories have been discovered.

9.4.3 The timeline view

The timeline view provides a highly abstracted overview of the dynamic network (R9.4.1). Metrics including betweenness centrality, closeness centrality, clustering coefficient, degree

assort coefficient, density, edge number, node number, average degree, and transitivity are calculated to show the state of dynamic networks at each time stamp. The timeline view contains two parts, an interactive time axis, and a pixel map. The pixel map visualizes metrics, which helps users to find interesting snapshots of dynamic networks. The interactive time axis (see Figure 9.3 (A)) enables users to select different snapshots (R9.4.1). After the time periods are submitted, the selected snapshots are extracted and merged accordingly. The data of merged snapshots are then visualized in the Matrices View to show the network data in detail.

Design Considerations We considered using three different visual designs in the timeline view to visualize the metrics: a line chart, a pixel map, and a glyph design. A line chart is intuitive to show time-varying data, while it lacks space efficiency. Using glyphs to show the metrics at each time stamp individually is space efficient while lacking intuitiveness. Thus, we choose to use a pixel map to show the metrics because a pixel map is more space efficient than line charts and more intuitive than a series of glyphs.

9.4.4 The matrices view

After time periods are selected in the timeline view, the data analysis module first aggregates snapshots of the dynamic network according to the selected time periods. The matrices view is designed for showing the dynamics of the network topology and the dynamics of selected rare category candidates. A hierarchical clustering algorithm [300], which builds a dendrogram based on network topology, is applied on each aggregated snapshot to reduce the number of entries in each matrix because a large matrix can hardly be visualized in a limited space with satisfactory detail. Same clusters at different time stamps are linked together to show the dynamics of the network. However, users cannot really explore and compare the neighborhood of rare category candidates in aggregated matrices because of the lack of detail. Therefore, a multi-focus tree cut algorithm is applied to each dendrogram to provide fine-grained detail of user-selected candidates and coarse-grained detail of other nodes. In this way, users are able to observe and compare the evolution pattern of rare category candidates (R9.4.1)

Multi-focus tree cutting. When users are interested in one or more rare category candidates, the dynamics of neighborhoods of these candidates are shown in the matrices view to support users to explore, compare, and identify rare categories among these candidates. We design a multi-focus tree cut algorithm to enable the matrices view to provide fine-grained details around selected nodes and coarse-grained details around unrelated nodes, which supports users in identifying rare categories among candidates (T9.3.2) by comparing

the features of candidates, labeled rare categories and non-rare categories. Different from existing multi-focus+context approaches [301, 302, 303], which work on the layout result of networks, our method directly works on the network topology and thus does not depend on the layout of networks.

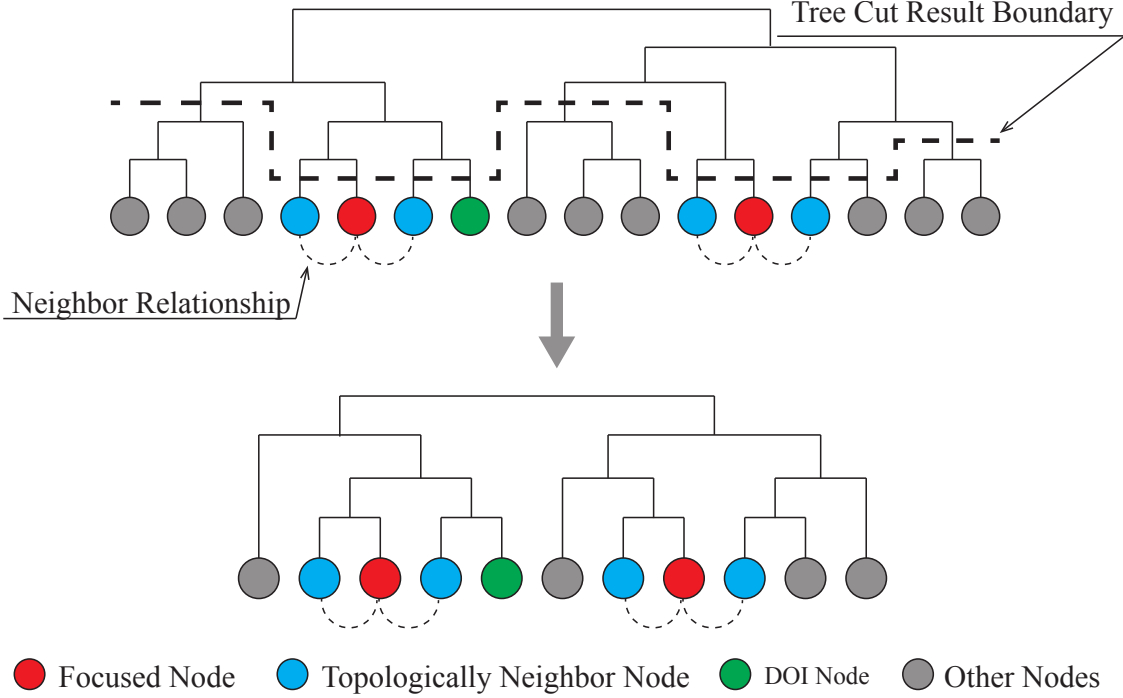


Figure 9.5: First stage of the tree cut algorithm: keep the details of all focused nodes.

Suppose we are given a dynamic network, which consists of a series of snapshots, $\tilde{\mathcal{G}} = \{S^{(1)}, S^{(2)}, \dots, S^{(t)}\}$. The multi-focus tree cutting algorithm works on each snapshot. The algorithm consists of two stages. In the first stage, details around all focused nodes are cut out from the tree; in the second stage, a merge operation is applied to prevent the result containing too many non-relevant single-node clusters.

First stage: multi-focus tree cutting. The procedure of the first stage is shown in Figure 9.5. For a specific snapshot $S^{(i)} = (V, E)$, hierarchical clustering is applied first to obtain a tree structure based on modularity [300]. In order to cut the tree with multiple focused nodes, we modified the original modularity. The set of focused nodes can be written as $F = \{n | \text{focused nodes}\}$. The cut of the tree structure is an optimization of an energy function based on the tree structure and the network topology. Suppose the cutting result is $C = \{N_1, N_2, \dots, N_m\}$, where N_i is a group of nodes in the tree. Then

$$C = \arg \min \sum_{i=1}^m (E(N_i)) \quad (9.2)$$

where

$$\begin{cases} E(N_i) &= \sum_{e \in N_i} \frac{D(e, N_i)}{\|N_i\|} - \sum_{e \in N_i} \left(\frac{S(e, N_i)}{\|N_i\|} \right)^2 \\ D(e, N) &= \begin{cases} Weight(e), & \text{if } \forall v \in e, v \in N_i \\ 0, & \text{else.} \end{cases} \\ S(e, N_i) &= \begin{cases} Weight(e), & \text{if } \exists v \in e, v \in N_i \\ 0, & \text{else.} \end{cases} \end{cases} \quad (9.3)$$

We defined the weight of an edge as the minimum of the weights of the node it links: suppose $e = (v1, v2)$, then $Weight(e) = \min(Weight(v1), Weight(v2))$. The weight of a node is defined based on the distance between the node and the focus nodes both in the tree structure and the network topology:

$$\begin{cases} Weight(v) &= \alpha_1 W_{DOI}(v) + \alpha_2 W_{Topology}(v) \\ W_{DOI}(v) &= \min_{n \in F} (D_{DOI}(n, v)) \\ W_{Topology}(v) &= \min_{n \in F} (D_{Topology}(n, v)) \end{cases} \quad (9.4)$$

, where $D_{DOI}(n, v)$ is the degree of interest distance between n and focused node v in the tree structure, $D_{Topology}(n, v)$ is the shortest distance between n and focused node v in the network topology, and α_1 and α_2 are weights of the two distances.

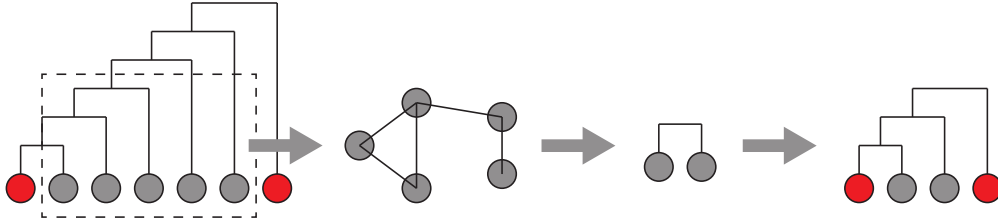


Figure 9.6: Second stage of tree cut algorithm: re-group the unrelated nodes according to the network structure.

Second stage: re-clustering of non-relevant nodes in the partial structure.

When the structure of a hierarchical clustering tree is partial and the focused nodes are deep in the tree, a large number of non-relevant nodes might be cut out from the tree, which increases the height of the cut result. To avoid this problem, we apply a re-cluster procedure to the non-relevant nodes. The continuous non-relevant single node sequences are first detected and cut out from the tree. Then the tree cut algorithm is applied again to the

sub-tree based on the network topology. Last, hierarchies are inserted back into the tree. The procedure of this stage is shown in Figure 9.6.

Visual designs in the matrices view. We use a combination of matrix, Sankey diagram and dendrogram as the basic representation of dynamic networks(see Figure 9.1). Sankey diagrams are added between each pair of adjacent matrices to show the evolution of these groups. The hierarchy of clusters represents the relationships among clusters and the structure of the network. In the RCAnalyzer, all networks are treated as undirected networks, and thus the adjacent matrices are symmetric. We use dendrograms to replace the upper (lower) triangular matrices and show the hierarchy of clustering result for space efficiency. The sequence of upper and lower triangular matrices are laid in a zigzag shape (see Figure 9.1).

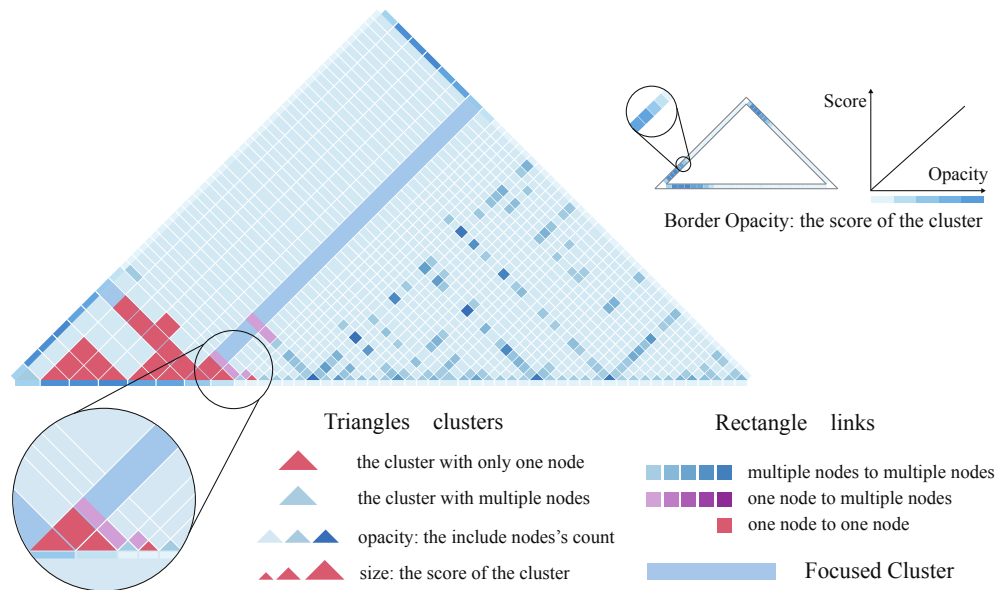


Figure 9.7: Visual encodings inside a matrix. Triangles represent a single node (red) or a group of nodes (gradient blue showing size). A red rectangle represents the connection between two single nodes; a purple rectangle represents the connections between a single node and a group of nodes; a blue rectangle represents the connections between two groups of nodes. Scores are encoded both by size of rectangles and triangles and the color on the matrix border.

Due to the tree cut algorithm, there are different granularity details. This leads to different numbers of nodes in different clusters. The opacity and color of triangles on the diagonal of

matrices encode the number of nodes, as shown in Figure 9.7. We use blue and red (shown in Figure 9.7) to distinguish a group of nodes and a single node. The gradient of blue in Figure 9.7 is used to encode the number of nodes in groups. Rectangles inside matrices represent three categories of connections: a single node to a single node, a single node to a group of nodes, and a group of nodes to a group of nodes. For consistency, we use blue to encode group-to-group relations, orange to encode one-to-one relations, and purple to encode one-to-group relations. The gradient of colors (Figure 9.7) represents the actual number of connections between the corresponding nodes.

Due to the importance of node anomalies in this work, we decide to use the size of triangles on the diagonal of matrices to encode the anomalous scores output by the BIRD algorithm (R9.4.1). If a large number of clusters is generated by the tree cut algorithm, sizes of single node clusters will be small under the limited size of matrices, which impedes the analysis of the nodes in which users are interested. We use three methods simultaneously to solve this problem. First, freely zooming and dragging are supported in this view. When the matrices are enlarged, the sequence of matrices cannot be fully displayed because of the limitation of space. Thus, we implement a special scale interaction with the scale functions shown in Figure 9.8 to enable local scaling without changing the size of matrices.

When the scale interaction is activated, the distortion of the size of the triangles and rectangles may mislead users, although we maintain the size ratio in the scaled local area. Thus, we encode the scores on the borders of the matrices by color, which brings two benefits: 1. users will clearly distinguish to which clusters the bands in Sankey diagrams belong when matrices are sparse; 2. users will observe the changes of scores over time stamps more easily.

Design Considerations Node-link Diagram and matrix representation are two common techniques to visualize networks. We choose the matrix as the basic representation of networks instead of the node-link diagram because the matrix representation can be better combined with a dendrogram. Although same clusters or nodes can be linked together in a series of node-link diagrams to visualize a dynamic network, overlap of lines in this solution will be severe and significantly reduce the readability of the visualization.

9.4.5 The rare category candidates view

The rare category candidates view is designed to reveal the features of candidates (R9.4.1). It contains two components: small multiples of candidate feature panels, which visualize the neighborhood information of candidates, and a sub-network view, which shows the sub-network formed by all detected candidates and their first-hop friends.

Representation of Ego Network of candidates consists of two visualization forms: a

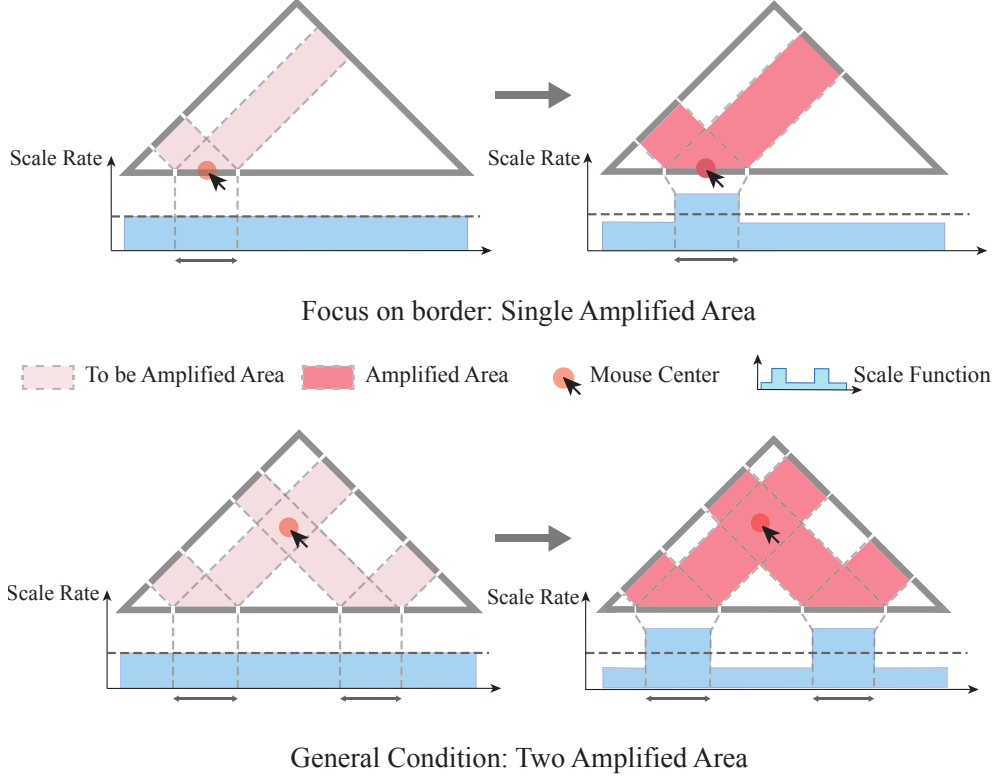


Figure 9.8: Scale functions when focus on the border of a matrix and focus inside a matrix.

node-link diagram and a matrix. The coexistence of node-link diagrams and matrices is not considered as redundant because we think the two visualization forms have different emphases: the former emphasizes vertices while the latter emphasizes links. Because BIRD detects rare categories between two time steps, changes of the candidates' ego networks at the two time steps are shown in Figure 9.3. The state of vertices and links are encoded by colors: blue indicates appearance, green indicates disappearance, and grey indicates fixedness.

Sub-network of Candidates shows the query process of BIRD by visualizing all the candidates together with their first-hop-neighbors (R9.4.1) and helps users to compare the candidates in the local area of the network. The color encoding is similar to the encoding in ego networks. Except for the color of links and nodes, we use red border of nodes to demonstrate the candidates detected in the current iteration and light red border of nodes to demonstrate the candidates detected in previous iterations. When an instance is hovered, both itself and its kNN will be enlarged, as shown in Figure 9.9.

9.4.6 Other panels

The Label Result View The label result view shows detected rare categories by recording label results of rare category candidates in a list of candidate feature panels (R9.4.1), as

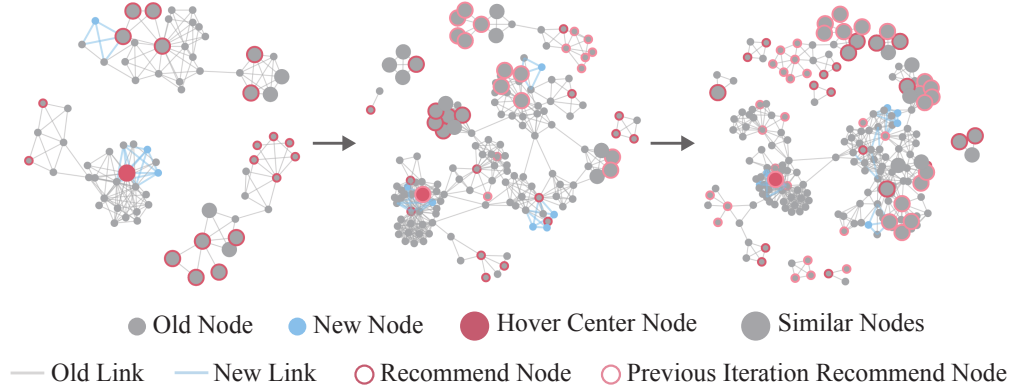


Figure 9.9: The sub-network view shows the query process of the BIRD algorithm by a node-link diagram formed by all the candidates ever queried by BIRD.

shown in Figure 9.3. Users can review the detected rare categories at any time during the analysis procedure.

The encoding panel shows the color encodings used in the system (see Figure 9.3 (G)). **The information panel** shows the detail information of selected blocks in the matrices view, as shown in Figure 9.3 (H). When hovering on triangles on the diagonal of a matrix, node count and node list are shown in the panel. When hovering on rectangles inside a matrix, the information panel is divided into two parts, each of which shows the node count and the nodes that have connections to the other cluster. The link count between two clusters is also shown (see Figure 9.3 (H)).

9.4.7 User interaction

The system implements a series of user interactions to support users to analyze the rare categories.

Detail on demand The instance view and the matrices view show the information of rare candidates at different levels of detail. Once nodes are selected in the instance view, the tree cut algorithm will be applied and the detail information of the selected candidates and their related nodes will be shown in the matrices sequence view with the context of the entire dynamic network.

Highlighting & Pinning All views in the RCanalyzer are linked. Whenever and wherever a node is hovered over by users, other views will highlight the node and its related nodes. Users can pin the block by clicking on it and then explore the details in the information panel.

Dragging & Zooming The matrices view supports users in freely dragging and zooming the matrices sequence.

Rare Category Labeling Users can label each candidate with a specific number, which helps BIRD distinguish different rare categories in the feature panel.

9.5 SYSTEM EVALUATION

In this section, we conducted one use scenario and a controlled user study to demonstrate the effectiveness of the RCanalyzer. The use scenario is based on a dynamic network extracted from the collaboration among authors of visualization publications [304].

We developed a prototype system to do all the experiments. The RCanalyzer is a web application which supports multiple users in analyzing the rare categories in dynamic networks. The front-end visualization is implemented by AngularJS, D3, and CSS. The back-end server is implemented by Python with Flask, Neo4J, numpy, igraph, and networkx. Use scenarios and the user study run on a PC with Intel(R) Core(TM) i7-4770 CPU, 20 GB RAM, and Windows10.

9.5.1 Use scenario: collaboration network in visualization publications

Dataset We extract all co-authorship in IEEE VIS dataset [304] from 1990 to 2015. An incremental collaboration network is constructed based on co-authorship, in which a link at timestamp t indicates two authors have coauthored at t or before t . We filtered the authors by taking the largest connected component in 2015 and there are 3640 authors left in the network. The number of links varies from 43(1990) to 11848(2015).

The timeline view and the matrices view show the basic information of the network (see Figure 9.3 (a) and (b)). Note that the time axis is initially divided into 5 segments to show the condition of the dynamic network in periods of time. The heatmap and the matrices show that before 2000, both the number and the increment of nodes and links are small; after 2000, the network grows faster, and after 2004, the network grows significantly.

After initializing the BIRD with the data in 2014 and 2015, W. D., X. W., and H., L. are selected to be the focused nodes in the instance view, as shown in Figure 9.3. They and their neighbors form a compact area in the sub-network view (Figure 9.3 (i)). Their surrounding areas from 2013 to 2015 are shown in the matrices view. Focused nodes are highlighted by the blue lines. Area (j) in Figure 9.3 is their surrounding area in 2013. The large link density in this area indicates that nodes in this area have close collaboration relationships. Thus, these nodes can be regarded as a small collaboration group. The Sankey diagram between 2013 and 2014 shows area (k) is almost the same as area (j). A dense structure in area (l) appeared beside area (k). Meanwhile, area (m) shows that two nodes, including X. W.,

in area (k) connect to most nodes in area (l). The blank of the Sankey diagram (labeled by (n)) on the left of the matrix in 2014 indicates that 8 nodes in area (l) are new nodes. The clique structure in area (l) indicates these nodes collaborated in the same paper. Large numbers of authors of the paper indicates that the paper might be the result of multilateral cooperation. The appearance of this uncommon cooperation causes W. D., X. W., H. L. to be identified as a rare category.

Between 2012 and 2013, D. J., S. A., and I. P. constitute a large and dense sub-network (Figure 9.2). However, there is a small gap between the first three authors (Figure 9.2 (A)) and the last author (Figure 9.2 (B)). Thus, whether they belong to the same category cannot be decided. The matrices view shows the dynamic changes in surrounding areas around them. In 2011, I. P. is in the area (A), and D. J. and S. A. are in area (B). It is clear that these two areas have no connections. In 2012, area (C) shows that the two areas in 2011 merged into one because of the new connections in area (D). However, a large number of new connections appeared in area E in 2013. From the Sankey diagram between 2013 and 2014, we know that authors newly connected to D., J. and S., A. in 2013 also appeared in the area G in 2014. From the matrix of 2014, we can see that area G and area H are separated from each other. Thus, the merging and splitting behaviors of the surrounding areas of D. J., S. A., and I. P. along time are the reasons why D. J., S. A., and I. P. are identified as a rare category.

9.5.2 User Study

We conducted a user study to verify the usability of the RCAnalyzer. We introduce the user study following the order of assumptions, datasets, participants, procedure, and result.

Assumptions. As there is no existing work supporting similar tasks to the RCAnalyzer, we do not use a baseline system in this user study and only test if the RCAnalyzer could help users to explore, analyze, and identify rare categories in dynamic networks and collect users' qualitative feedback. We first make three assumptions about the usability of the RCAnalyzer.

1. RCAnalyzer helps users identify examples of rare categories among the query result of the BIRD algorithm in each iteration.
2. RCAnalyzer helps users distinguish examples of rare categories and examples of major categories.
3. RCAnalyzer helps users distinguish examples of different rare categories.

For a dataset with ground truth, we can count the minimal number of iterations within which the BIRD algorithm can detect at least one example in each of the rare categories in

the dataset. By comparing this minimal number and the actual number of iterations users use in the study, we can validate the assumption 1. If the number of iterations used by users is close to the minimal number, the RCAnalyzer efficiently supports users to identify rare categories. We validate assumption 2 and 3 by calculating the accuracy of the rare categories labels labeled by users in the user study.

Synthetic Data. Because of the high complexity of the real datasets used in the case studies, it is hard to control the test and quantify the actual efficiency of rare category detection with the RCAnalyzer. Thus, we use synthetic datasets in the user study. All the synthetic datasets have two time stamps. Each synthetic dataset is constructed by the following procedure: 1) generating a grid network with N nodes at each time stamp; 2) adding edges among nodes in the network to form four different special structures: a clique, a bipartite graph, a star structure, and a circle, at the second time stamp. Special structures are treated as rare categories and other nodes are treated as the major category. We constructed four synthetic datasets with $N = 100, 200, 500$, and 1000 . The dataset with $N = 100$ is used in the tutorial of the user study. The minimal numbers of iterations on datasets with $N = 200, 500$, and 1000 are 5, 5, and 11 respectively.

Participants. We recruited 12 participants for the evaluation, including 9 males and 3 females. All of them have background in visualization, and one of them has a background in anomaly detection.

Tasks. The participants are asked to complete the following tasks in the user study:

- T1. Identify rare categories in the examples detected by BIRD in each iteration.
- T2. Label examples identified as rare categories.

Procedure. The user study has three stages. In the first stage, we introduce the basic concept of this work and the tasks of the user study to participants with a 10-minute tutorial. In the second stage, we introduce the RCAnalyzer to participants and let them explore the system with the synthetic dataset with $N = 100$ for 15 minutes. Participants are allowed to ask any questions about the system and the tasks in the first and the second stages. In the third stage, participants are asked to analyze the synthetic datasets with $N = 200, 500$, and 1000 , label rare categories they identified in the RCAnalyzer, and write down their labeling results on an answer sheet. In order to ensure that participants will not give answers arbitrarily, they are asked to describe the reason why a detected example is identified as a rare category. **Result** The accuracy of labeling rare categories is shown in Table 9.1. The results show that the detection of the clique, bipartite graph, and star graph is accurate (86.11%, 86.11%, and 91.67%) while the accuracy of detection of the circle is not very good

Size	Clique	Bipartite	Star	Circle
200	91.67%	83.33%	83.33%	58.33%
500	83.33%	83.33%	100.00%	91.67%
1000	83.33%	91.67%	91.67%	83.33%
Avg.	86.11%	86.11%	91.67%	77.78%

Table 9.1: Accuracy of labeling.

(77.87%). Detection of a circle structure is really hard because the surrounding area of a node on the circle is unobtrusive in the matrices view and nodes on the circle are queried by BIRD discontinuously, forming several segments of line instead of a circle in the sub-network view. To identify the circle structure, participants need to select a series of instances on the circle, but some of the participants missed too many instances on the circle, and thus were not able to label the circle structure correctly. The distribution of participants' query number is shown in Figure 9.10. The result shows that participants can finish the labeling process in 4-5 iterations in datasets with 200 and 500 nodes. For the dataset with 1000 nodes, many of the participants can finish the labeling processing in 11 to 15 iterations, while two outliers finished the labeling process in 2 and 6 iterations. This was because they labelled normal nodes as rare categories. Some of the participants finished the labeling result after over 20 iterations. This is because they did not label the rare categories promptly. Overall, the accuracy and the average query numbers show that most of the participants are able to identify rare categories promptly and correctly.

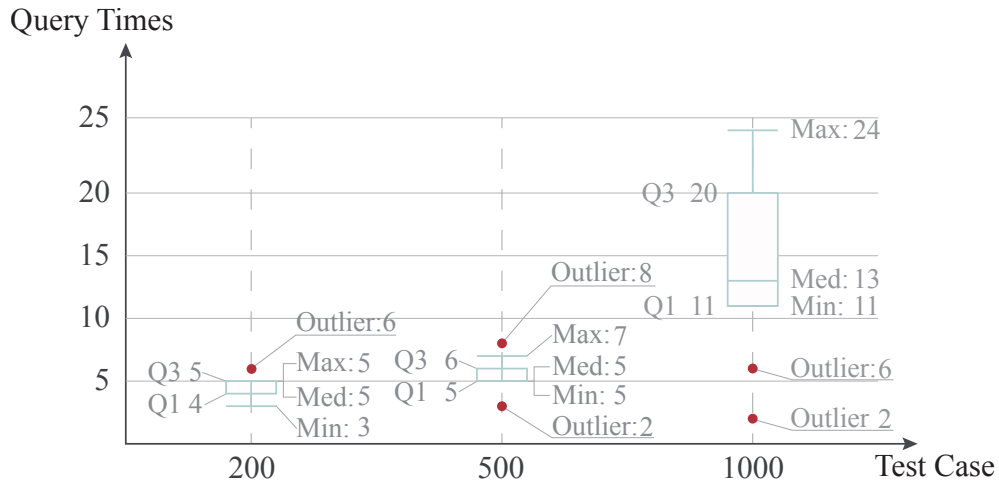


Figure 9.10: The query numbers of participants when they labeled all rare categories in the data.

Qualitative feedback In order to assess the learnability, usability and other perception aspects of the RCanalyzer, users were asked to give some qualitative feedback after the formal user study. The most frequent complaint was that the encodings in our matrices view were too complex. We used both the size and the color of each cell to encode different information. Users had to recognize all the encodings at the beginning of the user study. It would lead to confusion because they would forget the encodings. Some users said that the parameters were hard to comprehend. They said that it was hard to learn what will happen if the parameters were adjusted. It took a long time for them to learn how the system worked. Learnability and usability were both important problems which were hard to cover. One of the solutions for improvement is to reduce the complexity of our visual design. However, it takes much more time to know which visual design is less efficient and can be abandoned. In the future, we will redesign our visual design based on more user behaviors. For example, the color encoding on the border can be removed if users do not care about the border color encoding.

9.6 DISCUSSION

Generalizability In this chapter, we used a collaboration network to evaluate our system. However, the RCanalyzer supports rare category analysis in other networks. Although we only support the BIRD algorithm in our system, the RCanalyzer can work based on other RCD algorithms as long as they are based on the topology of dynamic networks. The matrices view with the tree cut algorithm can be applied in other applications for analyzing dynamic networks. For example, tracking the time-varying pattern of multiple nodes and comparing the change of ego-networks of multiple nodes. We believe that the combination of matrix sequence and multi-focus tree cut algorithm is a useful method as it enables simultaneous comparison of multiple nodes.

Scalability In use scenarios, we tested the effectiveness of the RCanalyzer on a network with 8319 nodes, 210625 edges, and 6 time steps, which indicates that the RCanalyzer has good scalability on large datasets. As for larger datasets, the major bottleneck would be the running time of initialization of the BIRD algorithm and tree cut algorithm due to the limitation of execution efficiency of Python. In the future, we plan to use pre-computation and server-side cache to support the analysis of larger datasets. As for the scalability of our visual design, it is related to the granularity of our tree cut algorithm, and the scale of the input dynamic network. From our experience, it is hard to show more than 6 time-steps with around 50 rows in each matrix at the same time in the matrices view (with $1,360 \times 635$ pixels). Interactions such as dragging and zooming to improve the readability of matrices

have been discussed in 9.4.4. For dynamic graphs with more time-steps, the tree cut algorithm should be more coarse-grained to show all time-steps in the meantime. However, the coarse-grained tree cut algorithm reduces the information of the dynamic networks.

Limitations Although the RCanalyzer is able to help users to analyze and label rare categories in dynamic networks, it still has several limitations. First, more interactions should be supported, such as querying and filtering. Interactions in the the RCanalyzer are enough to support the detection of rare categories, but more complete interactions can significantly improve user experience. Second, the process of interactions and visual encodings in the RCanalyzer are a little complicated. During the user study, it takes 15-25 minutes to train subjects to let them fully understand how to use the system. Third, the RCanalyzer only supports screens with 1920×1080 resolution. More adaptive layout should be supported to enable users to label rare categories at different resolutions.

Future Work First, we plan to add context information of nodes in the RCanalyzer. The RCanalyzer is based on the topology of dynamic networks currently because the BIRD algorithm detects rare categories by checking the changes of topological structure around nodes. However, nodes with the same topology may have completely different context information. We believe context information will help users distinguish different rare categories. Second, we plan to add data filtering to the RCanalyzer. Sometimes, users might be interested in only a special area in the network. A data filtering module can help them analyze the desired areas of data.

9.7 SUMMARY

In this chapter, we present the RCanalyzer, a novel visual analytics system which helps oracles to analyze the result of RCD methods and label the rare categories in dynamic networks. It consists of five linked views: a timeline view, a matrices view, an instance view, a sub-network view, and a label result view, and it shows the information of rare categories in different levels of detail. In addition, we present a multi-focus tree cut algorithm and a tree-structure constrained layout optimization algorithm to support the comparison of instances in the context of their surrounding structures. We use one use scenario, and one user study to demonstrate the usability and effectiveness in analyzing rare categories in dynamic networks.

CHAPTER 10: TEMPORAL INTERACTION NETWORK GENERATION

10.1 OVERVIEW AND MOTIVATION

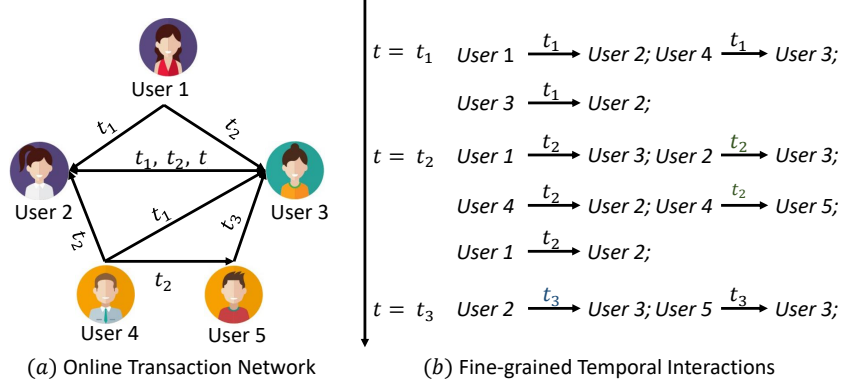


Figure 10.1: An example of *temporal interaction networks*. (a) An online transaction network with five users. (b) The corresponding system logs presented in the form of time-stamped edges between users.

Graph presents a fundamental abstraction for modeling complex systems in a variety of domains, ranging from chemistry [305], security [3, 306, 307], recommendation [308, 309], and social science [310]. Therefore, mimicking and generating realistic graphs have been extensively studied in the past. The traditional graph generative models are mostly designed to model a particular family of graphs based on some specific structural assumptions, such as heavy-tailed degree distribution [311], small diameter [312], local clustering [313], etc. In addition to the traditional graph generative models, a surge of research efforts on deep generative models [314, 315] have been recently observed in the task of graph generation. These approaches [154, 316] are trained directly from the input graphs without incorporating prior structural assumptions and often achieve promising performance in preserving diverse network properties of real networks.

Despite the initial success of deep generative models on graphs, most of the existing techniques are designed for static networks. Nonetheless, many real networks are intrinsically dynamic and stored as a collection of system logs (i.e., timestamped edges between entities). For example, in Figure 10.1, an online transaction network can be intrinsically presented as a sequence of timestamped edges (i.e., financial transactions) between users. When an online transaction is completed, a system log file (i.e., a timestamped edge from one account to another) will be automatically generated and stored in the system. A conventional way of modeling such dynamic systems is to construct time-evolving graphs [29, 98] by aggregating

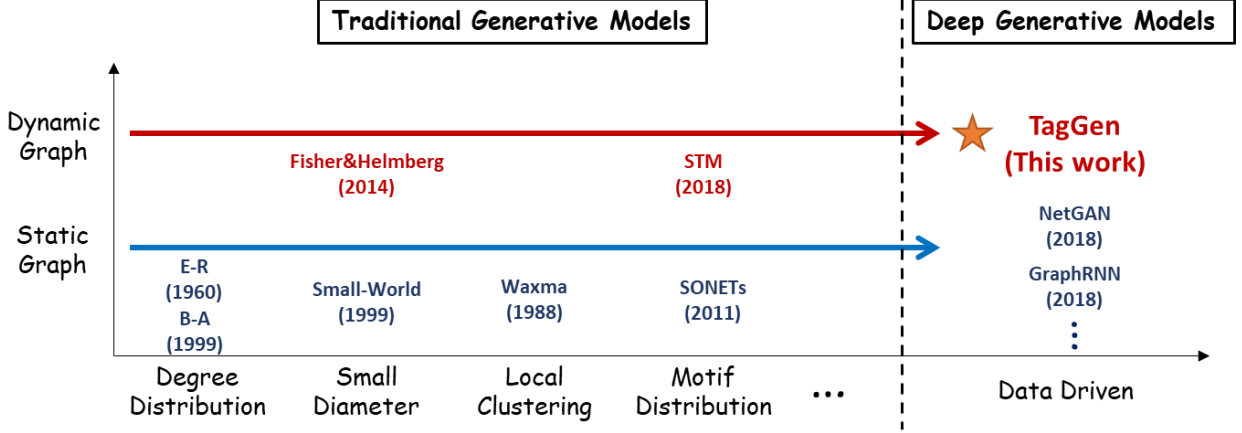


Figure 10.2: A two-dimensional conceptual space of graph generative models.

timestamps into a sequence of snapshots. One drawback comes from the uncertainty of defining the proper resolution of the time-evolving graphs. If the resolution is too fine, the massive number of snapshots will bring intractable computational cost when training deep generative models; if the resolution is too coarse, fine-grained temporal context information (e.g., the addition/deletion of nodes and edges) might be lost during the time aggregation.

Figure 10.2 compares various graph generators in a two-dimensional conceptual space in order to demonstrate the limitation of existing techniques as compared to ours. In this chapter, for the first time, we aim to address the following three open challenges: (*Q.1*) Can we directly learn from the raw temporal networks (i.e., temporal interaction network) represented as a collection of timestamped edges (see Figure 10.1 (b)) instead of constructing the time-evolving graphs? (*Q.2*) Can we develop an end-to-end deep generative model that can ensure the generated graphs preserve the structural and temporal characteristics (e.g., the heavy tail of degree distribution, and shrinking network diameter over time) of the original data?

To this end, we propose TAGGEN, a deep graph generative model for *temporal interaction networks* to tackle all of the aforementioned challenges. We first propose a random walk sampling strategy to jointly extract the key structural and temporal context information from the input graphs. On top of that, we develop a bi-level self-attention mechanism which can be directly trained from the extracted temporal random walks while preserving temporal interaction network properties. Moreover, we designed a novel network context generation scheme, which defines a family of local operations to perform *addition* and *deletion* of nodes and edges, thus mimicking the evolution of real dynamic systems. In particular, TAGGEN maintains the state of the graph and generates new temporal edges by training from the extracted temporal random walks [66]; the *addition* operation randomly chooses a node to

be connected with another one at a timestamp t ; the *deletion* operation randomly terminates the interaction between two nodes at timestamp t ; all the proposed operations are either accepted or rejected by a discriminator module in TAGGEN based on the current states of the constructed graph. At last, the selected plausible temporal random walks will be fed into an assembling module to generate temporal networks.

The main contributions of this chapter are summarized below.

- **Problem.** We formally define the problem of *temporal interaction network* generation and identify its unique challenges arising from real applications.
- **Algorithm.** We propose an end-to-end learning framework for *temporal interaction network* generation, which can (1) directly learn from a series of timestamped nodes and edges and (2) preserve the structural and temporal characteristics of the input data.
- **Evaluations.** We perform extensive experiments and case studies on seven real data sets, showing that TAGGEN achieves superior performances compared with the previous methods in the tasks of temporal graph generation and data augmentation.

The rest of the chapter is organized as follows. In Section 10.2, we review the existing literature. Problem definition is introduced in Section 10.3, followed by the details of our proposed framework TAGGEN in Section 10.4. Experimental results are reported in Section 10.5, before we conclude this chapter in Section 10.6.

10.2 RELATED WORK

In this section, we briefly review the related works regarding dynamic network mining and graph generative model.

Dynamic Network Mining. Recently, significant research interests have been observed in developing deep models for dynamic networks. Most existing work models the dynamic networks as time-evolving graphs, which aggregate temporal information into a sequence of snapshots. For instance, [317] proposes a network embedding approach for modeling the linkage evolution in the dynamic network; [13] proposes a graph attention neural mechanism to learn from the spatial-temporal context information of the time-evolving graphs; [318] proposes Spatio-Temporal Graph Convolutional Networks with complete convolutional structures, enabling faster training speed while tackling the issue of the high non-linearity and complexity of traffic flow. However, these approaches may not be able to fully capture the rich temporal context information in the data due to the aggregation over time. For

this reason, the authors of [66] proposed to learn network embedding for temporal interaction networks by developing a family of temporally increasing random walks to extract network context information. In this chapter, we propose a generic framework to further model and generate the temporal interaction networks by mimicking the network evolution process in real dynamic systems. To the best of our knowledge, TAGGEN is the first deep graph generative model designed for temporal networks.

Graph Generative Model. Early studies of graph generative models include the explicit probabilistic models [224, 319], stochastic block models [320], preferential attachment models [311, 321, 322], exponential random graph models [323], the small-world model [324], and Kronecker graphs [325]. In addition to the static models, some attempts have also been made for generating dynamic graphs. For instances, [312] proposes a dynamic graph generation framework that is able to control the network diameter for a long-time horizon; [326] develops a graph generator that models the temporal motif distribution. However, all of the aforementioned approaches basically generate graphs relying on some prior structural assumptions. Hence, such methods are often hand-engineered and cannot directly learn from the data without prior knowledge or assumptions. The recent progress in deep generative models (e.g., [314, 315]) has attracted a surge of attention to model the graph-structured data. For example, in [154], the authors aim to capture the topology of a graph by learning a distribution over the random walks in an adversarial setting; in [316], the authors propose a framework named Graph-RNN to decompose graph generation into two processes: one is to generate a sequence of nodes, and the other is to generate a sequence of edges for each newly added node. This chapter proposes a deep generative framework to model dynamic systems and generate the temporal interaction networks via a family of local operations to perform the addition and deletion of nodes and edges.

10.3 PRELIMINARIES

We formalize the graph generation problem for temporal interaction networks [66, 327, 328], and present our learning problem with inputs and outputs. Different from conventional dynamic graphs that are defined as a sequence of discrete snapshots, the temporal interaction network is represented as a collection of temporal edges. Each node is associated with multiple timestamped edges at different timestamps, which results in the different occurrences of node $v = \{v^{t_1}, \dots, v^T\}$. For example, in Figure 10.3, the node v_a is associated with three occurrences $\{v_a^{t_1}, v_a^{t_2}, v_a^{t_3}\}$ that appear at timestamps t_1 , t_2 and t_3 . The formal definitions of temporal occurrence and temporal interaction network are given as follows.

Definition 10.1 (Temporal Occurrence). In a temporal interaction network, a node v is associated with a bag of temporal occurrences $v = \{v^{t_1}, v^{t_2}, \dots\}$, which instance the occurrences of node v at timestamps $\{t_1, t_2, \dots\}$ in the network.

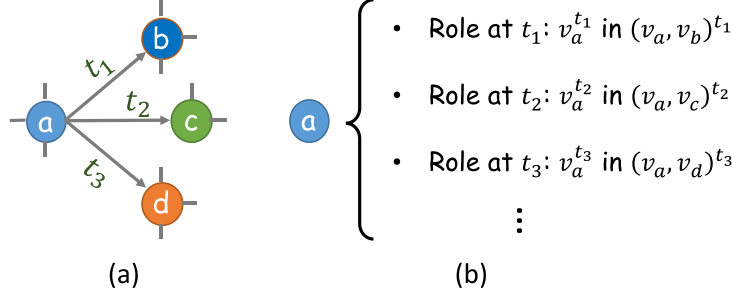


Figure 10.3: An example of node v_a and its temporal occurrences. (a) A miniature of a temporal interaction network. (b) The occurrences of node v_a that appear at t_1 , t_2 and t_3 .

Definition 10.2 (Temporal Interaction Network). A temporal interaction network $\tilde{\mathcal{G}} = (\tilde{\mathcal{V}}, \tilde{\mathcal{E}})$ is formed by a collection of nodes $\tilde{\mathcal{V}} = \{v_1, v_2, \dots, v_n\}$ and a series of timestamped edges $\tilde{\mathcal{E}} = \{e_1^{t_{e_1}}, e_2^{t_{e_2}}, \dots, e_m^{t_{e_m}}\}$, where $e_i^{t_{e_i}} = (u_{e_i}, v_{e_i})^{t_{e_i}}$.

In the static setting, existing works [227] define the network neighborhood $\mathcal{N}(v)$ of node v as a set of nodes that are generated through some neighborhood sampling strategies. Here, we generalize the notion of network neighborhood to the temporal interaction network setting as follows.

Definition 10.3 (Temporal Network Neighborhood). Given a temporal occurrence v^{t_v} at timestamp t_v , the neighborhood of v^{t_v} is defined as $\mathcal{N}_{FT}(v^{t_v}) = \{v_i^{t_{v_i}} | f_{sp}(v_i^{t_{v_i}}, v^{t_v}) \leq d_{\mathcal{N}_{FT}}, |t_v - t_{v_i}| \leq t_{\mathcal{N}_{FT}}\}$, where $f_{sp}(\cdot | \cdot)$ denotes the shortest path between two nodes, $d_{\mathcal{N}_{FT}}$ is the user-defined neighborhood range, and $t_{\mathcal{N}_{FT}}$ refers to the user-defined neighborhood time window.

In [66], the authors define the notion of *Temporal Walk*, which is presented as a sequence of vertices following a time-order constraint. In this chapter, we relax such a constraint by considering that all the nodes within a neighborhood time window $[t_v - t_{\mathcal{N}_{FT}} + 1, t_v + t_{\mathcal{N}_{FT}}]$ are the temporal neighbors of v^{t_v} and can be accessed from v via a random walk. Here, we formally define the k -Length Temporal Walk as follows.

Definition 10.4 (k -Length Temporal Walk). Given a temporal interaction network $\tilde{\mathcal{G}}$, a k -length temporal walk $W = \{w_1, \dots, w_k\}$ is defined as a sequence of incident temporal walks traversed one after another, i.e., $w_i = (u_{w_i}, v_{w_i})^{t_{w_i}}$, $i = 1, \dots, k$, where u_{w_i} and v_{w_i} are the source node and destination node of the i^{th} temporal walk w_i in W respectively.

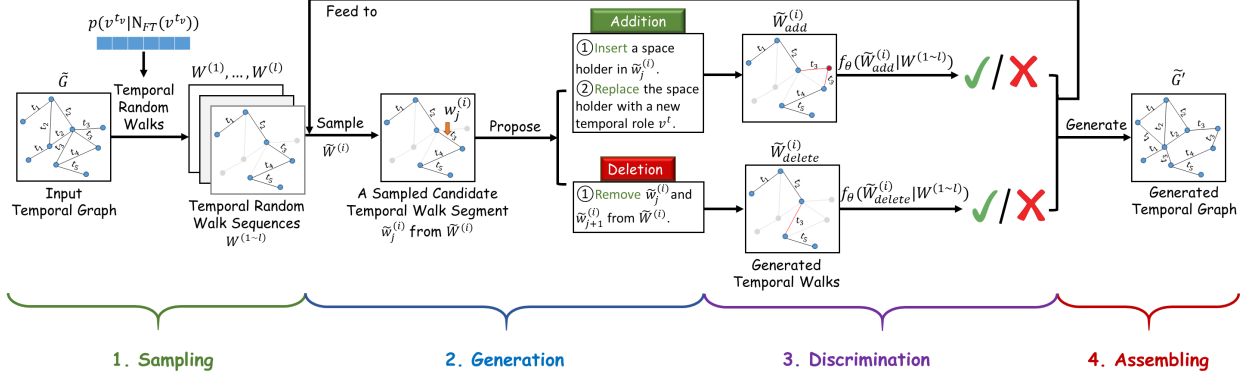


Figure 10.4: The proposed TAGGEN framework.

With all the aforementioned notions, we are ready to formalize the temporal interaction network generation problem as follows.

Problem 10.1. Temporal Interaction Network Generation

Input: a temporal interaction network $\tilde{\mathcal{G}}$, which is presented as a collection of timestamped edges $\{(u_{e_1}, v_{e_1})^{t_{e_1}}, \dots, (u_{e_m}, v_{e_m})^{t_{e_m}}\}$.

Output: a synthetic temporal interaction network $\tilde{\mathcal{G}}' = (\tilde{\mathcal{V}}', \tilde{\mathcal{E}}')$ that accurately captures the structural and temporal properties of the observed temporal network $\tilde{\mathcal{G}}$.

10.4 ALGORITHM

In this section, we introduce TAGGEN, a graph generative model for temporal interaction networks. The core idea of TAGGEN is to train a bi-level self-attention mechanism together with a family of local operations to model and generate temporal random walks for assembling temporal interaction networks. In particular, we first introduce the overall learning framework of TAGGEN. Then, we discuss the technical details of TAGGEN regarding context sampling, sequence generation, sample discrimination, and graph assembling in temporal interaction networks. At last, we present an end-to-end optimization algorithm for training TAGGEN.

10.4.1 A Generic Learning Framework

An overview of our proposed framework is presented in Figure 10.4, which consists of four major stages. Given a temporal interaction network defined by a collection of temporal edges (i.e., time-stamped interactions), we first extract network context information of temporal

interaction networks by sampling a set of temporal random walks [66] via a novel sampling strategy. Second, we develop a deep generative mechanism, which defines a set of simple yet effective operations (i.e., addition and deletion over temporal edges) to generate synthetic random walks. Third, a discriminator is trained over the sampled temporal random walks to determine whether the generated temporal walks follow the same distributions as the real ones. At last, we generate temporal interaction network, by collecting the qualified synthetic temporal walks via the discriminator. In the following subsections, we describe each stage of TAGGEN in details.

Context sampling. Inspired by the advances of network embedding approaches [227], we view the problem of temporal network context sampling as a form of local exploration in network neighborhood \mathcal{N}_{FT} via temporal random walks [66]. Specifically, given a temporal occurrence v^{t_v} , we aim to extract a set of sequences that are capable of generating its neighborhood $\mathcal{N}_{FT}(v^{t_v})$. Notice that in order to fairly and effectively sample neighborhood context, we should select the most representative temporal occurrences to serve as initial nodes from the entire data. Here we propose to estimate the *context importance* via computing the conditional probability $p(v^{t_v}|\mathcal{N}_{FT}(v^{t_v}))$ of each temporal occurrence v^{t_v} given its temporal network neighborhood context $\mathcal{N}_{FT}(v^{t_v})$ as follows.

$$p(v^{t_v}|\mathcal{N}_{FT}(v^{t_v})) = p(v^{t_v}|\mathcal{N}_S(v^{t_v}), \mathcal{N}_T(v^{t_v})) \quad (10.1)$$

where $\mathcal{N}_T(v^{t_v})$ and $\mathcal{N}_S(v^{t_v})$ denote the temporal neighborhood and structural neighborhood of v^{t_v} respectively.

$$\mathcal{N}_T(v^{t_v}) = \{v_i^{t_{v_i}} | |t_v - t_{v_i}| \leq t_{\mathcal{N}_{FT}}\} \quad (10.2)$$

$$\mathcal{N}_S(v^{t_v}) = \{v_i^{t_{v_i}} | f_{sp}(v_i^{t_{v_i}}, v^{t_v}) \leq d_{\mathcal{N}_{FT}}\} \quad (10.3)$$

Intuitively, when $p(v^{t_v}|\mathcal{N}_{FT}(v^{t_v}))$ is high, it turns out that v^{t_v} is a representative node in its neighborhood, which could be a good initial point for random walks; when $p(v^{t_v}|\mathcal{N}_{FT}(v^{t_v}))$ is low, it is highly possible that $p(v^{t_v})$ is an outlier point, whose behaviors deviate from its neighbors. A key challenge here is how to estimate $p(v^{t_v}|\mathcal{N}_{FT}(v^{t_v}))$. If $p(v^{t_v}|\mathcal{N}_T(v^{t_v}))$ and $p(v^{t_v}|\mathcal{N}_S(v^{t_v}))$ are independent to each other, it is easy to see

$$p(v^{t_v}|\mathcal{N}_{FT}(v^{t_v})) = p(v^{t_v}|\mathcal{N}_T(v^{t_v}))p(v^{t_v}|\mathcal{N}_S(v^{t_v})) \quad (10.4)$$

where $p(v^{t_v}|\mathcal{N}_T(v^{t_v}))$ and $p(v^{t_v}|\mathcal{N}_S(v^{t_v}))$ can be estimated via some heuristic methods [66, 227]. However, in real networks, the topology context and temporal context are correlated to some extent, which has been observed in [329]. For instance, the high-degree nodes

(i.e., $p(v^{t_v}|\mathcal{N}_S(v^{t_v}))$ is high) have a high probability to be active in a future timestamp (i.e., $p(v^{t_v}|\mathcal{N}_T(v^{t_v}))$ is high), and vice versa. These observations allow us to state a weak dependence [71] between the topology neighborhood distribution and temporal neighborhood distribution.

Definition 10.5 (Weak Dependence). For any $v^{t_v} \in \tilde{\mathcal{V}}$, the corresponding temporal neighborhood distribution and topology neighborhood distribution are weakly dependent on each other, such that, for $\delta > 0$, $p(v^{t_v}|\mathcal{N}_{FT}(v^{t_v})) \geq \delta[p(v^{t_v}|\mathcal{N}_T(v^{t_v}))p(v^{t_v}|\mathcal{N}_S(v^{t_v}))]$.

Lemma 10.1. 1 For any $v^{t_v} \in \tilde{\mathcal{V}}$, if the temporal neighborhood distribution $p(v^{t_v}|\mathcal{N}_T(v^{t_v}))$ and topology neighborhood distribution $p(v^{t_v}|\mathcal{N}_S(v^{t_v}))$ are weakly dependent on each other, then the following inequality holds:

$$\begin{aligned} p(v^{t_v}|\mathcal{N}_{FT}(v^{t_v})) & \\ \geq \alpha \frac{p(v^{t_v}|\mathcal{N}_S(v^{t_v}))p(v^{t_v}|\mathcal{N}_T(v^{t_v}))p(\mathcal{N}_S(v^{t_v}))p(\mathcal{N}_T(v^{t_v}))}{p(\mathcal{N}_S(v^{t_v}), \mathcal{N}_T(v^{t_v}))} \end{aligned} \quad (10.5)$$

where $\alpha = \frac{\delta}{p(v^{t_v})}$.

Proof. For any $v^{t_v} \in \tilde{\mathcal{G}}$, the context importance $p(v^{t_v}|\mathcal{N}_{FT}(v^{t_v}))$ can be estimated as

$$\begin{aligned} p(v^{t_v}|\mathcal{N}_{FT}(v^{t_v})) &= p(v^{t_v}|\mathcal{N}_S(v^{t_v}), \mathcal{N}_T(v^{t_v})) \\ &= \frac{p(v^{t_v}, \mathcal{N}_S(v^{t_v}), \mathcal{N}_T(v^{t_v}))}{p(\mathcal{N}_S(v^{t_v}), \mathcal{N}_T(v^{t_v}))} \end{aligned} \quad (10.6)$$

Since the corresponding temporal neighborhood distribution $p(v^{t_v}|\mathcal{N}_T(v^{t_v}))$ and topology neighborhood distribution $p(v^{t_v}|\mathcal{N}_S(v^{t_v}))$ satisfy a weak dependence, we can easily have

$$\begin{aligned} p(v^{t_v}|\mathcal{N}_{FT}(v^{t_v})) & \\ \geq \delta \frac{p(v^{t_v})p(\mathcal{N}_S(v^{t_v})|v^{t_v})p(\mathcal{N}_T(v^{t_v})|v^{t_v})}{p(\mathcal{N}_S(v^{t_v}), \mathcal{N}_T(v^{t_v}))} & \\ = \delta \frac{p(v^{t_v}) \frac{p(v^{t_v}|\mathcal{N}_S(v^{t_v}))p(\mathcal{N}_S(v^{t_v}))}{p(v^{t_v})} \frac{p(v^{t_v}|\mathcal{N}_T(v^{t_v}))p(\mathcal{N}_T(v^{t_v}))}{p(v^{t_v})}}{p(\mathcal{N}_S(v^{t_v}), \mathcal{N}_T(v^{t_v}))} & \\ = \alpha \frac{p(v^{t_v}|\mathcal{N}_S(v^{t_v}))p(v^{t_v}|\mathcal{N}_T(v^{t_v}))p(\mathcal{N}_S(v^{t_v}))p(\mathcal{N}_T(v^{t_v}))}{p(\mathcal{N}_S(v^{t_v}), \mathcal{N}_T(v^{t_v}))} \end{aligned} \quad (10.7)$$

QED.

Following [227], we assume $p(v^{t_v}|\mathcal{N}_S(v^{t_v}))$ and $p(v^{t_v}|\mathcal{N}_T(v^{t_v}))$ follow a uniform distribution, where all the temporal entities in a local region are equally important. Then, by computing $p(\mathcal{N}_S(v^{t_v}))$, $p(\mathcal{N}_T(v^{t_v}))$ and $p(\mathcal{N}_S(v^{t_v}), \mathcal{N}_T(v^{t_v}))$ (e.g., via kernel density estimation approaches [330]), we can infer the *context importance* $p(v^{t_v}|\mathcal{N}_{FT}(v^{t_v}))$ based on Eq. 10.5 for selecting initial nodes.

After selecting the initial temporal occurrence, we use the biased temporal random walk [66] to extract a collection of temporal walks for training TAGGEN. The key reasons for using random walk based sampling approaches are their flexibility of controlling sequence length and the capability of jointly capturing structural and temporal neighborhood context information, as shown in [66, 227, 229].

Sequence generation. To generate the synthetic temporal random walks, a straightforward solution is to train a sequence model by learning from the extracted random walks [154]. However, in the temporal network setting, it is unclear how to mimic the network evolution and produce temporal interaction networks. Therefore, in this chapter, we design a family of local operations, i.e., $\text{Action} = \{\text{add}, \text{delete}\}$, to perform *addition* and *deletion* of temporal entities and mimic the evolution of real dynamic networks. In particular, given a k -length temporal random walk $\widetilde{W}^{(i)} = \{\widetilde{w}_1^{(i)}, \dots, \widetilde{w}_k^{(i)}\}$, we first sample a candidate temporal walk segment $\widetilde{w}_j^{(i)} \in \widetilde{W}^{(i)}$ following a user-defined prior distribution $p(\widetilde{W}^{(i)})$. In this chapter, we assume $p(\widetilde{W}^{(i)})$ follows a uniform distribution, although the proposed techniques can be naturally extended to other types of prior distribution. Then, we randomly perform one of the following operations with probability $p_{\text{action}} = \{p_{\text{add}}, p_{\text{delete}}\}$.

- *add* : The *add* operation is done in a two-step fashion. First, we insert a place holder token in the candidate temporal walk segment $\widetilde{w}_j^{(i)} = (u_{\widetilde{w}_j^{(i)}}, v_{\widetilde{w}_j^{(i)}})^{t_{\widetilde{w}_j^{(i)}}}$, and then replace a new temporal entity $v^{t_{v^*}}$ with the place holder token such that $\widetilde{w}_j^{(i)}$ is broken into $\{(u_{\widetilde{w}_j^{(i)}}, v^*)^{t_{v^*}}, (v^*, v_{\widetilde{w}_j^{(i)}})^{t_{\widetilde{w}_j^{(i)}}}\}$. The length of the modified temporal random walk sequence $\widetilde{W}_{\text{add}}^{(i)}$ would be $k + 1$.
- *delete* : The *delete* operation removes the candidate temporal walk segment $\widetilde{w}_j^{(i)}$ from $\widetilde{W}^{(i)} = \{\widetilde{w}_1^{(i)}, \dots, \widetilde{w}_j^{(i)}, \dots, \widetilde{w}_k^{(i)}\}$, such that the length of the modified temporal random walk $\widetilde{W}_{\text{delete}}^{(i)}$ would be $k - 1$.

Sample discrimination. To ensure the generated graph context follows the similar global structure distribution as the input, TAGGEN is equipped with a discriminator model $f_\theta(\cdot)$, which aims to distinguish whether the generated temporal networks follow the same distribution as the original graphs. For each generated temporal random walk $\widetilde{W}_{\text{action}}^{(i)}$ after a certain operation $\text{action} = \{\text{add}, \text{delete}\}$, TAGGEN computes the conditional probability

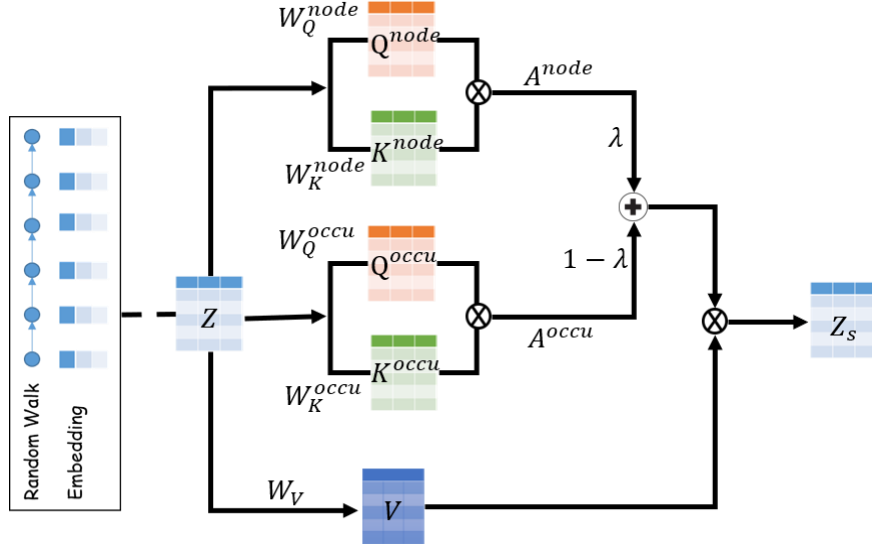


Figure 10.5: Bi-level self-attention.

$p(\widetilde{W}_{action}^{(i)}|W^{(1\sim l)})$ given the extracted real temporal random walks $W^{(1\sim l)} = \{W^{(1)}, \dots, W^{(l)}\}$ as follows.

$$p(\widetilde{W}_{action}^{(i)}|W^{(1\sim l)}) \propto p_{action}(action)f_{\theta}(\widetilde{W}_{action}^{(i)}) \quad (10.8)$$

where $f_{\theta}(\cdot)$ computes the likelihood of observing $\widetilde{W}_{action}^{(i)}$ given the training data $W^{(1\sim l)} = \{W^{(1)}, \dots, W^{(l)}\}$; $p_{action}(action)$ weights the proposed operation over $\widetilde{W}_{action}^{(i)}$.

Some recent graph generative frameworks (e.g., [154, 316]) model the extracted graph sequences via recurrent neural networks (RNNs) or long short-term memory (LSTM) architectures. However, such sequential nature inherently prevents parallelism and results in intractable running time for long sequence length [331]. For instance, GraphRNN [316] requires to map the n -node graph into length- n sequences for training purposes. Inspired by the recent advances of Transformer models in nature language processing [331], we propose to employ self-attention mechanisms to impose global dependencies among temporal entities (i.e., nodes and temporal occurrences) and reduce the overall sequential computation load. However, direct implementation with standard Transformer parameterization may fail to capture such bi-level dependencies (i.e., node-level dependencies and occurrence-level dependencies). Here, we propose a bi-level self-attention mechanism illustrated in Figure 10.5. In particular, given a k -length temporal random walk $\widetilde{W}^{(i)}$, we first obtain the d -dimensional representation $\mathbf{Z} \in \mathbb{R}^{n \times d}$ for each v^t (i.e., node v at timestamp t) via temporal network embedding approaches, e.g., [66]. As each node v is naturally represented as a bag of temporal

occurrences $v = \{v^{t1}, v^{t2}, \dots, v^T\}$, the bi-level self-attention mechanism is designed to jointly learn (1) the dependencies among nodes in $\tilde{\mathcal{G}}$ and (2) the dependencies among different temporal occurrences. Following the notations in [331], we define the occurrence-level attention $\mathbf{A}^{\text{occu}} \in \mathbb{R}^{n_r \times n_r}$ and node-level attention $\mathbf{A}^{\text{node}} \in \mathbb{R}^{n_r \times n_r}$ as follows.

$$\mathbf{A}^{\text{occu}}(v_i^{t1}, v_j^{t2}) = \frac{(\mathbf{z}_i^{t1} \mathbf{W}_Q^{\text{occu}}) \odot (\mathbf{z}_j^{t2} \mathbf{W}_K^{\text{occu}})}{\sqrt{d_k}} \quad (10.9)$$

$$\mathbf{A}^{\text{node}}(v_i^{t1}, v_j^{t2}) = \frac{(f_{agg}(\mathbf{z}_i^{t1}) \mathbf{W}_Q^{\text{node}}) \odot (f_{agg}(\mathbf{z}_j^{t2}) \mathbf{W}_K^{\text{node}})}{\sqrt{d_k}} \quad (10.10)$$

where $\mathbf{z}_i^{t1} \in \mathbb{R}^{1 \times d}$ is the d -dimensional embedding of node v_i^{t1} ; $\mathbf{W}_Q^{\text{occu}} \in \mathbb{R}^{d \times d_k}$ and $\mathbf{W}_K^{\text{occu}} \in \mathbb{R}^{d \times d_k}$ are the occurrence-level query weight matrix and key weight matrix, respectively; similarly, $\mathbf{W}_Q^{\text{node}} \in \mathbb{R}^{d \times d_k}$ and $\mathbf{W}_K^{\text{node}} \in \mathbb{R}^{d \times d_k}$ are the node-level query weight matrix and key weight matrix, respectively; d_k is a scaling factor; $f_{agg}(\cdot)$ is an aggregation function that summarizes all the occurrence-level information for each node. For implementation purposes, we define $f_{agg}(v_i^t) = \sum_{v_i^{t'} \in v_i^t} \mathbf{z}_i^{t'}$, such that $f_{agg}(v_i^{t1}) = f_{agg}(v_i^{t2})$ when $t1 \neq t2$. In this way, the entries (i.e., rows) in \mathbf{A}^{occu} and \mathbf{A}^{node} are exactly aligned. Moreover, we introduce a coefficient $\lambda \in [0, 1]$ to balance the occurrence-level attention and node-level attention and obtain the final bi-level self-attention \mathbf{Z}_s as follows.

$$\mathbf{Z}_s = [\lambda \times \text{softmax}(\mathbf{A}^{\text{node}}) + (1 - \lambda) \times \text{softmax}(\mathbf{A}^{\text{occu}})] \mathbf{V} \quad (10.11)$$

where $\mathbf{V} = \mathbf{W}_V \mathbf{Z}$ and \mathbf{W}_V denotes the value weight matrix.

With the single head attention described in Figure 10.5, we employ $h = 4$ parallel attention layers (i.e., heads) in discriminator $f_\theta(\cdot)$ for selecting the qualified synthetic random walks $\widetilde{W}_{action}^{(i)}$. The update rule of the hidden representations in $f_\theta(\cdot)$ is the same as the standard Transformer model defined in [331]. At the end of the stage 3, all of the selected synthetic temporal random walks via the $f_\theta(\cdot)$ will be fed to the beginning of Stage 2 (see Figure 10.4) to gradually modify these sequences until the user-defined stopping criteria are met and the sequences are ready for assembling (Stage 4).

Graph assembling. In the previous stage, we generate synthetic temporal random walks by gradually performing local operations on the extracted real temporal random walks. In this stage, we assemble all the generated temporal random walks and generate the temporal interaction networks. In particular, we first compute the frequency counts $s(e^{te})$ of

Algorithm 10.1: The TAGGEN Learning Framework.

Require: Temporal interaction network $\tilde{\mathcal{G}}$ and parameters including neighborhood range $d_{\mathcal{N}_{FT}}$, neighborhood time window $t_{\mathcal{N}_{FT}}$, number of initial node l , walks per initial temporal occurrences γ , walk length k and constants c_1 and $\xi \in (0.5, 1)$.

Ensure: Synthetic temporal interaction network $\tilde{\mathcal{G}}'$.

- 1: Sample l initial temporal occurrences based on Eq. 10.5.
 - 2: Sample γ temporal random walks starting from each initial temporal occurrence with neighborhood range $d_{\mathcal{N}_{FT}}$ and neighborhood time window $t_{\mathcal{N}_{FT}}$, and store them in \mathcal{S} .
 - 3: Train discriminator f_θ based on \mathcal{S} .
 - 4: Let $\mathcal{S}' = \{\}$.
 - 5: **for** $i = 1 : \gamma \times l$ **do**
 - 6: Initialize $\tilde{W}^{(i)}$ with the first entry in $W^{(i)}$, i.e., $\tilde{W}^{(i)} = \{w_1^{(i)}\}$.
 - 7: **for** $c = 1 : c_1$ **do**
 - 8: Sample a candidate temporal walk segment $\tilde{w}_j^{(i)}$ from $\tilde{W}^{(i)}$.
 - 9: Draw a number $random \sim Unif(0, 1)$.
 - 10: If $random < \xi$, perform *add* operation on $w_j^{(i)}$; if $random \leq \xi$, perform *delete* operation on $w_j^{(i)}$.
 - 11: If discriminator f_θ approves the proposal $\tilde{W}_{action}^{(i)}$, replace $\tilde{W}^{(i)}$ with $\tilde{W}_{action}^{(i)}$; if not, continue.
 - 12: **end for**
 - 13: Add $\tilde{W}^{(i)}$ into \mathcal{S}' .
 - 14: **end for**
 - 15: Construct $\tilde{\mathcal{G}}'$ based on \mathcal{S}' by ensuring all the temporal occurrences and timestamps are included in $\tilde{\mathcal{G}}'$.
-

each temporal edge $e^{te} = (u, v)^{te}$ in the generated temporal random walks. To ensure the frequency counts are reliable, we use a larger number of the extract temporal random walks from the original graphs to avoid the case where some unrepresented temporal occurrences (i.e., with a small degree) are not sampled. In order to transform these frequency counts to discrete temporal edges, we use the following strategies: (1) we firstly generate at least one temporal edge starting from each temporal occurrence v^{tv} with probability $p(v^{tv}, v^* \in \mathcal{N}_S(v^{tv})) = \frac{s(e^{te=(v, v^*)^{tv}})}{\sum_{v^* \in \mathcal{N}_S(v^{tv})} s(e^{te=(v, v^*)^{tv}})}$ to ensure all the observed temporal occurrences in $\tilde{\mathcal{G}}$ are included; (2) then we generate at least one temporal edge at each timestamp with probability $p(e^{te}) = \frac{s(e^{te})}{\sum_{e_i^{te_i}} s(e_i^{te_i})}$; (3) we generate the temporal edges with the largest counts until the generated graph has the same edge density as the original one. Note that the first two steps can be considered as pre-processing steps, which are independent from the sequence generation (Stage 2) in Figure 10.4.

10.4.2 Optimization Algorithm

To optimize TAGGEN, we use stochastic gradient descent [257] (SGD) to learn the hidden parameters of TAGGEN. The optimization algorithm is described in Alg. 10.1. The given inputs include the Temporal interaction network $\tilde{\mathcal{G}}$, neighborhood range $d_{\mathcal{N}_{FT}}$, neighborhood time window $t_{\mathcal{N}_{FT}}$, number of initial nodes l , walks per initial nodes γ , walk length k , the number of operations per sequence c_1 , and constant parameters $\xi \in (0.5, 1)$. With $\xi > 0.5$, we enforce the number of *add* operation to be larger than the number of *delete* operation. In this way, we can avoid the case of generating zero-entry temporal random walk sequences. From Step 1 to Step 3, we sample a set of temporal random walks \mathcal{S} from the input data and train the discriminator $f_\theta(\cdot)$. Step 4 to Step 14 is the main body of TAGGEN, which generates the exactly sample number of temporal random walks as in \mathcal{S} . We firstly initial each synthetic walk $\tilde{W}^{(i)}$ with first entry in $W^{(i)}$, i.e., $\tilde{W}^{(i)} = \{w_1^{(i)}\}$. From Step 7 to Step 12, we perform c_1 times operations (i.e., *add* and *delete*) to generate context for each synthetic walk $\tilde{W}^{(i)}$ and use discriminator $f_\theta(\cdot)$ to select the qualified temporal random walks to be stored in \mathcal{S}' . In the end, Step 15 constructs the \tilde{G}' based on \mathcal{S}' by ensuring all the temporal occurrences and timestamps are included in \tilde{G}' as discussed in the previous subsection regarding Stage 4.

Network	Nodes	Temporal Edges	Timestamps
EMAIL	986	332,334	26
DBLP	1,909	8,237	15
WIKI	7,118	95,333	6
MSG	1,899	20,296	28
BITCOIN	3,783	24,186	117
SO	3,262	13,077	36
MO	13,840	195,330	20

Table 10.1: Statistics of the network data sets.

10.5 EXPERIMENTAL EVALUATION

In this section, we demonstrate the performance of our proposed TAGGEN framework across seven real temporal networks in graph generation and data augmentation. Additional results regarding scalability analysis are reported in Appendix E.

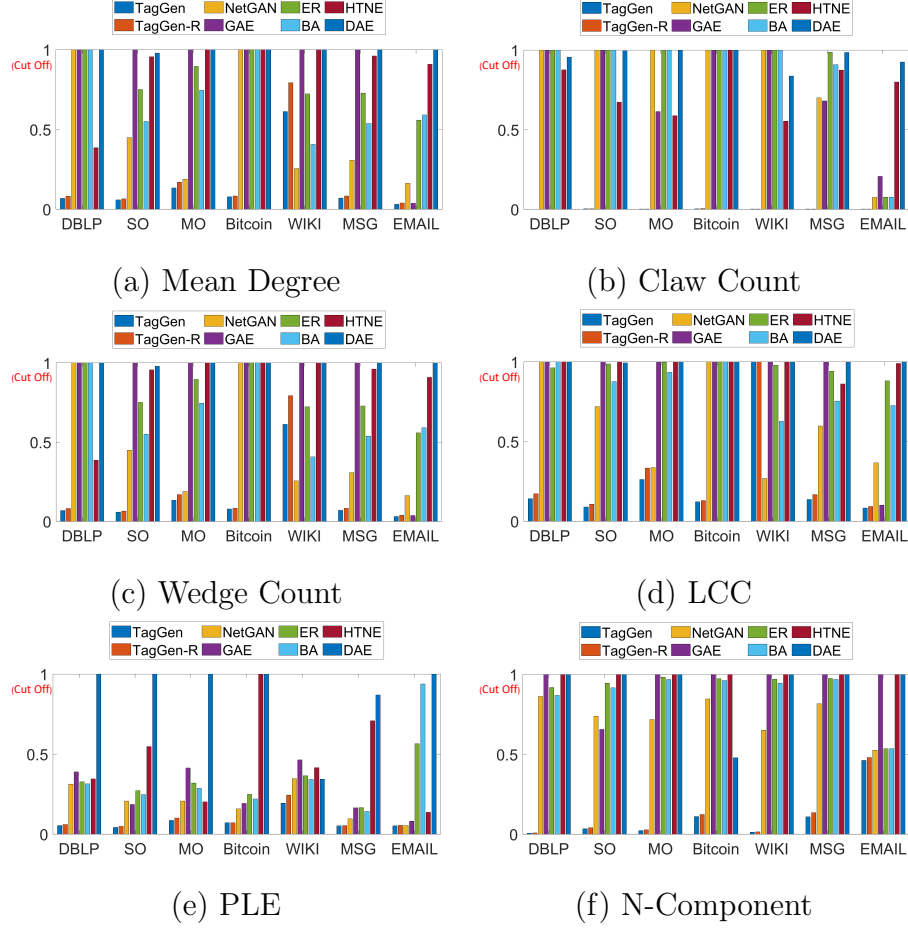


Figure 10.6: Average score $f_{avg}(\cdot)$ comparison with six metrics across seven temporal networks. Best viewed in color. We cut off high values for better visibility. (Smaller metric values indicate better performance)

10.5.1 Experiment Setup

Data Sets: We evaluate TAGGEN on seven real temporal networks. Specifically, DBLP [4] is a citation network that contains bibliographic information of the publications in IEEE Visualization Conference from 1990 to 2015; SO [4] and MO [327] are two collaboration networks where each node represents a user, and the edge represents one user’s comments on another user; WIKI [174] is a voting network, where each edge exists if the contributors vote to elect the administrators; EMAIL [327] and MSG [332] are two communication networks, where an edge exists if one person sends at least one email/message to another person at a certain timestamp; BITCOIN [333] is a who-trusts-whom network where people trade with bitcoins on a Bitcoin Alpha platform. The statistics of data sets are summarized in Table 10.1.

Metric name	Computation	Description
Mean Degree	$\mathbb{E}[d(v)]$	Mean degree of nodes in the graph.
Claw Count	$\sum_{v \in V} \binom{d(v)}{3}$	Number of the claw of the graph.
Wedge Count	$\sum_{v \in V} \binom{d(v)}{2}$	Number of wedges of the graph.
LCC	$\max_{f \in F} \ f\ $	Size of the largest connected component of the graph, where F is the set of all connected components in the graph.
PLE	$1 + n(\sum_{u \in V} \log(\frac{d(u)}{d_{min}}))^{-1}$	Exponent of the power-law distribution of the graph.
N-Component	$ F $	Number of connected components, where F is the set of all connected components in the graph.

Table 10.2: Graph statistics for measuring network properties.

Comparison Methods: We compare TAGGEN with two traditional graph generative models (i.e., Erdős-Rényi (ER) [224] and Barabási-Albert (BA) [311]), two deep graph generative models (GAE [334], NetGAN [154]), and two dynamic graph generators based on temporal network embedding approaches (HTNE [335], DAE [336]). Note that HTNE and GAE are not designed for graph generation. To generate temporal networks, we utilize the learned temporal network embedding to construct the adjacency matrix at each timestamp.

Evaluation Metrics: We evaluate the quality of the generated graphs by computing six network properties: (a) Mean Degree: the average degree of all nodes in the graph; (b) Claw Count: the claw count of the graph; (c) Wedge Count: the wedge count of the graph; (d) PLE: the exponent of the power-law distribution of the graph; (e) LCC: the size of the largest connected component of the graph; (f) N-Component: the number of connected components. (6) Assortativity: the degree assortativity of an input graph, which measures the similarity of connections in the graph concerning the node degree; (7) Gini_Coef: the Gini coefficient of the degree distribution of the graph; We provide the computation formula and description regarding these six metrics in Table 10.2. As all of these metrics are designed for static graphs, here we generalize the aforementioned metrics to the dynamic setting in the form of mean value and median value. In particular, given the real network $\tilde{\mathcal{G}}$, the synthetic one $\tilde{\mathcal{G}}'$ and a user-specific metric $f_m(\cdot)$, we first construct a sequence of snapshots \tilde{S}^t (\tilde{S}'^t), $t = 1, \dots, T$, of $\tilde{\mathcal{G}}$ ($\tilde{\mathcal{G}}'$) by aggregating from the initial timestamp to the current timestamp

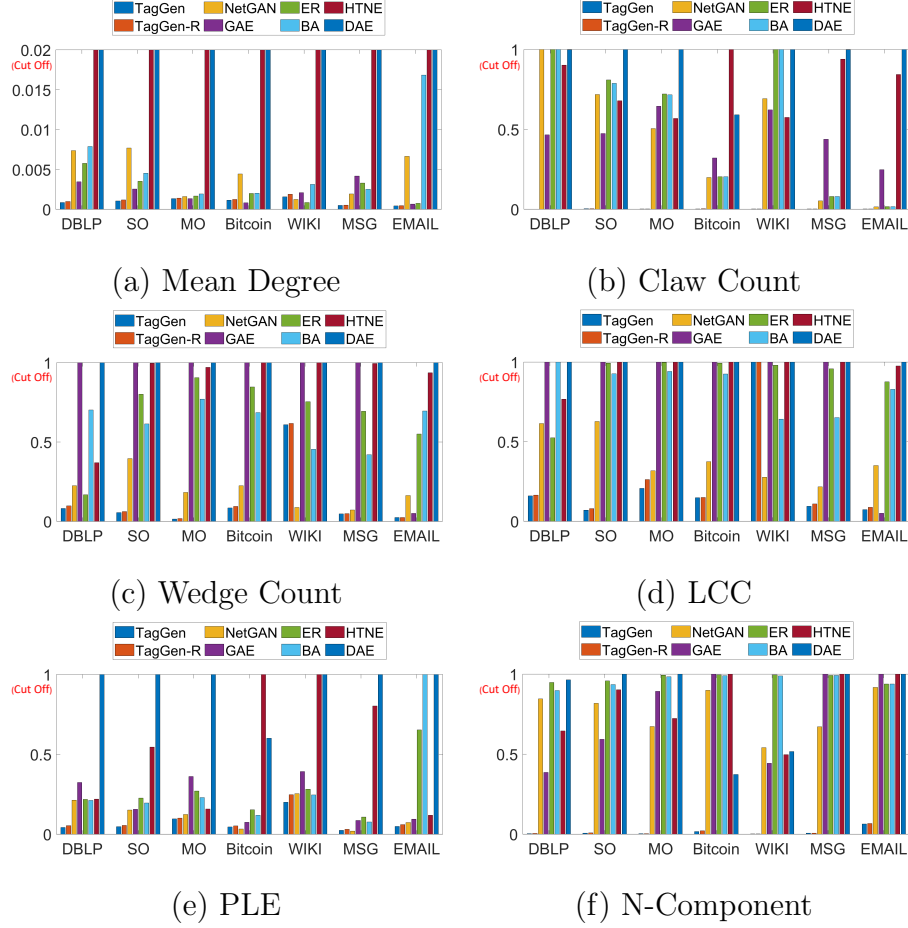


Figure 10.7: Median score $f_{med}(\cdot)$ comparison on six metrics across seven temporal networks. Best viewed in color. We cut off high values for better visibility. (Smaller metric values indicate better performance)

t . Then, we measure the averaged/median discrepancy (in percentage) between the original graph and the generated graph in terms of the given metric $f_m(\cdot)$ as follows.

$$f_{avg}(\tilde{\mathcal{G}}, \tilde{\mathcal{G}}', f_m) = Mean_{t=1:T}(|\frac{f_m(\tilde{\mathcal{S}}^t) - f_m(\tilde{\mathcal{S}}'^t)}{f_m(\tilde{\mathcal{S}}^t)}|) \quad (10.12)$$

$$f_{med}(\tilde{\mathcal{G}}, \tilde{\mathcal{G}}', f_m) = Median_{t=1:T}(|\frac{f_m(\tilde{\mathcal{S}}^t) - f_m(\tilde{\mathcal{S}}'^t)}{f_m(\tilde{\mathcal{S}}^t)}|) \quad (10.13)$$

10.5.2 Quantitative Results for Graph Generation

We compare TAGGEN with six baseline methods across seven dynamic networks regarding six network property metrics in the form of $f_{avg}(\cdot)$ and $f_{med}(\cdot)$ are shown in Figure 10.6 and

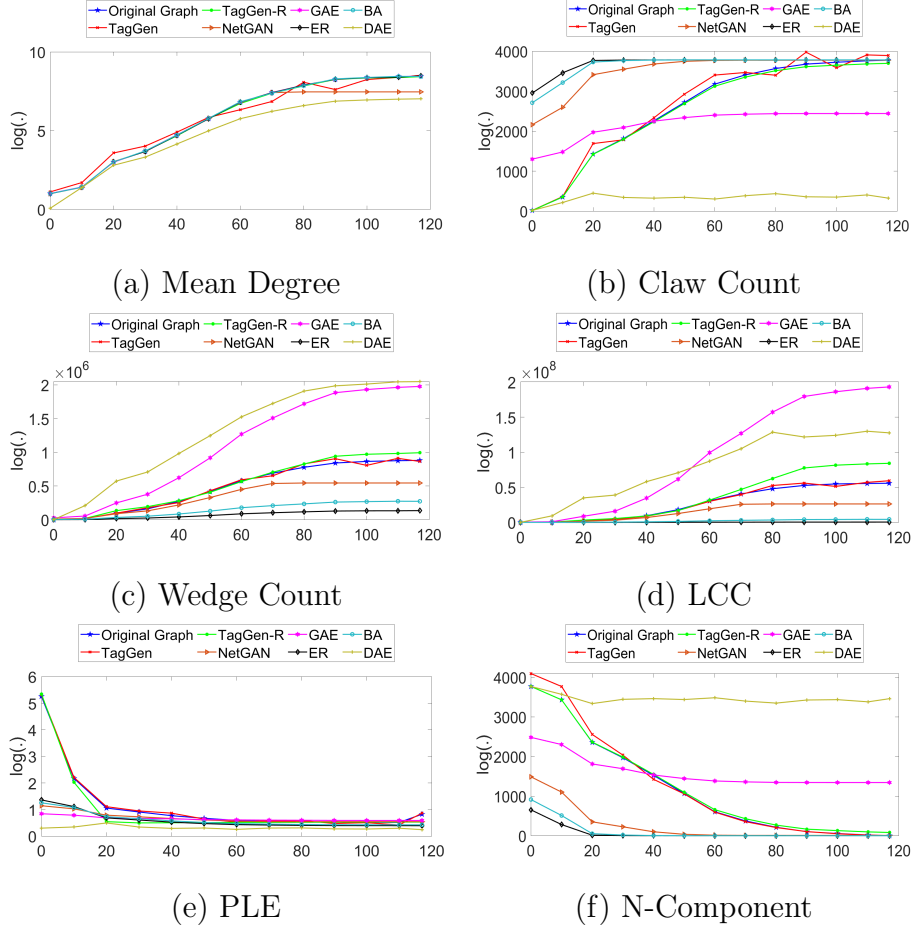


Figure 10.8: The comparison results on the six evaluation metrics across 117 timestamps in BITCOIN data set. Best viewed in color. The algorithm better fitting the curve of the original graph (colored in blue) is better.

Figure10.7. For the static methods, we apply them on the constructed graph snapshots at each timestamp and then report the results. In all of these figures, the performance is the smaller metric values, the better. For the sake of better visualization, the values of the scores are set to be one if any value is greater than 1. We draw several interesting observations from these results. (1) TAGGEN outperforms the baseline methods across the six evaluation metrics and seven data sets in most of the cases. (2) The random graph algorithms (i.e., ER and BA) perform well (i.e., close to TAGGEN and better than NetGAN and GAE) with Mean Degree (shown in Figure 10.6 (a) and Figure 10.7 (a)), but perform worse than the competitors with most of the other metrics. This is because such random graph algorithms are often designed to model a certain structural distribution (e.g., degree distribution) while falling short of capturing many other network properties (e.g., LCC and wedge count).

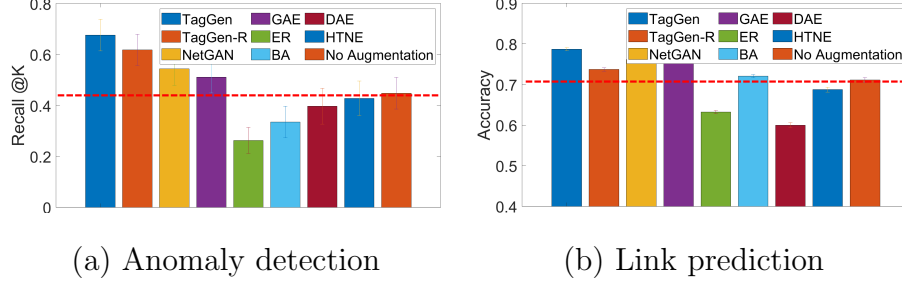


Figure 10.9: Data augmentation in SO

To further demonstrate the performance of TAGGEN, we experiment with the BITCOIN data set and evaluate the performance of all algorithms in terms of six different metrics in each timestamp. By doing this, we want to explore how the performances of the different methods change over 117 timestamps in the BITCOIN data set. The experimental results are shown in Figure 10.8, where the X-axis represents timestamp, and the Y-axis represents the value of a metric (labeled under each figure). In general, we observe (1) all the methods perform similarly well on Mean Degree metric; (2) TAGGEN consistently performs better than the baseline methods across six metrics and 117 timestamps as TAGGEN (colored in red) better fits the curves of the original graph (colored in blue). A simple guess here is that TAGGEN is the only dynamic graph generative model that can better track the trend of network evolution.

10.5.3 Case Studies in Data Augmentation

Anomaly Node Detection: In real-world networks, the performance of anomaly detection algorithms is often degraded due to data sparsity. Here, we conduct a case study of boosting the performance of anomaly node detection in SO data set via data augmentation. In particular, we select the labeled network SO as our evaluation data and consider a minority class (8%) in SO as the anomalies. In particular, we conduct 10-fold cross-validation and employ Recall@ K as the evaluation metric, where K is the total number of anomaly nodes in the test set. To assess the performance of anomaly node detection with data augmentation, we use the generative models to synthesize temporal edges and inject them into the original graph. Then, we encode the augmented temporal network into a node-wised representations [66], which is fed into the logistic regression model as inputs for classifying the malicious nodes. The experimental results are shown in Figure 10.9 (a), where No Augmentation (red dotted line) shows the result (Recall@ $K = 44.8\%$) of logistic regression directly trained on the embedding of the original graph without augmentation. The height of the

bars indicates the average value of Recall@ K , and the error bars represent the standard deviation in 10 runs. We observe that our proposed method boosts Recall@ K to 67.6% (22.8% improvement over the base model No Augmentation), while our best competitor NetGAN only achieves 54.3% (9.5% improvement over No Augmentation).

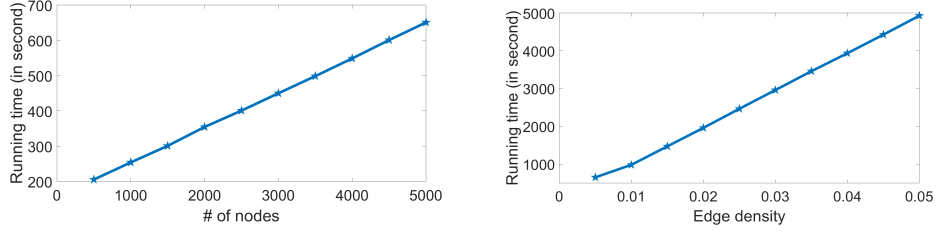
Link Prediction: In this experiment, we randomly select 50% of edges as the training data and the rest as the test data. Then, we compute the node embedding of both the original graph and the generated graph via CTDNE [66]. At last, we concatenate the two sets of node embedding and feed them into a logistic regression model to perform link prediction on the test data. In Figure 10.9 (b), the height of the bars indicates the average value of accuracy, and the error bars represent the standard deviation in 10 runs. It can be seen that NetGAN and GAE barely improve the performance of link prediction, while our proposed method TAGGEN increases the accuracy rate by 2.7% over the base model without data augmentation.

10.5.4 Scalability Analysis

We analyze the scalability of TAGGEN, by recording the running time (i.e., the sum of the training time and the time for graph generation) of TAGGEN on a series of synthetic graphs with increasing size. To be specific, we generate the synthetic graphs via ER algorithm [224], by which we can easily control the number of nodes and the number of edges in a graph. In the experiments, we set the batch size to be 128, the length of the random walk to be 20, the number of epochs to be 30, i.e., the same parameter settings as in the previous subsection. In Figure 10.10 (a), we fix the edge density to be 0.005, set the initial number of nodes to be 500, and increase the number of nodes by 500 each time. In Figure 10.10 (b), we fix the number of nodes to be 5,000 and increase the edge density from 0.005 to 0.05. Based on the results in Figure 10.10, we observe that the complexity of the proposed method is almost linear to the number of nodes. Besides, when we fix the number of nodes and increase the edge density, the running time also increases linearly.

10.6 SUMMARY

In this chapter, we propose TAGGEN - the first attempt to generate temporal networks by directly learning from a collection of timestamped edges. TAGGEN is able to generate graphs that capture important structural and temporal properties of the input data via a novel context sampling strategy together with a bi-level self-attention mechanism. We present comprehensive evaluations of TAGGEN by conducting the quantitative evaluation



(a) Running time vs. # of nodes (b) Running time vs. edge density

Figure 10.10: Scalability analysis

in temporal graph generation and two case studies of data augmentation in the context of anomaly detection and link prediction. We observe that: (1) TAGGEN consistently outperforms the baseline methods in seven data sets with six metrics; (2) TAGGEN boosts the performance of anomaly detection and link prediction approaches via data augmentation. However, key challenges remain in this space. One possible future direction is to develop generative models that can jointly model the evolving network structures and node attributes in order to generate attributed networks in the dynamic setting.

CHAPTER 11: FAIR GENERATION FOR RARE CATEGORIES ON GRAPHS

11.1 OVERVIEW AND MOTIVATION

The ever-increasing size of graphs, together with the difficulty of releasing and sharing them, has made the graph generation a fundamental problem that is key in many high-impact applications, including data augmentation [337], anomaly detection [338], drug design [339, 340], recommendation [154], and many more. For instance, financial institutes would like to share their transaction data or user networks with their partners to improve their service. However, directly releasing the real data would result in serious privacy issues. In this case, graph generative models provide an alternative solution without privacy concerns, by generating high-quality synthetic graphs for replacement. The classic graph-property oriented models are usually built upon succinct and elegant mathematical formula to preserve important structural properties, e.g., power-law degree distribution [311, 321, 325, 341], small world phenomena [324], shrinking diameters of dynamic graphs [312, 324], local clustering [313], motif distributions [342], etc. More recently, the data-driven models [154, 316, 343, 344, 345, 346, 347] have attracted much attention, which directly extract the contextual information from the input graphs and approximate their structure distribution with minimal prior assumptions.

Despite the tremendous success of existing graph generation techniques, the vast majority of existing graph generators are unsupervised and independent of downstream learning tasks. They are able to produce general-purpose graphs without considering any label information. However, in many real graphs, labels, such as identities of users [348] or community memberships [349], are available and could have a profound impact on the performance of downstream learning tasks. Considering an online transaction network (e.g., PayPal) that allows real-time money transfer among users and merchants, most of the transactions are normal while only a small amount of transactions are red-flagged (i.e., fraudulent transactions) by domain experts. Such label information could play a pivotal role in financial fraud detection (e.g., money laundering detection, identity theft detection). Therefore, if the graph generators neglect such label information, it is likely to negatively impact the downstream learning tasks (e.g., fraud detection).

Moreover, as the importance of model fairness has been widely recognized in the machine learning community, it is highly desirable to ensure certain parity or preference constraints in the learning process of generative models [350]. It is of key importance to ensure the protected group (e.g., the African Americans) and the unprotected group (e.g., the non-African

Americans) are treated equally in the generation process, especially when the generated data will be used for developing realistic AI systems (e.g., Correctional Offender Management Profiling for Alternative Sanctions (COMPAS) [351]). However, most, if not all, of the existing graph generative models are designed either prior to or in parallel with downstream tasks without considering model fairness in the generative process. The statistical nature of these models are designed to focus on the frequent patterns (i.e., the unprotected group), and as such, might overlook the underrepresented patterns (i.e., the protected group) in the observed data. As the protected groups contribute less to the general learning objective (e.g., minimizing the expected reconstruction loss [154]), they tend to suffer from the systematically higher errors. We refer to this phenomenon as *representation disparity* [352]. What is worse, as the protected groups are typically more scarce compared to the unprotected groups, it can be much more expensive to obtain label information from these groups than the unprotected groups in practice. As a consequence, the representation disparity issue could be further exacerbated when the models are learned from the highly skewed label information.

Therefore, in this chapter, we aim to tailor graph generation to downstream learning tasks, by incorporating label information and parity constraints. To this end, we have identified the following challenges. First (*Task Guidance*), how to train graph generative models under the guidance of ground-truth labels, so that the generated graphs are better suited for the downstream mining tasks comparing to the ones using general-purpose graph generators? Second (*Representation Disparity*), how to enforce the fairness constraints on the graph generative model so that the protected group and the unprotected group are treated equally in the generated graphs? Third (*Label Scarcity*), given limited label information (especially for the protected group), how to accurately capture the class-memberships of the protected groups in the input data and preserve them in the generated graph?

To this end, we propose a deep generative model named FAIRNET, which jointly trains a label-informed graph generation module and a fairness representation learning module in a mutually beneficial way. Moreover, To mitigate the representation disparity, FAIRNET integrates the self-paced learning paradigm in the graph generation process. It starts with few-shot labeled examples and then gradually propagates the labels from the ‘easy’ concepts to the ‘hard’ ones in order to accurately capture the behavior of the protected and unprotected groups in the input graphs. Moreover, to control the risk of protected groups, we propose a novel context sampling strategy for graph generative models, which is proven to be capable of capturing the context of each group S with a probability of at least $1 - T\delta\phi(S)$, where T is the maximum random walk length, $\phi(S)$ is the conductance of subgraph S , and δ is a positive constant.

The main contributions of this chapter are summarized below.

Problem. We formalize the *fair graph generation* problem and identify unique challenges motivated from the real applications.

Algorithms. We propose a self-paced graph generative model named FAIRNET, which incorporates the label information and fairness constraints to produce task-specific graphs.

Evaluation. We perform extensive experiments on seven real networks, which demonstrate that FAIRNET (1) achieves comparable performance as state-of-the-art graph generative models in terms of six widely-used metrics; (2) largely alleviates the representation disparity in the generated graphs; (3) significantly boosts the performance of subgraph detection via data augmentation.

The rest of the chapter is organized as follows. We review the related literature in Section 11.2. In Section 11.3, we introduce the notation and problem definition, followed by the details of our proposed framework FAIRNET in Section 11.4. Experimental results are reported in Section 11.5. Finally, we conclude this chapter in Section 11.6.

11.2 RELATED WORK

In this section, we provide a brief literature review regarding the topics of graph generative model, and fair machine learning.

11.2.1 Graph Generative Model

Graph generative models have a longstanding history, with applications in biology [353], chemistry [316], and social sciences [353]. Classic graph generators are often designed as network-property oriented models, which capture and reproduce one or more important structure properties, e.g., power law degree distribution [224, 311], small diameters [324], motif distribution [325, 326], and densification in graph evolution [312]. More recently, deep generative models [154, 316, 343] for graphs have received much research interest. For example, in [343], the authors propose a variational auto-encoders based framework named GraphVAE, which is designed to generate a number of small graphs and then employ a subgraph matching algorithm to assemble them into a complete graph with the same size of the original network; [305] studies the problem of molecular graph generation by proposing a novel generative model that can generate graphs with desired molecule structures and

physical properties; [316] proposes a deep autoregressive model that consists of a graph-level Recurrent neural network and an edge-level recurrent neural network to generate sequences of nodes and edges; [154] proposes a GAN-based graph generative model, where a generator is defined to generate synthetic random walks, and a discriminator is defined to distinguish them from the real ones that are sampled from the input graphs. As mentioned before, most of the existing works are predominately designed for producing general-purpose graphs. They overlook the label information and fairness requirements.

11.2.2 Fair Machine Learning

Fair machine learning aims to amend the biased machine learning models to be fair or invariant regarding specific variables. A surge of researches in fair machine learning has been done in the machine learning community. For example, [354] presents a learning algorithm for fair classification by enforcing group fairness and individual fairness in the obtained data representation; [355] proposes approaches to quantify and reduce bias in word embedding vectors that are trained from real-world data; in [352], the authors develop a robust optimization framework that minimizes the worst case risk over all distributions and preserves the minority group in an imbalanced data set; in [356], the authors present an adversarial-learning based framework for mitigating the undesired bias in modern machine learning models. in [357], the authors study the problem of automatic paper matching by proposing a local fairness formulation to guide the paper-assignment process; in [358], the authors propose an adversarial framework that provides flexibility to the end-users to accommodate different combinations of fairness constraints in learning network representations on knowledge graphs. To the best of our knowledge, we are the first to study the problem of debiasing representation disparity in the graph generative models.

11.3 PRELIMINARIES

Throughout the chapter, we use regular letters to denote scalars (e.g., α), boldface lowercase letters to denote vectors (e.g., \mathbf{v}), and boldface uppercase letters to denote matrices (e.g., \mathbf{A}). We formalize the graph generation problem in the context of undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} consists of n vertices, and \mathcal{E} consists of m edges. We let $\mathbf{A} \in \mathbb{R}^{n \times n}$ denote the adjacency matrix, $\mathbf{D} \in \mathbb{R}^{n \times n}$ denote the diagonal matrix of vertex degrees, and $\mathbf{I} \in \mathbb{R}^{n \times n}$ denote the identity matrix. The transition probability matrix \mathbf{M} of \mathcal{G} can be obtained by $\mathbf{M} = (\mathbf{A}\mathbf{D}^{-1} + \mathbf{I})/2$. We define an indicator vector $\chi_{\mathbf{S}} \in \mathbb{R}^n$ which is supported on a set of nodes $S \subset \mathcal{V}$, i.e., $\chi_{\mathbf{S}}(v) = 1$ iff $v \in S$; $\chi_{\mathbf{S}}(v) = 0$ otherwise.

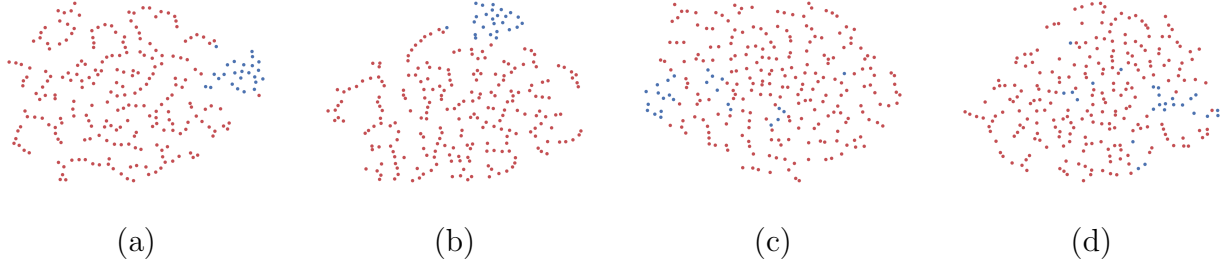


Figure 11.1: An illustrative example of representation disparity in deep graph generative models. The protected group is colored in blue while the unprotected group is colored in red. (a) Original graph; (b) NetGAN with 500 iterations; (c) NetGAN with 1000 iterations; (d) NetGAN with 2000 iterations.

In our problem setting, we are given a handful of labeled examples from C classes, as well as the membership of a protected group. Without loss of generality, we let $L = \{x_1, x_2, \dots, x_{|L|}\}$ denote the set of labeled vertices, which includes at least one from each class $y = 1, \dots, C$, $U = \{x_{|L|+1}, x_{|L|+2}, \dots, x_{|L|+|U|}\}$ denote the unlabeled vertices, $S^+ \subseteq \mathcal{V}$ denote the set of protected group vertices, and $S^- \subseteq \mathcal{V}$ denote the set of unprotected group vertices. Note that $S^- = \{x | x \in \mathcal{V} \text{ and } x \notin S^+\}$. Next, we elaborate on the background and motivation in the paragraphs below.

Learning Graph Generator from Random Walks. To generate a graph with n nodes, the generator has to output $O(n^2)$ variables to specify the adjacency matrix $\tilde{\mathbf{A}}$ of the synthetic graph $\tilde{\mathcal{G}}$. Instead of directly modeling the adjacency matrix, an alternative way [154, 316] is to train the generative model from the sequence representation of the observed graphs. In this chapter, we follow the idea of NetGAN [154], which trains over the k random walk sequences $\mathbb{W} = \{\mathbf{w}_1, \dots, \mathbf{w}_k\}$ extracted from the input graphs \mathcal{G} . Each random walk sequence \mathbf{w}_i , $i = 1, \dots, k$, consists of T incident nodes traversed one after another, i.e., $\mathbf{w}_i = \{x_{i,1}, \dots, x_{i,T}\}$, where $x_{i,j} \in \mathcal{V}$, $j = 1, \dots, T$. The learning objectives are defined to minimize the reconstruction error of generating synthetic random walks: $\tilde{\mathbf{w}} \sim g_\theta(\mathbb{W})$, where $\tilde{\mathbf{w}} = \{\tilde{x}_1, \dots, \tilde{x}_T\}$ is the synthetic random walk consisting of T vertices $\tilde{x}_i \in \mathcal{V}$, $i = 1, \dots, T$, g_θ denotes the recurrent neural network [359, 360] parameterized by θ . At last, all of the generated random walks will be used to assemble the adjacency matrix of the output synthetic graph $\tilde{\mathcal{G}}$.

Representation Disparity: Consider a standard graph generative model that is trained to minimize the reconstruction error of the input graph \mathcal{G} . Given the memberships of the protected group S^+ , we define the general graph reconstruction loss $R(\theta)$ and the group-wise graph reconstruction loss $R_S(\theta)$ as follows.

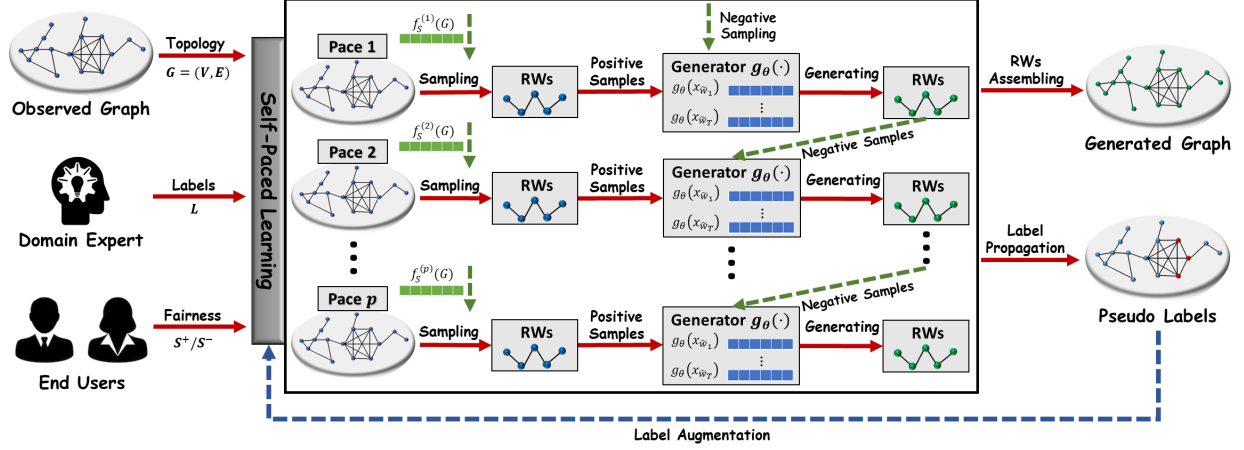


Figure 11.2: Overview of the proposed FAIRNET framework.

$$R(\theta) = -\mathbb{E}_{\mathbf{w} \in \mathcal{G}} \left[\sum_{t=1}^T \log g_{\theta}(w_t | \mathbf{w}_{<t}) \right] \quad (11.1)$$

$$R_S(\theta) = -\mathbb{E}_{\mathbf{w} \in \mathcal{G}_S} \left[\sum_{t=1}^T \log g_{\theta}(w_t | \mathbf{w}_{<t}) \right] \quad (11.2)$$

where \mathcal{G}_S refers to a subgraph in \mathcal{G} that is composed of a group of vertices $S \subseteq \mathcal{G}$; w_t and $\mathbf{w}_{<t}$ represent the t^{th} node and the first $(t-1)^{\text{th}}$ nodes in a sampled random walk \mathbf{w} . The objective of existing graph generative models typically aim to minimize Eq. 11.1 while ignoring the existence of the protected group S^+ that is typically under-represented. As the protected group S^+ contributes less to Eq. 11.1, it receives less attention from the generative model. As a result, the status quo of generative models may obtain a very low $R(\theta)$ but high $R_{S=S^+}(\theta)$. Following [352, 361], we refer to this phenomenon as the *representation disparity* in graph generative models. Figure 11.1 shows a motivating example of the representation disparity in graph generation. We consider one of the most popular deep graph generative models (i.e., NetGAN [154]) for the case study. We observe that the generated graphs in (b) initially maintains fairness (i.e., the protected group is well preserved in the embedding space), but the protected group in (c) and (d) becomes less prominent with more and more iterations. Here we formally define our problem below.

Problem 11.1. Fair Graph Generation

Input: (i) an observed undirected graph \mathcal{G} , (ii) few-shot labeled examples $L = \{x_1, \dots, x_{|L|}\}$,

(iii) the memberships of the protected group S^+ and the unprotected group S^- .

Output: the generated graph $\tilde{\mathcal{G}}$ that (i) captures the general structural properties of the input graph \mathcal{G} , (ii) agrees with the label information, and (iii) fairly preserves the contextual information of the protected group and the unprotected group.

11.4 ALGORITHM

In this section, we introduce our proposed FAIRNET framework. We first present an overview of FAIRNET together with its learning objective. Then we discuss its components on (1) label-informed graph generation, (2) mitigating the representation disparity in graph generation, and (3) the impact of self-paced learning. At last, we present an end-to-end optimization algorithm and graph assembling strategy for fair graph generation.

11.4.1 A Generic Joint Learning Framework

Given a graph \mathcal{G} associated with a handful of labeled nodes L and the membership of protected group S^+ , the goal of our framework is to learn a graph generator g_θ that agrees with the known label information, and fairly preserves the network context (i.e., structures and label information) of the protected group and the unprotected group in the generated graphs. With these design objectives in mind, we formulate FAIRNET as an optimization problem in Eq. 11.3 as follows.

$$\begin{aligned}
& \underset{\theta, \mathbf{v}^{(1)}, \dots, \mathbf{v}^{(C)}}{\operatorname{argmin}} \quad \mathcal{J} = \mathcal{J}_{\mathcal{G}} + \mathcal{J}_{\mathcal{P}} + \mathcal{J}_L + \mathcal{J}_F + \mathcal{J}_S \tag{11.3} \\
& = \underbrace{- \mathbb{E}_{\mathbf{w} \sim f_S(\mathcal{G})} \left[\sum_{t=1}^T \log g_\theta(w_t | \mathbf{w}_{<t}) \right]}_{\mathcal{J}_{\mathcal{G}}: \text{label informed generative model}} + \underbrace{\alpha \sum_{i=1}^L \xi_{x_i} d_\theta(x_i, y_i)}_{\mathcal{J}_{\mathcal{P}}: \text{prediction model}} \\
& \quad - \underbrace{\beta \sum_{i=1}^{L+U} \sum_{c=1}^C v_i^{(c)} \log \Pr(\hat{y}_i = c | x_i)}_{\mathcal{J}_L: \text{label propagation model}} + \underbrace{\gamma \sum_{c=1}^C \|m_c^+ - m_c^-\|}_{\mathcal{J}_F: \text{fairness regularizer}} - \underbrace{\lambda \sum_{i=1}^{L+U} \sum_{c=1}^C v_i^{(c)}}_{\mathcal{J}_S: \text{self-paced learning}}
\end{aligned}$$

It consists of five terms. The first term $\mathcal{J}_{\mathcal{G}}$ corresponds to the label informed generative model that minimizes the expected reconstruction error of the sampled random walk sequence \mathbf{w} using the label informed sampling strategy f_S (described later by Algorithm 11.1 in Subsection 3.1). The second term $\mathcal{J}_{\mathcal{P}}$ minimizes the weighted prediction loss for the training data L , where the function ξ_{x_i} defines the cost-sensitive ratios regarding the protected group

and unprotected group as follows.

$$\xi_{x_i} = \begin{cases} 1/|S^+| & x_i \in S^+ \\ 1/|S^-| & \text{Otherwise.} \end{cases} \quad (11.4)$$

where $|S^+|$ ($|S^-|$) denotes the cardinality of S^+ (S^-). Intuitively, as the protected group often corresponds to the minorities, i.e., $|S^+| \ll |S^-|$, we set $\xi_{x_i} \gg \xi_{x_j}$ for $x_i \in S^+$ and $x_j \in S^-$. By enforcing a higher loss of mis-classifying protected group nodes, the predictor d_θ tends to pay more attention to the underrepresented protected group S^+ . The third term \mathcal{J}_L corresponds to the label propagation model that maximizes the likelihood of observing x_i in its predicted class $\hat{y}_i = c$. The fourth term \mathcal{J}_F is the fairness regularizer, where m_c^+ and m_c^- denote the statistical parity measure [354] regarding the protected group S^+ and the unprotected group S^- , respectively; the last term is the self-paced regularizer, which globally maintains the learning pace of graph generation and label propagation, in which $\mathbf{v}^{(c)} \in \{0, 1\}^{n \times 1}$ denotes the self-paced vectors regarding the class $c = 1, \dots, C$; $\alpha, \beta, \gamma, \lambda$ are positive constants to balance the impact of each term on the overall objective function. We estimate the posterior probability $Pr(y_i|x_i)$ in Eq. 11.3 via the softmax function as

$$Pr(y_i|x_i) = \frac{\exp[\mathbf{h}^k(x_i)'\theta_{\mathbf{y}_i}]}{\sum_{y'_i} \exp[\mathbf{h}^k(x_i)'\theta_{\mathbf{y}'_i}]} \quad (11.5)$$

where $\mathbf{h}^k(x_i)$ denotes the k^{th} hidden representation of node x_i learned by d_θ .

An overview of FAIRNET is presented in Figure 11.2. It consists of three major components, including (M1) label informed graph generator module (i.e., \mathcal{J}_G), (M2) fairness learning module (i.e., $\mathcal{J}_P + \mathcal{J}_L + \mathcal{J}_F$), and (M3) self-paced learning paradigm (i.e., \mathcal{J}_S). In particular, M1 aims to address task guidance, by incorporating the label information from downstream tasks in the graph generation process; M2 targets to address representation disparity and label scarcity, by maintaining an ‘accurate and fair’ label propagation; and M3 serves as an intermediary agent, which globally maintains the learning pace of M1 and M2 by learning from the ‘easy’ concepts to the ‘hard’ ones. By the end of each cycle (i.e., learning pace [210]), the generated random walks and propagated pseudo labels will be fed to the next cycle as inputs for updating M1 and M2, respectively. At last, all the generated random walks will be fed into an assembling module for generating the final graph. Next, we will discuss the three modules in detail.

M1. Label-informed graph generation: The existing RNN-based graph generative models [154, 316] often suffer from the long training process when modeling large-scale

networks. Inspired by the Transformer model [331, 362, 363] in modeling long symbolic sequences, we formulate the generator as

$$\arg \min_{\theta} -\mathbb{E}_{\mathbf{w} \sim f_S(\mathcal{G})} \left[\sum_{t=1}^T \log g_{\theta}(w_t | \mathbf{w}_{<t}) \right] \quad (11.6)$$

where g_{θ} is the Transformer-based generator [331]; $f_S(\cdot)$ is a label-informed context sampling function; w_t and $\mathbf{w}_{<t}$ represent the t^{th} node and the first $t-1$ nodes in a specific random walk \mathbf{w} . Different from the general graph reconstruction loss $R(\theta)$ that is described in Eq. 11.1, we propose $\mathcal{J}_{\mathcal{G}}$ to approximately minimize $R_S(\theta)$ in Eq. 11.2 across both protected group (i.e., $S = S^+$) and unprotected group (i.e., $S = S^-$) via $f_S(\cdot)$. In particular, $f_S(\cdot)$ is designed to extract two types of context information from the input data. The first type of context is based on the graph \mathcal{G} , which encodes the general structure distribution by minimizing $R(\theta)$ in Eq. 11.1. The second type of context is based on the label information, which encodes the group-wise context information. In Figure 11.3, we present an example of extracting two types of random walks via $f_S(\cdot)$ on a toy graph. Without loss of generality, we assume that all the labeled examples are representative, i.e., located within the diffusion cores [364] of the corresponding classes, as defined below.

Definition 11.1. [Diffusion Core] For any subgraph $S \subseteq \mathcal{G}$, the (δ, t) -diffusion core of S is defined as $C^S = \{x \in S | 1 - \chi_S M^t \chi_x < \delta \phi(S)\}$, where $\delta \in (0, 1)$, $M = (AD^{-1} + I)/2$ is the transition probability matrix, χ_S and χ_x are two indicator vectors supported on S and $\{x\}$, and $\phi(S)$ denotes the conductance of S in \mathcal{G} .

Note that $1 - \chi_S M^t \chi_x$ computes the probability of a random walk starting from node $x \in S$ and escaping S after t steps. Roughly speaking, C^S is the set of nodes that are connected with each other within the cluster S . Next, in Lemma 11.1, we show that if the labeled example is located in the diffusion core of S , the extracted random walk sequences will purely preserve the context information within S with a high probability. For example, in Figure 11.3, we can see label informed random walks (colored in red) starting from a labeled example (indicated by the green arrow) in C^S (i.e., the clique bounded by the orange box) traverse within the cluster S (bounded by the blue box).

Lemma 11.1. 1 If the labeled example x_i is located in the diffusion core of a cluster S , i.e., $x_i \in C^S$, then the sampled T -length random walks starting from x_i only capture the context information within S with a probability of at least $1 - T\delta\phi(S)$.

Proof. To ensure the sampled random walks \mathbf{w} only preserves the the context information of S , we need \mathbf{w} stays entirely inside of S . Note that $M\chi_x$ is the distribution mass that a

one-step random walk start from x_i and $\text{diag}(\chi_S)M\chi_x$ is the truncated distribution when the \mathbf{w} stays inside S . Therefore, the probability of the extracted T -length random walks entirely staying inside of cluster S is $\mathbf{1}'(\text{diag}(\chi_S)M)^t\chi_x$.

For any $1 \leq t \leq T$, we can have

$$\begin{aligned}
& \mathbf{1}'(\text{diag}(\chi_S)M)^{t-1}\chi_x - \mathbf{1}'(\text{diag}(\chi_S)M)^t\chi_x \\
&= \mathbf{1}'(I - \text{diag}(\chi_S)M)(\text{diag}(\chi_S)M)^{t-1}\chi_x \\
&= \mathbf{1}'(M - \text{diag}(\chi_S)M)(\text{diag}(\chi_S)M)^{t-1}\chi_x \\
&= \mathbf{1}'(I - \text{diag}(\chi_S))M(\text{diag}(\chi_S)M)^{t-1}\chi_x \\
&= \chi_{\bar{S}}M(\text{diag}(\chi_S)M)^{t-1}\chi_x \\
&\leq \delta\phi(S)\chi_{\bar{S}}M^t\chi_x
\end{aligned} \tag{11.7}$$

Based on Def. 11.1, we have

$$\mathbf{1}'(\text{diag}(\chi_S)M)^{t-1}\chi_x - \mathbf{1}'(\text{diag}(\chi_S)M)^t\chi_x \leq \delta\phi(S) \tag{11.8}$$

For $t = 1, \dots, T$, the Eq. 11.8 can be written as follows.

$$1 - \mathbf{1}'(\text{diag}(\chi_S)M)^1\chi_x \leq \delta\phi(S) \tag{11.9}$$

$$\mathbf{1}'(\text{diag}(\chi_S)M)^1\chi_x - \mathbf{1}'(\text{diag}(\chi_S)M)^2\chi_x \leq \delta\phi(S) \tag{11.10}$$

\vdots

$$\mathbf{1}'(\text{diag}(\chi_S)M)^{T-1}\chi_x - \mathbf{1}'(\text{diag}(\chi_S)M)^T\chi_x \leq \delta\phi(S) \tag{11.11}$$

By adding up the above T inequalities, we have

$$1 - \mathbf{1}'(\text{diag}(\chi_S)M)^T\chi_x \leq T\delta\phi(S) \tag{11.12}$$

Thus, we has proofed that \mathbf{w} only preserves the context information of S with the probability of $\mathbf{1}'(\text{diag}(\chi_S)M)^T \geq 1 - T\delta\phi(S)$.

QED.

In practice, we want to control S to be compact, such that (1) $\phi(S)$ is small and $1 - T\delta\phi(S)$ is close to 1; (2) the extracted group-wise contextual information to be meaningful. We describe the technical details of $f_S(\cdot)$ by Algorithm 11.1 in this subsection. We first sample a random number $r \in [0, 1]$. Then, with probability r , we uniformly sample a T -length

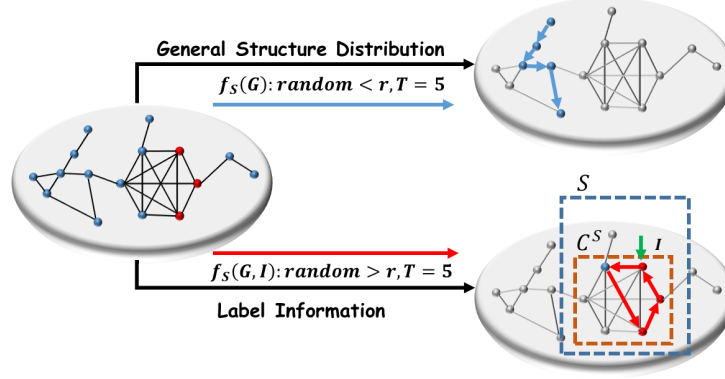


Figure 11.3: An illustrative example of random walk extraction via $f_S(\cdot)$, where the red dots represent the labeled examples, and the blue dots represent the unlabeled examples. With probability r , $f_S(\cdot)$ samples random walks (colored blue) for capturing general structure distribution; with probability $1 - r$, $f_S(\cdot)$ samples random walks (colored red) starting from a labeled example (pointed by a green arrow).

Algorithm 11.1: Label Informed Context Sampling.

Require:

Graph \mathcal{G} , indicator vector \mathbf{v}_{agg} and parameters T, r .

Ensure:

The set of sampled random walks \mathbb{W} .

- 1: Draw a number $random \sim Unif(0, 1)$.
 - 2: **if** $random < r$ **then**
 - 3: Uniformly sample a random walk \mathbf{w} of length T by existing methods, e.g., [229].
 - 4: **else**
 - 5: Randomly select an initial vertex x_i from the nonzero elements in \mathbf{v}_{agg} and conduct a random walk \mathbf{w} of length T .
 - 6: **end if**
-

random walk \mathbf{w} via the biased second-order random walk sampling strategy [229]; with probability $1 - r$, we sample graph context with the guidance of label information. Due to (C3) label scarcity of protected groups, directly sampling random walks starting from the labeled examples may result in amplifying the representation disparity in the generated graphs. Here we propose an indicator vector

$$\mathbf{v}_{agg} = \sum_{c=1}^C \mathbf{v}^{(c)} \quad (11.13)$$

where $\mathbf{v}^{(c)} \in \{0, 1\}^{n \times 1}$ is the self-paced vectors for the c^{th} class. In this way, we collect all the labeled examples together with the pseudo labeled nodes to be seed nodes for context sampling via $f_S(\cdot)$. The details of computing $\mathbf{v}^{(c)}$ will be discussed in M3 below. All the

sampled random walks via $f_S(\cdot)$ will be used to train g_θ by minimizing \mathcal{J}_G .

M2. Mitigating representation disparity in graph generation: Through M1, we encode the general structure distribution and the label information of the input data into the graph generator g_θ via $f_S(\cdot)$. Nevertheless, simply minimizing the reconstruction loss defined in Eq. 11.6 may overlook the protected group nodes, due to the imbalanced nature between the protected and unprotected groups, i.e., the majority of labeled examples come from the unprotected groups. To minimize the risk of representation disparity in g_θ , we propose a self-paced label propagation to gradually generate ‘*accurate and fair*’ labels to be fed to $f_S(\cdot)$ for label-informed context sampling. In particular, given a handful of labeled examples together with the membership of the protected group, the learning objective of this module is to minimize the following three terms (i.e., $\mathcal{J}_P + \mathcal{J}_L + \mathcal{J}_F$) below.

$$\begin{aligned} \arg \min_{\theta} \quad & \alpha \sum_{i=1}^L \xi_{x_i} d_\theta(x_i, y_i) - \beta \sum_{i=1}^{L+U} \sum_{c=1}^C v_i^{(c)} \log \Pr(\hat{y}_i = c | x_i) \\ & + \gamma \sum_{c=1}^C \|m_c^+ - m_c^-\| \end{aligned} \quad (11.14)$$

where the posterior probability $\Pr(y_i | x_i)$ is computed via softmax function shown in Eq. 11.5. In Eq. 11.14, the first term is the standard supervised loss function with cost-sensitive ratio ξ_{x_i} , where d_θ can be formulated as cross-entropy loss, mean squared loss or hinge loss. The second term ensures the label propagation is ‘*accurate*’ by maximizing the likelihood probability $\Pr(\hat{y}_i = c | x_i)$ that can be computed as the softmax of the output of d_θ . Different from many label propagation algorithms on graphs [365, 366], we incorporate the label examples to regularize our self-paced label propagation, by forcing $v_i^{(c)} = 1$ for all the labeled examples $x_i \in L$ with class label $y_i = c, c = 1, \dots, C$. The last term guarantees the label propagation is ‘*fair*’ via statistical parity constraint [354], where

$$m_c^+ = \frac{1}{|S^+|} \sum_{x_i \in S^+} \log \Pr(\hat{y}_i = c | x_i) \quad (11.15)$$

$$m_c^- = \frac{1}{|S^-|} \sum_{x_j \in S^-} \log \Pr(\hat{y}_j = c | x_j) \quad (11.16)$$

Intuitively, we would like to ensure the expected probability of a protected group node $x_i \in S^+$ from a particular class $\hat{y}_i = c$ is close to the expected probability of an unprotected group node $x_j \in S^-$ belonging to the same class $\hat{y}_j = c$. For example, in a professional

network, we want to ensure the female programmers (protected group S^+) have the same chance to be promoted to the position of the principal scientist as the male programmers (unprotected group S^-) in an IT company. As shown in Figure 11.2, in each iteration, FAIRNET feeds the generated pseudo labels in M2 as well as the ground truth labels to $f_S(\cdot)$ for training g_θ via negative sampling [249, 367].

M3. The impact of self-paced learning: Here we present the self-paced learning module that globally maintains the learning pace of the graph context extraction in M1 and the label propagation in M2, such that the two modules are trained in a mutually beneficial way. To be specific, at each self-paced cycle $l = 1, \dots, p$ shown in Figure 11.2, M3 first computes the self-paced vectors $\mathbf{v}^{(c)}$, $c = 1, \dots, C$, to assign pseudo labels to a set of unlabeled vertices using the self-paced threshold λ and the learned predictive model d_θ in the last cycle $l - 1$; then M1 samples new random walks based on the updated self-paced vectors $\mathbf{v}^{(c)}$ and updates the generative model in Eq. 11.6 via negative sampling [229]. In particular, at each cycle $l = 2, \dots, p$, we treat the newly sampled random walks via $f_S(\cdot)$ as positive samples and the generated random walks from last cycle $l - 1$ as negative samples. By this way, we gradually increase the learning difficulty of g_θ and force it to distinguish the characteristics of the real random walks from the fake ones, in order to better model the distribution of the protected and the unprotected groups; in the meanwhile, M2 updates the predictive model by learning from the augmented training data (i.e., labeled data and pseudo labeled data) that is preserved in the updated self-paced vectors $\mathbf{v}^{(c)}$.

As we can see, the self-paced vectors $\mathbf{v}^{(c)}$, $c = 1, \dots, C$, serve as a key component for training M1 and M2. The general philosophy of self-paced learning [210] is to learn from the ‘easy’ concepts to the ‘hard’ ones following the cognitive mechanism of human beings. In particular, we gradually increase the value of λ for increasing the learning difficulty, which will be used to update the self-paced vectors in the next learning cycle. By taking the partial derivative of \mathcal{J} in Eq. 11.3, $\mathbf{v}^{(c)}$ can be written as follows.

$$\frac{\partial \mathcal{J}}{\partial v_i^{(c)}} = -\log \Pr(\hat{y}_i = c | \mathbf{x}_i) - \lambda \quad (11.17)$$

Thus, the closed-form solution of updating $v_i^{(c)}$ can be obtained as

$$v_i^{(c)} = \begin{cases} 1 & -\log \Pr(\hat{y}_i = c | x_i) < \lambda \\ 0 & \text{Otherwise} \end{cases} \quad (11.18)$$

Intuitively, λ serves as a learning threshold for selecting the nodes with less prediction loss to be labeled. In particular, when $v_i^{(c)} = 1$, it indicates FAIRNET classifies x_i to class c with a high confidence $\log \Pr(\hat{y}_i = c | x_i) > -\lambda$; when $v_i^{(c)} = 0$, it indicates the prediction loss

Algorithm 11.2: The FAIRNET Learning Framework.

Require:

- (i) an undirected graph \mathcal{G} , (ii) few-shot labeled examples $L = \{x_1, \dots, x_{|L|}\}$,
- (iii) the memberships of the protected group vertices S^+ . (iv) parameters T , K , T_1 , N_1 , p , r , α , β , γ , λ .

Ensure:

- Generative model g_θ , predictive model d_θ , self-paced vectors $\mathbf{v}^{(1)}, \dots, \mathbf{v}^{(C)}$.
- 1: Initialize the predictive model $d_\theta(\cdot)$ and the self-paced vectors $\mathbf{v}^{(1)}, \dots, \mathbf{v}^{(C)}$ based on the labeled vertices L .
 - 2: Sample K positive random walks via Algorithm 11.1 and store them in \mathcal{N}^+ ; sample K negative random walks based on [229] and store them in \mathcal{N}^- .
 - 3: **for** $l = 1, \dots, p$ **do**
 - 4: Update the hidden parameters θ of the generator g_θ by training from \mathcal{N}^+ and \mathcal{N}^- .
 - 5: Sample K positive random walks by Algorithm 11.1 with the updated self-paced vectors $\mathbf{v}^{(1)}, \dots, \mathbf{v}^{(C)}$ and add them to \mathcal{N}^+ .
 - 6: Sample K negative random walks using the current generative model g_θ and add them to \mathcal{N}^- .
 - 7: Augment the value of λ .
 - 8: Update $\mathbf{v}^{(1)}, \dots, \mathbf{v}^{(C)}$ based on Eq. 11.18 and augment L with the pseudo labeled vertices.
 - 9: **for** $t = 1 : T_1$ **do**
 - 10: Sample N_1 labeled vertices from L and update hidden layers' parameters θ by taking a gradient step with respect to $\mathcal{J}_P + \mathcal{J}_L + \mathcal{J}_F$.
 - 11: **end for**
 - 12: **end for**
-

$-\log \Pr(\hat{y}_i = c|x_i)$ is higher than the threshold λ . By monitoring the increase rate of λ over self-paced cycles $l = 1, \dots, p$, the end users can easily control the learning pace and learning difficulty. In fact, FAIRNET propagates pseudo labels to the unlabeled vertices from the easy (i.e., the ones with a small loss $-\log \Pr(\hat{y}_i = 1|\mathbf{x}_i)$) to the hard (i.e., the ones with a large loss $-\log \Pr(\hat{y}_i = 1|\mathbf{x}_i)$) with a controllable pace. Next, we present the overall optimization algorithm to jointly train the aforementioned three modules.

11.4.2 Optimization algorithm

To optimize the objective function described in Eq. 11.3, we present the optimization algorithm in Algorithm 11.2 for learning FAIRNET framework. The inputs include an undirected graph \mathcal{G} together with the labeled vertices L , the memberships of the protected group S^+ , the length of random walks T , the number of sampled random walks K , batch iterations T_1 ,

batch size N_1 , the number of self-paced cycles p and parameters $r, \alpha, \beta, \gamma, \lambda$. In Step 1, we first initialize the predictive model $d_\theta(\cdot)$ and the self-paced vectors $\mathbf{v}^{(1)}, \dots, \mathbf{v}^{(C)}$ based on the labeled vertices L . Specifically, we let $\mathbf{v}_i^{(c)} = 1$ for all the vertices x_i labeled as class c ; otherwise, $\mathbf{v}_i^{(c)} = 0$. Step 2 samples K positive random walks and K negative random walks and stores them in \mathcal{N}^+ and \mathcal{N}^- respectively. Step 3 to Step 12 is the main body of Algorithm 11.2. In particular, at each self-paced cycle $l = 1, \dots, p$, Step 4 updates the generative model $g_\theta(\cdot)$ by learning from \mathcal{N}^+ and \mathcal{N}^- . Step 5 and Step 6 sample new positive random walks and negative random walks for training $g_\theta(\cdot)$ in the next cycle $l+1$. By adding the generated random walks to \mathcal{N}^- , we are increasing the difficulty of training $g_\theta(\cdot)$. In this way, we enforce $g_\theta(\cdot)$ to distinguish the characteristics of the real random walks from the fake ones and then generate better random walks that are plausible in the real graph. Step 7 and Step 8 update the self-paced vectors and λ , which will be used to augment the training set L with the pseudo labeled vertices. At last, from Step 9 to Step 11, we employ stochastic gradient descent (SGD) [257] to minimize the objective function of $M2$.

11.4.3 Fair Network (FAIRNET) Assembling

After obtaining g_θ and d_θ by optimizing FAIRNET, we construct a score matrix $\mathbf{S} \in \mathbb{R}^{n \times n}$ to infer the adjacency matrix $\tilde{\mathbf{A}}$ of the output graph $\tilde{\mathcal{G}}$. In particular, we let the learned generative model g_θ continuously generate synthetic random walks $\tilde{\mathbf{w}}$, and then collect the counts of each observed edge (i, j) to be stored in $\mathbf{S}(i, j)$. However, simply thresholding \mathbf{S} to produce $\tilde{\mathbf{A}}$ may lead to the low-degree nodes or protected groups nodes being left out. Here, we propose the following assembling criteria: (1) the protected group S^+ in the generated graph $\tilde{\mathcal{G}}$ should have a similar/ identical volume (total number of edges) as the original graph \mathcal{G} ; (2) each node should have at least one connected edge in the generated graph $\tilde{\mathcal{G}}$. Typically, we generate a much larger number of random walks than the sampled ones, which is beneficial to ensure the overall quality and to reduce the randomness of the generated graphs. At the end, we threshold \mathbf{S} to produce $\tilde{\mathbf{A}}$, which has the identical number of edges in \mathbf{A} .

11.5 EXPERIMENTAL EVALUATION

We empirically demonstrate the performance of FAIRNET on both synthetic and real graphs to evaluate the following aspects:

- Graph Generation: We compare the quality of the generated graphs with 5 baseline

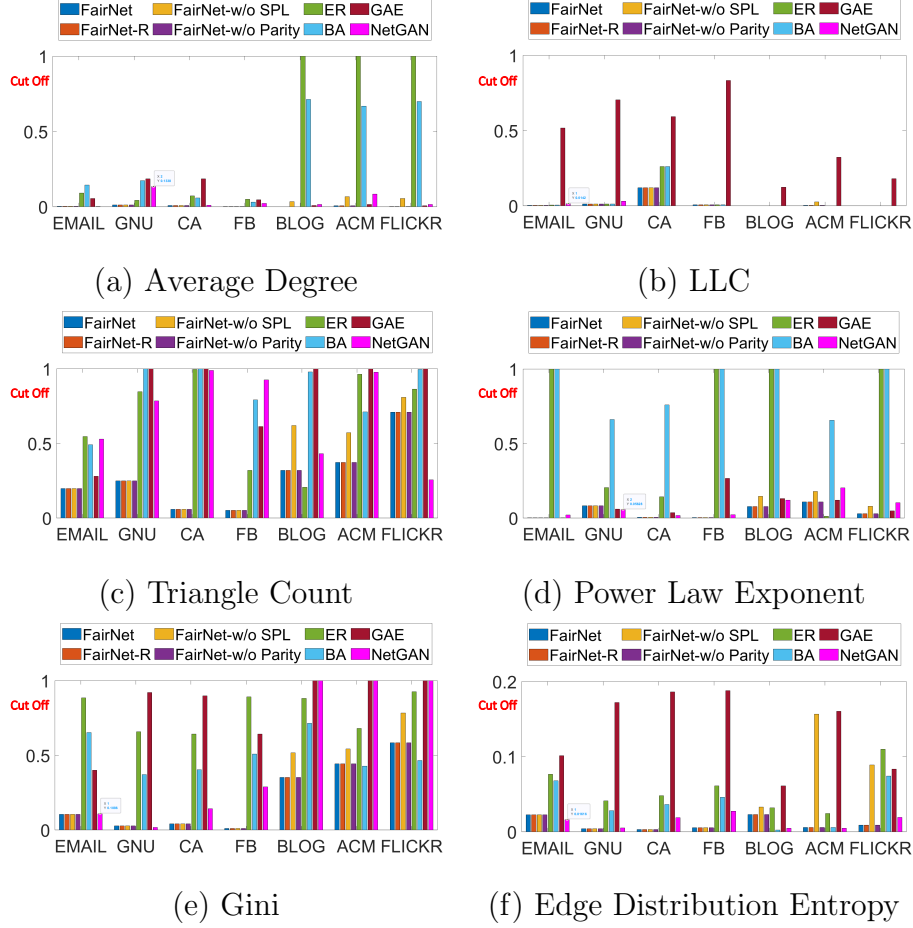


Figure 11.4: Overall discrepancy $R(\mathcal{G}, \tilde{\mathcal{G}}, f_m)$ regarding 6 metrics across 7 real networks. We cut off high values for better visibility. (Smaller metric values indicate better performance)

methods at both the scope of entire graph $\tilde{\mathcal{G}}$ and the scope of the protected group S^+ in terms of six conventional graph properties introduced in Subsection 4.2.

- **Data Augmentation:** We test the data augmentation capability of FAIRNET in the task of subgraph detection in Subsection 4.3.

11.5.1 Experiment Setup

Data Sets: We evaluate our proposed algorithm on seven real-world graphs. The statistics of these data sets are summarized in Table 11.1. Email [174] is a student-to-student communication network, where each node represents a student and an edge exists if one student sends one email to another student; FB [174] and BLOG [368] are social networks,

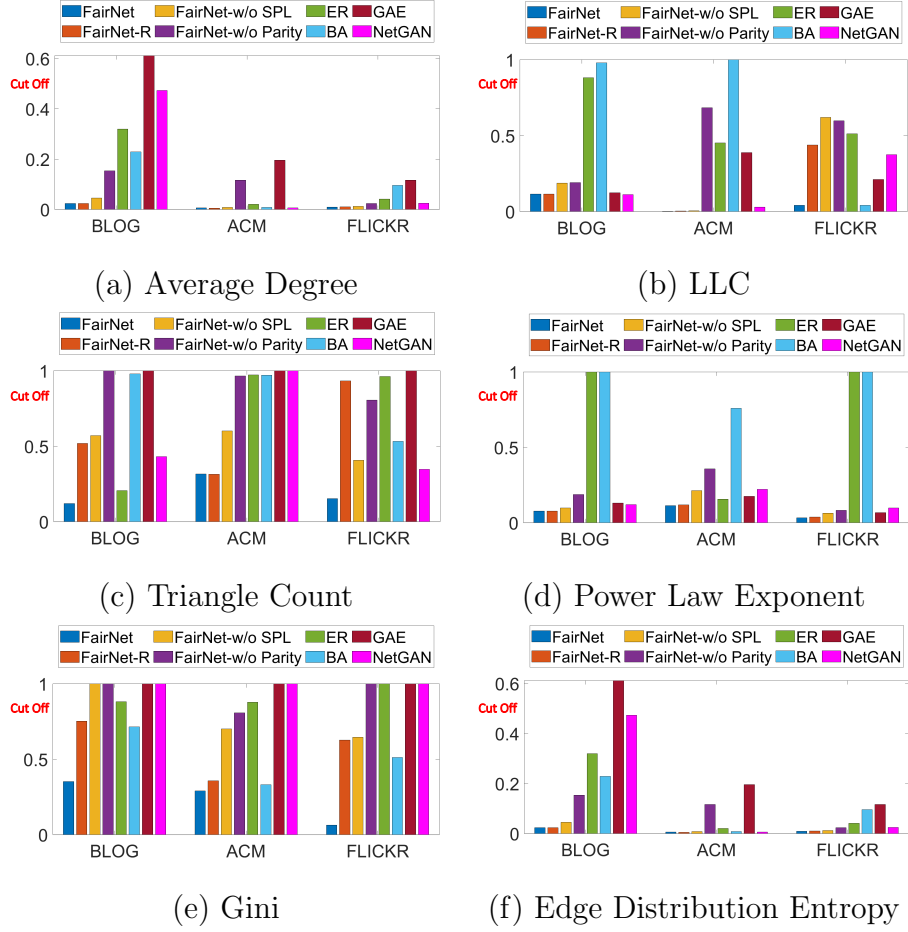


Figure 11.5: Protected group discrepancy $R^+(\mathcal{G}, \tilde{\mathcal{G}}, S^+, f_m)$ regarding 6 metrics across 3 real graphs. We cut off high values for better visibility. (Smaller metric values indicate better performance)

Category	Network	Nodes	Edges	Class	Protected Group
Communication	Email	1,005	25,571	N/A	N/A
Social Network	FB	4,039	88,234	N/A	N/A
	BLOG	5,196	360,166	6	300
Fie-Sharing	GNU	6,301	20,777	N/A	N/A
	FLICKR	7,575	501,983	9	450
Collaboration	CA	5,242	14,496	N/A	N/A
	ACM	16,484	197,560	9	600

Table 11.1: Statistics of the data sets.

where each node represents a user and each edge represents one user connected with another user; GNU [174] and FLICKR [368] are file-sharing networks, where each node represents a host and each edge indicates the connection between two hosts; CA [174] and ACM [368] are collaboration networks, where each node represents an author and each edge indicates a collaboration between two authors. Particularly, in ACM, BLOG, and FLICKR data sets, the nodes come with the class labels and the memberships of the protected group S^+ and unprotected group S^- .

Comparison Methods: We compare FAIRNET with multiple static graph generative models, including two random graph models, i.e., Erdős-Rényi (ER) model [224] and Barabási-Albert (BA) model [311], two deep graph generative models, i.e., GAE [334], NetGAN [154]. To investigate the contribution of different parts of FAIRNET, we conduct ablation study by introducing three variations of FAIRNET, including FAIRNET-R that samples random walks via uniform distribution, FAIRNET-w/o-SPL that trains without self-paced learning, and FAIRNET-w/o-Parity that trains without fairness constraint.

Evaluation: We present the results regarding the following metrics: (1) Average Degree: the average node degree; (2) LCC: the size of the largest connected component; (3) Triangle Count: the count of three mutually connected nodes; (4) Power Law Exponent: the exponent of the power law distribution of \mathcal{G} ; (5) Gini: the Gini coefficient of the degree distribution; (6) Edge Distribution Entropy: the relative edge distribution entropy of \mathcal{G} . The formulations of the six metrics are available in Table 11.2. For the sake of easy comparison, we measure the overall discrepancy $R(\mathcal{G}, \tilde{\mathcal{G}}, f_m)$ and the protected set discrepancy $R^+(\mathcal{G}, \tilde{\mathcal{G}}, S^+, f_m)$ between the original graph and the generated graph in terms of the above metrics $f_m(\cdot)$ as follows.

$$R(\mathcal{G}, \tilde{\mathcal{G}}, f_m) = \left\| \frac{f_m(\mathcal{G}) - f_m(\tilde{\mathcal{G}})}{f_m(\mathcal{G})} \right\| \quad (11.19)$$

$$R^+(\mathcal{G}, \tilde{\mathcal{G}}, S^+, f_m) = \left\| \frac{f_m(\mathcal{G}_{S^+}) - f_m(\tilde{\mathcal{G}}_{S^+})}{f_m(\mathcal{G}_{S^+})} \right\| \quad (11.20)$$

where \mathcal{G}_{S^+} and $\tilde{\mathcal{G}}_{S^+}$ denote the subgraphs that consist of the protected group vertices S^+ in \mathcal{G} and $\tilde{\mathcal{G}}$, respectively.

11.5.2 Graph Generation

Network Properties. We compare the quality of the generated graphs with 7 baseline methods at the level of both the entire graph $\tilde{\mathcal{G}}$ and the protected group S^+ in terms of 6 classic graph properties. We fit all the models on the 7 real-world graphs and report

Metric name	Computation	Description
Average Degree	$\mathbb{E}[d(v)]$	Average node degree.
LCC	$\max_{f \in F} f $	Size of the largest connected component in \mathcal{G} .
Triangle Count	$\frac{ \{(u,v,w) \{(u,v),(v,w),(u,w)\} \subseteq \mathcal{E}\} }{6}$	Number of the triangles.
Power Law Exponent	$1 + n(\sum_{u \in \mathcal{V}} \log(\frac{d(u)}{d_{min}}))^{-1}$	Exponent of the power-law distribution of \mathcal{G} .
Gini	$\frac{2 \sum_{i=1}^n i \hat{d}_i}{n \sum_{i=1}^n \hat{d}_i} - \frac{n+1}{n}$	Inequality measure for degree distribution.
Edge Distribution Entropy	$\frac{1}{\ln n} \sum_{v \in \mathcal{V}} -\frac{d(v)}{ \mathcal{E} } \ln \frac{d(v)}{ \mathcal{E} }$	Entropy of degree distribution.

Table 11.2: Graph statistics for measuring network properties.

the statistics of the generated graphs in Figure 11.4 and Figure 11.5. In Figure 11.4, we provide the comparison results in terms of the overall discrepancy $R(\mathcal{G}, \tilde{\mathcal{G}}, f_m)$ and have the following observations. (1) FAIRNET achieves comparable and even better performance than the baseline methods in most cases. (2) The traditional random graph models (i.e., ER, BA) excel at recovering the corresponding structural properties (e.g., Poisson degree distribution and heavy-tailed degree distribution) that they aim to model, whereas they fail to deal with the ones (e.g., triangle count) that they do not account for. (3) The deep graph generative models (e.g., FAIRNET, NetGAN) have better generalization to different network properties than the random graph models. (4) NetGAN performs better than FAIRNET in the data sets that provide labels and the protected group information, such as the FLICKR data set in Figure 11.4 (c). This is consistent with the objective of FAIRNET, which is not merely minimizing the overall reconstruction loss of the observed graphs. By incorporating the label information and fairness constraint to protect the protected group nodes, FAIRNET sacrifices the overall discrepancy to some extent. Then, in Figure 11.5, we demonstrate how well the protected group is preserved in the generated graphs. In particular, we compute the protected group discrepancy $R^+(\mathcal{G}, \tilde{\mathcal{G}}, S^+, f_m)$, and we observe that FAIRNET consistently outperforms all the other methods across all 7 data sets on all 6 metrics.

11.5.3 Data Augmentation

Here, we conduct case studies to evaluate the capability of FAIRNET in augmenting the performance of a prediction model for subgraph detection. We employ a logistic regression classifier as our base model, which is trained on the learned graph embedding of the original graph via node2vec [229]. For augmenting the data, we insert 5% more edges into the original graphs, which are produced by one particular graph generator. Then, we retrain the node2vec

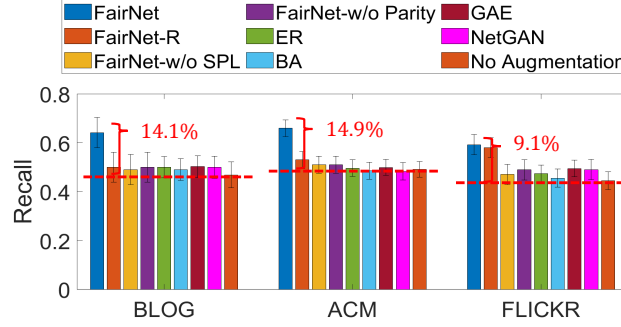


Figure 11.6: Data augmentation for subgraph detection. The red dotted line shows the performance without data augmentation. (Higher recall indicates better performance)

on the augmented graphs and use the learned logistic regression model to predict the target subgraphs. In our experiments, we split the data set into ten folds, with 90% for training and 10% for testing. In Figure 11.6, we provide the recall rate (i.e., bar height) as well as the standard deviation (i.e., error bars) in the task of subgraph detection on BLOG, ACM and FLICKR data sets. In general, we observed that: (1) FAIRNET significantly outperforms all the other graph generative models regarding performance improvement; (2) the baseline methods (e.g., GAE, NetGAN, etc.) without label information can only marginally increase the performance. For example, in the ACM data set, FAIRNET boosts the performance to 14.9%, while the best competitor FAIRNET-R and the second-best competitor GAE only achieve 0.9% and 0.8% improvement in recall over the performance of no augmentation, respectively.

11.6 SUMMARY

In this chapter, we introduce FAIRNET - a novel generative model that incorporates the label information and fairness constraints in the graph generation process. FAIRNET is developed based on a self-paced learning paradigm that globally maintains a label-informed graph generation module and a fairness learning module to extract graph context information. It is designed to gradually mitigate representation disparity by learning from the ‘easy’ concepts to the ‘hard’ ones to accurately capture the behavior of the protected groups and unprotected groups. The experimental results demonstrate the effectiveness of FAIRNET in generating high-quality graphs, alleviating the representation disparity, and enabling effective data augmentation for downstream applications.

CHAPTER 12: CONCLUSION AND FUTURE WORK

12.1 CONCLUSION

This thesis proposes a generic learning mechanism for rare category analysis on complex data. The learning mechanism boils down into three core learning modules: **(M1) Rare Category Characterization** - given scarce label information, characterizing and identifying complex rare examples (e.g., multi-view patterns, time-evolving patterns, high-order connectivity patterns, etc.) in a highly-skewed data distribution; **(M2) Rare Category Explanation** - providing the end-users a proper lens (e.g., visualization, relevant clues/interpretation) to diagnose the given data and prediction models; **(M3) Rare Category Generation** - mimicking the pattern and distribution of rare examples on complex data (e.g., temporal interaction network, attributed network) and generating synthetic rare category examples that resemble the real ones. Moreover, the learning mechanism automatically operates through a mutually beneficial synergy, which has been shown in Figure 1.3. The key philosophy of our learning mechanism lies in “all for one and one for all” - each module makes unique contributions (e.g., pseudo labels from M1, model interpretation from M2, and data augmentation from M3) to the whole learning mechanism, and receives support from the other two modules. Furthermore, to make real-world impacts, we have transferred some of our developed techniques (e.g., [15, 16]) to real systems and have also launched one of the first visual analytic systems (RCAnalyzer) for rare category analysis in the dynamic environment which allows the end-users to investigate and study rare category patterns without expertise in machine learning and data mining.

12.2 VISION AND FUTURE WORK

In the future, in response to the emerging challenges (e.g., privacy, security, and label scarcity) arising from high impact domains, I am passionate about the potential of human-like conscious thinking to improve the process of machine learning in various real-world applications. Specifically, in the short term (the first 3-5 years), I will continue working along with the theme of rare category analysis and building unified frameworks to *learn more* (knowledge) *from small* (rare observations) for complex data. Meanwhile, I will branch out to explore diverse applications related to finance, security, interpretable machine learning, algorithmic fairness, etc.; in the long term, I am interested in developing fundamental theories and principle solutions in the context of conscious machine learning and actively engaging

in the interdisciplinary research with experts from different domains.

Short Term Plan #1: Self-Supervised “Long-Tail” Category Analysis. In the past decade, deep learning has achieved remarkable success in various learning tasks (e.g., image classification, speech recognition, link prediction) through training “big models” upon “big data”. However, beyond these well-studied tasks (e.g., image classification over domestic cats and wild cats) with rich training data, the vast majority of real-world entities and patterns (e.g., identification over honest employees and malicious insiders in a large institution) are less-explored and lack of observational and annotated data, which often corresponds to the “long-tail” categories. Unlike the existing work on rare category analysis that focuses on one or a few rare categories, here we are facing a massive amount of under-represented categories from a “long-tail” distribution. Moreover, the current machine learning systems are mostly tailored to specific learning scenarios, making them fail to deliver their promises in detecting the targets of interest in the presence of distribution changes (e.g., dynamic systems).

This research is expected to answer the following questions, which are largely remained nascent in machine learning: (Q1) *How to comprehend such massive “long-tail” categories in the inherent paucity of observational and annotated data?* (Q2) *How to capture the targets of interest given a novel data distribution?* To answer these questions, I will address the following two critical aspects in developing a self-supervised mechanism to represent and infer the behaviors of “long-tail” category examples in the complex real-world scenarios. First (*self-supervised learning*), to alleviate the label scarcity, one potential solution is to design proxy objectives between the input features and the self-defined signals to capture the footprints of target signals and model the similar traits among different “long-tail” categories. In particular, for image data, the target signal could range from a particular character to a single pixel in images; for graph data, the target signals could be nodes, edges, and the associated attributes; for text data, the target signals could be masked words in a corpus. To jointly model the signals in different formats from multiple sources, I would like to design collective probabilistic inference techniques to extract and aggregate the features and the contextual information of the target of interest, in order to correctly detect the (potential) “long-tail” category examples. Second (*invariant representation learning*), in the presence of novel categories or data distribution changes, it is crucial to learn invariant representation to enable knowledge transfer from the source domain (i.e., the base domain with rich acquired knowledge) to the target domain (i.e., the novel domain with limited knowledge). To this end, I plan to develop theories and algorithms to study the capability of invariant representation in transferring knowledge and show how to predict “long-tail” categories with high accuracy.

Short Term Plan #2: Robust Rare Category Analysis. Here I propose to study

a fundamental while quite open research problem (i.e., robustness) in rare category analysis, which is attracting a surge of attention from many high-impact domains (e.g., spam detection, financial fraud, and system diagnosis). For example, in financial fraud detection, *how can we measure the entity sensitivity, algorithmic robustness, task hardness, and model generalization, given a prediction model? How can we achieve operational robustness and adversarial robustness in the presence of the external disturbance (e.g., noise, missing values, outliers, adversarial attacks)?* Despite the extensive work on adversarial machine learning, the vast majority of the previous works assume a balanced data distribution while neglecting realistic cases where the data is highly skewed, and the targets of interest are under-represented. Comparing to the conventional machine learning tools, rare category analysis models could be more sensitive and vulnerable in the presence of adversarial attacks, due to the rarity (C1), non-separability (C2), and label scarcity (C3) of rare category examples.

To fill this gap, I plan to develop fundamental theories and algorithms for robust rare category analysis, and finally enable them to function in our ultimate rare category analysis system. I believe the study of robust rare category analysis could play a *symbiotic role* with the current rare category analysis modules (Figure 1.3) - robust rare category analysis enables the flexibility to measure and compare the robustness and generalization ability of different models across different tasks, which allows downstream rare category applications (e.g., financial fraud detection) to provide more reliable and higher-quality services; the distilled knowledge (e.g., prediction, explanation, etc.) captured via other rare category analysis modules provides rich contextual information and clues to enhance operational robustness and adversarial robustness in rare category analysis. This work is expected to significantly (1) broaden the horizon of adversarial machine learning by studying data with highly skewed distribution, and (2) deepen the fundamental understanding of reliability and capability of machine learning models in rare category analysis.

Short Term Plan #3: Facilitating AI for Secure, Explainable and Ethical Financial Services. The recent advances in AI and machine learning technologies, together with the advent of the big data era, have gained an unprecedented impact on various kinds of applications (e.g., recommendation, machine transition, visual question answering) in the IT industry. Yet, many traditional industries like finance are still hesitated to fully engage with AI, due to the concerns regarding: (Q1) Security - *how to make sure the data (e.g., customers' bank accounts) and operations (e.g., financial services) upon AI systems are safe and private?* (Q2) Explainability - *how to make the AI systems interpretable as a real agent, by providing intuitive and understandable ways to reveal the underline process of model predictions (e.g., identifying malicious activities)?* (Q3) Ethics - *how to make the AI systems*

follow regulated process and meet ethical compliance (e.g., fairness, social goodness)?

During my Ph.D. program, I had several delightful and productive collaboration experiences with financial experts from industry (e.g., Thomas J. Watson Research Center, Three Bridges Capital, Early Warning Services, Ant Group, etc.), which gave me the opportunities to access real financial data and deal with real financial problems (e.g., money laundering detection, synthetic identity detection, stock forecasting, etc.). Moving forward, I will continue working on real problems and try to develop basic theories and algorithms to achieve the three aforementioned goals. To achieve data security (Q1), my investigations will expand in two folds: (1) *financial crime detection* - one potential strategy is to cast this problem as a rare category analysis problem [148], where the various kinds of financial crimes correspond to different rare categories. I will develop crime-specific algorithms to characterize their trails, detect their identities, and track their path in the dynamic settings. (2) *privacy-preserving learning* - I am interested in providing privacy-preserving solutions to allow financial institutions to collaborate with the outsiders (e.g., universities, third-party companies) in a way that could guarantee the safety and privacy of the data. In this context, I plan to extend the current deep generative models [12, 30] and design privacy-preserving data generators to produce synthetic financial data that is accessible for outsiders. To achieve explainability (Q2) and ethics (Q3) compliance, I will establish theories and metrics to quantify interpretability, fairness, and bias in the context of financial services. Moreover, I am excited to generalize my work on explainable representation learning [13] by leveraging ethical constraints (e.g., individual fairness, group fairness, counterfactual fairness) in real financial problems.

12.2.1 Long-Term Plan

Long Term Plan #1: Comprehending World Model. The current machine learning techniques are mostly designed in a closed or domain-specific environment, thus lacking understanding about the complicated and dynamic world. As a result, current AI systems are often vulnerable to the open environment with noisy data, out of order distributions, and adversarial attacks. In my future research, I am interested in advancing the frontier of AI from the lens of *world model*. Distinguished from most domain-specific machine learning models, the *world model* is domain-agnostic and capable of accommodating a variety of contributing models by connecting them with abundant pathways in order to provide trustworthy services cross multiple domains. To achieve this goal, I believe the world model should maintain a comprehensive knowledge base and a domain-invariant learning framework, where the former one enables to store and integrate a variety of real-world objects together (e.g., data, human knowledge, pieces of information, and models) and the latter one allows cross-domain knowl-

edge translation for downstream applications. On top of that, I would like to theoretically investigate the general formulation of the trade-off between intra-domain utility and inter-domain generalization of the world model. In particular, given a budget of inter-domain invariance of the world model, how can we achieve the maximum intra-domain utility for a particular task; given a lower-bound of intra-domain utility expectation, what is the optimal inter-domain invariance we can hope to achieve? Those problems are both fascinating and challenging, but I believe that being able to solve them could lead artificial intelligence systems to be more robust and capable of handling complex and dynamic environments.

Long Term Plan #2: Conscious Machine Learning. As described by the Turing Award Laureate Dr. Yoshua Bengio, the current machine learning systems mainly work in a way that “*we do intuitively, unconsciously, that we cannot explain verbally, in the case of behavior, things that are habitual*”. Compared to human intelligence, artificial intelligence learns knowledge from the outside world in a very narrow way - current AI systems can learn complex concepts between input data and target signals, but lack a conscious way to adapt, reason, and use logic to explore unknown factors from what has been learned. Inspired by cognitive neuroscience theories of consciousness, I plan to develop fundamental theories and principle solutions to couple consciousness with artificial intelligence. In particular, I am highly interested in exploring the inherent uncertainty and the causal relations that ubiquitously exist among entities in the open environment. Unlike the existing causal analysis that heavily relies on solid prior knowledge and assumptions, I would like to bridge the gap between causal models and big data by developing fundamental algorithms to tackle the potential challenges. I am also excited about exploring the potential applications of conscious machine learning in open-environment systems, such as question answering, financial trading, and medical diagnosis.

REFERENCES

- [1] B. W. Lyke, A. N. Higley, J. McLane, D. P. Schurhammer, A. D. Myers, A. J. Ross, K. Dawson, S. Chabanier, P. Martini, H. D. M. Des Bourbonx et al., “The sloan digital sky survey quasar catalog: Sixteenth data release,” *The Astrophysical Journal Supplement Series*, vol. 250, no. 1, p. 8, 2020.
- [2] D. Zhou, J. He, K. S. Candan, and H. Davulcu, “MUVIR: multi-view rare category detection,” in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, Q. Yang and M. J. Wooldridge, Eds. AAAI Press, 2015. [Online]. Available: <http://ijcai.org/Abstract/15/575> pp. 4098–4104.
- [3] S. Zhang, D. Zhou, M. Y. Yildirim, S. Alcorn, J. He, H. Davulcu, and H. Tong, “Hidden: hierarchical dense subgraph detection with application to financial fraud detection,” in *Proceedings of the 2017 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, 2017, pp. 570–578.
- [4] Dawei Zhou, J. He, H. Yang, and W. Fan, “SPARC: self-paced network representation for few-shot rare category characterization,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, Y. Guo and F. Farooq, Eds. ACM, 2018. [Online]. Available: <https://doi.org/10.1145/3219819.3219968> pp. 2807–2816.
- [5] D. Zhou, J. Luo, V. M. B. Silenzio, Y. Zhou, J. Hu, G. Currier, and H. A. Kautz, “Tackling mental health by integrating unobtrusive multimodal sensing,” in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, B. Bonet and S. Koenig, Eds. AAAI Press, 2015. [Online]. Available: <http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9546> pp. 1401–1409.
- [6] D. Zhou, J. He, Y. Cao, and J. Seo, “Bi-level rare temporal pattern detection,” in *IEEE 16th International Conference on Data Mining, ICDM 2016, December 12-15, 2016, Barcelona, Spain*, F. Bonchi, J. Domingo-Ferrer, R. Baeza-Yates, Z. Zhou, and X. Wu, Eds. IEEE Computer Society, 2016. [Online]. Available: <https://doi.org/10.1109/ICDM.2016.0083> pp. 719–728.
- [7] D. Zhou, A. Karthikeyan, K. Wang, N. Cao, and J. He, “Discovering rare categories from graph streams,” *Data Min. Knowl. Discov.*, vol. 31, no. 2, pp. 400–423, 2017. [Online]. Available: <https://doi.org/10.1007/s10618-016-0478-6>
- [8] J. Pan, D. Han, F. Guo, Dawei Zhou, N. Cao, J. He, M. Xu, and W. Chen, “Rcanalyzer: visual analytics of rare categories in dynamic networks,” *Frontiers Inf. Technol. Electron. Eng.*, vol. 21, no. 4, pp. 491–506, 2020. [Online]. Available: <https://doi.org/10.1631/FITEE.1900310>

- [9] D. Zhou, L. Zheng, J. Han, and J. He, “A data-driven graph generative model for temporal interaction networks,” in *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, R. Gupta, Y. Liu, J. Tang, and B. A. Prakash, Eds. ACM, 2020. [Online]. Available: <https://doi.org/10.1145/3394486.3403082> pp. 401–411.
- [10] D. Fu, D. Zhou, and J. He, “Local motif clustering on time-evolving graphs,” in *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, R. Gupta, Y. Liu, J. Tang, and B. A. Prakash, Eds. ACM, 2020. [Online]. Available: <https://doi.org/10.1145/3394486.3403081> pp. 390–400.
- [11] Dawei Zhou*, Z. Liu*, Y. Zhu, J. Gu, and J. H. . equal contribution), “Towards fine-grained temporal network representation via time-reinforced random walk,” in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*. AAAI Press, 2020. [Online]. Available: <https://aaai.org/ojs/index.php/AAAI/article/view/5936> pp. 4973–4980.
- [12] Dawei Zhou, L. Zheng, J. Xu, and J. He, “Misc-gan: A multi-scale generative model for graphs,” *Frontiers Big Data*, vol. 2, p. 3, 2019. [Online]. Available: <https://doi.org/10.3389/fdata.2019.00003>
- [13] Dawei Zhou*, Z. Liu*, and J. H. . equal contribution), “Towards explainable representation of time-evolving graphs via spatial-temporal graph attention networks,” in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, W. Zhu, D. Tao, X. Cheng, P. Cui, E. A. Rundensteiner, D. Carmel, Q. He, and J. X. Yu, Eds. ACM, 2019. [Online]. Available: <https://doi.org/10.1145/3357384.3358155> pp. 2137–2140.
- [14] P. Yang, H. Yang, H. Fu, D. Zhou, J. Ye, T. Lappas, and J. He, “Jointly modeling label and feature heterogeneity in medical informatics,” *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 10, no. 4, p. 39, 2016.
- [15] Dawei Zhou, L. Zheng, Y. Zhu, J. Li, and J. He, “Domain adaptive multi-modality neural attention network for financial forecasting,” in *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, Y. Huang, I. King, T. Liu, and M. van Steen, Eds. ACM / IW3C2, 2020. [Online]. Available: <https://doi.org/10.1145/3366423.3380288> pp. 2230–2240.
- [16] Dawei Zhou, S. Zhang, M. Y. Yildirim, S. Alcorn, H. Tong, H. Davulcu, and J. He, “A local algorithm for structure-preserving graph cut,” in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017. [Online]. Available: <https://doi.org/10.1145/3097983>

- [17] Dawei Zhou, J. He, H. Davulcu, and R. Maciejewski, “Motif-preserving dynamic local graph cut,” in *IEEE International Conference on Big Data, Big Data 2018, Seattle, WA, USA, December 10-13, 2018*, N. Abe, H. Liu, C. Pu, X. Hu, N. K. Ahmed, M. Qiao, Y. Song, D. Kossmann, B. Liu, K. Lee, J. Tang, J. He, and J. S. Saltz, Eds. IEEE, 2018. [Online]. Available: <https://doi.org/10.1109/BigData.2018.8622263> pp. 1156–1161.
- [18] Dawei Zhou, S. Zhang, M. Y. Yildirim, S. Alcorn, H. Tong, H. Davulcu, and J. He, “High-order structure exploration on massive graphs: A local graph clustering perspective,” *ACM Trans. Knowl. Discov. Data*, 2020.
- [19] D. M. Hawkins, *Identification of Outliers*, ser. Monographs on Applied Probability and Statistics. Springer, 1980. [Online]. Available: <https://doi.org/10.1007/978-94-015-3994-4>
- [20] A. Zimek, E. Schubert, and H.-P. Kriegel, “A survey on unsupervised outlier detection in high-dimensional numerical data,” *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 5, no. 5, pp. 363–387, 2012.
- [21] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection for discrete sequences: A survey,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 5, pp. 823–839, 2010.
- [22] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, “Outlier detection for temporal data: A survey,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 9, pp. 2250–2267, 2013.
- [23] L. Akoglu, H. Tong, and D. Koutra, “Graph based anomaly detection and description: a survey,” *Data mining and knowledge discovery*, vol. 29, no. 3, pp. 626–688, 2015.
- [24] C. Phua, V. Lee, K. Smith, and R. Gayler, “A comprehensive survey of data mining-based fraud detection research,” *arXiv preprint arXiv:1009.6119*, 2010.
- [25] M. E. Edge and P. R. F. Sampaio, “A survey of signature based methods for financial fraud detection,” *computers & security*, vol. 28, no. 6, pp. 381–394, 2009.
- [26] D. Pelleg and A. W. Moore, “Active learning for anomaly and rare-category detection,” in *Advances in neural information processing systems*, 2005, pp. 1073–1080.
- [27] Dawei Zhou, J. He, K. S. Candan, and H. Davulcu, “MUVIR: multi-view rare category detection,” in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, Q. Yang and M. J. Wooldridge, Eds. AAAI Press, 2015. [Online]. Available: <http://ijcai.org/Abstract/15/575> pp. 4098–4104.

- [28] S. Zhang, Dawei Zhou, M. Y. Yildirim, S. Alcorn, J. He, H. Davulcu, and H. Tong, “Hidden: Hierarchical dense subgraph detection with application to financial fraud detection,” in *Proceedings of the 2017 SIAM International Conference on Data Mining, Houston, Texas, USA, April 27-29, 2017*, N. V. Chawla and W. Wang, Eds. SIAM, 2017. [Online]. Available: <https://doi.org/10.1137/1.9781611974973.64> pp. 570–578.
- [29] D. Zhou, K. Wang, N. Cao, and J. He, “Rare category detection on time-evolving graphs,” in *2015 IEEE International Conference on Data Mining, ICDM 2015, Atlantic City, NJ, USA, November 14-17, 2015*, C. C. Aggarwal, Z. Zhou, A. Tuzhilin, H. Xiong, and X. Wu, Eds. IEEE Computer Society, 2015. [Online]. Available: <https://doi.org/10.1109/ICDM.2015.120> pp. 1135–1140.
- [30] Dawei Zhou, L. Zheng, and J. He, “A data-driven graph generative model for temporal interaction networks,” in *KDD ’20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, R. Gupta, Y. Liu, J. Tang, and B. A. Prakash, Eds. ACM, 2020. [Online]. Available: <https://dl.acm.org/doi/10.1145/3394486.3403082> pp. 401–411.
- [31] J. He and J. G. Carbonell, “Nearest-neighbor-based active learning for rare category detection,” in *Advances in neural information processing systems*, 2008, pp. 633–640.
- [32] J. He and J. Carbonell, “Prior-free rare category detection,” in *Proceedings of the 2009 SIAM International Conference on Data Mining*. SIAM, 2009, pp. 155–163.
- [33] J. He, Y. Liu, and R. Lawrence, “Graph-based rare category detection,” in *Data Mining, 2008. ICDM’08. Eighth IEEE International Conference on*. IEEE, 2008, pp. 833–838.
- [34] J. He and J. Carbonell, “Coselection of features and instances for unsupervised rare category analysis,” *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 3, no. 6, pp. 417–430, 2010.
- [35] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf, “Ranking on data manifolds,” in *Advances in neural information processing systems*, 2004, pp. 169–176.
- [36] J. Sun, H. Qu, D. Chakrabarti, and C. Faloutsos, “Neighborhood formation and anomaly detection in bipartite graphs,” in *Data Mining, Fifth IEEE International Conference on*. IEEE, 2005, pp. 8–pp.
- [37] H. Tong and C.-Y. Lin, “Non-negative residual matrix factorization with application to graph anomaly detection,” in *Proceedings of the 2011 SIAM International Conference on Data Mining*. SIAM, 2011, pp. 143–153.
- [38] E. Manzoor, S. M. Milajerdi, and L. Akoglu, “Fast memory-efficient anomaly detection in streaming heterogeneous graphs,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 1035–1044.

- [39] S. Deng, H. Rangwala, and Y. Ning, “Learning dynamic context graphs for predicting social events,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2019, pp. 1007–1016.
- [40] J.-X. Zhong, N. Li, W. Kong, S. Liu, T. H. Li, and G. Li, “Graph convolutional label noise cleaner: Train a plug-and-play action classifier for anomaly detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1237–1246.
- [41] H. Jagadish, N. Koudas, and S. Muthukrishnan, “Mining deviants in a time series database.” in *VLDB*, vol. 99, 1999, pp. 7–10.
- [42] X. Li and J. Han, “Mining approximate top-k subspace anomalies in multi-dimensional time-series data,” in *Proceedings of the 33rd international conference on Very large data bases*. VLDB Endowment, 2007, pp. 447–458.
- [43] S. Muthukrishnan, R. Shah, and J. S. Vitter, “Mining deviants in time series data streams,” in *Scientific and Statistical Database Management, 2004. Proceedings. 16th International Conference on*. IEEE, 2004, pp. 41–50.
- [44] H. Moradi Koupaie, S. Ibrahim, and J. Hosseinkhani, “Outlier detection in stream data by clustering method,” *International Journal of Advanced Computer Science and Information Technology (IJACSIT) Vol*, vol. 2, pp. 25–34, 2014.
- [45] N. Begum and E. Keogh, “Rare time series motif discovery from unbounded streams,” *Proceedings of the VLDB Endowment*, vol. 8, no. 2, pp. 149–160, 2014.
- [46] L. Cao, D. Yang, Q. Wang, Y. Yu, J. Wang, and E. A. Rundensteiner, “Scalable distance-based outlier detection over high-volume data streams,” in *2014 IEEE 30th International Conference on Data Engineering*. IEEE, 2014, pp. 76–87.
- [47] F. Schnitzler, T. Liebig, S. Mannor, G. Souto, S. Bothe, and H. Stange, “Heterogeneous stream processing for disaster detection and alarming,” in *IEEE International Conference on Big Data*. IEEE Press, 2014, pp. 914–923.
- [48] C.-C. M. Yeh, Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. A. Dau, D. F. Silva, A. Mueen, and E. Keogh, “Matrix profile i: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets,” in *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE, 2016, pp. 1317–1322.
- [49] A. Nairac, N. Townsend, R. Carr, S. King, P. Cowley, and L. Tarassenko, “A system for the analysis of jet engine vibration data,” *Integrated Computer-Aided Engineering*, vol. 6, no. 1, pp. 53–66, 1999.
- [50] K. Sequeira and M. Zaki, “Admit: anomaly-based data mining for intrusions,” in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2002, pp. 386–395.

- [51] U. Rebbapragada, P. Protopapas, C. E. Brodley, and C. Alcock, “Finding anomalous periodic time series,” *Machine learning*, vol. 74, no. 3, pp. 281–313, 2009.
- [52] S. Guha, N. Mishra, G. Roy, and O. Schrijvers, “Robust random cut forest based anomaly detection on streams,” in *International conference on machine learning*, 2016, pp. 2712–2721.
- [53] A. Tuor, S. Kaplan, B. Hutchinson, N. Nichols, and S. Robinson, “Deep learning for unsupervised insider threat detection in structured cybersecurity data streams,” in *Workshops at the Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
- [54] D. Li, D. Chen, L. Shi, B. Jin, J. Goh, and S.-K. Ng, “Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks,” *arXiv preprint arXiv:1901.04997*, 2019.
- [55] K. Sricharan and K. Das, “Localizing anomalous changes in time-evolving graphs,” in *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. ACM, 2014, pp. 1347–1358.
- [56] S. Ranshous, S. Shen, D. Koutra, S. Harenberg, C. Faloutsos, and N. F. Samatova, “Anomaly detection in dynamic networks: a survey,” *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 7, no. 3, pp. 223–247, 2015.
- [57] J. D. Wilson, N. T. Stevens, and W. H. Woodall, “Modeling and estimating change in temporal networks via a dynamic degree corrected stochastic block model,” *arXiv preprint arXiv:1605.04049*, 2016.
- [58] Y. Yasami and F. Safaei, “A statistical infinite feature cascade-based approach to anomaly detection for dynamic social networks,” *Computer Communications*, vol. 100, pp. 52–64, 2017.
- [59] A. Hollocou, J. Maudet, T. Bonald, and M. Lelarge, “A streaming algorithm for graph clustering,” *CoRR*, vol. abs/1712.04337, 2017. [Online]. Available: <http://arxiv.org/abs/1712.04337>
- [60] K. H. Lee, L. Xue, and D. R. Hunter, “Model-based clustering of time-evolving networks through temporal exponential-family random graph models,” *arXiv preprint arXiv:1712.07325*, 2017.
- [61] M. Macha and L. Akoglu, “Explaining anomalies in groups with characterizing subspace rules,” *Data Mining and Knowledge Discovery*, vol. 32, no. 5, pp. 1444–1480, 2018.
- [62] P. Agarwal, R. Verma, A. Agarwal, and T. Chakraborty, “Dyperm: Maximizing permanence for dynamic community detection,” in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2018, pp. 437–449.

- [63] A. P. Appel, R. L. Cunha, C. C. Aggarwal, and M. M. Terakado, “Temporally evolving community detection and prediction in content-centric networks,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2018, pp. 3–18.
- [64] M. Yoon, B. Hooi, K. Shin, and C. Faloutsos, “Fast and accurate anomaly detection in dynamic graphs with a two-pronged approach,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2019, pp. 647–657.
- [65] B. Yu, H. Yin, and Z. Zhu, “Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting,” in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, J. Lang, Ed. ijcai.org, 2018. [Online]. Available: <https://doi.org/10.24963/ijcai.2018/505> pp. 3634–3640.
- [66] G. H. Nguyen, J. B. Lee, R. A. Rossi, N. K. Ahmed, E. Koh, and S. Kim, “Continuous-time dynamic network embeddings,” in *Companion of the The Web Conference 2018 on The Web Conference 2018, WWW 2018, Lyon , France, April 23-27, 2018*, P. Champin, F. Gandon, M. Lalmas, and P. G. Ipeirotis, Eds. ACM, 2018. [Online]. Available: <https://doi.org/10.1145/3184558.3191526> pp. 969–976.
- [67] J. Sun, C. Faloutsos, C. Faloutsos, S. Papadimitriou, and P. S. Yu, “Graphscope: parameter-free mining of large time-evolving graphs,” in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2007, pp. 687–696.
- [68] M. Araujo, S. Papadimitriou, S. Günnemann, C. Faloutsos, P. Basu, A. Swami, E. E. Papalexakis, and D. Koutra, “Com2: fast automatic discovery of temporal (‘comet’) communities,” in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2014, pp. 271–283.
- [69] M. Amjadi and T. Tulabandhula, “Block-structure based time-series models for graph sequences,” *arXiv preprint arXiv:1804.08796*, 2018.
- [70] A. Blum and T. Mitchell, “Combining labeled and unlabeled data with co-training,” in *Proceedings of the eleventh annual conference on Computational learning theory*. ACM, 1998, pp. 92–100.
- [71] S. P. Abney, “Bootstrapping,” in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. ACL, 2002. [Online]. Available: <https://aclanthology.org/volumes/P02-1/>
- [72] M.-F. Balcan, A. Blum, and K. Yang, “Co-training and expansion: Towards bridging theory and practice,” in *Advances in neural information processing systems*, 2004, pp. 89–96.

- [73] V. Sindhwani and D. S. Rosenberg, “An rkhs for multi-view learning and manifold co-regularization,” in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 976–983.
- [74] L. Song, A. Anandkumar, B. Dai, and B. Xie, “Nonparametric estimation of multi-view latent variable models,” *arXiv preprint arXiv:1311.3287*, 2013.
- [75] S. Günnemann, I. Färber, M. Rüdiger, and T. Seidl, “Smvc: semi-supervised multi-view clustering in subspace projections,” in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 253–262.
- [76] W. Eberle, J. Graves, and L. Holder, “Insider threat detection using a graph-based approach,” *Journal of Applied Security Research*, vol. 6, no. 1, pp. 32–81, 2010.
- [77] S. Dasgupta and D. Hsu, “Hierarchical sampling for active learning,” in *International Conference on Machine Learning*. ACM, 2008, pp. 208–215.
- [78] J. He, “Rare category analysis,” Ph.D. dissertation, Carnegie Mellon University, School of Computer Science, Machine Learning . . . , 2010.
- [79] Z. Liu, K. Chiew, Q. He, H. Huang, and B. Huang, “Prior-free rare category detection: More effective and efficient solutions,” *Expert Systems with Applications*, vol. 41, no. 17, pp. 7691–7706, 2014.
- [80] K. Yamanishi, J.-I. Takeuchi, G. Williams, and P. Milne, “On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms,” *Data Mining and Knowledge Discovery*, vol. 8, no. 3, pp. 275–300, 2004.
- [81] C. C. Aggarwal and S. Y. Philip, “On clustering massive text and categorical data streams,” *Knowledge and Information Systems*, vol. 24, no. 2, pp. 171–196, 2010.
- [82] D. J. Hill, B. S. Minsker, and E. Amir, “Real-time bayesian anomaly detection for environmental sensor data,” in *Congress-International Association for Hydraulic Research*, vol. 32, no. 2. Citeseer, 2007, p. 503.
- [83] L. M. Bettencourt, A. A. Hagberg, and L. B. Larkey, “Separating the wheat from the chaff: practical anomaly detection schemes in ecological applications of distributed sensor networks,” in *Distributed Computing in Sensor Systems*. Springer, 2007, pp. 223–239.
- [84] C. Franke and M. Gertz, “Detection and exploration of outlier regions in sensor data streams,” in *IEEE International Conference on Data Mining Workshops*. IEEE, 2008, pp. 375–384.
- [85] U. Kang, M. McGlohon, L. Akoglu, and C. Faloutsos, “Patterns on the connected components of terabyte-scale graphs,” in *IEEE International Conference on Data Mining*. IEEE, 2010, pp. 875–880.

- [86] K. Henderson, T. Eliassi-Rad, C. Faloutsos, L. Akoglu, L. Li, K. Maruhashi, B. A. Prakash, and H. Tong, “Metric forensics: a multi-level approach for mining volatile graphs,” in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2010, pp. 163–172.
- [87] L. Akoglu, M. McGlohon, and C. Faloutsos, “Oddball: Spotting anomalies in weighted graphs,” in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2010, pp. 410–421.
- [88] U. Kang, C. E. Tsourakakis, A. P. Appel, C. Faloutsos, and J. Leskovec, “Hadi: Mining radii of large graphs,” *ACM Transactions on Knowledge Discovery from Data*, vol. 5, no. 2, p. 8, 2011.
- [89] M. Gupte and T. Eliassi-Rad, “Measuring tie strength in implicit social networks,” in *Annual ACM Web Science Conference*. ACM, 2012, pp. 109–118.
- [90] E. Muller, P. I. Sánchez, Y. Mulle, and K. Bohm, “Ranking outlier nodes in subspaces of attributed graphs,” in *Data Engineering Workshops (ICDEW), 2013 IEEE 29th International Conference on*. IEEE, 2013, pp. 216–222.
- [91] J. Gao, F. Liang, W. Fan, C. Wang, Y. Sun, and J. Han, “On community outliers and their efficient detection in information networks,” in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2010, pp. 813–822.
- [92] M. Davis, W. Liu, P. Miller, and G. Redpath, “Detecting anomalies in graphs with numeric labels,” in *ACM International Conference on Information and Knowledge Management*. ACM, 2011, pp. 1197–1202.
- [93] J. Leskovec, J. Kleinberg, and C. Faloutsos, “Graphs over time: densification laws, shrinking diameters and possible explanations,” in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2005, pp. 177–187.
- [94] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan, “Group formation in large social networks: membership, growth, and evolution,” in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2006, pp. 44–54.
- [95] R. Kumar, M. Mahdian, and M. McGlohon, “Dynamics of conversations,” in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2010, pp. 553–562.
- [96] M. Berlingerio, D. Koutra, T. Eliassi-Rad, and C. Faloutsos, “Netsimile: a scalable approach to size-independent network similarity,” in *arXiv preprint arXiv:1209.2684*, 2012.
- [97] D. Koutra, T.-Y. Ke, U. Kang, D. H. P. Chau, H.-K. K. Pao, and C. Faloutsos, “Unifying guilt-by-association approaches: Theorems and fast algorithms,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2011, pp. 245–260.

- [98] H. Tong, S. Papadimitriou, P. S. Yu, and C. Faloutsos, “Proximity tracking on time-evolving bipartite graphs,” in *Proceedings of the SIAM International Conference on Data Mining*. SIAM, 2008. [Online]. Available: <https://doi.org/10.1137/1.9781611972788>
- [99] D. Koutra, E. E. Papalexakis, and C. Faloutsos, “Tensorsplat: Spotting latent anomalies in time,” in *Panhellenic Conference on Informatics*. IEEE, 2012, pp. 144–149.
- [100] W. Fan, X. Wang, and Y. Wu, “Incremental graph pattern matching,” *ACM Transactions on Database Systems*, vol. 38, no. 3, p. 18, 2013.
- [101] L. Akoglu, R. Khandekar, V. Kumar, S. Parthasarathy, D. Rajan, and K.-L. Wu, “Fast nearest neighbor search on large time-evolving graphs,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2014, pp. 17–33.
- [102] J. Sherman and W. J. Morrison, “Adjustment of an inverse matrix corresponding to a change in one element of a given matrix,” *The Annals of Mathematical Statistics*, vol. 21, no. 1, pp. 124–127, 1950.
- [103] J. J. Oresko, Z. Jin, J. Cheng, S. Huang, Y. Sun, H. Duschl, and A. C. Cheng, “A wearable smartphone-based platform for real-time cardiovascular disease detection via electrocardiogram processing,” *Information Technology in Biomedicine, IEEE Transactions on*, 2010.
- [104] D. L. Holloway and A. Anderson, “Online payment system for merchants,” June 16 2006, uS Patent App. 11/922,346.
- [105] S. Pan, Q. Ye, S. Liu, and D. Zhou, “Joint resource allocation for wlan&wcdma integrated networks based on spectral bandwidth mapping,” *Journal of Electronics (China)*, vol. 28, no. 4-6, pp. 474–482, 2011.
- [106] L. Chen, J. Warner, P. L. Yung, D. Zhou, W. Heinzelman, L. Demirkol, U. Muncuk, K. Chowdhury, and S. Basagni, “Reach²-mote: A range extending passive wake-up wireless sensor node,” *ACM Transactions on Sensor Networks, Vol. V, No. N, Article A*, 2015.
- [107] J. Li, X. Hu, J. Tang, and H. Liu, “Unsupervised streaming feature selection in social media,” in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 2015, pp. 1041–1050.
- [108] X. Zhang, G. Xue, R. Yu, D. Yang, and J. Tang, “Truthful incentive mechanisms for crowdsourcing,” in *2015 IEEE Conference on Computer Communications (INFOCOM)*, 2015, pp. 2830–2838.
- [109] Y. Zhou and J. He, “Crowdsourcing via tensor augmentation and completion,” in *IJCAI*, 2016.

- [110] K. Shu, P. Luo, W. Li, P. Yin, and L. Tang, “Deal or deceit: detecting cheating in distribution channels,” in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, 2014, pp. 1419–1428.
- [111] Y. Wang, Q. Zhang, and B. Li, “Efficient unsupervised abnormal crowd activity detection based on a spatiotemporal saliency detector,” in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016, pp. 1–9.
- [112] X. Pan, J. Tan, S. Kavulya, R. Gandhi, and P. Narasimhan, “Ganesha: blackbox diagnosis of mapreduce systems,” *ACM SIGMETRICS*, 2010.
- [113] V. Chandola, V. Mithal, and V. Kumar, “Comparative evaluation of anomaly detection techniques for sequence data,” in *ICDM*, 2008.
- [114] F. A. González and D. Dasgupta, “Anomaly detection using real-valued negative selection,” *Genetic Programming and Evolvable Machines*, 2003.
- [115] D. J. Hill and B. S. Minsker, “Anomaly detection in streaming environmental sensor data: A data-driven modeling approach,” *Environmental Modelling and Software*, 2010.
- [116] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, “Solving the multiple instance problem with axis-parallel rectangles,” *Artificial intelligence*, 1997.
- [117] O. Maron and T. Lozano-Pérez, “A framework for multiple-instance learning,” *Advances in neural information processing systems*, 1998.
- [118] Q. Zhang and S. A. Goldman, “Em-dd: An improved multiple-instance learning technique,” in *Advances in neural information processing systems*, 2001.
- [119] S. Andrews, I. Tsochantaridis, and T. Hofmann, “Support vector machines for multiple-instance learning,” in *Advances in neural information processing systems*, 2002, pp. 561–568.
- [120] Z.-H. Zhou, Y.-Y. Sun, and Y.-F. Li, “Multi-instance learning by treating instances as non-iid samples,” in *ICML*, 2009.
- [121] L. E. Baum and T. Petrie, “Statistical inference for probabilistic functions of finite state markov chains,” *The annals of mathematical statistics*, 1966.
- [122] W. Khreich, E. Granger, A. Miri, and R. Sabourin, “A survey of techniques for incremental learning of hmm parameters,” *Information Sciences*, 2012.
- [123] Z.-Q. Luo and P. Tseng, “On the convergence of the coordinate descent method for convex differentiable minimization,” *Journal of Optimization Theory and Applications*, 1992.
- [124] P. Tseng, “Convergence of a block coordinate descent method for nondifferentiable minimization,” *Journal of optimization theory and applications*, 2001.

- [125] Y. Xu and W. Yin, “A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion,” *SIAM Journal on imaging sciences*, 2013.
- [126] L. R. Rabiner, “A tutorial on hidden markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, 1989.
- [127] E. L. Russell, L. H. Chiang, and R. D. Braatz, *Data-driven methods for fault detection and diagnosis in chemical processes*. Springer Science & Business Media, 2012.
- [128] Y. Chen, B. Hu, E. Keogh, and G. E. Batista, “Dtw-d: time series semi-supervised learning from a single example,” in *SIGKDD*, 2013.
- [129] F. J. Ordóñez, P. de Toledo, and A. Sanchis, “Activity recognition using hybrid generative/discriminative models on home environments using binary sensors,” *Sensors*, 2013.
- [130] S. E. Schaeffer, “Graph clustering,” *Comput. Sci. Rev.*, vol. 1, no. 1, pp. 27–64, 2007. [Online]. Available: <https://doi.org/10.1016/j.cosrev.2007.05.001>
- [131] Z. Bu, H. Li, C. Zhang, J. Cao, A. Li, and Y. Shi, “Graph k-means based on leader identification, dynamic game, and opinion dynamics,” *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 7, pp. 1348–1361, 2020. [Online]. Available: <https://doi.org/10.1109/TKDE.2019.2903712>
- [132] H.-J. Li and L. Wang, “Multi-scale asynchronous belief percolation model on multiplex networks,” *New Journal of Physics*, vol. 21, no. 1, p. 015005, 2019.
- [133] C. Klymko, D. F. Gleich, and T. G. Kolda, “Using triangles to improve community detection in directed networks,” *CoRR*, vol. abs/1404.5874, 2014. [Online]. Available: <http://arxiv.org/abs/1404.5874>
- [134] K.-K. R. Choo, “Money laundering risks of prepaid stored value cards.” *Trends and Issues in Crime and Criminal Justice*, no. 363, pp. 1–6, 2008.
- [135] C. J. Hoofnagle, “Identity theft: Making the known unknowns known,” *Harv. JL & Tech.*, vol. 21, p. 97, 2007.
- [136] A. R. Benson, D. F. Gleich, and J. Leskovec, “Tensor spectral clustering for partitioning higher-order network structures,” in *Proceedings of the 2015 SIAM International Conference on Data Mining, Vancouver, BC, Canada, April 30 - May 2, 2015*, S. Venkatasubramanian and J. Ye, Eds. SIAM, 2015. [Online]. Available: <https://doi.org/10.1137/1.9781611974010.14> pp. 118–126.

- [137] T. Wu, A. R. Benson, and D. F. Gleich, “General tensor spectral co-clustering for higher-order data,” in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, Eds., 2016. [Online]. Available: <http://papers.nips.cc/paper/6376-general-tensor-spectral-co-clustering-for-higher-order-data> pp. 2559–2567.
- [138] N. K. Ahmed, J. Neville, R. A. Rossi, and N. G. Duffield, “Efficient graphlet counting for large networks,” in *2015 IEEE International Conference on Data Mining, ICDM 2015, Atlantic City, NJ, USA, November 14-17, 2015*, C. C. Aggarwal, Z. Zhou, A. Tuzhilin, H. Xiong, and X. Wu, Eds. IEEE Computer Society, 2015. [Online]. Available: <https://doi.org/10.1109/ICDM.2015.141> pp. 1–10.
- [139] M. Bressan, F. Chierichetti, R. Kumar, S. Leucci, and A. Panconesi, “Counting graphlets: Space vs time,” in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 2017, Cambridge, United Kingdom, February 6-10, 2017*, M. de Rijke, M. Shokouhi, A. Tomkins, and M. Zhang, Eds. ACM, 2017. [Online]. Available: <https://doi.org/10.1145/3018661.3018732> pp. 557–566.
- [140] W. Li and M. K. Ng, “On the limiting probability distribution of a transition probability tensor,” *Linear and Multilinear Algebra*, vol. 62, no. 3, pp. 362–385, 2014.
- [141] C. E. Tsourakakis, J. Pachocki, and M. Mitzenmacher, “Scalable motif-aware graph clustering,” in *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, R. Barrett, R. Cummings, E. Agichtein, and E. Gabrilovich, Eds. ACM, 2017. [Online]. Available: <https://doi.org/10.1145/3038912.3052653> pp. 1451–1460.
- [142] D. A. Spielman and S. Teng, “A local clustering algorithm for massive graphs and its application to nearly linear time graph partitioning,” *SIAM J. Comput.*, vol. 42, no. 1, pp. 1–26, 2013. [Online]. Available: <https://doi.org/10.1137/080744888>
- [143] R. Andersen, F. R. K. Chung, and K. J. Lang, “Local graph partitioning using pagerank vectors,” in *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings*. IEEE Computer Society, 2006. [Online]. Available: <https://doi.org/10.1109/FOCS.2006.44> pp. 475–486.
- [144] R. Andersen, F. R. K. Chung, and K. J. Lang, “Local partitioning for directed graphs using pagerank,” in *Algorithms and Models for the Web-Graph, 5th International Workshop, WAW 2007, San Diego, CA, USA, December 11-12, 2007, Proceedings*, ser. Lecture Notes in Computer Science, A. Bonato and F. R. K. Chung, Eds., vol. 4863. Springer, 2007. [Online]. Available: https://doi.org/10.1007/978-3-540-77004-6_13 pp. 166–178.

- [145] D. F. Gleich, L. Lim, and Y. Yu, “Multilinear pagerank,” *SIAM J. Matrix Anal. Appl.*, vol. 36, no. 4, pp. 1507–1541, 2015. [Online]. Available: <https://doi.org/10.1137/140985160>
- [146] J. Li, H. Dani, X. Hu, and H. Liu, “Radar: Residual analysis for anomaly detection in attributed networks,” in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, C. Sierra, Ed. ijcai.org, 2017. [Online]. Available: <https://doi.org/10.24963/ijcai.2017/299> pp. 2152–2158.
- [147] C. Chen, J. He, N. Bliss, and H. Tong, “On the connectivity of multi-layered networks: Models, measures and optimal control,” in *2015 IEEE International Conference on Data Mining, ICDM 2015, Atlantic City, NJ, USA, November 14-17, 2015*, C. C. Aggarwal, Z. Zhou, A. Tuzhilin, H. Xiong, and X. Wu, Eds. IEEE Computer Society, 2015. [Online]. Available: <https://doi.org/10.1109/ICDM.2015.104> pp. 715–720.
- [148] D. Zhou and J. He, “Gold panning from the mess: Rare category exploration, exposition, representation, and interpretation,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, A. Teredesai, V. Kumar, Y. Li, R. Rosales, E. Terzi, and G. Karypis, Eds. ACM, 2019. [Online]. Available: <https://doi.org/10.1145/3292500.3332268> pp. 3213–3214.
- [149] J. Cao, Z. Bu, Y. Wang, H. Yang, J. Jiang, and H.-J. Li, “Detecting prosumer-community groups in smart grids from the multiagent perspective,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 8, pp. 1652–1664, 2019.
- [150] H.-J. Li, Z. Bu, Z. Wang, J. Cao, and Y. Shi, “Enhance the performance of network computation by a tunable weighting strategy,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 3, pp. 214–223, 2018.
- [151] Z. Bu, H.-J. Li, J. Cao, Z. Wang, and G. Gao, “Dynamic cluster formation game for attributed graph clustering,” *IEEE transactions on cybernetics*, vol. 49, no. 1, pp. 328–341, 2017.
- [152] Z. Bu, Z. Wu, J. Cao, and Y. Jiang, “Local community mining on distributed and dynamic networks from a multiagent perspective,” *IEEE Transactions on cybernetics*, vol. 46, no. 4, pp. 986–999, 2015.
- [153] S. Zhang, D. Zhou, M. Y. Yildirim, S. Alcorn, J. He, H. Davulcu, and H. Tong, “Hidden: Hierarchical dense subgraph detection with application to financial fraud detection,” in *Proceedings of the 2017 SIAM International Conference on Data Mining, Houston, Texas, USA, April 27-29, 2017*, N. V. Chawla and W. Wang, Eds. SIAM, 2017. [Online]. Available: <https://doi.org/10.1137/1.9781611974973.64> pp. 570–578.

- [154] A. Bojchevski, O. Shchur, D. Zügner, and S. Günnemann, “Netgan: Generating graphs via random walks,” in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, ser. Proceedings of Machine Learning Research, J. G. Dy and A. Krause, Eds., vol. 80. PMLR, 2018. [Online]. Available: <http://proceedings.mlr.press/v80/bojchevski18a.html> pp. 609–618.
- [155] Y. Zhou, L. Ying, and J. He, “Multic²: an optimization framework for learning from task and worker dual heterogeneity,” in *Proceedings of the 2017 SIAM International Conference on Data Mining, Houston, Texas, USA, April 27-29, 2017*, N. V. Chawla and W. Wang, Eds. SIAM, 2017. [Online]. Available: <https://doi.org/10.1137/1.9781611974973.65> pp. 579–587.
- [156] Y. Zhou and J. He, “Crowdsourcing via tensor augmentation and completion,” in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, S. Kambhampati, Ed. IJCAI/AAAI Press, 2016. [Online]. Available: <http://www.ijcai.org/Abstract/16/347> pp. 2435–2441.
- [157] S. O. Gharan and L. Trevisan, “Approximating the expansion profile and almost optimal local graph clustering,” in *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*. IEEE Computer Society, 2012. [Online]. Available: <https://doi.org/10.1109/FOCS.2012.85> pp. 187–196.
- [158] R. Andersen, S. O. Gharan, Y. Peres, and L. Trevisan, “Almost optimal local graph clustering using evolving sets,” *J. ACM*, vol. 63, no. 2, pp. 15:1–15:31, 2016. [Online]. Available: <https://doi.org/10.1145/2856030>
- [159] H. Yin, A. R. Benson, J. Leskovec, and D. F. Gleich, “Local higher-order graph clustering,” in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*. ACM, 2017. [Online]. Available: <https://doi.org/10.1145/3097983.3098069> pp. 555–564.
- [160] Y. Shi, X. He, N. Zhang, C. Yang, and J. Han, “User-guided clustering in heterogeneous information networks via motif-based comprehensive transcription,” in *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2019, Würzburg, Germany, September 16-20, 2019, Proceedings, Part I*, ser. Lecture Notes in Computer Science, U. Brefeld, É. Fromont, A. Hotho, A. J. Knobbe, M. H. Maathuis, and C. Robardet, Eds., vol. 11906. Springer, 2019. [Online]. Available: https://doi.org/10.1007/978-3-030-46150-8_22 pp. 361–377.
- [161] M. Rosvall, A. V. Esquivel, A. Lancichinetti, J. D. West, and R. Lambiotte, “Memory in network flows and its effects on spreading dynamics and community detection,” *Nature communications*, vol. 5, no. 1, pp. 1–13, 2014.

- [162] F. Chierichetti, R. Kumar, P. Raghavan, and T. Sarlós, “Are web users really markovian?” in *Proceedings of the 21st World Wide Web Conference 2012, WWW 2012, Lyon, France, April 16-20, 2012*, A. Mille, F. L. Gandon, J. Misselis, M. Rabinovich, and S. Staab, Eds. ACM, 2012. [Online]. Available: <https://doi.org/10.1145/2187836.2187919> pp. 609–618.
- [163] A. E. Raftery, “A model for high-order markov chains,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 47, no. 3, pp. 528–539, 1985.
- [164] S. Adke and S. Deshmukh, “Limit distribution of a high order markov chain,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 50, no. 1, pp. 105–108, 1988.
- [165] W.-K. Ching and M. K. Ng, “Markov chains,” *Models, algorithms and applications*, 2006.
- [166] A. R. Benson, D. F. Gleich, and L. Lim, “The spacey random walk: A stochastic process for higher-order data,” *SIAM Review*, vol. 59, no. 2, pp. 321–345, 2017. [Online]. Available: <https://doi.org/10.1137/16M1074023>
- [167] B. Bollobás, *Modern Graph Theory*, ser. Graduate Texts in Mathematics. Springer, 2002, vol. 184. [Online]. Available: <https://doi.org/10.1007/978-1-4612-0619-4>
- [168] J. Síma and S. E. Schaeffer, “On the np-completeness of some graph cluster measures,” in *SOFSEM 2006: Theory and Practice of Computer Science, 32nd Conference on Current Trends in Theory and Practice of Computer Science, Merín, Czech Republic, January 21-27, 2006, Proceedings*, ser. Lecture Notes in Computer Science, J. Wiedermann, G. Tel, J. Pokorný, M. Bieliková, and J. Stuller, Eds., vol. 3831. Springer, 2006. [Online]. Available: https://doi.org/10.1007/11611257_51 pp. 530–537.
- [169] L. Lovász and M. Simonovits, “The mixing rate of markov chains, an isoperimetric inequality, and computing the volume,” in *31st Annual Symposium on Foundations of Computer Science, St. Louis, Missouri, USA, October 22-24, 1990, Volume I*. IEEE Computer Society, 1990. [Online]. Available: <https://doi.org/10.1109/FSCS.1990.89553> pp. 346–354.
- [170] L. Lovász and M. Simonovits, “Random walks in a convex body and an improved volume algorithm,” *Random Struct. Algorithms*, vol. 4, no. 4, pp. 359–412, 1993. [Online]. Available: <https://doi.org/10.1002/rsa.3240040402>
- [171] S. M. Ross, *Introduction to probability models*. Academic press, 2014.
- [172] A. B. Hollingshead et al., *Four factor index of social status*. New Haven, CT, 1975.
- [173] S. Fortunato, “Community detection in graphs,” *Physics reports*, 2010.
- [174] J. Leskovec and A. Krevl, “SNAP Datasets: Stanford large network dataset collection,” <http://snap.stanford.edu/data>, June 2014.

- [175] F. M. Harper and J. A. Konstan, “The movielens datasets: History and context,” *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, pp. 19:1–19:19, 2016. [Online]. Available: <https://doi.org/10.1145/2827872>
- [176] C. E. Tsourakakis, F. Bonchi, A. Gionis, F. Gullo, and M. A. Tsiarli, “Denser than the densest subgraph: extracting optimal quasi-cliques with quality guarantees,” in *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013*, I. S. Dhillon, Y. Koren, R. Ghani, T. E. Senator, P. Bradley, R. Parekh, J. He, R. L. Grossman, and R. Uthurusamy, Eds. ACM, 2013. [Online]. Available: <https://doi.org/10.1145/2487575.2487645> pp. 104–112.
- [177] C. H. Q. Ding, T. Li, and M. I. Jordan, “Nonnegative matrix factorization for combinatorial optimization: Spectral clustering, graph matching, and clique finding,” in *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy*. IEEE Computer Society, 2008. [Online]. Available: <https://doi.org/10.1109/ICDM.2008.130> pp. 183–192.
- [178] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. [Online]. Available: <https://openreview.net/forum?id=SJU4ayYgl>
- [179] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph attention networks,” in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. [Online]. Available: <https://openreview.net/forum?id=rJXMpikCZ>
- [180] W. L. Hamilton, R. Ying, and J. Leskovec, “Representation learning on graphs: Methods and applications,” *IEEE Data Eng. Bull.*, vol. 40, no. 3, pp. 52–74, 2017. [Online]. Available: <http://sites.computer.org/debull/A17sept/p52.pdf>
- [181] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, “How powerful are graph neural networks?” in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. [Online]. Available: <https://openreview.net/forum?id=ryGs6iA5Km>
- [182] M. Guye, G. Bettus, F. Bartolomei, and P. J. Cozzone, “Graph theoretical analysis of structural and functional connectivity mri in normal and pathological brain networks,” *Magnetic Resonance Materials in Physics, Biology and Medicine*, vol. 23, no. 5-6, pp. 409–421, 2010.

- [183] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, “Neural message passing for quantum chemistry,” in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 2017. [Online]. Available: <http://proceedings.mlr.press/v70/gilmer17a.html> pp. 1263–1272.
- [184] E. Choi, M. T. Bahadori, L. Song, W. F. Stewart, and J. Sun, “GRAM: graph-based attention model for healthcare representation learning,” in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*. ACM, 2017. [Online]. Available: <https://doi.org/10.1145/3097983.3098126> pp. 787–795.
- [185] K. He, R. B. Girshick, and P. Dollár, “Rethinking imagenet pre-training,” in *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 2019. [Online]. Available: <https://doi.org/10.1109/ICCV.2019.00502> pp. 4917–4926.
- [186] T. Mikolov, E. Grave, P. Bojanowski, C. Puhersch, and A. Joulin, “Advances in pre-training distributed word representations,” in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*, N. Calzolari, K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, and T. Tokunaga, Eds. European Language Resources Association (ELRA), 2018. [Online]. Available: <http://www.lrec-conf.org/proceedings/lrec2018/summaries/721.html>
- [187] N. Navarin, D. V. Tran, and A. Sperduti, “Pre-training graph neural networks with kernels,” *arXiv preprint arXiv:1811.06930*, 2018.
- [188] Z. Hu, C. Fan, T. Chen, K.-W. Chang, and Y. Sun, “Pre-training graph neural networks for generic structural feature extraction,” *arXiv preprint arXiv:1905.13728*, 2019.
- [189] W. Hu, B. Liu, J. Gomes, M. Zitnik, P. Liang, V. S. Pande, and J. Leskovec, “Strategies for pre-training graph neural networks,” in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. [Online]. Available: <https://openreview.net/forum?id=HJIWWJSFDH>
- [190] M. T. Rosenstein, Z. Marx, L. P. Kaelbling, and T. G. Dietterich, “To transfer or not to transfer,” in *NIPS 2005 workshop on transfer learning*, vol. 898, 2005, pp. 1–4.

- [191] S. Hu, Z. Xiong, M. Qu, X. Yuan, M. Côté, Z. Liu, and J. Tang, “Graph policy network for transferable active learning on graphs,” in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020. [Online]. Available: <https://proceedings.neurips.cc/paper/2020/hash/73740ea85c4ec25f00f9acbd859f861d-Abstract.html>
- [192] D. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre, R. Gómez-Bombarelli, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, “Convolutional networks on graphs for learning molecular fingerprints,” in *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds., 2015, pp. 2224–2232.
- [193] L. Jiang, Z. Zhou, T. Leung, L. Li, and L. Fei-Fei, “Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels,” in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, ser. Proceedings of Machine Learning Research, J. G. Dy and A. Krause, Eds., vol. 80. PMLR, 2018. [Online]. Available: <http://proceedings.mlr.press/v80/jiang18c.html> pp. 2309–2318.
- [194] Z. Hu, Y. Dong, K. Wang, K. Chang, and Y. Sun, “GPT-GNN: generative pre-training of graph neural networks,” in *KDD ’20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, R. Gupta, Y. Liu, J. Tang, and B. A. Prakash, Eds. ACM, 2020. [Online]. Available: <https://doi.org/10.1145/3394486.3403237> pp. 1857–1867.
- [195] J. Qiu, Q. Chen, Y. Dong, J. Zhang, H. Yang, M. Ding, K. Wang, and J. Tang, “GCC: graph contrastive coding for graph neural network pre-training,” in *KDD ’20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, 2020.
- [196] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, “Supervised contrastive learning,” in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020. [Online]. Available: <https://proceedings.neurips.cc/paper/2020/hash/d89a66c7c80a29b1bdbab0f2a1a94af8-Abstract.html>
- [197] F. Sun, J. Hoffmann, V. Verma, and J. Tang, “Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization,” in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. [Online]. Available: <https://openreview.net/forum?id=r1lff2NYvH>

- [198] H. Zhao, R. T. des Combes, K. Zhang, and G. J. Gordon, “On learning invariant representations for domain adaptation,” in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 2019. [Online]. Available: <http://proceedings.mlr.press/v97/zhao19a.html> pp. 7523–7532.
- [199] B. Li, Y. Wang, S. Zhang, D. Li, T. Darrell, K. Keutzer, and H. Zhao, “Learning invariant representations and risks for semi-supervised domain adaptation,” *CoRR*, 2020.
- [200] J. Lee, H. Kim, J. Lee, and S. Yoon, “Transfer learning for deep learning on graph-structured data,” in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, S. P. Singh and S. Markovitch, Eds. AAAI Press, 2017. [Online]. Available: <http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14803> pp. 2154–2160.
- [201] R. Levie, M. M. Bronstein, and G. Kutyniok, “Transferability of spectral graph convolutional neural networks,” *CoRR*, vol. abs/1907.12972, 2019.
- [202] M. Wu, S. Pan, C. Zhou, X. Chang, and X. Zhu, “Unsupervised domain adaptive graph convolutional networks,” in *WWW ’20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*. ACM / IW3C2, 2020, pp. 1457–1467.
- [203] T. N. Kipf and M. Welling, “Variational graph auto-encoders,” *CoRR*, vol. abs/1611.07308, 2016. [Online]. Available: <http://arxiv.org/abs/1611.07308>
- [204] W. L. Hamilton, Z. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017. [Online]. Available: <https://proceedings.neurips.cc/paper/2017/hash/5dd9db5e033da9c6fb5ba83c7a7e9-Abstract.html> pp. 1024–1034.
- [205] P. Velickovic, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, “Deep graph infomax,” in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. [Online]. Available: <https://openreview.net/forum?id=rklz9iAcKQ>
- [206] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, “LINE: large-scale information network embedding,” in *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, A. Gangemi, S. Leonardi, and A. Panconesi, Eds. ACM, 2015. [Online]. Available: <https://doi.org/10.1145/2736277.2741093> pp. 1067–1077.

- [207] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, “A theory of learning from different domains,” *Mach. Learn.*, vol. 79, no. 1-2, pp. 151–175, 2010. [Online]. Available: <https://doi.org/10.1007/s10994-009-5152-4>
- [208] H. Zhao, S. Zhang, G. Wu, J. M. F. Moura, J. P. Costeira, and G. J. Gordon, “Adversarial multiple source domain adaptation,” in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., 2018. [Online]. Available: <https://proceedings.neurips.cc/paper/2018/hash/717d8b3d60d9eea997b35b02b6a4e867-Abstract.html> pp. 8568–8579.
- [209] Z. Ying, J. You, C. Morris, X. Ren, W. L. Hamilton, and J. Leskovec, “Hierarchical graph representation learning with differentiable pooling,” in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., 2018. [Online]. Available: <https://proceedings.neurips.cc/paper/2018/hash/e77dbaf6759253c7c6d0efc5690369c7-Abstract.html> pp. 4805–4815.
- [210] M. P. Kumar, B. Packer, and D. Koller, “Self-paced learning for latent variable models,” in *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada*, J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, Eds. Curran Associates, Inc., 2010. [Online]. Available: <https://proceedings.neurips.cc/paper/2010/hash/e57c6b956a6521b28495f2886ca0977a-Abstract.html> pp. 1189–1197.
- [211] X. Zhu, “Machine teaching: An inverse problem to machine learning and an approach toward optimal education,” in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, B. Bonet and S. Koenig, Eds. AAAI Press, 2015. [Online]. Available: <http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9487> pp. 4083–4087.
- [212] Y. Fan, F. Tian, T. Qin, X. Li, and T. Liu, “Learning to teach,” in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. [Online]. Available: <https://openreview.net/forum?id=HJewuJWCZ>
- [213] L. Jiang, D. Meng, Q. Zhao, S. Shan, and A. G. Hauptmann, “Self-paced curriculum learning,” in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, B. Bonet and S. Koenig, Eds. AAAI Press, 2015. [Online]. Available: <http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9750> pp. 2694–2700.

- [214] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” in *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, ser. ACM International Conference Proceeding Series, A. P. Danyluk, L. Bottou, and M. L. Littman, Eds., vol. 382. ACM, 2009. [Online]. Available: <https://doi.org/10.1145/1553374.1553380> pp. 41–48.
- [215] P. Tseng, “Convergence of a block coordinate descent method for nondifferentiable minimization,” *Journal of optimization theory and applications*, vol. 109, no. 3, pp. 475–494, 2001.
- [216] P. Velickovic, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, “Deep graph infomax,” *CoRR*, vol. abs/1809.10341, 2018. [Online]. Available: <http://arxiv.org/abs/1809.10341>
- [217] Q. Lu and L. Getoor, “Link-based classification,” in *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA*, T. Fawcett and N. Mishra, Eds. AAAI Press, 2003. [Online]. Available: <http://www.aaai.org/Library/ICML/2003/icml03-066.php> pp. 496–503.
- [218] G. Namata, B. London, L. Getoor, B. Huang, and U. EDU, “Query-driven active surveying for collective classification,” in *10th International Workshop on Mining and Learning with Graphs*, vol. 8, 2012.
- [219] W. Hamilton, J. Zhang, C. Danescu-Niculescu-Mizil, D. Jurafsky, and J. Leskovec, “Loyalty in online communities,” in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 11, no. 1, 2017.
- [220] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.
- [221] J. Wang, X. Peng, W. Peng, and F.-X. Wu, “Dynamic protein interaction network construction and applications,” *Proteomics*, 2014.
- [222] D. Fu and J. He, “DPPIN: A biological dataset of dynamic protein-protein interaction networks,” *CoRR*, 2021.
- [223] L. van der Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008. [Online]. Available: <http://jmlr.org/papers/v9/vandermaaten08a.html>
- [224] P. Erdős and A. Rényi, “On random graphs, i,” *Publicationes Mathematicae (Debrecen)*, 1959.
- [225] E. M. Fich and A. Shivdasani, “Financial fraud, director reputation, and shareholder wealth,” *Journal of financial Economics*, vol. 86, no. 2, pp. 306–336, 2007.

- [226] N. Jindal and B. Liu, “Review spam detection,” in *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*, C. L. Williamson, M. E. Zurko, P. F. Patel-Schneider, and P. J. Shenoy, Eds. ACM, 2007. [Online]. Available: <https://doi.org/10.1145/1242572.1242759> pp. 1189–1190.
- [227] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: Online learning of social representations,” in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 701–710.
- [228] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, “LINE: large-scale information network embedding,” in *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, A. Gangemi, S. Leonardi, and A. Panconesi, Eds. ACM, 2015. [Online]. Available: <https://doi.org/10.1145/2736277.2741093> pp. 1067–1077.
- [229] A. Grover and J. Leskovec, “node2vec: Scalable feature learning for networks,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, B. Krishnapuram, M. Shah, A. J. Smola, C. C. Aggarwal, D. Shen, and R. Rastogi, Eds. ACM, 2016. [Online]. Available: <https://doi.org/10.1145/2939672.2939754> pp. 855–864.
- [230] Z. Yang, W. W. Cohen, and R. Salakhutdinov, “Revisiting semi-supervised learning with graph embeddings,” in *Proceedings of the 33rd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, 2016, pp. 40–48.
- [231] T. Chen and Y. Sun, “Task-guided and path-augmented heterogeneous network embedding for author identification,” in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 2017, Cambridge, United Kingdom, February 6-10, 2017*, M. de Rijke, M. Shokouhi, A. Tomkins, and M. Zhang, Eds. ACM, 2017. [Online]. Available: <https://doi.org/10.1145/3018661.3018735> pp. 295–304.
- [232] X. Wang, P. Cui, J. Wang, J. Pei, W. Zhu, and S. Yang, “Community preserving network embedding,” in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, S. P. Singh and S. Markovitch, Eds. AAAI Press, 2017. [Online]. Available: <http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14589> pp. 203–209.

- [233] X. Ren and L. Bo, “Discriminatively trained sparse code gradients for contour detection,” in *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., 2012. [Online]. Available: <https://proceedings.neurips.cc/paper/2012/hash/16a5cdac362b8d27a1d8f8c7b78b4330-Abstract.html> pp. 593–601.
- [234] W. Shen, X. Wang, Y. Wang, X. Bai, and Z. Zhang, “Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection,” in *IEEE CVPR (2015)*, 2015.
- [235] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: synthetic minority over-sampling technique,” *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002. [Online]. Available: <https://doi.org/10.1613/jair.953>
- [236] B. Zadrozny, J. Langford, and N. Abe, “Cost-sensitive learning by cost-proportionate example weighting,” in *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM 2003), 19-22 December 2003, Melbourne, Florida, USA*. IEEE Computer Society, 2003. [Online]. Available: <https://doi.org/10.1109/ICDM.2003.1250950> p. 435.
- [237] H. He and E. A. Garcia, “Learning from imbalanced data,” *IEEE Transactions on knowledge and data engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.
- [238] V. Hodge and J. Austin, “A survey of outlier detection methodologies,” *Artificial intelligence review*, vol. 22, no. 2, pp. 85–126, 2004.
- [239] S. Li, M. Shao, and Y. Fu, “Multi-view low-rank analysis for outlier detection,” in *Proceedings of the 2015 SIAM International Conference on Data Mining*. SIAM, 2015, pp. 748–756.
- [240] S. Li, M. Shao, and Y. Fu, “Multi-view low-rank analysis with applications to outlier detection,” *TKDD*, 2018.
- [241] Y. Sun, M. S. Kamel, and Y. Wang, “Boosting for learning multiple classes with imbalanced class distribution,” in *Data Mining, 2006. ICDM’06. Sixth International Conference on*. IEEE, 2006, pp. 592–602.
- [242] G. Wu and E. Y. Chang, “Adaptive feature-space conformal transformation for imbalanced-data learning,” in *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 2003, pp. 816–823.
- [243] J. He, H. Tong, and J. G. Carbonell, “Rare category characterization,” in *ICDM 2010, The 10th IEEE International Conference on Data Mining, Sydney, Australia, 14-17 December 2010*, G. I. Webb, B. Liu, C. Zhang, D. Gunopulos, and X. Wu, Eds. IEEE Computer Society, 2010. [Online]. Available: <https://doi.org/10.1109/ICDM.2010.154> pp. 226–235.

- [244] C. Huang, Y. Li, C. Change Loy, and X. Tang, “Learning deep representation for imbalanced classification,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5375–5384.
- [245] M. Belkin and P. Niyogi, “Laplacian eigenmaps and spectral techniques for embedding and clustering,” in *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds. MIT Press, 2001. [Online]. Available: <https://proceedings.neurips.cc/paper/2001/hash/f106b7f99d2cb30c3db1c3cc0fde9ccb-Abstract.html> pp. 585–591.
- [246] J. B. Kruskal, *Multidimensional scaling*. Sage, 1978, no. 11.
- [247] S. T. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
- [248] J. B. Tenenbaum, V. De Silva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” *science*, vol. 290, no. 5500, pp. 2319–2323, 2000.
- [249] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in Neural Information Processing Systems*, C. J. C. Burges, L. Bottou, Z. Ghahramani, and K. Q. Weinberger, Eds., 2013. [Online]. Available: <https://proceedings.neurips.cc/paper/2013>
- [250] R. Guo, J. Li, and H. Liu, “INITIATOR: noise-contrastive estimation for marked temporal point process,” in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, J. Lang, Ed. ijcai.org, 2018. [Online]. Available: <https://doi.org/10.24963/ijcai.2018/303> pp. 2191–2197.
- [251] J. Wu, J. He, and Y. Liu, “Imverde: Vertex-diminished random walk for learning network representation from imbalanced data,” *CoRR*, vol. abs/1804.09222, 2018. [Online]. Available: <http://arxiv.org/abs/1804.09222>
- [252] Y. Bengio, “Evolving culture versus local minima,” in *Growing Adaptive Machines - Combining Development and Learning in Artificial Neural Networks*, ser. Studies in Computational Intelligence, T. Kowaliw, N. Bredèche, and R. Doursat, Eds. Springer, 2014, vol. 557, pp. 109–138. [Online]. Available: https://doi.org/10.1007/978-3-642-55337-0_3
- [253] F. Khan, X. J. Zhu, and B. Mutlu, “How do humans teach: On curriculum learning and teaching dimension,” in *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. C. N. Pereira, and K. Q. Weinberger, Eds., 2011. [Online]. Available: <https://proceedings.neurips.cc/paper/2011/hash/f9028faec74be6ec9b852b0a542e2f39-Abstract.html> pp. 1449–1457.

- [254] V. I. Spitzkovsky, H. Alshawi, and D. Jurafsky, “From baby steps to leapfrog: How ”less is more” in unsupervised dependency parsing,” in *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 2-4, 2010, Los Angeles, California, USA*. The Association for Computational Linguistics, 2010. [Online]. Available: <https://aclanthology.org/N10-1116/> pp. 751–759.
- [255] F. Ma, D. Meng, Q. Xie, Z. Li, and X. Dong, “Self-paced co-training,” in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 2017. [Online]. Available: <http://proceedings.mlr.press/v70/ma17b.html> pp. 2275–2284.
- [256] S. Wu, Q. Ji, S. Wang, H. Wong, Z. Yu, and Y. Xu, “Semi-supervised image classification with self-paced cross-task networks,” *IEEE Trans. Multim.*, vol. 20, no. 4, pp. 851–865, 2018. [Online]. Available: <https://doi.org/10.1109/TMM.2017.2758522>
- [257] L. Bottou, “Large-scale machine learning with stochastic gradient descent,” in *19th International Conference on Computational Statistics, COMPSTAT 2010, Paris, France, August 22-27, 2010 - Keynote, Invited and Contributed Papers*, Y. Lechevallier and G. Saporta, Eds. Physica-Verlag, 2010, pp. 177–186. [Online]. Available: https://doi.org/10.1007/978-3-7908-2604-3_16
- [258] L. v. d. Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.
- [259] J. Jovanovic, E. Bagheri, and D. Gasevic, “Comprehension and learning of social goals through visualization,” *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 4, pp. 478–489, 2015.
- [260] H. Huang, Q. He, K. Chiew, F. Qian, and L. Ma, “CLOVER: a faster prior-free approach to rare-category detection,” *Knowledge and information systems*, vol. 35, no. 3, pp. 713–736, 2013.
- [261] H. Huang, Q. He, J. He, and L. Ma, “RADAR: Rare category detection via computation of boundary degree,” in *Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, 2011, pp. 258–269.
- [262] F. Beck, M. Burch, S. Diehl, and D. Weiskopf, “The state of the art in visualizing dynamic graphs.” in *EuroVis (STARs)*. Citeseer, 2014.
- [263] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, “Network anomaly detection: methods, systems and tools,” *Ieee communications surveys & tutorials*, vol. 16, no. 1, pp. 303–336, 2013.
- [264] N. A. Heard, D. J. Weston, K. Platanioti, and D. J. Hand, “Bayesian anomaly detection methods for social networks,” *The Annals of Applied Statistics*, pp. 645–662, 2010.

- [265] Dawei Zhou, A. Karthikeyan, K. Wang, N. Cao, and J. He, “Discovering rare categories from graph streams,” *Data Min. Knowl. Discov.*, vol. 31, no. 2, pp. 400–423, 2017. [Online]. Available: <https://doi.org/10.1007/s10618-016-0478-6>
- [266] P. Vatturi and W.-K. Wong, “Category detection using hierarchical mean shift,” in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 847–856.
- [267] H. Lin, S. Gao, D. Gotz, F. Du, J. He, and N. Cao, “RCLens: Interactive rare category exploration and identification,” *IEEE transactions on visualization and computer graphics*, 2017.
- [268] T. Haberkorn, I. Koglbauer, and R. Braunstingl, “Traffic displays for visual flight indicating track and priority cues,” *IEEE transactions on human-machine systems*, vol. 44, no. 6, pp. 755–766, 2014.
- [269] Y. Liu, S. Dai, C. Wang, Z. Zhou, and H. Qu, “GenealogyVis: A system for visual analysis of multidimensional genealogical data,” *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 6, pp. 873–885, 2017.
- [270] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.
- [271] T. Zhang, X. Wang, Z. Li, F. Guo, Y. Ma, and W. Chen, “A survey of network anomaly visualization,” *Science China Information Sciences*, vol. 60, no. 12, p. 121101, 2017.
- [272] Jolliffe and Ian, “Principal component analysis,” *Springer Berlin*, vol. 87, no. 100, pp. 41–64, 1986.
- [273] A. Inselberg, *Parallel Coordinates*. Springer New York, 2009.
- [274] N. Cao, D. Gotz, J. Sun, and H. Qu, “DICON: Interactive visual analysis of multidimensional clusters,” *IEEE Transactions on Visualization & Computer Graphics*, vol. 17, no. 12, pp. 2581–2590, 2011.
- [275] P. Xu, H. Mei, R. Liu, and C. Wei, “ViDX: Visual diagnostics of assembly line performance in smart factories,” *IEEE Transactions on Visualization & Computer Graphics*, vol. 23, no. 1, p. 291, 2017.
- [276] E. Corchado and Á. Herrero, “Neural visualization of network traffic data for intrusion detection,” *Applied Soft Computing*, vol. 11, no. 2, pp. 2042–2056, 2011.
- [277] C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin, and W.-Y. Lin, “Intrusion detection by machine learning: A review,” *Expert Systems with Applications*, vol. 36, no. 10, pp. 11 994–12 000, 2009.
- [278] S. T. Teoh, K. L. Ma, S. F. Wu, and X. Zhao, “Case study: Interactive visualization for internet security,” in *Proceedings of the conference on Visualization’02*. IEEE Computer Society, 2002, pp. 505–508.

- [279] D. Thom, H. Bosch, S. Koch, M. Wörner, and T. Ertl, “Spatiotemporal anomaly detection through visual analysis of geolocated twitter messages,” in *Pacific visualization symposium (PacificVis), 2012 IEEE*. IEEE, 2012, pp. 41–48.
- [280] J. Zhao, N. Cao, Z. Wen, Y. Song, Y.-R. Lin, and C. Collins, “FluxFlow: Visual analysis of anomalous information spreading on social media,” *IEEE Transactions on Visualization & Computer Graphics*, vol. 20, no. 12, pp. 1773–1782, 2014.
- [281] N. Cao, C. Shi, S. Lin, J. Lu, Y.-R. Lin, and C.-Y. Lin, “TargetVue: Visual analysis of anomalous user behaviors in online communication systems,” *IEEE transactions on visualization & computer graphics*, vol. 22, no. 1, pp. 280–289, 2016.
- [282] C. Wang, Z. Xiao, Y. Liu, Y. Xu, A. Zhou, and K. Zhang, “SentiView: Sentiment analysis and visualization for internet popular topics,” *IEEE transactions on human-machine systems*, vol. 43, no. 6, pp. 620–630, 2013.
- [283] X. Fan, C. Li, X. Yuan, X. Dong, and J. Liang, “An interactive visual analytics approach for network anomaly detection through smart labeling,” *J. Vis.*, vol. 22, no. 5, pp. 955–971, 2019. [Online]. Available: <https://doi.org/10.1007/s12650-019-00580-7>
- [284] B. Bach, E. Pietriga, and J.-D. Fekete, “GraphDiaries: Animated transitions and temporal navigation for dynamic networks,” *IEEE transactions on visualization and computer graphics*, vol. 20, no. 5, pp. 740–754, 2013.
- [285] K.-P. Yee, D. Fisher, R. Dhamija, and M. Hearst, “Animated exploration of dynamic graphs with radial layout,” in *IEEE Symposium on Information Visualization*, 2001, pp. 43–43.
- [286] D. Archambault, H. Purchase, and B. Pinaud, “Animation, small multiples, and the effect of mental map preservation in dynamic graphs,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 4, pp. 539–552, 2011.
- [287] U. Brandes and B. Nick, “Asymmetric relations in longitudinal social networks,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2283–2290, 2011.
- [288] D. Oelke, D. Kokkinakis, and D. A. Keim, “Fingerprint Matrices: Uncovering the dynamics of social networks in prose literature,” in *Computer Graphics Forum*, vol. 32, no. 3pt4. Wiley Online Library, 2013, pp. 371–380.
- [289] M. Burch, B. Schmidt, and D. Weiskopf, “A matrix-based visualization for exploring dynamic compound digraphs,” in *2013 17th International Conference on Information Visualisation*. IEEE, 2013, pp. 66–73.
- [290] B. Bach, N. Henry-Riche, T. Dwyer, T. Madhyastha, J.-D. Fekete, and T. Grabowski, “Small MultiPiles: Piling time to explore temporal patterns in dynamic networks,” in *Computer Graphics Forum*, vol. 34, no. 3. Wiley Online Library, 2015, pp. 31–40.

- [291] B. Bach, E. Pietriga, and J.-D. Fekete, “Visualizing dynamic networks with matrix cubes,” in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM, 2014, pp. 877–886.
- [292] J. Zhao, Z. Liu, M. Dontcheva, A. Hertzmann, and A. Wilson, “MatrixWave: Visual comparison of event sequence data,” in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2015, pp. 259–268.
- [293] S. van den Elzen, D. Holten, J. Blaas, and J. J. van Wijk, “Reducing snapshots to points: A visual analytics approach to dynamic network exploration,” *IEEE transactions on visualization and computer graphics*, vol. 22, no. 1, pp. 1–10, 2015.
- [294] N. Henry, J.-D. Fekete, and M. J. McGuffin, “NodeTrix: a hybrid visualization of social networks,” *IEEE transactions on visualization and computer graphics*, vol. 13, no. 6, pp. 1302–1309, 2007.
- [295] R. Blanch, R. Dautriche, and G. Bisson, “Dendrogramix: A hybrid tree-matrix visualization technique to support interactive exploration of dendrograms,” in *2015 IEEE Pacific Visualization Symposium (PacificVis)*. IEEE, 2015, pp. 31–38.
- [296] C. Vehlow, F. Beck, P. Auwärter, and D. Weiskopf, “Visualizing the evolution of communities in dynamic graphs,” in *Computer Graphics Forum*, vol. 34, no. 1. Wiley Online Library, 2015, pp. 277–288.
- [297] M. Hlawatsch, M. Burch, and D. Weiskopf, “Visual adjacency lists for dynamic graphs,” *IEEE transactions on visualization and computer graphics*, vol. 20, no. 11, pp. 1590–1603, 2014.
- [298] P. Riehmann, M. Hanfler, and B. Froehlich, “Interactive sankey diagrams,” in *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005*. IEEE, 2005, pp. 233–240.
- [299] S. Havre, B. Hetzler, and L. Nowell, “ThemeRiver: Visualizing theme changes over time,” in *IEEE Symposium on Information Visualization 2000. INFOVIS 2000. Proceedings*. IEEE, 2000, pp. 115–123.
- [300] M. E. Newman and M. Girvan, “Finding and evaluating community structure in networks,” *Physical Review E Statistical Nonlinear & Soft Matter Physics*, vol. 69, no. 2, p. 026113, 2004.
- [301] E. R. Gansner, Y. Koren, and S. C. North, “Topological fisheye views for visualizing large graphs,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 4, pp. 457–468, 2005.
- [302] K.-C. Feng, C. Wang, H.-W. Shen, and T.-Y. Lee, “Coherent time-varying graph drawing with multifocus+ context interaction,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 8, pp. 1330–1342, 2012.

- [303] P. K. Sundararajan, O. J. Mengshoel, and T. Selker, “Multi-focus and multi-window techniques for interactive network exploration,” in *Visualization and Data Analysis 2013*, vol. 8654. International Society for Optics and Photonics, 2013, p. 86540O.
- [304] P. Isenberg, F. Heimerl, S. Koch, T. Isenberg, P. Xu, C. D. Stolper, M. M. Sedlmair, J. Chen, T. Möller, and J. Stasko, “vispubdata.org: A Metadata Collection about IEEE Visualization (VIS) Publications,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, 2017, to appear. [Online]. Available: <https://hal.inria.fr/hal-01376597>
- [305] J. You, B. Liu, Z. Ying, V. S. Pande, and J. Leskovec, “Graph convolutional policy network for goal-directed molecular graph generation,” in *Advances in Neural Information Processing Systems*, S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., 2018. [Online]. Available: <https://proceedings.neurips.cc/paper/2018>
- [306] J. Kang and H. Tong, “N2N: network derivative mining,” in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, W. Zhu, D. Tao, X. Cheng, P. Cui, E. A. Rundensteiner, D. Carmel, Q. He, and J. X. Yu, Eds. ACM, 2019. [Online]. Available: <https://dl.acm.org/citation.cfm?id=3357384>
- [307] Y. Ban, X. Liu, L. Huang, Y. Duan, X. Liu, and W. Xu, “No place to hide: Catching fraudulent entities in tensors,” in *The World Wide Web Conference*, L. Liu, R. W. White, A. Mantrach, F. Silvestri, J. J. McAuley, R. Baeza-Yates, and L. Zia, Eds. ACM, 2019. [Online]. Available: <https://doi.org/10.1145/3308558>
- [308] X. Liu, J. He, S. Duddy, and L. O’Sullivan, “Convolution-consistent collective matrix completion,” in *International Conference on Information and Knowledge Management*, W. Zhu, D. Tao, X. Cheng, P. Cui, E. A. Rundensteiner, D. Carmel, Q. He, and J. X. Yu, Eds. ACM, 2019. [Online]. Available: <https://dl.acm.org/citation.cfm?id=3357384>
- [309] H. Shao, D. Sun, J. Wu, Z. Zhang, A. Zhang, S. Yao, S. Liu, T. Wang, C. Zhang, and T. F. Abdelzaher, “paper2repo: Github repository recommendation for academic papers,” in *The Web Conference*, Y. Huang, I. King, T. Liu, and M. van Steen, Eds. ACM / IW3C2, 2020. [Online]. Available: <https://doi.org/10.1145/3366423>
- [310] H. Shao, S. Yao, Y. Zhao, C. Zhang, J. Han, L. M. Kaplan, L. Su, and T. F. Abdelzaher, “A constrained maximum likelihood estimator for unguided social sensing,” in *IEEE Conference on Computer Communications*. IEEE, 2018. [Online]. Available: <https://ieeexplore.ieee.org/xpl/conhome/8464035/proceeding>
- [311] R. Albert and A. Barabási, “Statistical mechanics of complex networks,” *CoRR*, vol. cond-mat/0106096, 2001.
- [312] F. Fischer and C. Helmberg, “Dynamic graph generation for the shortest path problem in time expanded networks,” *Math. Program.*, 2014.

- [313] B. M. Waxman, "Routing of multipoint connections," *IEEE J. Sel. Areas Commun.*, 1988.
- [314] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., 2014. [Online]. Available: <https://proceedings.neurips.cc/paper/2014>
- [315] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *2nd International Conference on Learning Representations*, Y. Bengio and Y. LeCun, Eds., 2014. [Online]. Available: <https://openreview.net/group?id=ICLR.cc/2014>
- [316] J. You, R. Ying, X. Ren, W. L. Hamilton, and J. Leskovec, "Graphrnn: Generating realistic graphs with deep auto-regressive models," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. G. Dy and A. Krause, Eds., vol. 80. PMLR, 2018. [Online]. Available: <http://proceedings.mlr.press/v80/>
- [317] T. Li, J. Zhang, P. S. Yu, Y. Zhang, and Y. Yan, "Deep dynamic network embedding for link prediction," *IEEE Access*, 2018.
- [318] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, J. Lang, Ed. ijcai.org, 2018. [Online]. Available: <http://www.ijcai.org/proceedings/2018/>
- [319] D. Chakrabarti, Y. Zhan, and C. Faloutsos, "R-MAT: A recursive model for graph mining," in *Proceedings of the Fourth SIAM International Conference on Data Mining*, M. W. Berry, U. Dayal, C. Kamath, and D. B. Skillicorn, Eds. SIAM, 2004. [Online]. Available: <https://doi.org/10.1137/1.9781611972740>
- [320] A. Goldenberg, A. X. Zheng, S. E. Fienberg, and E. M. Airoldi, "A survey of statistical network models," *Foundations and Trends in Machine Learning*, 2009.
- [321] L. Akoglu and C. Faloutsos, "RTG: a recursive realistic graph generator using random typing," *Data Min. Knowl. Discov.*, vol. 19, no. 2, pp. 194–209, 2009. [Online]. Available: <https://doi.org/10.1007/s10618-009-0140-7>
- [322] J. M. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins, "The web as a graph: Measurements, models, and methods," in *5th Annual International Conference of Computing and Combinatorics*, ser. Lecture Notes in Computer Science, T. Asano, H. Imai, D. T. Lee, S. Nakano, and T. Tokuyama, Eds., vol. 1627. Springer, 1999. [Online]. Available: <https://doi.org/10.1007/3-540-48686-0>
- [323] G. Robins, P. Pattison, Y. Kalish, and D. Lusher, "An introduction to exponential random graph (p^*) models for social networks," *Soc. Networks*, 2007.

- [324] C. Grabow, S. Grosskinsky, J. Kurths, and M. Timme, “Collective relaxation dynamics of small-world networks,” *CoRR*, vol. abs/1507.04624, 2015.
- [325] J. Leskovec, D. Chakrabarti, J. M. Kleinberg, C. Faloutsos, and Z. Ghahramani, “Kron-
ecker graphs: An approach to modeling networks,” *J. Mach. Learn. Res.*, 2010.
- [326] S. Purohit, L. B. Holder, and G. Chin, “Temporal graph generation based on a dis-
tribution of temporal motifs,” in *Proceedings of the 14th International Workshop on
Mining and Learning with Graphs*, 2018.
- [327] A. Paranjape, A. R. Benson, and J. Leskovec, “Motifs in temporal networks,” in
*Proceedings of the Tenth ACM International Conference on Web Search and Data
Mining*, M. de Rijke, M. Shokouhi, A. Tomkins, and M. Zhang, Eds. ACM, 2017.
[Online]. Available: <https://doi.org/10.1145/3018661>
- [328] S. Kumar, X. Zhang, and J. Leskovec, “Predicting dynamic embedding trajectory
in temporal interaction networks,” in *Proceedings of the 25th ACM SIGKDD
International Conference on Knowledge Discovery and Data Mining*, A. Teredesai,
V. Kumar, Y. Li, R. Rosales, E. Terzi, and G. Karypis, Eds. ACM, 2019. [Online].
Available: <https://doi.org/10.1145/3292500>
- [329] D. V. Buonomano and M. M. Merzenich, “Temporal information transformed into a
spatial code by a neural network with realistic properties,” *Science*, 1995.
- [330] G. R. Terrell and D. W. Scott, “Variable kernel density estimation,” *The Annals of
Statistics*, 1992.
- [331] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser,
and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information
Processing Systems*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach,
R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017. [Online]. Available:
<https://proceedings.neurips.cc/paper/2017>
- [332] P. Panzarasa, T. Opsahl, and K. M. Carley, “Patterns and dynamics of users’ behav-
ior and interaction: Network analysis of an online community,” *J. Assoc. Inf. Sci.
Technol.*, 2009.
- [333] S. Kumar, F. Spezzano, V. S. Subrahmanian, and C. Faloutsos, “Edge
weight prediction in weighted signed networks,” in *IEEE 16th International
Conference on Data Mining*, F. Bonchi, J. Domingo-Ferrer, R. Baeza-Yates,
Z. Zhou, and X. Wu, Eds. IEEE Computer Society, 2016. [Online]. Available:
<https://ieeexplore.ieee.org/xpl/conhome/7837023/proceeding>
- [334] T. N. Kipf and M. Welling, “Variational graph auto-encoders,” *CoRR*, vol.
abs/1611.07308, 2016.

- [335] Y. Zuo, G. Liu, H. Lin, J. Guo, X. Hu, and J. Wu, “Embedding temporal network via neighborhood formation,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Y. Guo and F. Farooq, Eds. ACM, 2018. [Online]. Available: <https://doi.org/10.1145/3219819>
- [336] P. Goyal, S. R. Chhetri, and A. Canedo, “dyngraph2vec: Capturing network dynamics using dynamic graph representation learning,” 2020.
- [337] D. Chakrabarti and C. Faloutsos, “Graph mining: Laws, generators, and algorithms,” *ACM Comput. Surv.*, 2006.
- [338] L. Akoglu, M. McGlohon, and C. Faloutsos, “RTM: laws and a recursive generator for weighted time-evolving graphs,” in *ICDM*. IEEE Computer Society, 2008.
- [339] J. M. Stokes, K. Yang, K. Swanson, W. Jin, A. Cubillos-Ruiz, N. M. Donghia, C. R. MacNair, S. French, L. A. Carfrae, Z. Bloom-Ackerman et al., “A deep learning approach to antibiotic discovery,” *Cell*, 2020.
- [340] W. Jin, R. Barzilay, and T. S. Jaakkola, “Junction tree variational autoencoder for molecular graph generation,” in *ICML*, 2018.
- [341] M. Kim and J. Leskovec, “Multiplicative attribute graph model of real-world networks,” *Internet Math.*, 2012.
- [342] L. Zhao, B. B. II, T. I. Netoff, and D. Q. Nykamp, “Synchronization from second order network connectivity statistics,” *Frontiers Comput. Neurosci.*, 2011.
- [343] M. Simonovsky and N. Komodakis, “Graphvae: Towards generation of small graphs using variational autoencoders,” in *Artificial Neural Networks and Machine Learning*, ser. Lecture Notes in Computer Science, V. Kurková, Y. Manolopoulos, B. Hammer, L. S. Iliadis, and I. Maglogiannis, Eds., vol. 11139. Springer, 2018. [Online]. Available: <https://doi.org/10.1007/978-3-030-01418-6>
- [344] X. Guo and L. Zhao, “A systematic survey on deep generative models for graph generation,” *arXiv preprint arXiv:2007.06686*, 2020.
- [345] Y. Li, O. Vinyals, C. Dyer, R. Pascanu, and P. W. Battaglia, “Learning deep generative models of graphs,” *CoRR*, 2018.
- [346] A. Grover, A. Zweig, and S. Ermon, “Graphite: Iterative generative modeling of graphs,” in *International conference on machine learning*. PMLR, 2019, pp. 2434–2444.
- [347] N. Goyal, H. V. Jain, and S. Ranu, “Graphgen: a scalable approach to domain-agnostic labeled graph generation,” in *Proceedings of The Web Conference 2020*, 2020, pp. 1253–1263.

- [348] R. Harrison and M. Thomas, “Identity in online communities: Social networking sites and language learning,” *International Journal of Emerging Technologies and Society*, 2009.
- [349] B. Wellman, “The network community: An introduction,” *Networks in the global village*, 1999.
- [350] P. Gajane and M. Pechenizkiy, “On formalizing fairness in prediction with machine learning,” *arXiv preprint arXiv:1710.03184*, 2017.
- [351] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, “A survey on bias and fairness in machine learning,” *CoRR*, 2019.
- [352] T. B. Hashimoto, M. Srivastava, H. Namkoong, and P. Liang, “Fairness without demographics in repeated loss minimization,” in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, ser. Proceedings of Machine Learning Research, J. G. Dy and A. Krause, Eds., vol. 80. PMLR, 2018. [Online]. Available: <http://proceedings.mlr.press/v80/hashimoto18a.html> pp. 1934–1943.
- [353] M. Ye, X. Liu, and W. Lee, “Exploring social influence for recommendation: a generative model approach,” in *SIGIR*. ACM, 2012.
- [354] R. S. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork, “Learning fair representations,” in *ICML*. JMLR.org, 2013.
- [355] T. Bolukbasi, K. Chang, J. Y. Zou, V. Saligrama, and A. T. Kalai, “Man is to computer programmer as woman is to homemaker? debiasing word embeddings,” in *NeurIPS*, 2016.
- [356] B. H. Zhang, B. Lemoine, and M. Mitchell, “Mitigating unwanted biases with adversarial learning,” in *AIES*, 2018.
- [357] A. Kobren, B. Saha, and A. McCallum, “Paper matching with local fairness constraints,” in *SIGKDD*, 2019.
- [358] A. J. Bose and W. L. Hamilton, “Compositional fairness constraints for graph embeddings,” in *ICML*, 2019.
- [359] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, “Recurrent neural network based language model,” in *INTERSPEECH*. ISCA, 2010.
- [360] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, 1997.
- [361] A. Bose and W. Hamilton, “Compositional fairness constraints for graph embeddings,” in *International Conference on Machine Learning*. PMLR, 2019, pp. 715–724.

- [362] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, J. Burstein, C. Doran, and T. Solorio, Eds. Association for Computational Linguistics, 2019. [Online]. Available: <https://aclanthology.org/volumes/N19-1/>
- [363] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, and Q. V. Le, “Xlnet: Generalized autoregressive pretraining for language understanding,” in *Advances in Neural Information Processing Systems*, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. B. Fox, and R. Garnett, Eds., 2019. [Online]. Available: <https://proceedings.neurips.cc/paper/2019>
- [364] S. Apers, “Expansion testing using quantum fast-forwarding and seed sets,” *CoRR*, 2019.
- [365] F. Wang and C. Zhang, “Label propagation through linear neighborhoods,” *TKDE*, 2007.
- [366] J. Ugander and L. Backstrom, “Balanced label propagation for partitioning massive graphs,” in *WSDM*, S. Leonardi, A. Panconesi, P. Ferragina, and A. Gionis, Eds. ACM, 2013.
- [367] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” in *1st International Conference on Learning Representations*, Y. Bengio and Y. LeCun, Eds., 2013. [Online]. Available: <https://openreview.net/group?id=ICLR.cc/2013>
- [368] K. Ding, J. Li, and H. Liu, “Interactive anomaly detection on attributed networks,” in *WSDM*, 2019.
- [369] W. Zhu, D. Tao, X. Cheng, P. Cui, E. A. Rundensteiner, D. Carmel, Q. He, and J. X. Yu, Eds., *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*. ACM, 2019. [Online]. Available: <https://dl.acm.org/citation.cfm?id=3357384>
- [370] Y. Huang, I. King, T. Liu, and M. van Steen, Eds., *WWW ’20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*. ACM / IW3C2, 2020. [Online]. Available: <https://doi.org/10.1145/3366423>
- [371] Y. Guo and F. Farooq, Eds., *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*. ACM, 2018. [Online]. Available: <https://doi.org/10.1145/3219819>