Trustworthiness, Diversity and Inference in Recommendation Systems

by

Cheng Chen
B.Sc., Beijing University of Posts and Telecommunications, 2010
M.Sc., University of Victoria, 2012

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Computer Science

Trustworthiness, Diversity and Inference in Recommendation Systems

by

Cheng Chen
B.Sc., Beijing University of Posts and Telecommunications, 2010
M.Sc., University of Victoria, 2012

Supervisory Committee

Dr. Kui Wu, Co-Supervisor
(Department of Computer Science)

Dr. Venkatesh Srinivasan, Co-Supervisor
(Department of Computer Science)

Dr. Alex Thomo, Departmental Member
(Department of Computer Science)

Dr. Hong-Chuan Yang, Outside Member
(Department of Electrical and Computer Engineering)

Supervisory Committee

Dr. Kui Wu, Co-Supervisor
(Department of Computer Science)

Dr. Venkatesh Srinivasan, Co-Supervisor
(Department of Computer Science)

Dr. Alex Thomo, Departmental Member
(Department of Computer Science)

Dr. Hong-Chuan Yang, Outside Member
(Department of Electrical and Computer Engineering)

ABSTRACT

Recommendation systems are information filtering systems that help users effectively and efficiently explore large amount of information and identify items of interest. Accurate predictions of users' interests improve user satisfaction and are beneficial to business or service providers. Researchers have been making tremendous efforts to improve the accuracy of recommendations. Emerging trends of technologies and application scenarios, however, lead to challenges other than accuracy for recommendation systems. Three new challenges include: (1) opinion spam results in untrustworthy content and makes recommendations deceptive; (2) users prefer diversified content; (3) in some applications user behavior data may not be available to infer users' preference.

This thesis tackles the above challenges. We identify features of untrustworthy commercial campaigns on a question and answer website, and adopt machine learning-based techniques to implement an adaptive detection system which automatically

detects commercial campaigns. We incorporate diversity requirements into a classic theoretical model and develop efficient algorithms with performance guarantees. We propose a novel and robust approach to infer user preference profile from recommendations using copula models. The proposed approach can offer in-depth business intelligence for physical stores that depend on Wi-Fi hotspots for mobile advertisement.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

The four-year journey of pursuing this PhD at the University of Victoria is a challenging but fascinating experience in my life. It is a great pleasure for me to express my gratitude to many people who made this thesis possible. I would like to sincerely thank:

my supervisor of six years, Dr. Kui Wu, for being a superb mentor to help me navigate through my entire graduate studies. After I finished the master thesis, you encouraged me to pursue this PhD and provided great amount of support. You showed me how to identify novel scientific problems and how to formulate the ideas and improve the writing. You always trust me and the work I have done, and believe that I can overcome whatever difficulties as long as I give all my effort.

my co-supervisor of six years, Dr. Venkatesh Srinivasan, for the great amount of guidance from a more theoretical perspective. You help me sharpen my theoretical skills by formulating research problems in rigorous mathematical frameworks, verifying symbols and proofs, which are fundamental throughout this thesis. Whenever I am in doubt, you are also very supportive to make me calm and sometimes share your personal experience.

my supervisory committee member, Dr. Alex Thomo, for introducing me to the world of data mining and those insightful discussions regarding manuscripts. You always respect thoughts I have and help me identify the most effective steps that should be taken to approach the problems. Your knowledge in data science helps me apply the theoretical works to important practical applications.

my wonderful wife, Fang, for the tremendous love of these years and the courage to be here. Together we share numerous memorable moments in this beautiful city. You also take good care of our daily lives, from organizing every little things to preparing delicious dinner. Your accompany helps me be fully devoted to the research.

my parents, Yanping and Shaohua, for supporting me to study abroad. My mother has always been there whenever I feel discouraging and depressed. My father often encourages me to talk about my research with him in plain language and sometimes even give me extra insights how to further improve my

work. Thank you both for always encouraging me to overcome the difficulties in my daily life.

DEDICATION

To the future of scientific research.

Chapter 1

Introduction

In this chapter we present a general overview of the thesis, describing the motivations, main research topics, giving an outline of the conducted study, and summarize contributions.

1.1 Motivation

In the last decade, the rapid development of the Internet, the mobile Internet and related technologies has brought fast growth in the number of Web services in different domains, including information retrieval, multimedia streaming, social networking, e-commerce, entertainment, etc. As of June 2016 [1], more than 3 billion people have become users of the Internet all over the world. Web services have greatly influenced their way of interacting with the world and have become an essential component of daily lives. In addition, smart devices (e.g., smartphones, tablets, and smart wearables) have become more prevalent than ever before. With wireless technologies, smart devices have provided people with ubiquitous access to the Internet, leading to an ever-growing ecosystem of the mobile Internet. Recent mobile marketing statistics shows that mobile users have outnumbered the desktop users worldwide and over 80% of mobile users access the Internet via smartphones [3].

While people enjoy the convenience of Web services, they are often overwhelmed by the numerous content delivered over the services, i.e., information overload. Based on recent estimations, Amazon sells hundreds of millions of products in the USA, several hundred hours of new videos are uploaded to Youtube per minute, and hundreds of millions of tweets are sent on Twitter per day. In such scenarios, exploring and

identifying valuable content of interest in an efficient way is imperative to both service providers and users.

Currently, recommendation systems (RSs) have become a vital and prevalent approach to address information overload. Given the interaction dataset of users and items (e.g., movie ratings), RSs discover hidden patterns, learn a model that characterizes user behaviors, and then provide users with suggested and often personalized products or services. Since the appearance of several earlier research works [120, 60], RSs have been an active area in both industry and academia.

While conventionally based on demographics and profiles, RSs have experienced remarkable development along with new trends of Web services. Nowadays, RSs are taking advantage of more diversified data, such as social information, localized information and personalized information. The new generation of RSs facilitated by the abundant information has broadened existing functionalities and has been widely used in almost all aspects of online activities to support numerous practical applications, such as *personalized* recommendations of books and other products by Amazon and eBay, nearby restaurants recommendations by Yelp, friends suggestions by Facebook, movies and TV shows by Netflix, news by Google, questions and answers by Quora and competitors matching by online gaming companies.

In addition to these online services, there is an emerging trend that RSs are gradually deployed to traditional business such as brick-and-mortar retailers (retailers with physical stores). RSs serve as a service provided together with the increasingly popular in-store wireless access points, such as Wi-Fi hotspots. Due to the prevalence of smart mobile devices (e.g., smartphones, tablets, and smart wearables), brick-and-mortar retailers are willing to invest money into the free access points to enrich customers' in-store experience. Nowadays, free Wi-Fi services are offered in many places, including cafes, airports, hotels, restaurants, cinemas, and shopping malls. Collecting usage data from the access point, in-store RSs can analyze customer behavior such as their geographic data and dwell times at different locations, and help retailers make informed marketing decisions and proactively engage customers by sending product recommendations. Currently, most existing industry solutions mine the collected data for basic customer demographics, presence analytics, Wi-Fi usage, and loyalty and engagement [4]. To obtain a better understanding of customer preference, new methodologies are needed for analyzing the unique data in this application scenario, such as data traffic. Compared to RSs of e-commerce, in-store RSs are still in its early stage and has not been fully studied in literature.

Along with the wide deployment, RS has also attracted an increasing level of interest in the academic community in the last two decades. Researchers have attempted to develop more advanced and sophisticated recommendation techniques that aim at accurate and timely recommendations. Collaborative filtering (CF) is nowadays a widely used technique by RSs to learn preference information from many users and then make predictions for a specific user. Since users' preference might change in various the surrounding environments, context-aware RS such as location-based, time-based and weather-based RSs appear to be new directions of future development of RSs and have the potential to be integrated into wearable devices, which will significantly benefit people's daily life.

Despite the massive efforts of pursuing advanced recommendation models and algorithms, there are still emerging open problems and controversial issues regarding various perspectives of recommendation that have not been fully studied and therefore inevitably limit the applicability and reliability of RS in practice. In this thesis, we identify particular challenges to RSs in trustworthiness, diversity and inference (of user preference profile). While each topic expands a large research space, we consider the following novel challenges, which arise from emerging trends and we believe are crucial for further advancement of RSs.

1.1.1 Trustworthiness

Recommendations should be trustworthy. In many applications such as product recommendation, content delivered by RSs generally relies on user generated data, which will be the ground truth input to RSs. Existing RSs may not work well in the presence of the so-called Internet water army, a large crowd of hidden paid posters who get paid to generate artificial content for commercial profits. Paid posters have become popular with the booming of crowd-sourcing marketing [135]. As confirmed in [135], crowd-sourcing systems such as Amazon's Mechanical Turk, Zhu Ba Jie (a similar Chinese crowd-sourcing site), have been broadly used for commercial campaigns. Due to the prevalent crowd-sourcing marketing strategy, huge amount of online information are generated for hype or commercial purposes in many domains. For example, reviews of a book or comments of a product might be written by so-called *paid posters*[30]. The content of their reviews does not need to be trustful, rather expressing attraction through exaggeration. Product recommendation based on these reviews will be misleading. Another example is the community question

and answer (CQA) portals where users can post and answer questions, such as Yahoo! Answers[①], Quora[②] and Baidu Zhidao[③]. Answers deliberately written by paid posters for product advertising purposes might contain doubtful information. Therefore, RSs trained on the malicious user generated data can be deceptive. Although large amount of efforts have been taken to improve the relevance of recommendations, it will be non-trivial for RSs to differentiate the trustworthy from malicious information without explicitly checking for validity of input data.

1.1.2 Diversity

RS should generate diverse recommendations while maintaining overall high user utility. Normally, items to be recommended can be classified into different groups, based on certain criteria, such as movies of similar genres and books of similar topics. The need of diversity arises from the recommendation situation where users generally have a broad and diverse range of interests with respect to item groups, and therefore a variety of recommendations is often desired. For book recommendation, a reader may not want all recommended books from the same subject but instead may prefer books of diverse subjects so that more interesting topics can be discovered. RS should allow a reader to constrain the number of books from the same subject. In other words, books from the same subject are in "conflict" with each other when being recommended to a reader, and the number of such conflicts should be below the reader's tolerance threshold. For RSs, however superior in terms of common recommendation metrics (e.g., accuracy), recommending only top items (determined by recommendation algorithms) without considering the conflict among them will result in lacking of diversity. This problem will be harmful to both users and service providers. On one hand, users might become overwhelmed by a few most popular items. On the other hand, service providers lose the chance to expose the large collection of available items and cannot make full use of a well-trained RS for potential profit. For an online service of millions of users, the balance between the overall user utility and diversity is therefore interesting and imperative. Existing works that focus on diversified recommendation generally lack a formal approach to quantify the diversity demand. The problem of formulating a theoretical framework that

[①] `https://answers.yahoo.com/`, June 2016
[②] `http://www.quora.com/`, June 2016
[③] `http://zhidao.baidu.com/`, June 2016

achieves overall high user utility of recommendation while satisfying various diversity requirements remains open and will be addressed in this thesis.

1.1.3 Inference of User Profiles

A user (preference) profile commonly indicates what information is of interest to this specific user. It is the result of the user modelling procedure, which is based on user-item interaction data. A critical problem for many applications is how to handle the sparsity of user behavior data when the access to user behavior of a particular domain of interest is limited. **Effective RSs should be able to infer user profiles in a robust way, when there is little ground truth about user behaviors.** Compared to online business (e.g., Amazon, eBay, Netflix, etc.), this problem is especially common in traditional business such as brick-and-mortar retailers. For online business, due to its popularity and accessibility, they can easily accumulate a large number of users and a large collection of user interactions with items, which will be used to model users' preferences, create detailed user profiles and therefore produce more accurate recommendations. Brick-and-mortar business, however, can collect very limited amount of information about their customers. Unlike online merchants, brick-and-mortar retailers often have limited physical space, limited choices and less connection to the customers. While a customer may have many valuable behaviors online (e.g., online purchases on all categories, movie ratings, etc.), this data is hidden to retailers. Retailers generally can only collect customers' interaction with their limited products and do not have a broad understanding of their customers. Lacking of sufficient ground truth data is a major bottleneck towards an effective RS service for retailers. How to leverage customers' hidden online behavior data is an open challenge and has not been exploited in literature. Since user profiling is the core to RSs and is indispensable for blooming brick-and-mortar business, dedicated approaches are required to address this problem and is a topic of this thesis.

1.2 Research Goals

This thesis aims to contribute to all three novel challenges mentioned in the Section 1.1. Each challenge is closely related to the most recent application trends of RSs and presents many opportunities for further innovation and extension to RSs of the state-of-the-art. In order to conduct studies from both empirical and theoretical

perspectives, we pursue the following research goals with respect to trustworthiness, diversity and inference in this thesis.

- Trustworthiness: **Identify the trustworthiness problem in a particular domain and develop a general methodology that explicitly filters out malicious data.** While most of current work is focused on the accuracy of recommendation algorithms, the trustworthiness problem has been overlooked. For services that depend on user generated content (UGC), malicious information used as the input to RSs can lead to deceptive recommendations. Explicit identification and removal of the malicious data is a key step for validation of ground truth data for RSs.

- Diversity: **Formulate the diversity requirements in the problem of utility maximization and develop algorithms with theoretical analysis.** There is a lack of theoretical study about utility maximization with flexible diversity in the literature. Furthermore, how to explicit specify diversity and how this specification affects the overall utility are not clear. An in-depth investigation of this problem is conducted in this thesis.

- Profile Inference: **For the typical application scenario of brick-and-mortar business where RSs face the data sparsity problem, develop a novel approach that takes advantage of emerging techniques to not only enrich the data for modelling, but also creatively infer user preference profiles.** Equipped with modern techniques such as Wi-Fi hotspots, traditional brick-and-mortar business represents a unique application scenario of RSs. Rethinking the user profiling procedure and a dedicated and creative profiling approach are keys for success of RSs deployed in this scenario.

1.3 Contributions

The work of this thesis contributes to the research of trustworthiness, diversity and inference in RSs.

Trustworthiness

In Chapter 2, we identify the trustworthiness problem caused by paid posters in the community question and answer (CQA) domain and make the following contributions:

- We discover that the behavioral features of paid posters are different in CQA forums when compared to other types of forums such as microblog and news reports. We identify the special features of paid posters in CQA forums that are useful for effective detection.
- Based on the identified special features, we design a supervised learning-based detection method and assigns credibility scores to each of the best answers by using semantic analysis and user features, such as users' history data.
- We implement an adaptive detection system which automatically analyzes the hidden patterns of commercial campaigns and raises alarms instantaneously to end users whenever a potential commercial campaign is detected. The system is adaptive and accommodates new evidence gathered by the detection algorithm over time.

Diversity

In Chapter 3, we study the utility maximization with diversity problem in the e-commerce application scenario, where items or sellers are often recommended to users for user utility maximization while capturing diversity across specified conflicts among entities of the same type (e.g., users/items/sellers). We formally define this problem by extending a classic graph-theoretic model. More specifically, we make the following contributions:

- We model the utility maximization recommendation as a weighted bipartite b-matching problem (WBM). We initiate the study of natural extensions of WBM for modeling the conflict-aware version (i.e., CA-WBM). In terms of the theoretical model, the problem we tackle is how to maximize the total weight when matching vertices are under both degree and conflict constraints.
- We present a general formulation of CA-WBM that directly models both the degree constraint on each vertex and conflicts between two vertices. We model it using semidefinite programming (SDP) and integer linear programming (ILP). To the best of our knowledge, this explicit modeling is completely new.
- We prove that CA-WBM is NP-hard and we present greedy and linear programming (LP) based algorithms that are scalable and close to optimal. We also provide a randomized algorithm that solves the online version of CA-WBM.

- We provide an extensive experimental evaluation on synthetic and real-world datasets of the e-commerce application scenario, validating our claims of scalability and optimality.

In CA-WBM, one can arbitrarily assign conflicts between two entities (e.g., users/items/sellers). This makes the model hard to solve and is often unnecessarily general, because in many applications the conflicts are transitive within *groups* (e.g., households, genres, topics, temporal ranges), i.e., the conflicts arrange in cliques. In Chapter 4, we propose two new models for diversity requirements that are based on the assumption that the conflicts arrange in cliques and make the following contributions:

- We prove that the CA-WBM problem [31] is hard to approximate by reducing from Maximum Weight Independent Set (Section 4.2).
- We introduce group-aware WBM subject to *degree constraints* (GA-WBM-D), together with a polynomial-time, exact linear programming algorithm and a scalable, 2-approximate greedy algorithm (Section 4.3).
- We introduce group-aware WBM subject to *budget ceilings* (GA-WBM-B) and prove it is NP-hard. We give a greedy algorithm using a k-extendible system to guarantee 3-approximate solutions (Section 4.4).
- We conduct an extensive experimental evaluation on e-commerce data showing that the linear programs and the greedy algorithms return excellent results on small inputs, and the greedy algorithms scale to bipartite graphs with over eleven million edges (Section 4.5).

Profile Inference

In Chapter 5, we introduce and study a novel value-added service, Recommendation to Profile Inference (Rec2PI), for Wi-Fi data mining. Rec2PI utilizes a new source of data, i.e., recommendations pushed to a user in a certain domain (e.g., books or movies), to infer the user's preference profile in that domain. More specifically, we make the following contributions:

- We initiate the study of a novel value-added service arising in Wi-Fi data mining: Without knowing the algorithms and the dataset used by a third-party RS, how can we infer users' behavior based on the recommendations from the third-party

RS? This is a reversed procedure of general RSs. To the best of our knowledge, we are the first to investigate this reversed learning problem.

- We propose a general framework, Rec2PI, that builds probabilistic inference models based on open datasets. In addition, we adopt a novel approach that incorporates copulas, a powerful statistical tool for dependence modeling, into the inference procedure.
- We perform extensive experimental evaluation on real-world datasets. We show that the performance of popular approaches in RSs, such as latent factor models (LFMs), is not stable when solving the reversed learning problem, i.e., the results exhibit high variance. In contrast, our copula-based solution is not only accurate but also much more stable.

1.4 Publications

Trustworthiness

- Cheng Chen, Kui Wu, Venkatesh Srinivasan, and R. Bharadwaj. "The best answers? think twice: online detection of commercial campaigns in the CQA forums," in *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 458-465, August 2013.
- Cheng Chen, Kui Wu, Venkatesh Srinivasan, R. Kesav Bharadwaj. "The Best Answers? Think Twice: Identifying Commercial Campagins in the CQA Forums," *Springer Journal of Computer Science and Technology*, vol. 30, no. 4, pp. 810-828, July 2015.

Diversity

- Cheng Chen, Lan Zheng, Venkatesh Srinivasan, Alex Thomo, Kui Wu and Anthony Sukow, "Conflict-Aware Weighted Bipartite B-Matching and Its Application to e-commerce," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 6, pp. 1475-1488, June 1 2016.
- Cheng Chen, Sean Chester, Venkatesh Srinivasan, Kui Wu and Alex Thomo, "Group-Aware Weighted Bipartite b-Matching," in *Proceedings of the 25th ACM Conference on Information and Knowledge Management*, October 2016.

Profile Inference

- Cheng Chen, Fang Dong, Kui Wu, Venkatesh Srinivasan and Alex Thomo, "From Recommendation to Profile Inference (Rec2PI): A Value-added Service to Wi-Fi Data Mining," in *Proceedings of the 25th ACM Conference on Information and Knowledge Management*, October 2016.

Chapter 2

Commercial Campaigns Detection in the Community Question and Answer Websites

2.1 Introduction

As a popular type of Web 2.0 websites replied on user generated content (UGC), CQA websites allow users to post and answer questions, such as Yahoo! Answers[①], Naver[②] and Baidu Zhidao[③]. Some CQA websites like Quora[④] attract users by offering professional answers, most of which come from verified people in reality. These websites gain popularity and trust by providing a sense of interaction between the questioner and the masses. With millions of archived Q&A sessions, CQA forums have become a major source of advice for many Internet users.

As a large knowledge base of crowds, the archived Q&A sessions have been used for automatic question answering and recommendation. Nevertheless, the quality of user-generated content in the Q&A sessions varies drastically. For instance, some answers do not match the questions and even contain spam and rude words. In recent years, tremendous efforts have been made to locate better answers and remove spam from the archived questions and answers resource. Techniques such as analysis of

[①] `https://answers.yahoo.com/`, June 2016
[②] `http://www.naver.com/`, June 2016
[③] `http://zhidao.baidu.com/`, June 2016
[④] `http://www.quora.com/`, June 2016

text, user-question-answer's link relationship, and user feedback features have been used in tools like PageRank to identify high-quality web pages [70, 73, 9].

Existing techniques, however, may not work well in the presence of the so-called Internet water army, a large crowd of hidden posters who get paid to generate artificial content in the social media for commercial profits. Paid posters have become popular with the booming of crowd-sourcing marketing. As confirmed in [135], crowd-sourcing systems such as Amazon's Mechanical Turk, Zhu Ba Jie (a similar Chinese crowd-sourcing site), have been broadly used for commercial campaigns. Due to their popularity, the CQA websites have become the targets of those campaigns that create untruthful Q&A sessions for commercial purpose. Consider the following example:

Question: I tried several methods to lose weight but all failed. What should I do? Please give me some advice!

Best answer: Don't worry, I have experienced the same pain as you. Firstly, you have to keep a healthy diet. Be careful about the nutrition in your food and never eat fast food. Secondly, don't sit too long in front of a computer. Finally, perform physical exercise everyday. What's more, you can also try a product named X. This product cotains ingredients such as ... and can help you lose weight without any risks.

The above Q&A session is actually generated by paid posters. The answer provides very practical advice at first and then gives suggestion on the product which needs to be promoted. The practical advice part is to earn the trust of the users. We have observed that fake answers generated by paid posters are often long enough and quite relevant to the questions, and some paid posters involved in the fake Q&A sessions are ranked high according to the website's reputation system.

Based on textual similarities, previous work [92, 20, 22] is likely to treat the above answer as of high quality due to the high relevance of textual features between the answer and the question content. As a result, the output may contain commercial spam, resulting in a credibility problem. Therefore, additional strategies, such as writing templates, public calls for commercial campaigns, and a poster's track reputation, should be integrated for the effective detection of paid posters. Furthermore, most existing work relies on offline analysis, while end users demand for instant help and should be warned of potential commercial campaigns when they browse a CQA forum. The call for a real-time response system that can detect potentially fake Q&A sessions on the fly is strong.

In this chapter, We tackle the trustworthiness challenges by providing a comprehensive study of machine learning-based methods to detect commercial campaigns and designing an adaptive detection system tailored specifically for CQA websites.

2.2 Data Collection and Labeling

2.2.1 Data Collection

We collect ground truth data for commercial campaigns detection on a popular CQA website of Chinese language, Baidu Zhidao. Users who register on Baidu Zhidao participate in various Q&A sessions, either as question askers or repliers. Since we know that paid posters who accept missions from crowd-sourcing sites create a variety of Q&A sessions on the site for product propaganda, the collecting process can be targeted directly to the product campaigns. In addition, since the readers tend to pay more attention to the best answers and also due to the manner in which online paid posters are supposed to work, we only collected the best answers and ignored other ones. This is to avoid collecting a large amount of irrelevant information for this study.

In order to collect campaign Q&A sessions, we first visited the crowd-sourcing websites, where the paid posters apply for campaign tasks and get paid, as stated in Section 2.1. From the campaigns calling for paid posters, we selected 11 closed requests because the paid posters who worked for the 11 products had finished the tasks. We extracted keywords for the 11 products and searched for Q&A sessions with them on Baidu Zhidao. We used a crawler to visit and download the web pages associated with searching result. These sessions included not only the campaign sessions, but also normal sessions containing the keywords. After parsing all the collected web pages, we obtained a group of target users, including both paid posters and normal users, as well as the links to the users' homepages hosted by Baidu Zhidao.

By following the users' homepages, we could find useful information for our research. For example, a user's homepage provides the Q&A sessions where this user posted his/her answers (the question answering records). The question-answer history provides a good knowledge on the multiple campaigns that a potential paid poster might have been involved. Having obtained the initial dataset of IDs and links, we then visited each user's homepage, and retrieved every Q&A session that the user participated in. We only collected the closed Q&A sessions (i.e., the best answer de-

termined). A closed Q&A session implies that users can no longer post new answers to the question, but they can click the "Like" button to support the posted answers, including the best answer and other answers.

From those Q&A sessions, we finally extracted information used in our analysis. The recorded information from those web pages includes questioner ID, answer ID, time, title, question content, answer content, user feedbacks (visited times, ratings). For text information (Q&A title and question/answer content), we have removed stopwords from the raw data.

From the Q&A website, *Baidu Zhidao*, we collected 6462 users' question-answer history records accumulated during a three-month period from October to December in 2011. For each user, we built a list of history information, showing the question, answer, participated user IDs, and other features. Associated with the 6462 user IDs, we have $75,200$ Q&A sessions in total, all having the best answer.

In the following, we describe a solicitation example of Q&A campaign.

A Solicitation Example

Mission title: a_brand's_name (brand A): Baidu Zhidao, 5 RMB (0.8 USD) per valid Q&A.

The title indicates that this is a Q&A campaign mission for brand A, conducted on Baidu Zhidao CQA website. The payment is 5 RMB (0.8 USD) for each approved Q&A session.

General requirements:

1. Normally, it takes three days to complete a Q&A campaign; Day 1: Post the question. Day 2: Use a different IP address and login account to answer the question. Day 3: Select the answer as the best one. The Q&A will be invalid (you will not get paid) if the answer is not selected as the best one, or the best answer is deleted within 72 hours.

2. One account can only be used to post/answer one question regarding the same solicitation.

3. If you answer a question posted by yourself, you must change your IP address and the account.

4. You must mention brand A in your answer.

5. Once you complete a Q&A, you need to send the link to the mission supervisor for evaluation. You will get paid once it is approved.

Keywords: detergent for car washes, car washing plant

Question template: What is a good detergent for car washes?

Answer template: There are many different detergents, such as brand A, brand B and brand C. The detergent of brand A is better because it does not need wiping and it takes only seven minutes to clean a car. Of course, you will need some washing equipment. If you want to open a car washing plant, I highly recommend brand A.

The question by paid poster 1: I recently bought a car. It gets dirty after some driving. I would like to know which detergent works well?

The answer by paid poster 1: Cleaning is necessary to keep a vehicle in good shape. Important electrical connectors should be protected before cleaning. Then you can use the detergent of brand A to wash every individual part. Do not use high pressure washer.

The question by paid poster 2: Which detergent is good for car wash?

The answer by paid poster 2: I always use the detergent of brand A in my car wash plant because customers are very satisfied with brand A. You do not need wiping. With the help of washing equipment, you can finish washing a vehicle in seven minutes.

Note that this example only shows one of many possible working patterns of campaign Q&A. In practice, paid posters do not have to post questions by themselves. They could find related questions posted by regular questioner and answer them according to campaign templates.

2.2.2 Manual Data Labeling

To get a sample dataset for feature analysis, campaign sessions should be differentiated from the normal ones. By reading the best answers and *cross-checking* the Q&A templates from the crowd-sourcing websites such as Zhubajie[⑤] and Tiancaicheng[⑥], we manually label the Q&A sessions in the dataset. We summarize the applied techniques below:

1. Since we have collected a list of 11 products which were hyped in the Baidu Zhidao, we could compare the Q&A content with the campaign templates. If the

[⑤] `http://www.zhubajie.com/`, June 2016

[⑥] `http://www.tiancaicheng.com/`, June 2016

product's name is in the 11 initial samples and the contents match the templates, such as the descriptive words and the organized pattern of sentences, we labeled it as a campaign Q&A session. We stress that there is difference between our work and related research which needs to judge the quality of answers. The evaluation of quality of answers is usually based on question-answer relevance, length of the texts, grammar correctness, politeness, and so on. To obtain a reliable dataset, researchers often rely on multiple assessors and are faced with the difficulty of reaching an agreement among the multiple evaluation results. Our labeling method differs from the above and largely avoids the annotation difficulty, because we know exactly the name of the hyped product and how paid posters would write the Q&A sessions.

2. When we encountered new products not in the list of 11 initial samples, we recorded the product's name and searched it in the crowd-sourcing websites. If we found the template of this product, we use the above method to compare their contents.

3. If a new product is listed in the campaign websites but the template is not available, we followed some special features normally found in Email spam to make a decision. For example, a spam may use different fonts to write the telephone numbers and insert special characters between the product's name. This type of operations is usually used to escape detection by the filter system. We labeled the session as campaign if the product's name is in a campaign list and the best answer has special features similar to Email spam.

4. If we could not find the new product in the campaign websites, we then tried to identify potential templates used in the same category of products and special features obvious in an Email spam. If none of those could be identified, we labeled the session as a normal session.

We have labeled 4998 samples in our dataset. Among these, 2147 samples are campaign Q&A sessions and the other 2851 samples are normal ones. The sample size is large enough for our current study. Since we selected 11 campaigns, which were posted on the crowdsourcing websites, as the seeds of our crawler, and we further encountered new products involved in campaigns, the proportion of campaign sessions is relatively high in the dataset.

When we manually labeled our datasets, we carefully read the contents of a user's post. The meaning can be understood by human but is hard to use in machine learning based classification. Even with the above template based labeling method, it is not easy to write an algorithm to automatically identify a campaign session because a poster may re-phrase the template in his/her own words. Due to these reasons, we need to search for statistical features that can be effectively used towards building a detection system.

2.3 Analysis of Statistical Features

2.3.1 Insufficiency of Existing Statistical Features

Here, we demonstrate the limitations of the features used in our previous work [30] on detection of Internet water army in news report towards addressing the problem we study in this chapter.

Interval Post Time

In [105], Arjun *et al.* defined several spamming indicators for modelling the behaviour of fake review writers. They found that spammers of a spam group tend to post reviews during a short time interval. This feature has been shown to be a good indicator to detect Internet water army in news report websites [30].

In our work, we consider two timestamps for a Q&A session: One is the time when the questioner posts the question topic (the ask time), and the other one is the time when the best answer is posted by a replier (the best answer posted time). We define *interval post time* as the latter timestamp minus the former one.

In Figure 2.1, we show the approximated probability distribution of interval post time with dot-dashed lines for campaign sessions and solid lines for non-campaign sessions. The x-axis is drawn by log scale.

From the figure, we find it difficult to tell the difference between campaign and non-campaign Q&A sessions. Two reasons may contribute to the above phenomenon. There are many normal users who spend much time on the Q&A website and try to post answers to *open* questions, especially those questions associated with some *rewards points*. These people are known as *bounty hunters*. Most bounty hunters post very good answers because they want to get more rewards points. On the other hand, online paid posters, before they post and choose the best answer, normally wait for

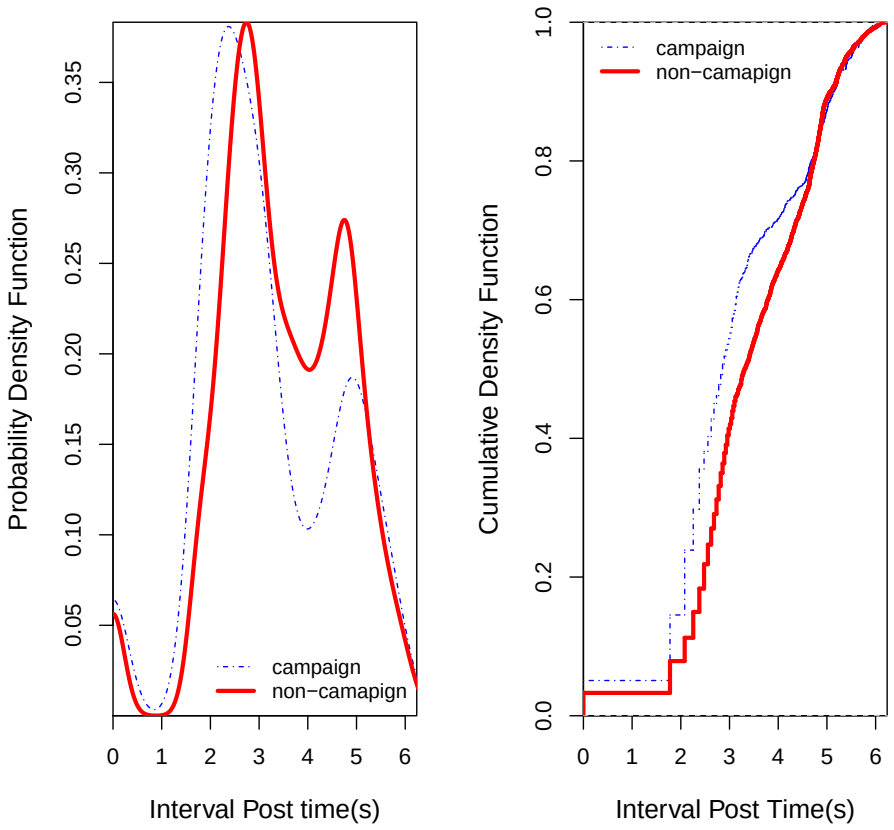


Figure 2.1: The PDF and CDF of the interval post time

some random time for other answers appearing in the session. This is to give readers a fake impression that the best answer is selected among many answers. While paid posters try to finish a job as quickly as possible in news review websites [30], the same behaviour does not exist here.

Number of Other Answers

Before the question is closed, users can post their own answers. This variable counts the number of answers other than the best one. Intuitively, if the paid posters create the sessions themselves, they may not have patience to wait for more replies. They could close the sessions and get paid as soon as possible. To test this conjecture, we show the probability distribution of this feature for campaign sessions and normal sessions in Figure 2.2 .

Similar to the interval post time, the number of other answers does not indicate much difference for the two types of Q&A sessions. This invalidates the above con-

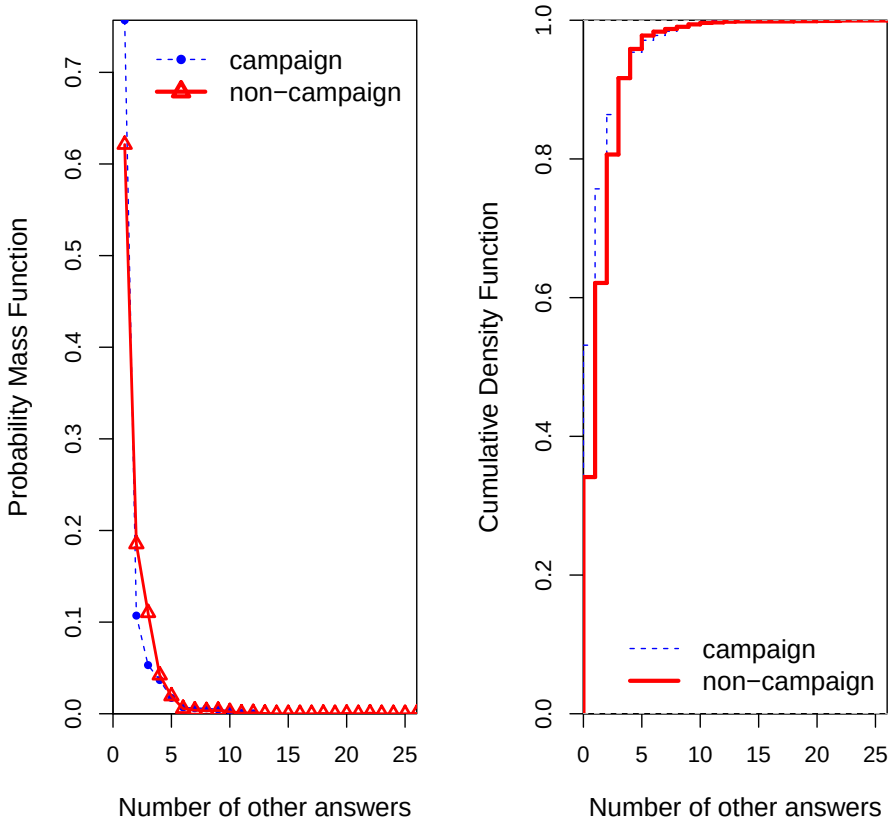


Figure 2.2: The PMF and CDF of the number of other answers

jecture and we do not consider it as a good feature for the detection of paid posters in CQA portals.

Number of Likes

Similar to the "Like" button in Facebook, if other readers find the best answer to be helpful, they may click the "like" button. The number on the button indicates the total number of clicks. Intuitively, this feature represents user's feedback and should be helpful in identifying trustful answers. The more "likes" an answer receives, the more likely it is a good answer. However, as shown in Figure 2.3, this is not a reliable feature. This is because the paid posters could click the button themselves and even use different user IDs to click multiple times. This behavior is also confirmed in [21] as the "vote spam attack".

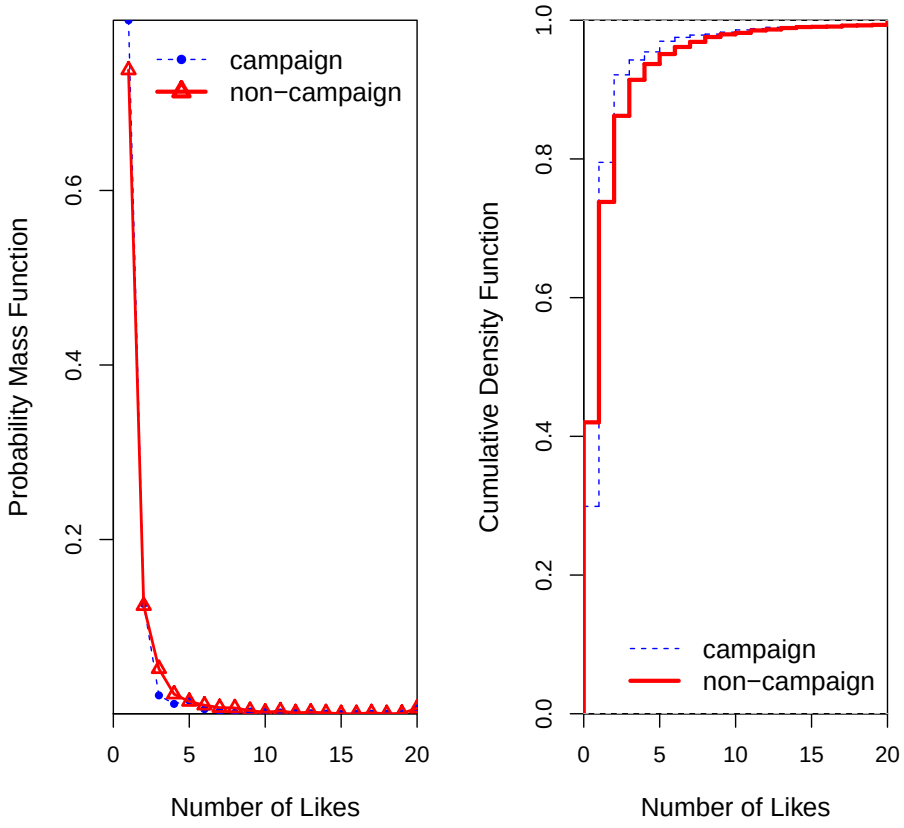


Figure 2.3: The PMF and CDF of the number of likes

Relevance between Questions and The Best Answers

This feature is extensively used before in identifying high-quality answers [92, 20, 9, 22]. The previous work is usually based on following assumptions:

1. Semantically high relevance between questions and answers indicates high quality.

2. Selected best answers should have higher quality than other answers.

The above assumptions are risky for the detection of potential campaigns created by paid posters. In commercial campaigns, answers with *high-quality* are rather misleading and would beat the retrieval mechanism. Many of the answers are well-organized and highly related to the questions. In this sense, a "high-quality" answer does not necessarily mean trustworthiness. Thus, we do not consider the relevance measure in our work.

2.3.2 Special Features for CQA Portals

The limitations of existing statistical features shown above lead us to look for new features specific to users in CQA websites.

Spam Grade of Questioner ID (SGqID)

It indicates whether the questioner tends to ask campaign questions. A paid poster may use multiple IDs (for questioning and answering, respectively) to complete a Q&A campaign. Therefore, a questioner ID which appears in many malicious Q&A sessions is more likely to be associated with a paid poster. For a given questioner ID (qID), we calculate the ratio of the number of campaign sessions and the total number of sessions in which the user has participated, as shown in Equation 2.1.

$$\text{SGqID} = \frac{q_1}{q_0 + q_1} \tag{2.1}$$

where q_0 and q_1 are the number of non-campaign and campaign sessions where the user appears as the questioner, respectively. To avoid 0 probability, we assign 0.5 to q_1 when $q_1 = 0$. This is a technique known as Laplace correction or Laplace estimator. It has been widely adopted to avoid zero frequency problem, which arises when an entity (qID in this case) does not occur in one of the classes (campaign or non-campaign sessions). Laplace correction assigns the entity of each class a fixed pseudocount of a certain number. 0.5 or 1 are commonly used as the pseudocount in practice. In addition, if the system does not have enough information for a certain user (i.e., the denominator is less than 5), we set its SGqID value to 0.5. This decision follows the Maximum Entropy Principle [76], i.e., we should "make use of all the information that is given and scrupulously avoid making assumptions about information that is not available." We refer to these two techniques (Laplace correction and Maximum Entropy Principle) as "data twist" in later sections.

Spam Grade of Answerer ID (SGaID)

It indicates whether the best answer poster tends to write campaign answers. The intuition behind is similar to that of SGqID. For a given answerer ID (aID), we calculate the ratio of the number of campaign sessions and the total number of sessions

in which the user has participated, as shown in Equation 2.2.

$$\text{SGaID} = \frac{a_1}{a_0 + a_1} \tag{2.2}$$

where a_0 and a_1 are the number of non-campaign and campaign sessions the user appears as the poster of the best answers, respectively. Similar to SGqID, to avoid 0 probability, we specify 0.5 to a_1 when $a_1 = 0$. If the system does not record enough information, we set its SGaID value to 0.5.

Spam Grade of the Text (SGtext)

It indicates whether the collection of words in sessions associated with a user tends to be campaign specific. For a given mission of Q&A campaign, different paid posters may share the same template, which is provided by the mission supervisor. Therefore we can expect similar words or expressions in the postings. To calculate this feature, we need to perform statistical analysis over the words. Text information of a Q&A session consists of the title, the content of question, and the content of the best answer. We remove the duplicate words so that we can get a collection of distinct words (word_1, word_2, word_3, ... , word_n) for each Q&A session. For each word, we calculate *spam grade* which characterizes the property of the word, i.e., whether it is more campaign oriented or non-campaign oriented. Words with higher benchmark are more likely to imply hidden promotion behavior, i.e., they appear in many campaigns sessions but few normal ones. To get rid of the impact of different length, we take the average value over the summation of the benchmarks of all words as the spam grade of the whole text. For each word, the definition of spam grade is defined in Equation 2.3.

$$SGword_i = \log\left(\frac{N+1}{n_i+1}\right) \times \frac{s_i+1}{S+1} \tag{2.3}$$

where N and S are the total number of non-campaign and campaign sessions in the databases and n_i and s_i are the number of non-campaign and campaign sessions where the $word_i$ appears. The intuition behind this definition is to obtain a weighting scheme showing whether a word tends to be campaign specific, based on the fraction of campaign and non-campaign Q&A sessions that contain the word. The definition of the spam grade takes both non-campaign and campaign sessions into consideration. This definition achieves desired effects: (1) the spam grade is scaled up when the word occurs fewer times in non-campaign sessions and occurs in many campaign sessions;

(2) the spam grade is scaled down when the word occurs more in non-campaign sessions and occurs fewer in campaign sessions; (3) the spam grade is neutralized when the word frequently occurs (or rarely occurs) in both non-campaign and campaign sessions.

We apply "log" to avoid a large value in the equation. The term "+1" is used to normalize the result in case of zero counts. Then the calculation of the spam grade of text with L distinct words is shown in Equation 2.4.

$$\text{SGtext} = \frac{SGword_1 + SGword_2 + ... + SGword_L}{L} \tag{2.4}$$

2.4 Detection Method

In this section, we firstly select features that are useful for detection. Then we introduce a supervised learning approach, logistic regression, to calculate campaign scores (which indicate whether a Q&A session tends to be a campaign) for Q&A sessions using the selected features. Based on the scores, we can distinguish normal answers from campaign answers.

2.4.1 Feature Selection

We sort the 4998 labelled samples by the timestamp of best answers and take 3500 of them as training set (1183 commercial campaigns and 2317 normal Q&A sessions) and the remaining 1498 as test set (964 commercial campaigns and 534 normal Q&A sessions). Note that the split of the dataset is arbitrary so that we can obtain a general result.

Using the training set, we extracted the most important features by calculating the information gain ratio between the class label (campaign or non-campaign) and each feature we proposed in Section 2.3. Information gain ratio is defined by Equation 2.5.

$$\text{Gain Ratio} = \frac{H(Y) + H(X) - H(Y, X)}{H(X)} \tag{2.5}$$

The gain ratios for features are shown in Table 2.1.

As shown in Table 2.1, the spam grade features are more significant in terms of information gain ratio. Therefore, the three "SG" features will be used to build the classification model.

Feature	Gain Ratio
Interval post time	0.04428713
Number of other answers	0.01636462
Number of likes	0.04459128
SGqid	0.21631413
SGaid	0.30249365
SGtext	0.17179217

Table 2.1: Information Gain Ratios for Each Feature

Figure 2.4 exhibits the values using the three "SG" features on the entire dataset.

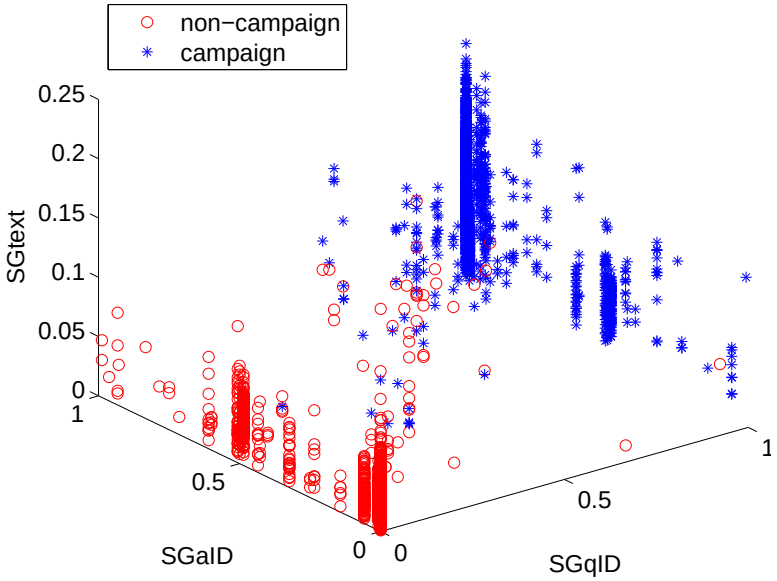


Figure 2.4: 4998 samples captured by SGqID, SGaID and SGtext

Through this figure, we can observe a clear gap between the campaign sessions and the non-campaign sessions. We can then apply the regression based approach to calculate the campaign score, which indicates whether a Q&A session tends to be a campaign.

2.4.2 The Algorithm

Figure 2.4 has already shown that the samples can be distinguished by the three selected features, SGqID, SGaID and SGtext. In order to get a score indicating whether a Q&A session is a potential commercial campaign or not, we apply logistic regression as the learning method. We can use it to calculate values of $P(Y = 1|\boldsymbol{x}, \boldsymbol{\theta})$ and $P(Y = 0|\boldsymbol{x}, \boldsymbol{\theta})$. Here, Y is a indicator variable, where $Y = 1$ and $Y = 0$

represent campaign and non-campaign Q&A sessions, respectively. $\boldsymbol{x}$ is a vector of three features for each session. $\boldsymbol{\theta}$ is a vector of model parameters, each associated with a session feature and including an individually constant item(also called intercept term) which is not related to the session features.

By applying the sigmoid function, the hypothesis $h_{\boldsymbol{\theta}}(\boldsymbol{x})$ which outputs a score of $P(Y = 1|\boldsymbol{x}, \boldsymbol{\theta})$ or $P(Y = 0|\boldsymbol{x}, \boldsymbol{\theta})$ (termed as *campaign score*) is defined as follows:

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^{\mathrm{T}} \cdot \boldsymbol{x}}}$$

where $\boldsymbol{\theta}^{\mathrm{T}} \cdot \boldsymbol{x} = \theta_1 + \theta_2 \times \mathrm{SGqID} + \theta_3 \times \mathrm{SGaID} + \theta_4 \times \mathrm{SGtext}$. To facilitate the vector calculation, we add a dummy attribute (with 1 as the value) to $\boldsymbol{x}$.

In practice, the higher the score, the higher the probability that the given session is a campaign session. The values of $\boldsymbol{\theta}$ will be learned by logistic regression. The objective then becomes an regression problem where we optimize the model so that the output campaign scores of sessions are close to their true labels (0 or 1).

The convex cost function of this optimization problem is given by

$$J(\boldsymbol{\theta}) = \frac{1}{m} \times \Sigma_{i=1}^{m} \log(1 + e^{-y^{(i)} \times (\boldsymbol{\theta}^{\mathrm{T}} \cdot \boldsymbol{x}^{(i)})}) + \frac{1}{2} \times \boldsymbol{\theta}^{\mathrm{T}} \cdot \boldsymbol{\theta}$$

where m is the number of samples in the training dataset and $\boldsymbol{x}^{(i)}$ is a vector consisting of m feature vectors of the i-th training sample. We use gradient descent method to find the minimum of the cost function and the corresponding values in $\boldsymbol{\theta}$.

2.4.3 Significance Test for Logistic Regression

In order to understand the relative contribution and overlap of the selected features, we perform a significance test for the proposed "SG" features in a full model (including all the three "SG" features). We also conduct multiple predictive comparisons between models which contain one or two of the "SG" features.

We use the "glm" function in R to train a full logistic regression model and examine p-values of the "SG" features. We find that the p-value of SGqid is 0.364, while p-values of SGaid and SGtext are both below 0.05. It suggests that SGqid is statistically insignificant in the full model. Furthermore, we also train different models using one or two of the "SG" features and report McFadden's R^2 [97] (a pseudo R^2 measure for logistic regression) over the training set in Table 2.2.

Features	McFadden's R^2
SGqid	0.09548916
SGaid	0.69939115
SGtext	0.50783348
SGqid + SGaid	0.70039091
SGqid + SGtext	0.54451565
SGaid + SGtext	0.76490791
SGqid + SGaid + SGtext	0.76510296

Table 2.2: McFadden's R^2 for Different Combinations of "SG" Features

In Table 2.2, we observe that the full model has the highest McFadden's R^2, which is only slightly better than the model with both SGaid and SGtext. Next, we compare the predictive power on the test set of high R^2 models (SGaid, SGqid + SGaid, SGqid + SGtext, SGaid + SGtext and the full model). Figures 2.5, 2.6, 2.7 and 2.8 show the corresponding ROC curves and values of the area under the curve (AUC). Since the curves and AUCs of SGaid + SGtext and the full model are nearly the same, we only show Figure 2.8 of the full model.

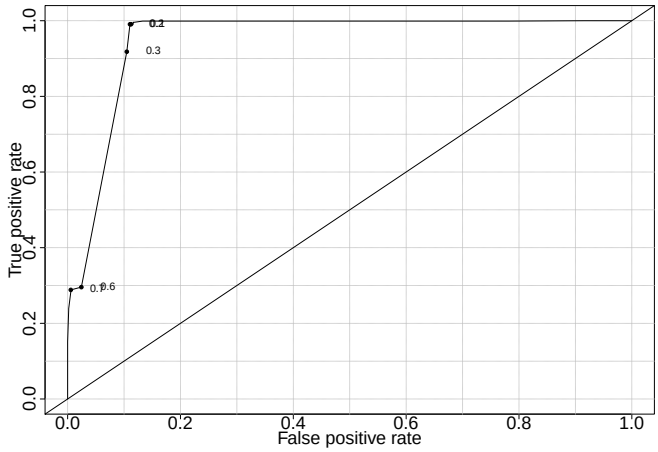


Figure 2.5: ROC curve of SGaid on sorted data, AUC = 0.950.

Figure 2.8 of the full model shows the overall best AUC (0.9830567). Considering that it also has the highest McFadden's R^2, we will take all "SG" features into consideration in the following sections.

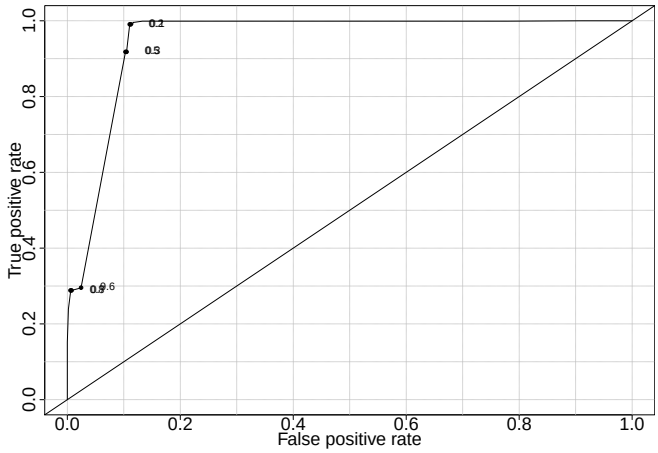


Figure 2.6: ROC curve of SGqid + SGaid on sorted data, AUC = 0.950.

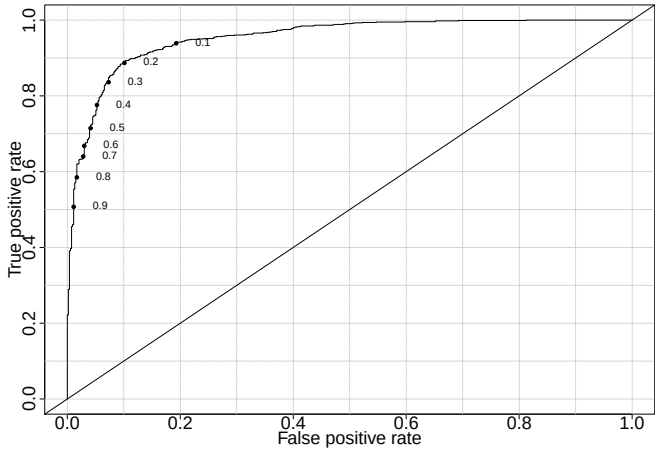


Figure 2.7: ROC curve of SGqid + SGtext on sorted data, AUC = 0.952.

2.4.4 Classification Threshold

The value of $h_{\boldsymbol{\theta}}$ should be carefully determined. When the $\boldsymbol{\theta}$ is optimized, we then calculate the campaign score of each Q&A session in the test dataset. The result is shown in Figure 2.8.

We observe that 0.4, 0.5 and 0.6 are closer to the top left of the figure than other values. Based on Figure 2.8, we set 0.5 as our threshold for $h_{\boldsymbol{\theta}}$. Note that, setting 0.5 as the classification threshold means that we would predict a positive label for a test sample when $\boldsymbol{\theta}^{\mathrm{T}} \cdot \boldsymbol{x}^{(i)} > 0$ while a negative label if $\boldsymbol{\theta}^{\mathrm{T}} \cdot \boldsymbol{x}^{(i)} < 0$.

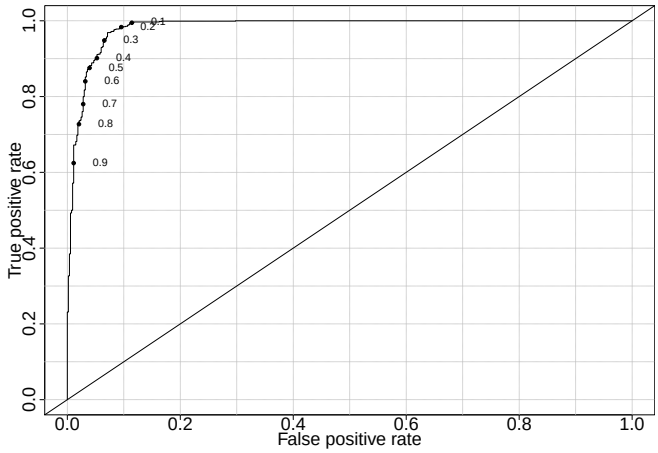


Figure 2.8: ROC Curve of all "SG" features on sorted data, AUC = 0.983.

2.5 Adaptive Detection System

In the previous section, we have shown that we can build a model to effectively calculate the campaign score and predict the labels of unknown sessions. In practice, newly emerging campaigns may have very different patterns of features as those used to build the model. It is necessary to develop an "adaptive" detection system that can update its database using new samples and evolve new model parameters, while maintaining stable detection performance over time. In this section, we present the design of such an adaptive detection system. We will evaluate its performance and assess whether manual labelling is necessary when adding new samples via an experiment based on a real-world dataset in Section 2.6.

The major components of the detection system include browser plugin and a remote server. Figure 2.9 shows the system architecture and the communication between the client plugin and the server.

The sequence of actions that take place when a user opens a Q/A session are:

1. The plugin first sends only the URL of the page to the server. The server searches for the URL in its database. If it is found, the server returns the score (spam rating) to the client. The client side script displays the result. This avoids unnecessarily sending complete web page to the server if it is already present in the database.

2. If the URL is not present, the server sends a response *not found* and the client after receiving the response sends the rest of the data to the server through another *XMLHTTPRequest* and waits for the server's response.

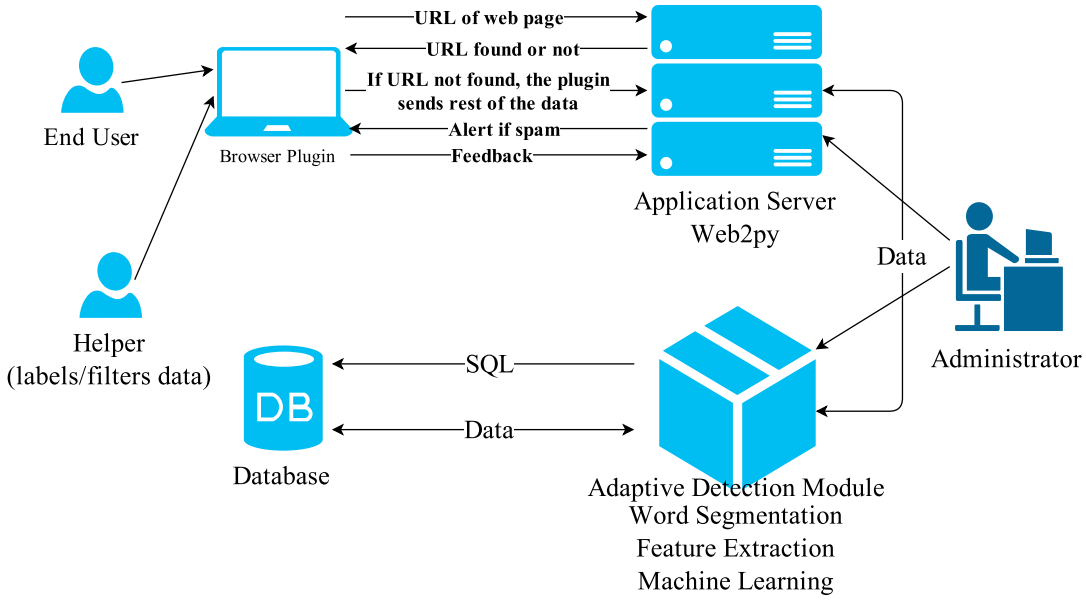


Figure 2.9: System architecture and communication between the client and the server.

3. The server receives the data, segments the text into words, and stores it in the database. The server then extracts the statistical features necessary for the analysis from the data. Logistic regression analysis is performed to predict the class of the session (spam or no spam). If the session is classified as a spam, an alert is returned back to the user.

4. The client-side script displays the result to the user.

5. (Optional) If the user is an authorized user, the user can provide feedback to the server (whether or not he/she feels the session is a campaign session). There are three types of users in the system: regular users are those who use our system and they are not granted the right to annotate sessions; helper users are those who have experience and are capable of helping label the data; the administrator is the person responsible for the management of the system. Note that helpers could be contracted out to employees of professional companies such as Rediff Shopping and eBay [105].

6. When newly labelled sessions are available, the system updates the detection model using existing and newly labelled data. Note that this step could be done regularly in a daily or even weekly basis.

2.6 Performance Evaluation

To evaluate the performance of online detection system, we use the collected data from *Baidu Zhidao* and replay the data in multiple iterations to simulate a real-world scenario. In particular, we pretend that initially we only have partial data and use the data as the training dataset to build a detection model. In each iteration, we add some new sessions and use them as the test dataset to test the performance of the detection system. At the end of an iteration, the new sessions are added into the training dataset, and the detection model is updated using the new training dataset. This step corresponds to the scenario that new data are labeled and added into the system. Then we repeat with another iteration. Note that we sort the Q&A sessions according to the timestamp when a session is closed. In this way, the performance is closer to that of a real-world scenario.

For the test, we begin with 200-sample training set and build an initial detection model. At each iteration, we add 200-sample test set. After evaluating the detection performance, we expand the training dataset with the 200 test samples, and update the detection model with the new training dataset. We repeat this process until we use up all 4998 samples.

Figure 2.10 shows the ratio of non-campaign and campaign Q&A sessions in each iteration.

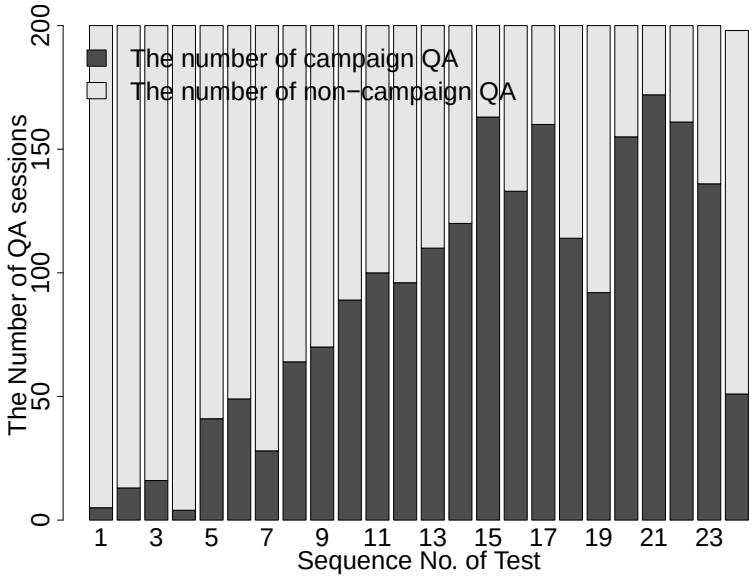


Figure 2.10: Ratio of non-campaign and campaign Q&A.

We evaluate the following four performance metrics:

$$\text{Precision} = \frac{TruePositive}{TruePositive + FalsePositive}$$
$$\text{Recall} = \frac{TruePositive}{TruePositive + FalseNegative}$$
$$\text{F measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$
$$\text{Accuracy} = \frac{TrueNegative + TruePositive}{TotalNumberofUsers}$$

In the following, we will perform comprehensive experiments on the dataset, comparing different methodologies. In particular, we assess whether manual labelling is necessary (Subsections 2.6.1 and 2.6.2), compare the adaptive model to the fixed model (Subsection 2.6.3) and linear classifiers as well as non-linear classifiers (Subsection 2.6.4). We further demonstrate the effectiveness of our model by showing that a model trained by non-twisted data and a model with text only information do not perform well. These results are described in Subsections 2.6.5 and 2.6.6, respectively.

2.6.1 Adaptive Model with Manual Labelling

We first conduct experiments of an adaptive model with manual labelling, i.e., the database is updated using manual labelled (ground-truth) new samples. Figure 2.11 and Figure 2.12 show the update of model parameters and the detection performance in each iteration, respectively.

In Figure 2.11, "Theta 1", "Theta 2", "Theta 3" and "Theta 4" are parameters for the dummy attribute, SGqID, SGaID, and SGtext, respectively. We can observe that the detection model tends to converge after enough sessions have been added into the database over several iterations. For example, after 10 iterations, the precision achieves 85% - 90%.

We also notice that there is a "degraded" point at the 15-th iteration in the recall, F measure and accuracy figures. After carefully checking the log file (true/false positive and true/false negative) of this iteration, we find out that the false negative is very high, which means a large number of campaign sessions are classified as non-campaign ones. In addition, the continuously generated test set keeps changing because we sort the Q&A sessions according to the timestamp when a session is

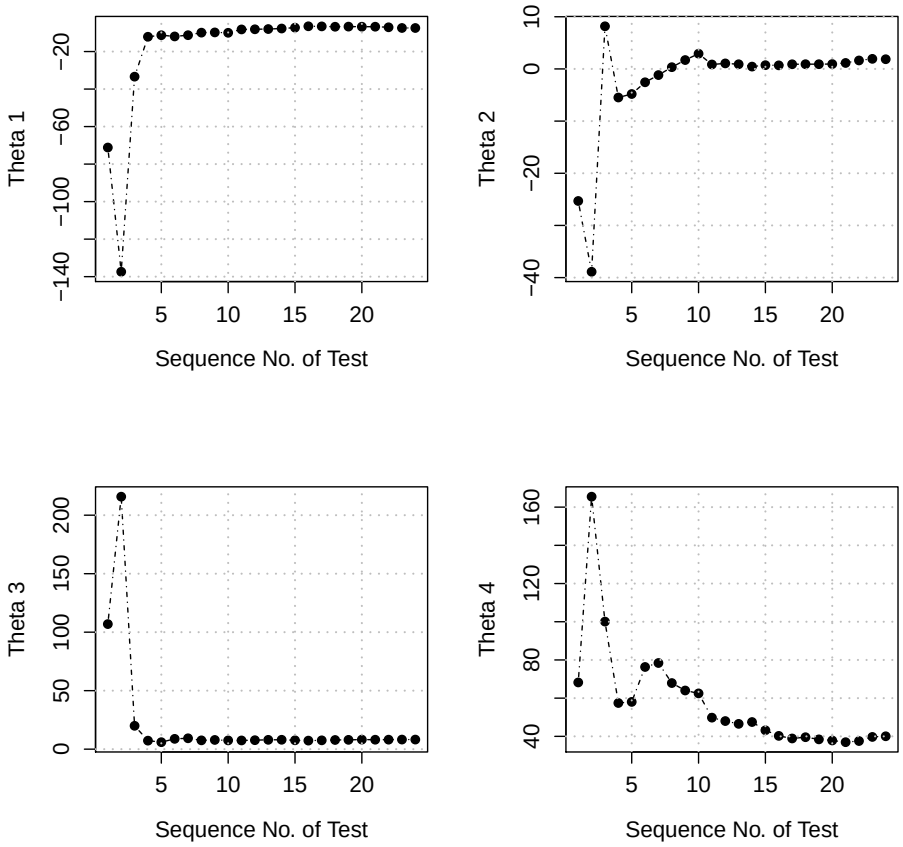


Figure 2.11: Adaptive changes of model parameters over time.

closed. Since the 200 samples of the 15-th test have very different patterns, it affects the performance of the detection model significantly.

Nonetheless, the system is able to recover from the bad performance and works well over all measures after more Q&A data is taken into account in training the model. The four metrics are all above 80% during the last few iterations. This test scenario is similar to the practical application where we predicate the unknown sessions using current knowledge and train a new model based on the sessions after we manually label them.

2.6.2 Adaptive Model without Manual Labelling New Samples

To illustrate the advantage of manual labelling, we also perform the experiment in which we update the model using the predictions of new samples. We use 200 manually labelled samples as the initial training data and build a model. At each iteration,

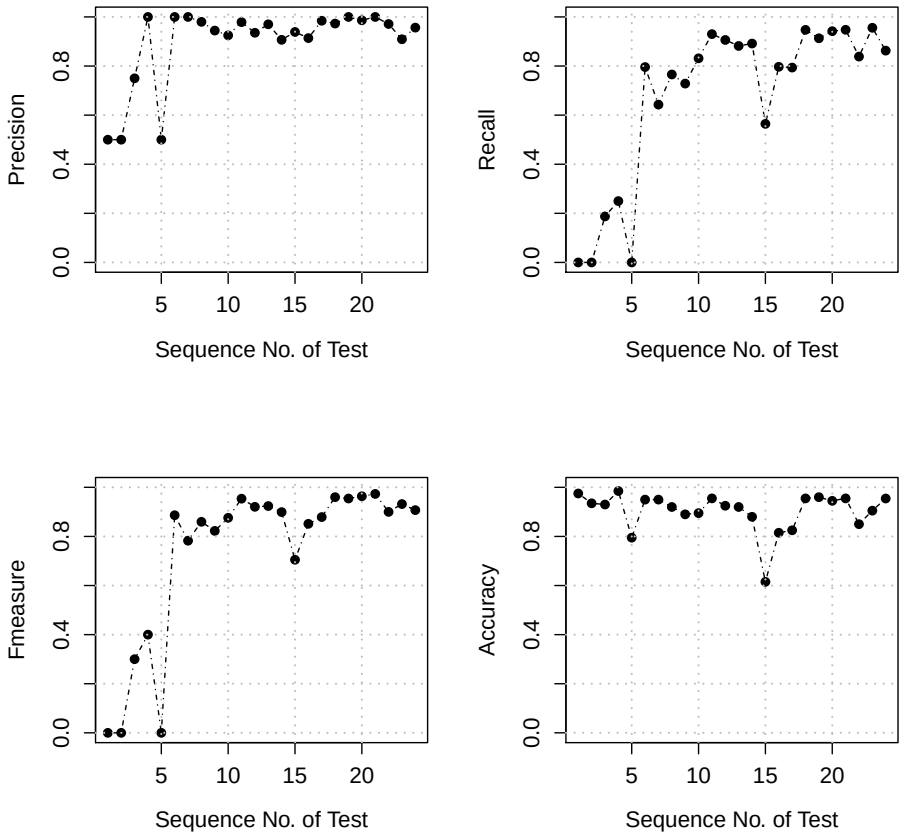


Figure 2.12: The performance of the adaptive detection system over time with manual labelling.

we test 200 new sessions using the model and insert the new samples associated with the predictions into the database. The results are shown in Figure 2.13.

In Figure 2.13, we observe that the recall measure is surprisingly high; it achieves 1.0 for most of iterations. It suggests that the false negative is very low. After checking its predictions of all iterations, we find that it outputs very few negative predictions, sometimes none at all. These predictions result in large performance fluctuation because the proportions of the campaign session in continual iterations are different. In addition, with more samples being used to train the model, the increasing trend in precision, F measure and accuracy are not as stable as they are in Figure 2.12. Therefore, manual labelling new samples is still critical for accurate predication in practice.

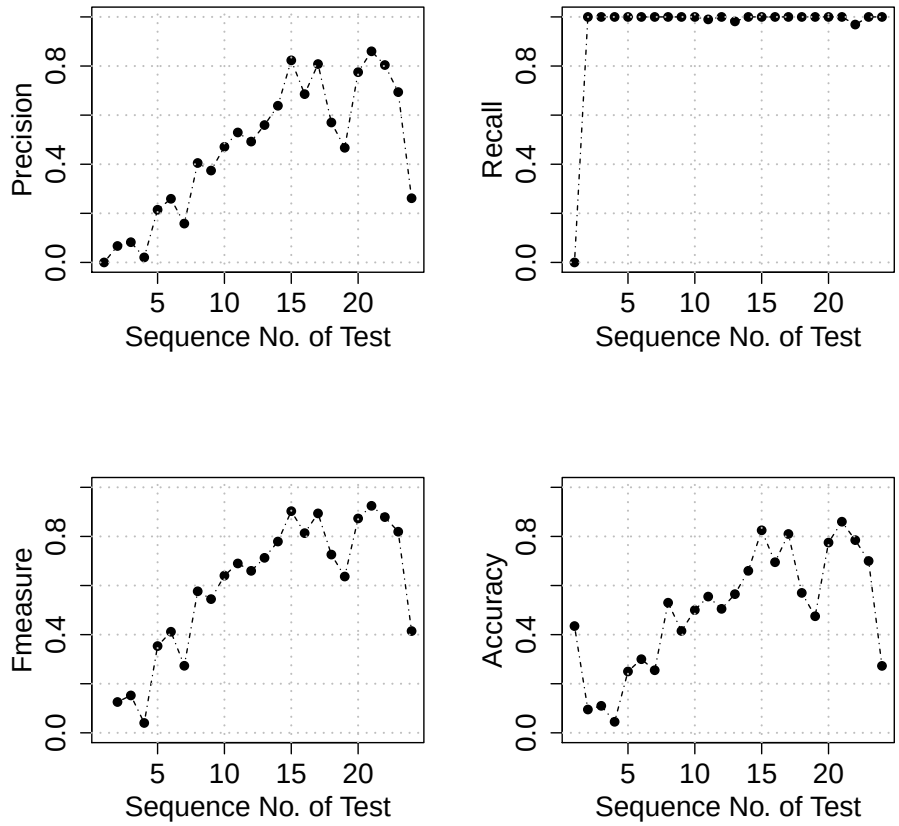


Figure 2.13: The performance of the adaptive detection system over time without manual labelling.

2.6.3 Fixed Model

To illustrate the advantage of adaptivity with respect to accumulated samples, we test two types of the fixed model in which we use a fixed size training set to train the model.

Fixed Training Set

We use the first 200 samples as the initial training data and build a model. We fix the model parameters, and at each iteration, we test 200 new sessions using the fixed model. The results are shown in Figure 2.14.

Since the parameters of the fixed model are only trained on the first set of training samples, we do not draw the changes of the model's parameters. The precision in some tests is nearly 50% and it even becomes very high in a few tests from the 15-th to the 20-th iterations. However, compared to Figure 2.12, we note that the recall values are always very low. It means that the false negative is high. The

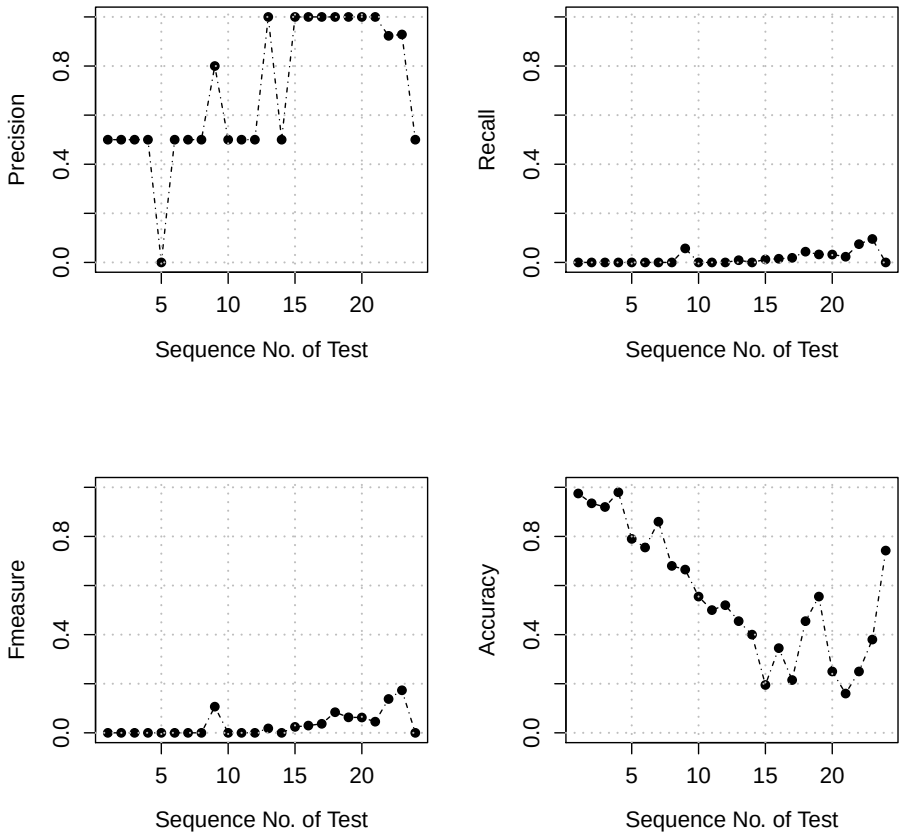


Figure 2.14: The performance of the fixed model.

low F measure values confirm this problem in the fixed model, i.e., the fixed model classifies many campaign Q&A sessions as the non-campaign sessions. Consequently, although the precision is high, other metrics indicate that the fixed model has obvious bias in classification. What is worse, this model cannot update itself by new samples because the parameters are only trained on the initial training dataset.

Moving Window of A Fixed Size

We use a moving window of a fixed size (200 samples) to train the model at each iteration. For example, at the first iteration, we train the model with samples $1-200$ and test it with samples $201-400$. At the second iteration, we train the model with samples $201-400$ and test it with samples $401-600$. We repeat this procedure for all iterations. The results are shown in Figure 2.15.

Figure 2.15 shows similar recall as in Figure 2.13; it achieves 1.0 for most of iterations. Again, we check predictions of all iterations and find that the model outputs very few negative predictions, sometimes none at all. In addition, the performances of

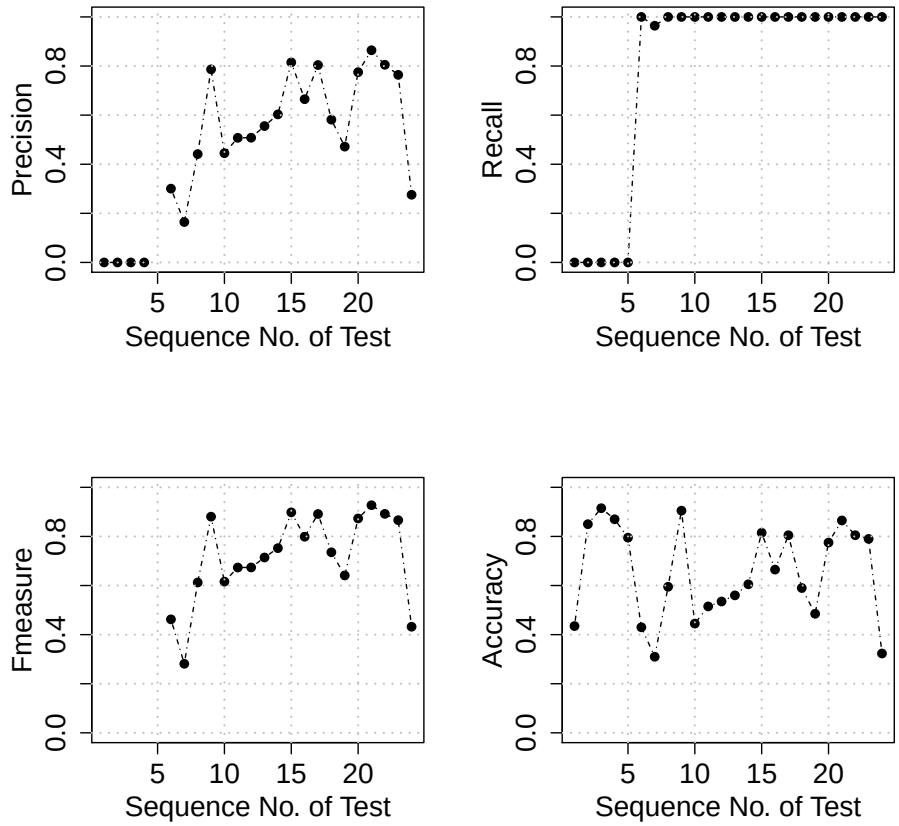


Figure 2.15: The performance of the fixed model with moving windows of a fixed size.

precision, F measure and accuracy are not stable. Therefore, training by the moving window of a fixed size restricts the model's ability to adapt to new samples. The sample accumulation (quantity) is necessary to improve the performance of the detection system.

2.6.4 Experiments with Different Models Using Two More Advanced Classification Packages

As evaluated in Subsection 2.6.1, a logistic regression based method shows satisfactory overall performance when distinguishing campaign answers from non-campaign ones. To further examine the effectiveness of this method, we explore different linear classifiers as well as non-linear ones on our dataset and compare their prediction performance in Subsection 2.6.4. We perform experiments using two popular machine learning libraries, LIBSVM[⑦] [29] and LIBLINEAR [42].

[⑦]Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`

LIBSVM is a general-purpose SVM solver, which supports kernel functions in order to train non-linear classifiers. On the other hand, LIBLINEAR is exclusively used for linear classification, i.e., it supports logistic regression and linear support vector machines. Without using kernels, LIBLINEAR can train a much larger set via a linear classifier. Consequently, LIBLINEAR is considered as a better choice over LIBSVM when handling large-scale datasets (e.g., document classification) for which using nonlinear mappings does not provide additional benefit.

Tables 2.3 and 2.4 list candidate models to be tested in the experiment.

Kernel type	Description
t0	linear: $\boldsymbol{u}^{\mathrm{T}} \cdot \boldsymbol{v}$
t1	polynomial: $(\gamma \times \boldsymbol{u}^{\mathrm{T}} \cdot \boldsymbol{v} + coef0)^{degree}$
t2	radial basis function (RBF): $\exp(-\gamma \times \lvert\boldsymbol{u} - \boldsymbol{v}\rvert^2)$
t3	sigmoid: $\tanh(\gamma \times \boldsymbol{u}^{\mathrm{T}} \cdot \boldsymbol{v} + coef0)$

Table 2.3: LIBSVM Kernel Types

Solver type	Description
s0	L2-regularized logistic regression (primal)
s1	L2-regularized L2-loss support vector classification (dual)
s2	L2-regularized L2-loss support vector classification (primal)
s6	L1-regularized logistic regression

Table 2.4: LIBLINEAR Solver Types

Figure 2.16 shows the running time of different models for LIBLINEAR and LIBSVM. The total time includes model training and sample prediction and it is evaluated on a sequence of 24 tests, as described in Section 2.6. We observe that both LIBLINEAR and LIBSVM are very fast; even the slowest model (LIBSVM with t2) takes less than 2 seconds on all 24 iterations. In addition, LIBLINEAR is substantially faster than LIBSVM. The reasons for these observations are that we have a limited number of features for each sample and linear classifiers can be trained more efficiently than non-linear ones.

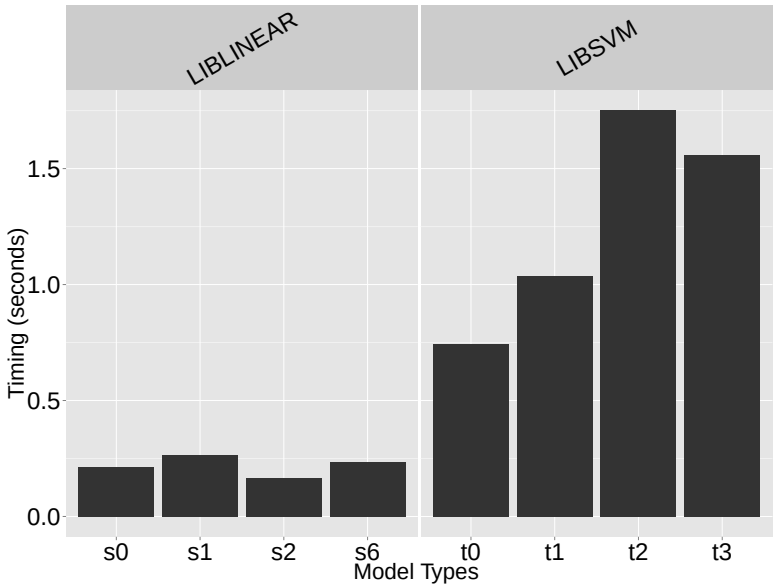


Figure 2.16: Timing of different model types for LIBLINEAR and LIBSVM.

We then show the performance metrics for the 8 tests. As for LIBSVM, since the metrics of kernel types t0, t2 and t3 are similar, we only draw Figure 2.17 and Figure 2.18 for kernel types t1 and t2, respectively.

Since the performance curves for all models of LIBLINEAR are similar, we only draw a figure (Figure 2.19) for the solver type s2, which is also the fastest.

From Figures 2.17, 2.18 and 2.19, we see that the recall and the accuracy of LIBSVM with polynomial kernel are worse than those with RBF kernel. On the other hand, the precision of linear classifier trained by LIBLINEAR is not as stable as that of LIBSVM with RBF kernel. When it comes to the other three metrics, we observe similar trends for LIBLINEAR and LIBSVM with RBF kernel. Since LIBLINEAR takes much less time than LIBSVM, we can conclude that a linear classifier is more suitable for our detection problem. In addition, note that metrics of precision and F measure in the three figures are not available for the first few tests (corresponding to the missing points in Figures 2.17, 2.18 and 2.19), because initially the models only return negative predictions.

2.6.5 Non-Twisted Data Based Results

We have done some twists for user's features in Section 2.3. For example, to avoid 0 probability, we specify 0.5 to $q1$ (the number of questions which have campaign answers for a specific user) when $q1 = 0$. If the system does not have enough infor-

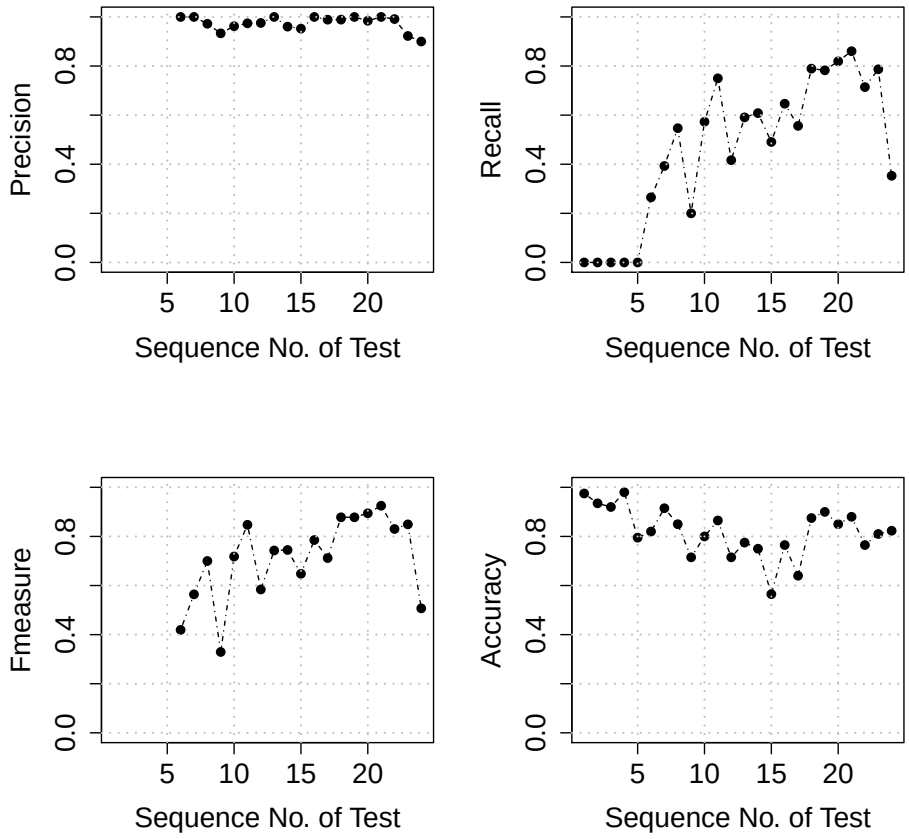


Figure 2.17: LIBSVM with polynomial kernel using default penalty and model parameters (t1).

mation for a certain user(i.e., the total number of questions is less than 5), we set its SGqID value to 0.5. In order to show the importance of twists, we train models based on raw data, i.e., with zero probability.

Figure 2.20 showed the performance metrics of this approach and it can be seen that fluctuation exists everywhere on performance curves. The overall performance is worse than that based on twisted data. It suggests that data correction should be considered before model training.

2.6.6 Experiments Using Only Text Information

We now use only text information (question, answers) for training the classifier. As a comparison to the previous method, we will use a typical information retrieval approach which consists of feature word selection, vectorization and classification.

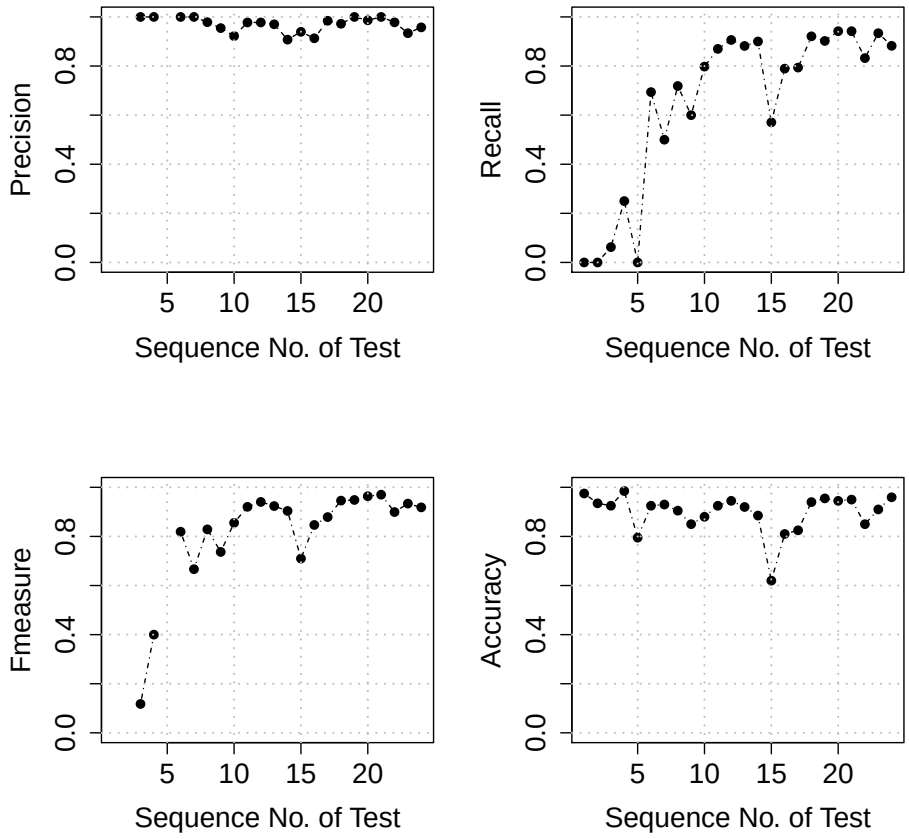


Figure 2.18: LIBSVM with RBF kernel using default penalty and model parameters (t2).

Feature Selection

We use the Chi-square method [142, 49] to retrieve a bag of feature words, a standard methodology of extracting features in documentation classification.

We define variables A, B, C and D in Table 2.5. For example, A is the number of campaign Q&As which have a specific word in the answers. D is the number of non-campaign Q&As which do not have the specific word.

Feature	campaign	non-campaign	Total
has word	A	B	$A+B$
not word	C	D	$C+D$
total	$A+C$	$B+D$	N

Table 2.5: Chi-square Feature Selection

After we collect the statistical information for every individual word, we can then compute Chi-square values. The Chi-square value of a word in the document collec-

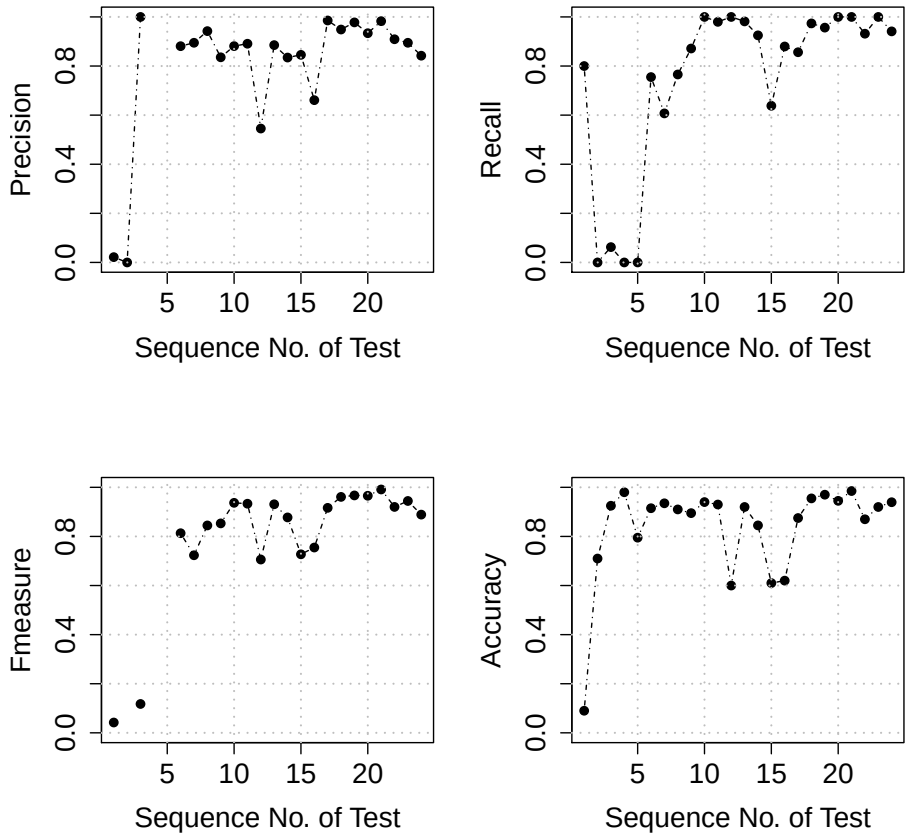


Figure 2.19: LIBLINEAR with L2-regularized L2-loss support vector classification (s2).

tion is defined as

$$\text{Chi-square}(word, classification) = \frac{(A \times D - B \times C)^2}{(A + B) \times (C + D)}$$

We compute the Chi-Square value for each word in the training document collection, sort them in descending order and retrieve the first d words as the bag of the most predictive features.

Vectorization

After selecting feature words from the document collection, we can then vectorize each document by associating it with a vector of dimension d. To preserve consistency, we compute the weight for each dimension in the same way as how the spam grade value is calculated in Section 2.3. Note that for each iteration of the 24 tests, we re-extract the features words and consequently re-compute the weight for each feature word.

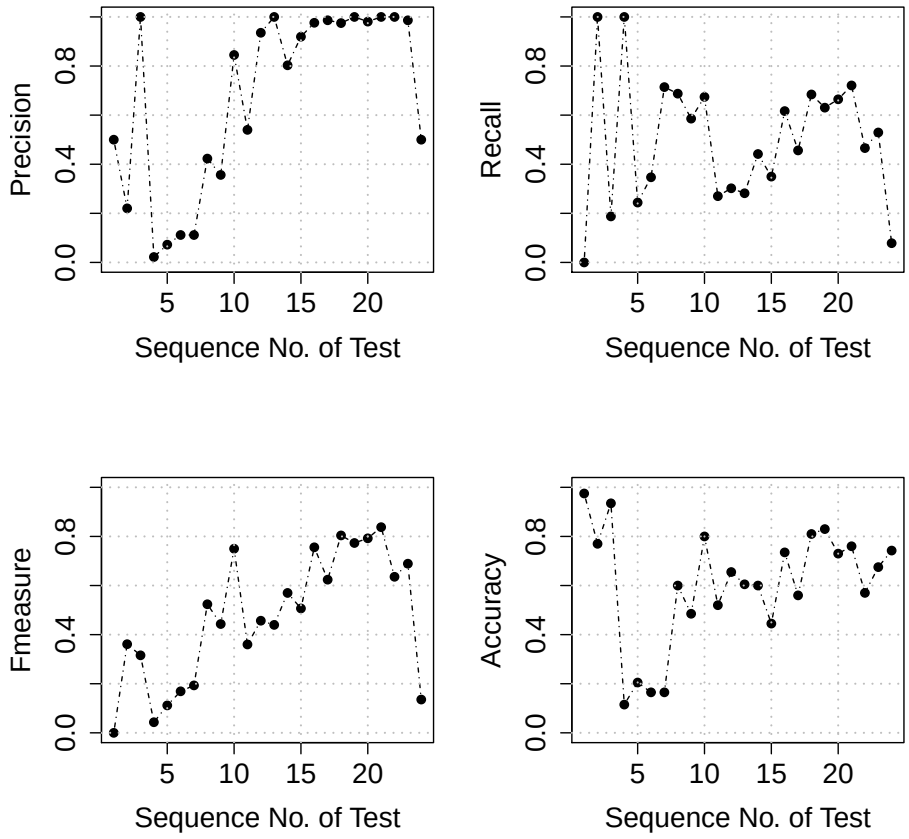


Figure 2.20: Performance metrics on features without data correction.

Performance Evaluation

In order to show the impact of different dimensions, we use three settings for the following tests, i.e., $d = 100$, $d = 150$ and $d = 200$. We test LIBSVM with t2 and LIBLINEAR with s2. The results are shown in Figure 2.21 (LIBSVM) and Figure 2.22 (LIBLINEAR).

In Figures 2.21 and 2.22, different dimensions do not produce very different curves. With increased training samples, a model with higher dimension (200) is only slightly better than models with lower dimension (100 and 150). After the 15-th iteration, recall and F measure of LIBLINEAR are better than those of LIBSVM.

We now compare performance of text-only features (Figure 2.21 and Figure 2.22) to that of user-text features (Figure 2.18 and Figure 2.19). We list two main observations as follows.

- For LIBSVM, precision is very high using either set of features. Recall and F measure are significantly improved using user-text features when the number

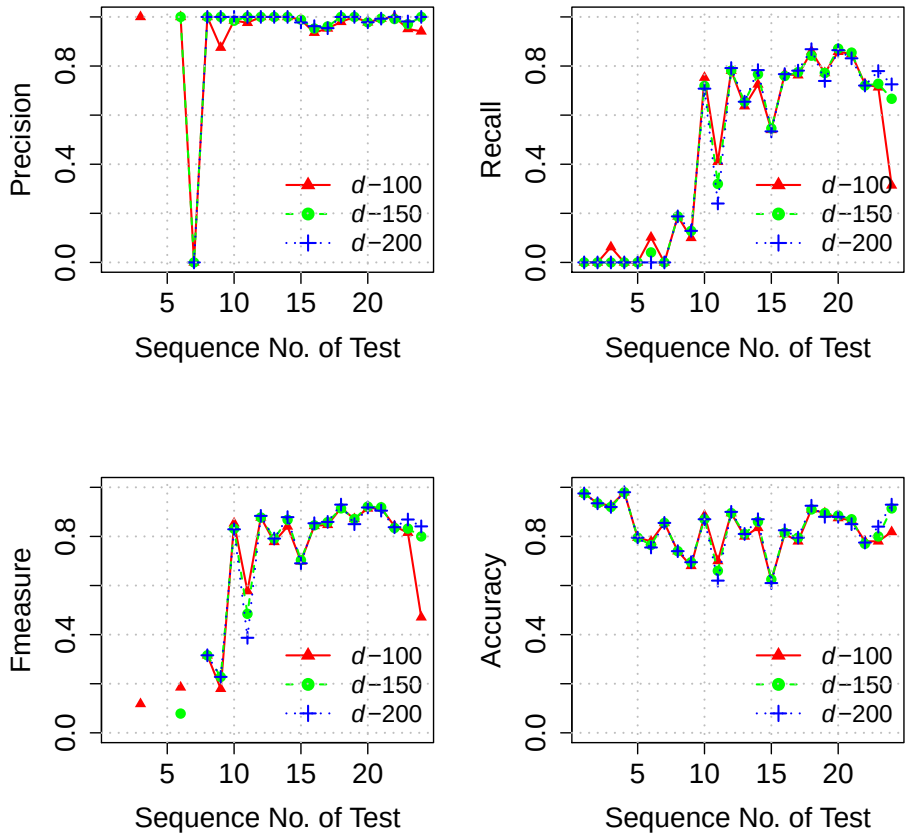


Figure 2.21: LIBSVM with RBF kernel (t2) using default penalty and model parameters.

of training samples is small, e.g., tests between the 5-th and the 10-th iteration. After the 15-th iteration, recall and F measure are also higher for user-text features ($0.9 \sim 1.0$), while text-only features vary from 0.8 to 0.9. In addition, user-text features provide more stable accuracy values than text-only features.

- For LIBLINEAR, we observe significant improvement of the four metrics between the 5-th and the 10-th iteration by using user-text features. After the 15-th iteration, user-text features lead to high values for all metrics that are close to 0.95. On the other hand, text-only features produce values around 0.9 which are slightly worse. Another interesting observation is that precision and F measure of 200-dimension text-only features fall below 0.8 at the last iteration, while those with 100-dimension features drop to 0.4. In contrast, user-text features show robustness during the last iteration.

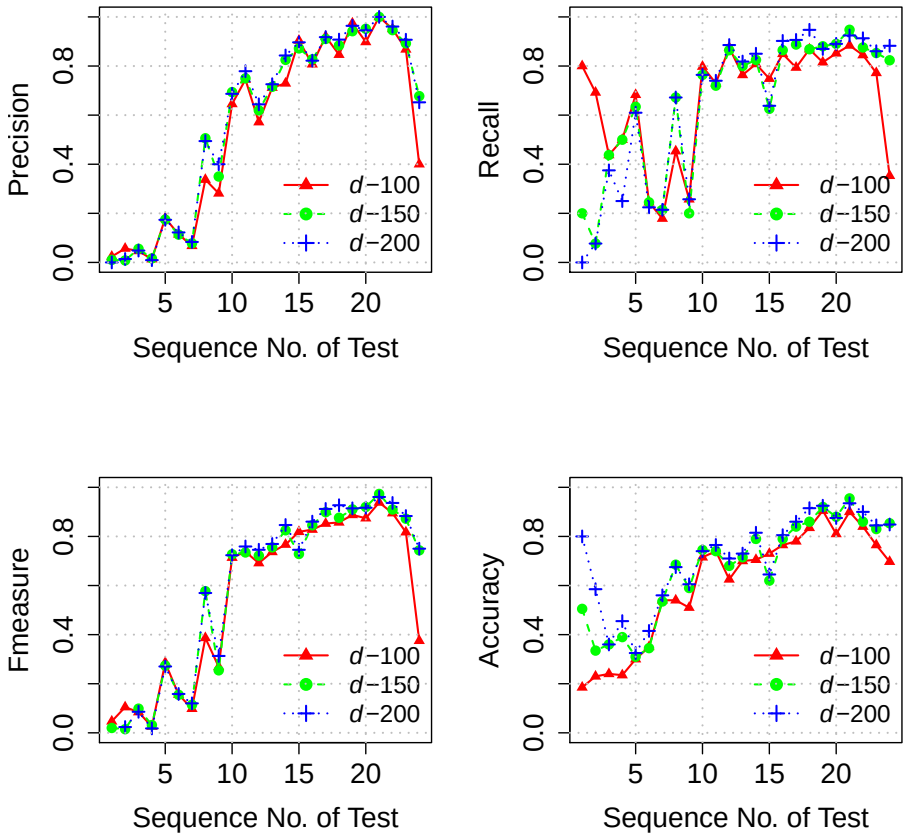


Figure 2.22: LIBLINEAR with L2-regularized L2-loss support vector classification (s2).

The overall performance of text-only features during the last few iterations is worse than that of user-text features. In addition, the number of features used in text-only approach (100, 150 and 200) is much more than that used in user-text method (3 as described in Section 2.3). This fact implies that the latter approach runs much faster while preserving high performance. To summarize, our proposed approach exhibits both effectiveness and efficiency, which are important factors in practice.

2.7 Conclusions

Detection of hidden campaigns can help remove deceptive information and improve users' experience when accessing recommended content. In this chapter, we disclose the behavior of a specific group of online paid posters who create commercial campaigns on the community Q&A websites. We collect real-world datasets and identify effective features to distinguish normal sessions and the campaigns. The performance

of our classifier, with integrated statistic and semantic analysis, is promising on the real-world case study. Based on a learning technique, we also implement a prototype of adaptive detection system which can retrieve the result in real time. The campaign scores and/or predicated labels can help users make better decisions when searching for answers on CQA portals and help the questioners select better answers as well.

Chapter 3

Conflict-Aware Weighted Bipartite b-Matching

3.1 Introduction

In this chapter, we study the utility maximization with diversity constraint problem in the e-commerce application scenario, where items or sellers are often recommended to users for user utility maximization while capturing diversity across specified conflicts among entities of the same type (e.g., users/items/sellers). We formally define this problem by extending a classic graph-theoretic model, weighted bipartite b-matching (WBM).

The weighted bipartite b-matching problem (WBM) is a classic optimisation problem that is ubiquitous in data management and e-commerce applications. Figure 3.1 illustrates the problem: the input is an edge-weighted, undirected, bipartite graph $G = ((U, V), E, W)$ and a maximum degree constraint (in red) for each vertex. The WBM problem seeks to match vertices in U to vertices in V so that each vertex is matched with no more vertices than its degree constraint allows. Equivalently stated, we want to compute a subgraph H of G with the maximum sum of edge weights where none of the degree constraints is violated. The solution to the given example is subgraph H with edges given in blue. The sum of edge weights of the subgraph H (score(H)) is 1.6.

Due to its expressive power, WBM and its variants find important applications in many areas, such as resource allocation [17, 100], scheduling [44], Internet advertising [98] and recommender systems [7]. Consider a problem of recommending items

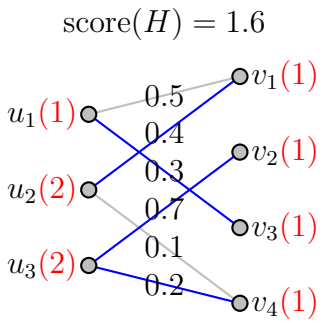


Figure 3.1: The WBM problem. The input graph has score 2.2, the sum of all its edge weights. The blue edges of the solution H yield the highest score, 1.6, of all subgraphs satisfying the red degree constraints.

(e.g., books) to readers as an example. Typically, the recommender system should satisfy three requirements: (1) degree constraints on the number of recommendations books are part of within the maximum availability, (2) degree constraints on the number of recommendations readers receive before they become overwhelmed, and (3) recommendation of books should be based on reader preferences. This problem can be naturally modeled by WBM, where the left-side nodes and right-side nodes of the bipartite graph denote readers and books, respectively, and each edge weight represents the preference of a book to a reader. Note that edge weights can be learned beforehand using collaborative filtering algorithms [129, 84, 127]. The goal is to find a subgraph such that the total weight (preference) of the matched edges in the subgraph is maximized, while satisfying all degree constraints.

An implicit assumption of WBM is that any two nodes on the same side do not interfere with each other, even if they share similar features. A problem with this is that WBM cannot model diversity constraints on the items matched to users. For example, a recommender system running WBM can recommend several books of the same subject to a reader, as long as the subject is his/her favorite and the availability constraints of the books are not violated. This, however, does not generate desired results in some real-world scenarios. For book recommendation, a reader may not want all recommended books from the same subject but instead may prefer books of diverse subjects so that more interesting topics can be discovered. The recommender system should allow a reader to constrain the number of books from the same subject. In other words, books from the same subject are in "conflict" with each other when being recommended to a reader, and the number of such conflicts should be below the reader's tolerance threshold. Hence, the interference issue would inevitably lead

to a new challenge in WBM when generating the matching result. Diversity is a critical consideration for most applications of WBM, from recommender systems [48] to web document ranking [115] to online shopping [134]. In particular, WBM cannot capture diversity across topics and genres [143], groups of redundant customers, such as households, nor temporal ranges [88].

In this chapter, we introduce a new generalization of WBM, Conflict-Aware Weighted Bipartite b-Matching (CA-WBM), that can address the conflict challenges mentioned above. For ease of understanding, we describe CA-WBM in the context of e-commerce. In this scenario, the input bipartite graph is composed of buyer vertices, seller vertices, and edges between buyers and sellers. From the point of view of an e-commerce company, an edge typically associates a profit weight for each pair of a buyer and a seller. Then the goal is to maximize the total profit from all potential transactions between buyers and sellers.

In the simplest form of CA-WBM, we assume that each buyer and seller has a specific "capacity". That is, a buyer cannot be matched to more sellers than his capacity, and a seller cannot be matched more buyers than he has budget (capacity) for. The matching result should maximize the total profitability (sum of recommendation weights), while respecting the capacity constraints of the buyers and sellers. It is easy to see that this is an instance of WBM. Next, we enrich the model in a natural way by allowing the system to accommodate various matching strategies in terms of conflict constraints. To capture these constraints, we assume that buyers (sellers) could be "in conflict" with other buyers (sellers). For example, it is not desirable to have two buyers in the same household[①]

We also study an online version of CA-WBM (online CA-WBM), where the buyers arrive in an online fashion and the corresponding edges are revealed when each buyer arrives. Specifically, online CA-WBM is relevant to Internet advertising [98], including online adwords [99, 34] and display advertising [45, 18]. Nevertheless, these problems do not consider conflict between entities (e.g., adwords). Therefore, online CA-WBM will be valuable in providing a more flexible service to Internet advertising. For example, if a bidder expects his/her products to be exposed to a wide range of adwords, he/she can set conflict constraint based on similarities between adwords.

The remainder of this chapter is organized as follows. We formally define WBM and CA-WBM in Section 3.2 and prove the NP-hardness of CA-WBM in Section 3.3.

[①]Such information could be identified in data, e.g., based on buyers' mailing address and home phone number.

In Section 3.4, we design different algorithms to solve CA-WBM using SDP and ILP. We also propose a greedy algorithm and prove its performance bound. In Section 3.5, we study an online version of CA-WBM and propose a randomized algorithm to solve the online problem. We bound the competitive ratio of the randomized algorithm. We present experimental evaluation in Section 3.6. We conclude the chapter in Section 3.8.

3.2 Problem Formulation

Throughout this chapter and Chapter 4, we assume that any graph $G = ((U, V), E, W)$ is edge-weighted, undirected, and bipartite. In other words, U and V are disjoint sets of vertices and the edge set $E \subseteq U \times V$ contains only edges between vertices in U and vertices in V. The edge-labelling function $W : E \to \mathbb{R}_+$ assigns a positive, real-valued weight to every edge. The problems in this chapter define a scoring function and constraints and find a subgraph $H = ((U, V), E', W)$ that maximizes the score while satisfying the constraints.

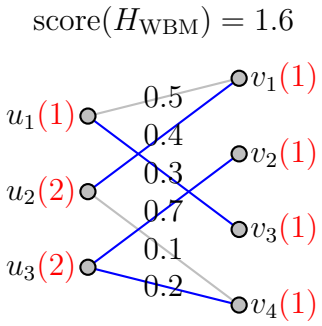
To facilitate understanding of CA-WBM, we start by first studying the simpler, weighted bipartite b-matching (WBM) problem without considering the conflict between buyers (sellers).

Weighted Bipartite B-Matching (WBM)
Given G and vertex-labelling function $B : U \bigcup V \to \mathbb{N}$, find a subgraph $H = ((U, V), E', W)$ maximising $\sum_{e \in E'} W(e)$ with every vertex $u \in U \bigcup V$ adjacent to at most $B(u)$ edges.

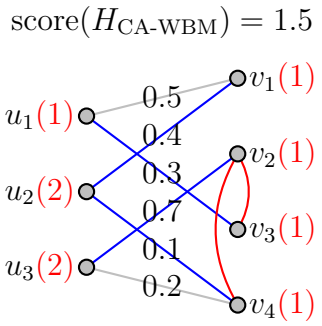
Figure 3.1 (WBM) is repeated in Figure 3.2a for ease of comparison. The matching excludes the high-weight (u_1, v_1) edge, because u_1 and v_1 are permitted only one edge and $\{(u_1, v_3), (u_2, v_1)\}$ produces a higher overall score.

For convenience, we slightly overload the notation and use an mn-dimensional vector to denote $\mathcal{E} = [e_{ij}]$, with $e_{ij} = 1$ indicating that there is an edge between buyer i and seller j and $e_{ij} = 0$ otherwise. Similarly, we use a vector to denote $W = [w_{ij}]$. If $e_{ij} = 0$, then $w_{ij} = 0$. When the subscripts are hard to read, as in the next section, we also use $W(i, j)$ to denote w_{ij}.

The weight of the edge between $i \in U$ and $j \in V$, w_{ij}, reflects the profitability value if i is matched to j. In practice, we may pre-process the bipartite graph based on various business models. For instance, we may order the buyers or sellers based on



(a) H_{WBM}: the WBM solution shown in Figure 3.1



(b) $H_{\text{CA-WBM}}$: the top score avoiding conflict edge (v_2, v_4)

Figure 3.2: The CA-WBM problem contrasted with WBM. Two conflicts, $(v_2, v_3), (v_2, v_4)$, are introduced, e.g., because the products are too similar. If u_3 has a conflict threshold $\tau(u_3) = 0$, then it cannot match both v_2 and v_4, leading to a lower score, but potentially more diverse, solution.

money spent and earned, recency of buys and sells, etc. In addition, we may constrain the edges from a buyer to a ranked range of sellers, so that the buyer is not matched to sellers well outside of her "tier". One possible such scenario is that we may need to match top buyers to top sellers, middle-tier buyers to middle-tier sellers, and so on, as this maximizes the chance of making both the buyers and sellers happy.

Let $m = |U|$ and $n = |V|$. We represent the degree constraint for all vertices, as specified by the vertex-labelling function B, with a $(m+n)$-dimensional column vector $D = [D(i)]^T$. We denote by $X = [x_{ij}]^T$ the mn-dimensional column vector of 0-1 variables, with $x_{ij} = 1$ indicating buyer i is matched to seller j and $x_{ij} = 0$ otherwise. Then WBM can be written as a linear program, i.e.,

$$\begin{aligned} \max_{X} \quad & WX \\ s.t. \quad & \mathbb{A}X(i) \leq D(i), \forall i, 1 \leq i \leq m+n \\ & x_{ij} \in \{0,1\}, \forall i,j, 1 \leq i \leq m, 1 \leq j \leq n, \end{aligned} \tag{3.1}$$

where matrix $\mathbb{A}$ is an $(m+n) \times mn$ matrix defined by

$$\underbrace{\begin{bmatrix} [e_{11}, \ldots, e_{1n}] & & & \\ & [e_{21}, \ldots, e_{2n}] & & \\ & & \ddots & [e_{m1}, \ldots, e_{mn}] \\ [e_{11}, 0, \ldots, 0] & & \ldots & [e_{m1}, 0, \ldots, 0] \\ \ddots & & \ddots & \ddots \\ [0, \ldots, 0, e_{1n}] & & \ldots & [0, \ldots, 0, e_{mn}] \end{bmatrix}}_{(m+n)\times mn}. \tag{3.2}$$

e_{ij} elements of the first m rows represent the adjacent edges for vertices in U and e_{ij} elements of the last n rows represent the adjacent edges for vertices in V. Note that $e_{ij} = 1$ indicates that there is an edge between $i \in U$ and $j \in V$ and $e_{ij} = 0$ otherwise. The degree constraints are given by $\mathbb{A}X(i) \leq D(i)$, where $\mathbb{A}X(i)$ denotes the i-th element in (vector) $\mathbb{A}X$ and $D(i)$ the i-th element in D.

It has been shown that the WBM problem (as a *bipartite maximum weight b-matching* problem) could be reduced to the transportation problem in operations research [10, 124], and as such we can obtain an LP formulation that can be solved efficiently by modern solvers. Furthermore, note that matrix $\mathbb{A}$ is the incidence matrix of the buyer-seller bipartite graph, which can be proven to be totally unimodular [13, 138]. As [124, 119] show, the polyhedron $P = \{X : \mathbb{A}X \leq D\}$ is integral, and there is a polynomial time algorithm which finds an integral optimal solution.

In order to capture the conflict constraint, we now consider a natural extension of the model above. In the following, we focus on integrating the conflict between buyers. Note that the formulation and proposed algorithms (with little modification) also apply to the case where we simultaneously consider the conflict on both buyer and seller sides. We omit the latter to avoid repeated depiction.

In some scenarios, sellers will prefer a *diverse* list of buyers that avoids certain redundancies. For example, a seller might prefer that their list does not include more than one buyer from each household. Advertising a given merchandise to more than one potential buyer in a household is, in most cases, unnecessary. Formally, *we say that two buyers are in conflict with each other if matching them to the same seller is not desirable.* We will represent the presence of such conflicts between two buyers using conflict edges (red edges in Figure 3.2b).

We call this problem conflict-aware constrained matching (CA-WBM). The goal is to *compute a maximum profit subgraph satisfying the degree constraints with the additional requirement that the number of conflict edges within a list of buyers matched to any particular seller is smaller than a threshold.*

CA-WBM additionally imposes a set of conflict pairs C, requiring that each $u \in U$ is adjacent to at most $\tau(u)$ of those pairs:

Conflict-Aware
Weighted Bipartite B-Matching (CA-WBM) [31]
Given G, vertex-labelling functions $B : U \bigcup V \to \mathbb{N}$, $\tau : U \to \mathbb{N}$, and a set of unordered pairs $C \subseteq V \times V$, find the subgraph $H = ((U, V), E', W)$ maximising $\sum_{e \in E'} W(e)$ with every vertex $u \in U \bigcup V$ adjacent to at most $B(u)$ edges and every vertex $u \in U$ adjacent to at most $\tau(u)$ pairs of vertices $v, v' \in V$ that appear as an unordered pair $(v, v') \in C$.

Figure 3.2b illustrates CA-WBM. The conflict pairs (v_2, v_3) and (v_2, v_4) are marked by red arcs and the threshold function is: $\tau(u) = 0, \forall u \in U$. Relative to WBM, the conflicts restrict the space of feasible solutions; e.g., the subgraph in Figure 3.2a would not solve the CA-WBM instance, because u_3 is matched to both vertices of the conflict pair (v_2, v_4).

It is easy to see that WBM is a special case of CA-WBM. We next show that the constraints in CA-WBM pose a great challenge and significantly increase the complexity of the matching problem.

3.3 NP-Hardness Result for CA-WBM

In this section, we provide strong evidence that CA-WBM is highly unlikely to have an efficient (i.e, polynomial time) algorithm by showing that it is NP-hard.

Theorem 1. CA-WBM *is NP-hard.*

Proof. We give a polynomial-time reduction from the NP-hard problem Revenue Maximization in Interval Scheduling [12, 19, 83].

Revenue Maximization in Interval Scheduling (RMIS)

Instance: A set $M = \{m_1, m_2, \ldots, m_t\}$ of t machines and a set $J = \{j_1, j_2, \ldots, j_n\}$ of n jobs. For each job j in J, we are given three parameters: (1) $S(j)$, the set of machines on which j can be processed (2) $R(j, -)$, the set of possible revenues obtained

when job j is processed on different machines (3) $I(j)$, the time interval during which job j must be processed.

Goal: Find a feasible schedule of a subset of jobs on the machines that maximizes the total revenue of the jobs scheduled.

We now describe a reduction from RMIS to CA-WBM. Given an instance I of the revenue maximization problem, construct a graph $G(I)$, which is an instance of CA-WBM. Define $G(I) = \langle (J, M), E, W \rangle$ with $E \subseteq J \times M$, and $C \subseteq J \times J$ and weights $W : E \to \mathbb{R}^+$ as follows.

- $E = \{(j_k, m_l) | m_l \in S(j_k)\}$.
- $C = \{(j_k, j_l) | I(j_k) \cap I(j_l) \neq \emptyset\}$.
- $W(j_k, m_l) = R(j_k, m_l)$.
- $B(j_k) = 1$ for all k and $B(m_l) = n$ for all l.
- $\tau = 0$.

We now explain the reduction above. There is an edge between job j_k and machine m_l if m_l belongs to $S(j_k)$, the set of machines in which job j_k can be processed. There is a conflict edge between job j_k and job j_l if their time intervals for processing overlap. The weight on an edge (j_k, m_l) represents the revenue obtained if job j_k is processed on machine m_l. Since each job can be assigned to at most one machine, their degree constraints are set to 1. There is no constraint on the number of jobs assigned to any machine and hence their degree constraint is set to n. Finally, there must be no conflict between two jobs assigned to same machine. Therefore, τ is set to 0.

It can be easily seen that an optimal solution for $G(I)$, an instance of CA-WBM yields an optimal solution for I, an instance of RMIS. In other words, a maximum weight subgraph of $G(I)$ satisfying the degree constraints and conflict constraints as described above exactly corresponds to a revenue maximizing schedule in I. Furthermore, we observe that the reduction above is a polynomial-time reduction. Therefore we conclude that CA-WBM is NP-hard. □

3.4 Algorithms for Solving CA-WBM

3.4.1 A SDP Algorithm for CA-WBM

In the previous section, we showed that CA-WBM is NP-hard. In this and following sections, we will design efficient algorithms for CA-WBM that provide high-quality solutions that are close to optimal.

Our first algorithm for CA-WBM is based on a semidefinite programming approach. To understand the motivation for this approach, recall that we described a IP formulation in Section 3.2. Using the terminology from Section 3.2 and 3.3, the conflict constraint can be described as follows:

$$\sum_{(j_k,j_l)\in C} x_{ki}x_{li} \leq \tau \ \forall \ i \in U \tag{3.3}$$

That is, the conflict constraint is quadratic. We use a single τ for illustration purpose. In practice, different sellers can have different values of τ. We will now show how to formulate CA-WBM as a semidefinite program. Define a $mn \times mn$ symmetric matrix $\mathbb{Y} = XX^T$ where X is as in Section 3.2. The CA-WBM problem can be described as

$$\begin{aligned} \max \quad & Trace(\mathbb{W}\mathbb{Y}) \\ s.t. \quad & Trace(\mathbb{D}_i^b\mathbb{Y}) \leq D(i), \forall \ i \in V \\ & Trace(\mathbb{D}_i^s\mathbb{Y}) \leq D(i), \forall \ i \in U \\ & Trace(\mathbb{C}_i\mathbb{Y}) \leq \tau, \forall \ i \in U \\ & \mathbb{Y} = XX^T \succeq 0 \end{aligned} \tag{3.4}$$

where $\mathbb{W}, \mathbb{D}_i^b, \mathbb{D}_i^s$ and $\mathbb{C}_i$ are suitably defined $mn \times mn$ symmetric matrices described below.

1. $\mathbb{W}$ is a diagonal matrix with diagonal weights w_{ij}.
2. $\mathbb{D}_i^b$ is diagonal matrix with a 1 for row indexed by (i,j) if $(i,j) \in E$ and 0 otherwise.
3. $\mathbb{D}_i^s$ is diagonal matrix with a 1 for row indexed by (k,i) if $(k,i) \in E$ and 0 otherwise.

4. Finally, $\mathbb{C}_i$ is a matrix with entries 1/2 and 0. An entry indexed by row (j,i) and column (k,i) is equal to 1/2 if $(j,i),(k,i) \in E$ and $(j,k) \in C$. It is 0 otherwise.

Our SDP based algorithm for CA-WBM is as follows:

1. Solve the semidefinite program relaxation to obtain optimal solution $\mathbb{Y}$. From now on, we refer to $Trace(\mathbb{W}\mathbb{Y})$ as the *SDP optimal*.

2. Using the Cholesky decomposition [114] of $\mathbb{Y}$, obtain the vectors x_{ij} corresponding to $\mathbb{Y}$.

3. Use a two-step rounding procedure, random projection [54] followed by threshold rounding, to obtain $0, 1$ values for x_{ij}. The result after this step is referred to as the *SDP with rounding*.

In step (1) of our SDP algorithm, the SDP described above is solved using a generic SDP solver. The output of step (1) is a semidefinite matrix $\mathbb{Y}$. In step (2) of our algorithm, we use a well-known fact that any semidefinite matrix $\mathbb{Y}$ can be written as $\mathbb{Y} = \mathbb{V}\mathbb{V}^T$ where $\mathbb{V}$ is a $mn \times mn$ lower triangular matrix. This decompostion is known as Cholesky decomposition of $\mathbb{Y}$ [114]. The columns of $\mathbb{V}$ give us a vector solution for the variables x_{ij}. Thus, the output of step (2) of our algorithm are mn vectors, one for each x_{ij}. These vectors correspond to the optimal solution of the SDP. In the last step of our algorithm, we convert the vectors x_{ij} to integral $\{0, 1\}$ values using a two-step rounding procedure. In the first step, we convert x_{ij}'s to fractional values by a random projection [54]. That is, we pick a random vector x of dimension mn by picking each of its coordinates from the normal distribution $N(0, 1)$ and define each $x_{i,j}$ as the length of its projection on to x. Finally, we sort x_{ij} and round each non-zero value to 1 provided doing so does not violate the degree constraints or the conflict constraints. Otherwise, we set it to 0.

3.4.2 ILP Formulation of CA-WBM

Solving the SDP formulation requires large physical memory to store the $mn \times mn$ matrix $\mathbb{Y}$. In practice (e.g., the eBay purchase graph of a certain category), large values for m (the number of buyers) and n (the number of sellers) inevitably restrict the applicability of the SDP approach. This limitation, however, can be alleviated if we could model CA-WBM as an integer linear programming (ILP) problem.

In order to achieve this goal, we introduce a new 0-1 variable $z_{i,(j,k)}$ to formulate Inequality 3.3 as a linear constraint. For each seller i, $z_{i,(k,l)}$ equals 1 if and only if there is a conflict edge between two buyers k and l, and both edge e_{ki} and e_{li} are recommended in the graph. Using the terminology from Sections 3.2 and 3.3, this constraint can be described as follows:

$$1 - x_{ki} - x_{li} + z_{i,(k,l)} \geq 0, \ \forall \ i \in U, \ \forall (k,l) \in C_i \tag{3.5}$$

$$x_{ki} + x_{li} - 2z_{i,(k,l)} \geq 0, \ \forall \ i \in U, \ \forall (k,l) \in C_i \tag{3.6}$$

$$\sum_{(k,l) \in C_i} z_{i,(k,l)} \leq \tau, \ \forall \ i \in U \tag{3.7}$$

In constraints (5), (6) and (7), C_i is defined as follows: $C_i = \{(k,l) \in C | (k,i) \in E \wedge (l,i) \in E\}$. That is, C_i represents the set of conflicts within the set of buyers linked to seller i.

The linear conflict constraints can be easily incorporated into Problem 3.1. Let c denote the total number of conflict constraints with respect to all sellers in the graph. Then we can obtain a linear programming formulation, where $X = [x_{ij}]^T$ is a $(mn + c)$-dimensional column vector of 0-1 variables. In addition, matrix $\mathbb{A}$ and vectors W and B can be changed accordingly.

By eliminating the need to store large dimensional matrices, now we can use an ILP solver to tackle CA-WBM problems of larger sizes. Since obtaining an integer solution in CA-WBM is NP-hard, in order to further improve efficiency, we use a rounding procedure after solving the linear program relaxation. Our LP based algorithm for CA-WBM is as follows:

1. Solve the linear program relaxation to obtain optimal solution X.

2. Sort the first mn elements of X from largest to smallest. We round each non-zero value to 1 provided doing so does not violate the degree constraints or the conflict constraints. Otherwise, we set it to 0. The result after this step is referred to as the *LP relaxation with rounding*.

3.4.3 A Greedy Algorithm for CA-WBM

In this section, we describe and study the performance of a simple greedy algorithm for this problem. This algorithm has the advantage that it is highly scalable and provides good quality solutions in practice.

The greedy algorithm, denoted as GREEDY, for CA-WBM is as follows:

1. Sort all the edges in E by weights from largest to smallest.
2. To construct the maximum weight subgraph H, consider every edge in the sorted list. Add this edge to H if doing so does not violate any degree constraint or conflict constraint.
3. Continue until we reach the end of the sorted list.

We will now prove a theoretical guarantee on the performance of GREEDY.

Theorem 2. *Let $d = \max_{v \in V} |\{(v, v') | (v, v') \in C\}|$. Algorithm GREEDY is a $(2+d)$-approximation algorithm.*

Proof. We use the concept of a *k-extendible system* to provide performance guarantees of a greedy algorithm. Mestre introduced the notion of a k-extendible system in his study of the performance of the greedy technique as an approximation algorithm [101].

Definition 1 (k-Extendible System [101])**.** *Let U be a finite set and $\mathcal{F}$, $\mathcal{F} \subseteq 2^U$, be a collection of subsets of U. Set system $(U, \mathcal{F})$ is called a* k-extendible system *if it satisfies the following properties:*

1. Downward-closure*: If $A \subseteq B$ and $B \in \mathcal{F}$, then $A \in \mathcal{F}$.*
2. Exchange*: Let $A, B \in \mathcal{F}$ with $A \subseteq B$, and let $x \in U - B$ be such that $A \cup \{x\} \in \mathcal{F}$. Then there exists $Y \subseteq B - A$, $|Y| \leq k$, such that $(B - Y) \cup \{x\} \in \mathcal{F}$. In other words, let us start with any choice of two sets A and B such that B is an extension of A. Suppose that there is an element x such that the set A with x added to it also belongs to $\mathcal{F}$. Then we will be able to find a subset Y inside B of size at most k such that if we remove the elements of Y from B and add the element x to the resulting set, it will also belong to the collection $\mathcal{F}$.*

Informally, Mestre showed that if the set of all feasible solutions forms a k-extendible system, algorithm GREEDY gives a k-approximation algorithm. That is, on any instance, the solution output by GREEDY differs from the optimal solution by a multiplicative factor of at most k. We now state his result more formally.

Theorem 3 (Mestre [101])**.** *Let $(U, \mathcal{F})$ be a k-extendible system for some k. Let $W : U \to \mathbb{R}^+$ be a positive weight function on U. The greedy algorithm gives a k-approximation algorithm for the optimization problem that asks to determine $\max_{F \in \mathcal{F}} W(F)$ where $W(F) = \sum_{s \in F} W(s)$ for any $F \in \mathcal{F}$.*

To apply this result to our problem, we will check that the set of all feasible solutions to CA-WBM forms a $(2+d)$-extendible system. For the CA-WBM problem, $U = E$ and $\mathcal{F}$ be the set of all subgraphs of G satisfying the degree and conflict constraints. Then it is easy to see that $(U, \mathcal{F})$ is downward closed. That is, removing an edge from a feasible solution H will always result in be a feasible solution as this will not cause any violation of constraints.

For the exchange property, consider the case when a new edge $e = (u, v)$, $u \in U$ and $v \in V$, is added to a feasible solution H. We make two observations: (1) Adding e could result in violation of degree constraint at u and v. However, this can be rectified by removing two other edges, one incident on u and other incident on v; (2) Adding e could result in a violation of the conflict constraint at v. Rectifying this could require removing at most d edges where $d = \max_{v \in V} |\{(v, v')|(v, v') \in C\}|$. Therefore, we obtain a $(2 + d)$-extendible system. □

We remark that this analysis is worst-case. In practice, GREEDY shows far superior performance, as demonstrated in our later test with real-world datasets.

3.5 Online CA-WBM and A Randomized Algorithm

In real-world scenarios, we may not know all buyers/sellers in advance. In online business, the set of sellers is relatively stable compared to the set of buyers. As such, we mainly focus on one typical problem where new buyers join the system as time goes. Other variations of the problem, e.g., both new buyers and new sellers joining the system, could be studied with the idea presented in this section, but their analysis remains challenging.

In the online version of CA-WBM, an algorithm for the problem only knows the set U (the fixed vertex set, e.g., sellers) when it starts, and the sequence of vertices belonging to V (the arriving vertices set, e.g., buyers $b_1, b_2, ..., b_m$) arrives online one by one. When a vertex $b_i \in V$ arrives, all edges incident to b_i, as well as their weights, and conflict edges associated with other buyer vertices which arrived earlier, are revealed. The algorithm should immediately make recommendation to b_i and cannot change the recommendation at a later time.

Ting and Xiang [131] proposed a near optimal randomized algorithm for online bipartite maximum weighted b-matching. According to their definition, while each

fixed vertex $s \in U$ can be matched to at most $B(s)$ arrival vertices, each arrival vertex can be matched to only one fixed vertex. We extend their algorithm and analysis by incorporating the conflict constraint and allowing the degree constraint of each arrival vertex to be larger than 1.

3.5.1 Assumptions and Settings

Fixed Vertex

Denote the degree constraint of a fixed $s \in U$ by $B(s)$ and we assume $B(s) \geq 1$. To ease the analysis, we make $B(s)$ copies of each fixed vertex s and form them as a group. An example is shown in Figure 3.3. In addition, recall that in CA-WBM, the number of conflict edges within a list of arriving vertices matched to any particular fixed vertex is smaller than a threshold τ.

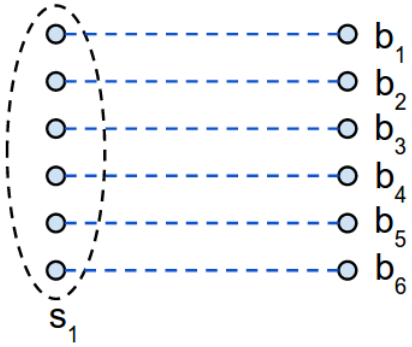


Figure 3.3: An example: copies of a fixed vertex s_1. $(b_1, ..., b_6)$ are adjacent arriving vertices of s_1. Assume that the degree constraint of s_1 is 6 and the conflict threshold τ is 0. Each copy of s_1 can be matched to only one arriving vertex. The 6 copies as a whole cannot be matched to any conflict pair of arriving vertices.

Arriving Vertex

Let $B(b)$ denote the degree constraint of an arriving vertex $b \in V$. In the following, we first study the special case where each arriving vertex can be matched to only one fixed vertex, i.e., the degree constraint of each b is 1. Later, we will extend our result to the general case where the degree constraint of each arriving vertex is more than one ($B(b) > 1$).

Edge Weight

In addition to the fixed vertex set U, we assume that the maximum weight of edges ($w_{\max}$) can be estimated and is known to the algorithm. In the matching scenario of e-commerce, this is a reasonable assumption since the system can estimate the maximum amount of money based on transaction records from buyers and sellers.

3.5.2 The Randomized Algorithm for $B(b) = 1$

In this special case, the degree constraint of each fixed vertex is 1. Inspired by the algorithm in [131], we propose a randomized algorithm for online CA-WBM, ***Randomized-CA-WBM***, that takes the constraint on conflicts into account.

Algorithm 1: ***Randomized-CA-WBM***

Input: A fixed vertex set U, the maximum edge weight $w_{\max}$
Output: Generate matched edges on the fly

Let $g = \lceil \ln(1 + w_{\max}) \rceil$, choose an integer k uniformly from $\{0, 1, 2, ..., g-1\}$;
Set $\gamma = e^k$;
while *a new vertex* $b \in V$ *arrives* **do**
    $T = \{\hat{s} \mid \hat{s}$ is a copy vertex of s, which is a fixed vertex incident to b in V and $w((s, b)) \geq \gamma\}$;
    if $T = \emptyset$ **then**
        leave b unmatched forever;
    else
        match b to an arbitrary vertex in T, if doing so does not violate its conflict constraint (When $T \neq \emptyset$, a vertex b may not be matched to any vertex due to the conflict constraint.)

The performance of an online algorithm is often analysed with competitive analysis, proposed by Sleator and Tarjan [128]. In our weight maximization problem, let $\mathcal{A}$ denote a randomized online algorithm, let σ denote the arrival sequence, and let $\mathbb{E}[w(\mathcal{A}(\sigma)]$ be the expected solution output by $\mathcal{A}$ when processing the arrival sequence σ. Let $w(\mathcal{OPT}(\sigma))$ denote the output of the optimal offline algorithm $\mathcal{OPT}$ when processing σ. We say that a randomized online algorithm $\mathcal{A}$ is *R-competitive*, if the ratio of $\mathbb{E}[w(\mathcal{A}(\sigma)]$ to $w(\mathcal{OPT}(\sigma))$ is at least $1/R$,

$$\frac{\mathbb{E}[w(\mathcal{A}(\sigma)]}{w(\mathcal{OPT}(\sigma))} \geq \frac{1}{R}$$

for all arrival sequences σ. The smallest such R is called the competitive ratio of $\mathcal{A}$. To evaluate ***Randomized-CA-WBM***, we compare its performance to that of the optimal offline algorithm. Theorem 4 gives the upper bound of the competitive ratio of ***Randomized-CA-WBM***.

Theorem 4. ***Randomized-CA-WBM*** *achieves a competitive ratio of* $(\alpha+1)e\lceil\ln(1+w_{\max})\rceil$, *where* $\alpha = \max(d_1-1, d_2)$, *and* $d_1 = \max_{s\in U} B(s)$ *and* $d_2 = \max_{b\in V} |\{(b,b')|(b,b') \in C\}|$.

In order to prove Theorem 4, we extend and adapt the analysis in [131] so that it applies to Algorithm 1. Let E' denote the set of edges output by ***Randomized-CA-WBM*** for the graph $G = \langle (U,V), E, W\rangle$ and $C \subseteq V \times V$ with any arrival sequence on V, and let $S(E') = \{s \in U \mid \exists\, b \text{ s.t.}(s,b) \in E'\}$ be the set of fixed vertices' ends of the edges in E'. Let $w(E') = \sum_{(s,b)\in E'} w((s,b))$ be the total weight of edges in E', where $w((s,b))$ is the weight of the edge (s,b). For any $i \geq 0$, let $E'_{\geq e^i}$ be the result if the threshold τ is e^i. Then the expectation of $w(E')$ for any arrival sequence is $\mathbb{E}[w(E')] = \sum_{0\leq i\leq g-1} w(E'_{\geq e^i})\frac{1}{g}$.

Denote O as the optimal maximum weight subgraph of G. Let $O_{[e^i,e^{i+1})} = \{x \in O | w(x) \in [e^i, e^{i+1})\}$ be the set of edges in O, with each edge's weight $w(x) \in [e^i, e^{i+1})$.

Lemma 1. $\forall i \in \{0,1,...,g-1\}$, $|S(O_{[e^i,e^{i+1})}) - S(E'_{\geq e^i})| \leq \alpha\, |S(E'_{\geq e^i})|$, *where* $\alpha = \max(d_1-1, d_2)$, *and* $d_1 = \max_{s\in U} B(s)$ *and* $d_2 = \max_{b\in V} |\{(b,b')|(b,b') \in C\}|$.

Proof. Consider a copy vertex $\hat{s}$ of any fixed vertex s, and $\hat{s} \in S(O_{[e^i,e^{i+1})}) - S(E'_{\geq e^i})$. Note that $\hat{s} \in S(O_{[e^i,e^{i+1})})$ but $\hat{s} \notin S(E'_{\geq e^i})$. The refusal of matching $\hat{s}$ to a valid arriving vertex b (since $w((\hat{s},b)) \geq e^i$) in the result output by ***Randomized-CA-WBM*** suggests two possible cases,

1. b is matched to another $\hat{s}' \in S(E'_{\geq e^i})$
2. b is unmatched due to the conflict constraint

For every vertex $\hat{s} \in S(O_{[e^i,e^{i+1})}) - S(E'_{\geq e^i})$ of the first case, each of them can be mapped to a unique vertex $\hat{s}' \in S(E'_{\geq e^i})$.

For the second case, we can analyze the matching result of the worst case for a certain fixed vertex s_1, as shown in Figure 3.4.

Let $degree_{s_1}$ be the degree constraint of s_1 and $conflict_{(s_1,b)}$ be the largest possible number of conflict edges associated with any s_1's neighboring arriving vertex. For

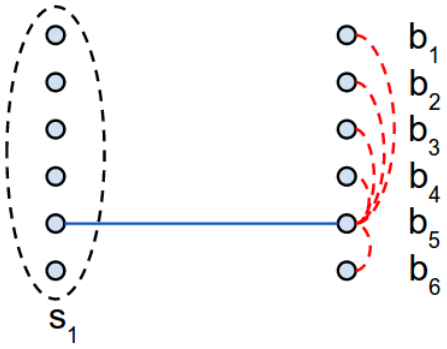


Figure 3.4: The worst case: if one of the copies is matched to an arriving vertex, none of the rest can be matched due to the conflict threshold (which is 0 in the worst case). b_5 arrives first and it is in conflict with other vertices.

example, in Figure 3.4, $degree_{s_1} = 6$ and $conflict_{(s_1,b)} = 5$ (b_5 is in conflict with 5 arriving vertices).

For a fixed vertex s_1, if $(degree_{s_1} - 1) \leq conflict_{(s_1,b)}$, the largest possible number of unmatched copies is $(degree_{s_1} - 1)$; otherwise, the number is $(conflict_{(s_1,b)})$.

If we consider the case when $d_1 = \max_{s \in U} B(s)$ and $d_2 = \max_{b \in V} |\{(b, b')|(b, b') \in C\}|$ are known ahead of time, up to $\max(d_1 - 1, d_2)$ vertices in $S(O_{[e^i,e^{i+1})}) - S(E'_{\geq e^i})$ can be mapped to a unique vertex $s' \in S(E'_{\geq e^i})$; then we have the lemma. □

We remark that this upper bound is tight. For example, in the worst case shown in Figure 3.4, $d_1 = 6$ and $d_2 = 5$. Suppose that in the optimal solution, b_5 is not matched to s_1, then all other arriving vertices are able to be matched. Thus, $|S(O_{[e^i,e^{i+1})}) - S(E'_{\geq e^i})| = 5$, $|S(E'_{\geq e^i})| = 1$, and $\max(d_1 - 1, d_2) = 5$. In addition, if the conflict threshold τ becomes larger than 0, more copies of s_1 will be allowed to be matched to arriving vertices. Then the number of unmatched copies is less than $\max(d_1 - 1, d_2)$. Therefore, the upper bound still holds.

Lemma 2. *$\forall i \in \{0, 1, ..., g - 1\}$, $w(E'_{\geq e^i}) \geq \frac{1}{(\alpha+1)e} w(O_{[e^i,e^{i+1})})$, where $\alpha = \max(d_1 - 1, d_2)$.*

Proof. Since each copy ($\hat{s}$) of any fixed vertex s can be matched to only one arriving vertex b, the number of matched fixed copies is equal to the number of matched edges.

By Lemma 1, for any $i \in \{0, 1, ..., g-1\}$, we have

$$\begin{aligned}
|O_{[e^i,e^{i+1})}| &= |S(O_{[e^i,e^{i+1}))}| \\
&= |S(O_{[e^i,e^{i+1})}) \cap S(E'_{\geq e^i})| + \\
&\quad |S(O_{[e^i,e^{i+1})}) - S(E'_{\geq e^i})| \\
&\leq (\max(d_1 - 1, d_2) + 1)|S(E'_{\geq e^i})| \\
&= (\max(d_1 - 1, d_2) + 1)|E'_{\geq e^i}|
\end{aligned}$$

then

$$\begin{aligned}
w(E'_{\geq e^i}) \geq e^i|E'_{\geq e^i}| &\geq \frac{e^i}{\max(d_1 - 1, d_2) + 1}|O_{[e^i,e^{i+1})}| \\
&\geq \frac{1}{(\max(d_1 - 1, d_2) + 1)e} w(O_{[e^i,e^{i+1})})
\end{aligned}$$

the lemma follows. □

Now, we can prove Theorem 4 using Lemma 2.

Proof.

$$\begin{aligned}
\mathbb{E}[w(E')] &= \sum\nolimits_{0 \leq i \leq g-1} w(E'_{\geq e^i}) \frac{1}{g} \\
&\geq \frac{1}{(\alpha + 1)eg} \sum\nolimits_{0 \leq i \leq g-1} w(O_{[e^i,e^{i+1})}) \\
&= \frac{1}{(\alpha + 1)eg} w(O).
\end{aligned}$$

□

3.5.3 The Randomized Algorithm for $B(b) \geq 1$

In general, each arriving vertex can be matched to more than one fixed vertex. By making copies of each arriving vertex upon arrival, Algorithm 1 for $B(b) = 1$ can be shown to adapt well with little modification. Let $B(b_1)$ denote the degree constraint of a arriving vertex b_1. Upon arrival of this arriving, the algorithm immediately makes $B(b_1)$ copies, $(b_{1,1}, ..., b_{1,B(b)})$. Instead of grouping the copies, the algorithm considers one copy at a time. We require that each arriving vertex copy can be matched to at most one fixed vertex, and two copies of the same arriving cannot be matched to the

same fixed vertex. Since the input of CA-WBM remain the same (i.e., a set of fixed vertices U and the maximum edge weight $w_{\max}$), the theoretical analysis also applies.

3.5.4 The Lower Bound on Competitive Ratio

Ting et al. [131] proved that for the online maximum weighted b-matching problem, no randomized algorithm can be better than $\frac{\lceil \log_2(w_{\max}+1)\rceil+1}{2}$-*competitive*. Since their problem, online maximum weighted b-matching problem, is a special case of our problem, we get the same lower bound as in [131] showing that the performance of our algorithm, Randomized CA-WBM, is near optimal.

Theorem 5. *For online CA-WBM, no randomized algorithm can achieve a competitive ratio better than* $\frac{\lceil \log_2(w_{\max}+1)\rceil+1}{2}$.

3.6 Experimental Evaluation

In this section, our main focus is to illustrate the proposed algorithms' optimality and scalability for CA-WBM and online CA-WBM. Specifically,

- CA-WBM. Since CA-WBM is proven to be NP-hard, we mainly focus on the performance of the proposed approximate algorithms. For the SDP formulation, we evaluate its performance by comparing the optimal solution and the solution obtained by the rounding procedure for several combinations of degree and conflict constraints. For the ILP formulation, we perform experiments to compare the integral solution with the result of the linear programming (LP) relaxation with rounding on much larger graphs. Finally, we demonstrate the scalability of our greedy algorithm.

- Online CA-WBM. We evaluate the proposed randomized algorithm by repeating experiments for different random arrival sequences. We calculate the competitive ratio for each run and compare the average behavior to the upper bound.

We conducted comprehensive experiments with eBay's transactional data provided by Terapeak. The original data consists of three-month transactions across all categories of eBay Canada and eBay US in 2013. We used the data of a specific category (Cell phones and Accessories) from both eBay Canada and eBay US. In order to

test the algorithms on datasets of different scales, we created three datasets for evaluation: a small-scale synthetic dataset (eBay Canada), a moderate-scale synthetic dataset (eBay Canada), and a large-scale real-world dataset (eBay US).

The datasets of eBay Canada are called "*synthetic*" because we changed the original graph structure with an imputation step to generate bipartite graphs of different sizes. The details about how we created synthetic datasets will be explained in Sections 3.6.1, 3.6.1 and 3.6.1. The real-world eBay US dataset contains purchase information of transactions between buyers and sellers in the category of cell phones and accessories. Note that even though we used data from Terapeak, similar transaction data can also be collected via the eBay API[②]. The basic information of each dataset is summarized in Table 3.1.

Experiments with the small-scale and moderate-scale datasets were run on a 64-bit Ubuntu 12.04 desktop of 3.40GHz * 8 Intel Core i7 CPU and 3.8 GB memory. Experiments with the large-scale dataset were run on a similar desktop with 12 GB memory.

	synthetic-small	synthetic-moderate	eBay US
Algorithms tested	SDP optimal, SDP with rounding, ILP, GREEDY	ILP, LP with rounding, GREEDY	GREEDY
Number of buyers	26	18,742	5,751,334
Number of sellers	5	1,884	126,101
Number of edges	50	56,520	11,387,517

Table 3.1: Basic Information of Synthetic and Real-world Datasets

3.6.1 CA-WBM

To create synthetic datasets for small-scale and moderate-scale experiments, we use eBay Canada data and firstly sort buyers, as well as sellers, by the total monetary purchases and total profits, respectively. Adding edges between sorted buyers and sellers, we can create bipartite graphs of different density settings. In the experiments, edges are generated in a manner that each seller has the same number of connected buyers. For example, if the density is set to 0.5%, each seller will be connected to roughly 90 buyers (calculated from the synthetic-moderate data in Table 3.1). The

[②] `https://go.developer.ebay.com/`

first seller (top ranked) is connected with the first 90 buyers (from 1 to 90), and the second seller is connected with buyers from 11 to 100, and so on. The weight of an edge is the sum of the buyer's total monetary purchases (on all sellers within the category) and the seller's total profits (from all buyers within the category). Of course, any monotone weight function can be used. For each buyer (seller) in the bipartite graph, the degree constraint ratio is the proportion of the maximum number of matched sellers (buyers) among all candidates. Every node (buyer or seller) in the experiment has the same degree constraint ratio.

We also introduce conflict edges between pairs of buyers. The requirement could be set by sellers who do not want to be recommended many buyers that are in conflict with each other, e.g., buyers who share similar features such as the same address. By limiting the number of conflicting buyers, we could promote diversity of the matching results for sellers.

In the following experiments, we randomly create conflicts between buyers according to different conflict pair ratios. This ratio is the number of buyer pairs in conflict divided by the total number of buyer pairs.

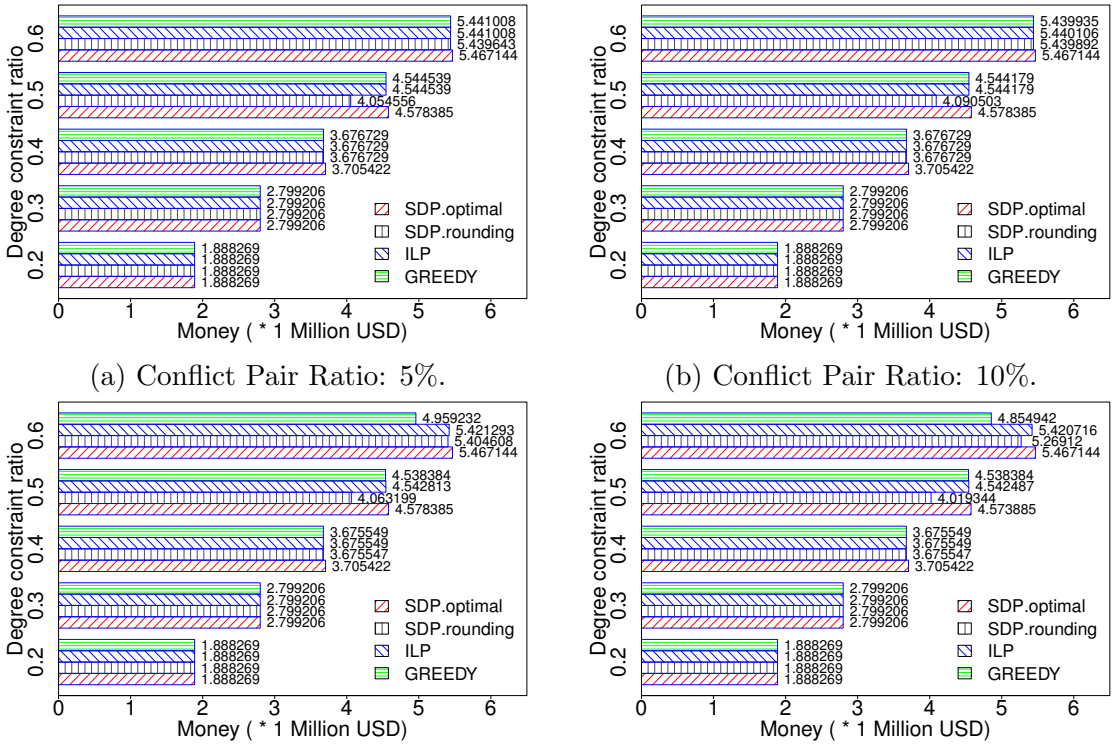
CA-WBM on Small-scale Synthetic Datasets

As discussed in Section 3.2, the SDP formulation of *CA-WBM* can be solved by a SDP solver plus a rounding procedure. We use SDPT3 [132, 133] as the SDP solver. For ILP, we use a fast linear programming (LP) solver, Gurobi[③] for Matlab, to solve *CA-WBM* at different scales.

The applicability of the SDP approach for *CA-WBM* severely suffers from the limitation of physical memory, i.e., the need to store a large-dimensional matrix [82]. In the real-world scenario, the dimension of the matrix can easily reach hundreds of thousands or more, which is beyond the capacity of a stand-alone machine. Due to this reason, we can only test the performance of the SDP approach in small-scale datasets.

We create a subgraph of small size consisting of top 5 sellers and top 26 buyers (ranked by money). Each seller is connected with 10 buyers so that each buyer can be assigned to multiple sellers. Specifically, the first seller is connected with buyers from 1 to 10, the second seller is connected with buyers from 5 to 14, and so on. We use two types of edge weight, money and node rank. Money weight is computed in the

[③]http://www.gurobi.com/



(a) Conflict Pair Ratio: 5%.

(b) Conflict Pair Ratio: 10%.

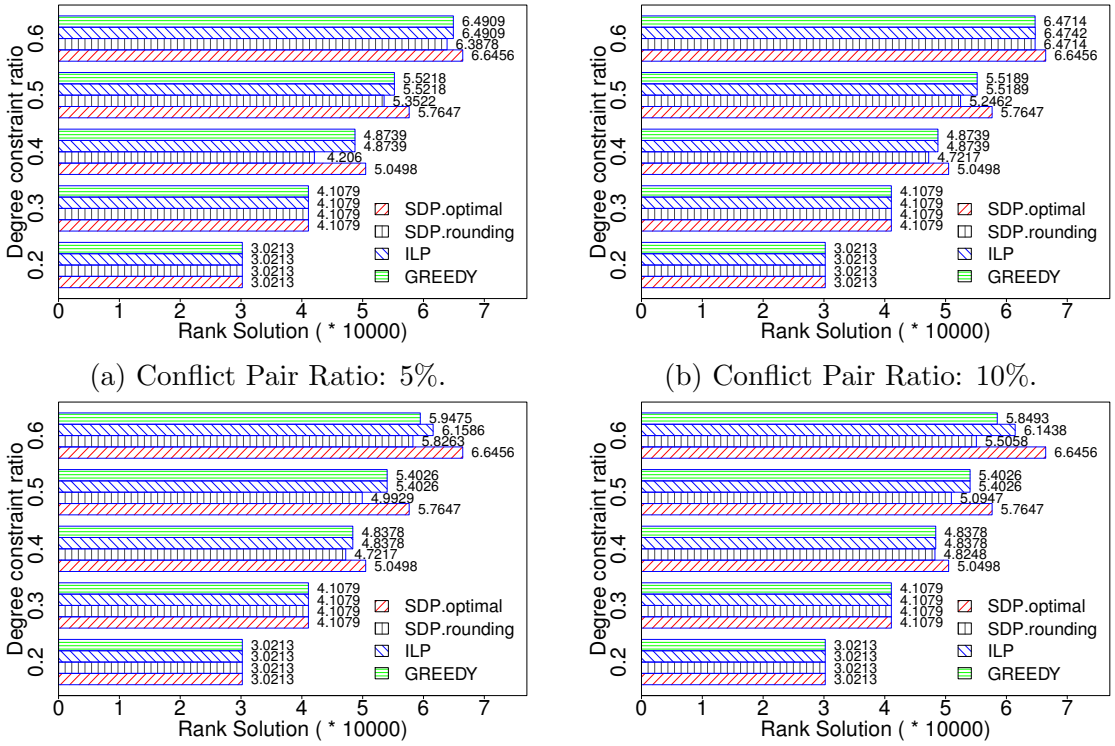
(c) Conflict Pair Ratio: 15%.

(d) Conflict Pair Ratio: 20%.

Figure 3.5: Money solution of different conflict pair ratios. When degree constraint ratio and conflict pair ratio are both low, GREEDY shows close to optimal solutions while SDP with rounding is weaker. Values of each bar are actual solutions.

same way described earlier, i.e., the sum of the buyer's total monetary purchases (on all sellers within the category) and the seller's total profits (from all buyers within the category). Of course, any monotone weight function can be used. Rank weight equals the multiplication of a big constant (the total number of buyer and seller nodes in the entire graph, $20,626$ in our case) and the reciprocal of the sum of the buyer's rank and the seller's rank.

We create different number of conflict buyer pairs by randomly sampling the number of buyer pairs with the following percentages of total possible buyer pairs, $\{5\%, 10\%, 15\%, 20\%\}$. We set different degree constraint ratios for each node, {20%, 30%, 40%, 50%, 60%}. In addition, we use a constant ("1") for each seller's conflict constraint, which means each seller can at most accept one conflict buyer pair. We use this constant to amplify the impact of conflict constraint on the small-scale subgraph. For SDP with rounding, we run the randomized rounding procedure 20 times and take the best solution.



(a) Conflict Pair Ratio: 5%.

(b) Conflict Pair Ratio: 10%.

(c) Conflict Pair Ratio: 15%.

(d) Conflict Pair Ratio: 20%.

Figure 3.6: Rank solution of different conflict pair ratios. GREEDY achieves close to optimal performance in all cases. The solutions obtained by SDP with rounding are slightly worse compared to GREEDY. Values of each bar are actual solutions.

Figure 3.5 and Figure 3.6 depict the comparison of different approaches, including SDP based approaches (Section 3.4.1), ILP (Section 3.4.2), and GREEDY (Section 3.4.3), for the money weight and rank weight, respectively. The result obtained by ILP is the integral optimal.

In Figure 3.5, we observe that regardless of conflict pair ratio, solutions of the different methods are very similar when degree constraint ratio is smaller than 60%, with SDP with rounding being slightly weaker than others. The reason is that when the degree constraint is tight, the conflict constraint of each seller is less likely to be activated, i.e., chances are rare for multiple conflict buyers to be matched to a seller. The difference arises when the degree constraint ratio and the conflict pair ratio are both weak and the higher conflict pair ratio results in larger performance drop for both SDP with rounding and GREEDY (e.g., comparing the sets of bars of 60% degree constraint ratio in (c) and (d)).

Figure 3.6 shows a similar performance change trend for the approaches. For example, the solution difference becomes larger when degree constraints are weaker and the performance of SDP with rounding and GREEDY gradually decreases as the conflict pair ratio increases. Meanwhile, we also observe that the performance of GREEDY is always superior to that of SDP with rounding.

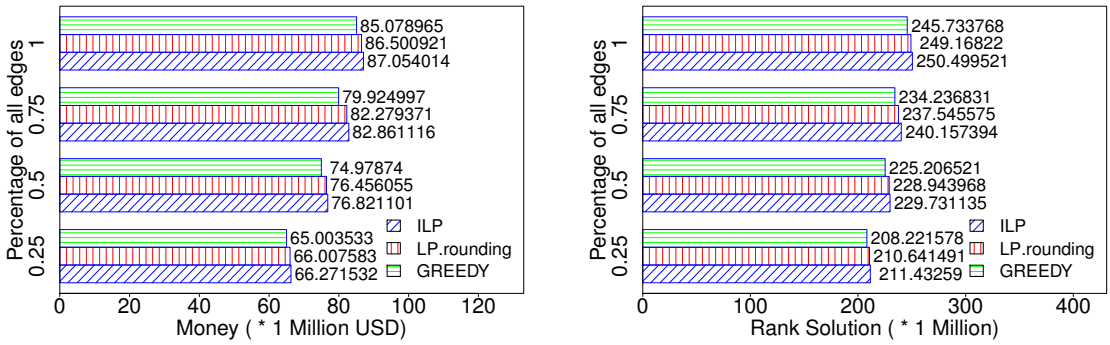
We also observe that both SDP with rounding and GREEDY achieve close to optimal solutions (compared to the result of ILP). Specifically in the experiments, GREEDY exhibits a superior performance compared to the theoretical analysis.

CA-WBM on Moderate-scale Synthetic Datasets

ILP formulation enables us to take full advantage of the LP solver (Gurobi) to solve CA-WBM problems with larger sizes. In this section, we perform moderate-scale experiments to compare solutions of different methods, i.e., ILP (Section 3.4.2), LP relaxation with rounding (Section 3.4.2) and GREEDY (Section 3.4.3).

We create a 0.16%-density bipartite graph using the same method described in Section 3.6.1. Note that the density of the corresponding real-world buyer-seller graph is 0.016%. The full graph consists of 18742 buyers, 1884 sellers and 56520 edges. We also extract three subsets of different sizes from the full graph, i.e., using 25%, 50% and 75% of the total number of edges (the number of buyer and seller nodes decreases accordingly). The degree constraint ratio, conflict pair ratio are 50% and 10%, respectively. The conflict threshold τ (refer to Section 3.2) for each seller is set to be 50% of the total number of conflicting buyer pairs associated to the seller. Figure 3.7 shows the solution comparison of different methods on different datasets.

Figure 3.7 shows very promising results for LP relaxation with rounding and GREEDY algorithms on both money weights and rank weights; they are only slightly worse than the optimal integral solution obtained by ILP. Comparing to the SDP experiments, this experiment also verifies the effectiveness of both LP relaxation with rounding and GREEDY on moderate datasets because the graph we use in this section is considerably larger than the one used for SDP experiments. The density (0.16%) is also 10 times larger than the corresponding real-world buyer-seller graph (0.016%). Therefore, our ILP formulation improves the scalability of solving moderate-scale CA-WBM problems.



(a) Money Solution.

(b) Rank Solution.

Figure 3.7: ILP experiments of CA-WBM on moderate-scale datasets. Both LP relaxation with rounding and GREEDY achieve close to optimal solutions. Values of each bar are actual solutions.

CA-WBM on Large-scale Real-world Dataset

When the size of the graph grows larger, however, the ILP formulation and the corresponding LP relaxation become untameable with existing ILP/LP solvers due to the enormous number of variables (as shown in Table 3.2) and prohibitive computational requirement. In this case, GREEDY shows its most important advantage that it scales very well when applied to even larger datasets. We run GREEDY on the large-scale eBay US dataset, which contains $5,751,334$ buyers, $126,101$ sellers and $11,387,517$ edges. The weight of edge between a buyer and a seller represents the total amount of money spent by the buyer on this seller. To show its scalability as graph size increases, we extract three subsets of different sizes from the full graph, i.e., using 25%, 50% and 75% of the total number of edges. The degree constraint ratio, conflict pair ratio are 20% and 1%, respectively. A larger conflict pair ratio results in more z variables. The conflict threshold τ (refer to Section 3.2) for each seller is set to be 20% of the total number of conflicting buyer pairs associated to the seller. Table 3.3 summarizes the statistical information of each subset. The running time of GREEDY for each of them is shown in Figure 3.8.

	25%	50%	75%	100%
# constraints	$6,022,924$	$6,860,587$	$8,488,620$	$10,043,227$
# variables	$2,935,592$	$6,183,430$	$9,843,523$	$13,467,108$
# z variables	$88,712$	$489,671$	$1,302,885$	$2,079,591$

Table 3.2: Problem Size of LP Formulation for Each Subset of eBay US

	25%	50%	75%	100%
# buyers	1,574,114	2,988,717	4,300,322	5,751,334
# sellers	66,751	90,925	109,511	126,101
# edges	2,846,880	5,693,759	8,540,638	11,387,517

Table 3.3: Basic Information of Three Subsets of eBay US

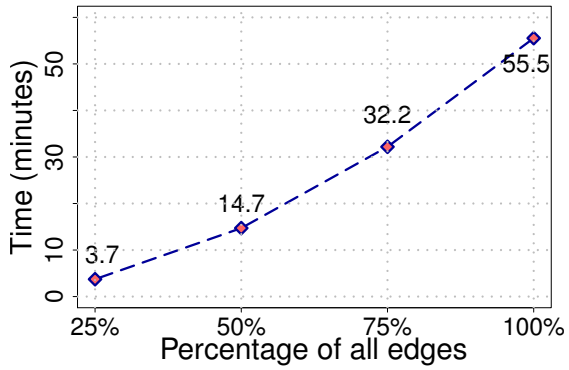


Figure 3.8: Greedy algorithm on large-scale datasets showing its scalability.

In Figure 3.8, the running time increases nearly linearly, and it only requires less than one hour to get a solution using a single desktop computer when the number of edges is considerably as large as 11,387,517.

3.6.2 Online CA-WBM

We use the same 0.16%-density bipartite graph as in Section 3.6.1 and we retain all edges (full size). The settings for degree constraint ratio, conflict pair ratio and conflict threshold τ also remain the same. We repeat 10,000 tests for money weight and rank weight, respectively. In each run, we randomly generate the arrival sequence for buyers and we compute its competitive ratio after it finishes. Figure 3.9 shows the box plot of competitive ratios for money and rank weight. In both cases, the majority of competitive ratios are located under 4. The average competitive ratio out of 10000 tests is 3.25 for money weight, and 3.63 for rank weight, respectively. Therefore, on average, the online algorithm indeed shows promising performance.

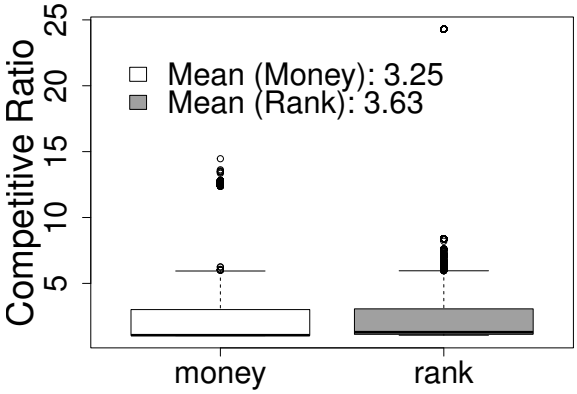


Figure 3.9: Competitive ratios of 10000 runs for Alg. 1. For money weight, first quartile, median and third quartile are 1.047, 1.067 and 3.012, respectively. For rank weight, the values are 1.14, 1.296 and 3.068, respectively.

3.7 More Discussions

3.7.1 Further Extensions of CA-WBM

The problem we propose and study in this chapter, CA-WBM (including its online version), is general and expressive. With further extension, it is able to model new recommendation (business) strategies. We list two modelling extensions of CA-WBM as follows.

1. Minimum degree constraint. For example, a seller may want his degree be larger than some constant c. That is, his product is recommended to at least c buyers. We use a $(m+n)$-dimensional column vector $LD : U \cup V \rightarrow \mathbb{N}$ to denote the lower bound on degree constraints of seller/buyer nodes. The values in LD can be greater than or equal to 0. Using LD, we can incorporate the following constraint in Problem 3.1

$$-\mathbb{A}X(i) \leq -LD(i), \forall i, 1 \leq i \leq m+n \tag{3.8}$$

2. Seller-dependent conflict constraint. Different sellers can specify different types of conflict. For example, some sellers prefer buyers from different cities and others may prefer buyers with different ages. In Section 3.4.2 of the main paper, the 0-1 variable $z_{i,(j,k)}$ only represents one type of conflict. If two or more types of conflict exist, we can create a list of z variables, such as $\mathbf{z} = [z^{(1)}_{i,(j,k)}, z^{(2)}_{i,(j,k)}, \cdots, z^{(Q)}_{i,(j,k)}]$, where Q is the number of different conflict types.

Consequently, constraints (5), (6), (7) in the TKDE main paper becomes:

$$1 - x_{ki} - x_{li} + z^{(j)}_{i,(k,l)} \geq 0,\ \forall\ i \in U,\ \forall (k,l) \in C^{(j)}_i \tag{3.9}$$

$$x_{ki} + x_{li} - 2z^{(j)}_{i,(k,l)} \geq 0,\ \forall\ i \in U,\ \forall (k,l) \in C^{(j)}_i \tag{3.10}$$

$$\sum_{(k,l) \in C^{(j)}_i} z^{(j)}_{i,(k,l)} \leq \tau^{(j)},\ \forall\ i \in U, \tag{3.11}$$

where $C^{(j)}_i = \{(k,l) \in C^{(j)} | (k,i) \in E \wedge (l,i) \in E\}$. That is, $C^{(j)}_i$ represents the set of type-j conflicts within the set of buyers linked to seller i. Furthermore, $\tau^{(j)}$ denotes the conflict threshold for conflict of type j.

Since the two extensions can be integrated into the ILP formulation of CA-WBM, we perform a small-scale, proof-of-concept experiment to evaluate the correctness of the formulation under varying constraints. We use the general LP solver, Gurobi [58], to obtain the optimal solution whenever feasible. The dataset we use is similar to the synthetic-small dataset in Table 3.1. Figure 3.10 shows the structure of a complete bipartite graph. It consists of 5 sellers and 26 buyers. Blue dashed lines and red dotted lines on the buyer side represent two different types of conflict. We assume that buyers $1, 2, 3, 4$ and 5 are in conflict with all other buyers in "blue" conflict, and buyer 26 is in conflict with the rest in "red" conflict. Edge weights are arbitrarily chosen from the set $\{1, 2, 3\}$. We use a constant ("1") for each seller's conflict threshold, which means that each seller can accept at most one conflict buyer pair.

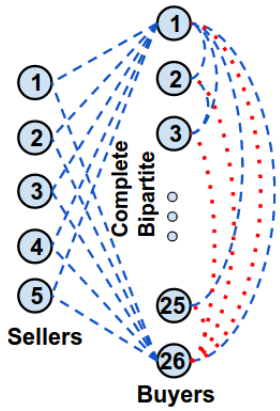


Figure 3.10: A bipartite graphs with two types of conflict. Edge weights are arbitrarily chosen from the set $\{1, 2, 3\}$.

We perform 4 tests, using different settings, on the extended formulation that includes the minimum degree constraint and seller-dependent conflict constraint. Ta-

	Test 1		Test 2		Test 3		Test 4	
Max. Seller Degree (MAXSD)	21		21		21		21	
Min. Seller Degree (MINSD)	16		17		17		18	
Seller-dependent Conflict (SDC)	Blue	Red	Blue	Red	Blue	Red	Blue	Red
1	✓		✓		✓		✓	
2	✓		✓		✓		✓	
3	✓		✓		✓		✓	
4	✓		✓		✓		✓	
5	✓		✓			✓		✓
ILP Solution	126		Infeasible		136		Infeasible	

Table 3.4: Experimental Settings and Optimal Solutions

ble 3.4 shows the ILP solutions with respect to different minimum degree constraints and seller-dependent conflict constraints. All sellers use the same maximum or minimum degree constraint setting. In Test 1 and Test 2, all sellers specify the same type-blue conflict. When the minimum seller degree constraint is 16, the ILP solver obtains the optimal solution 126. However, the problem becomes infeasible if the minimum degree constraint is increased to 17. In Test 3 and Test 4, seller 5 specifies the type-red conflict, which is different from the rest. Since type-red conflict only involves 26 buyer pairs, Test 3 and Test 4 have less conflict constraints. The Minimum Seller Degree (MINSD) remains the same as 17 in Test 3 and we obtain the optimal solution 136. Nonetheless, if MINSD is increased to 18, the problem becomes infeasible.

3.7.2 Difficulties of Adapting Existing Algorithms to CA-WBM

CA-WBM is a variant of a classical problem, WBM, or the corresponding unweighted version, BM. In this section, we discuss why algorithms for WBM using a well-known technique cannot be applied to CA-WBM, without substantial modification.

To ease the illustration, we study the unweighted version, CA-BM, a special case of CA-WBM. We view this problem as the generalization of maximum bipartite b-matching (BM). Given an unweighted bipartite graph $G = (V_1, V_2, E)$, where V_1, V_2

and E represent left vertices, right vertices and edges, respectively, BM is to find a maximum size b-matching $M \subseteq E$ such that every vertex v in the induced subgraph $H \subseteq G$ is incident to at most $B(v)$ edges (degree constraints).

Using reduction [51] from BM to the problem of computing maximum matchings, algorithms, such as the famous Hopcroft-Karp algorithm [62] and the blossom algorithm (developed by Edmonds [39]), that can solve maximum matching in polynomial time, for bipartite graphs and general graphs, respectively can be used to solve BM with only slight changes [51]. The core technique used in these algorithms is that they repeatedly increase the size of a partial matching by finding augmenting paths in the graph. We will show why this technique does not work for CA-BM. We first briefly introduce important concepts and then describe a basic augmenting path algorithm for computing a maximum size b-matching on a bipartite graph.

Definition 2 (Augmenting Path (AP))**.** *Given a graph $G = (V, E)$ and a b-matching $M \subseteq E$, a path P is an augmenting path with respect to M if:*

1. *P starts from and ends on unsaturated vertices,*
2. *Both the first and last edges of P are not in M,*
3. *The edges of P alternate between edges belonging to M and edges not belonging to M.*

A vertex v is said to be unsaturated if there are less than $B(v)$ edges of M that are incident on v. The definition also implies that the length of AP must be odd.

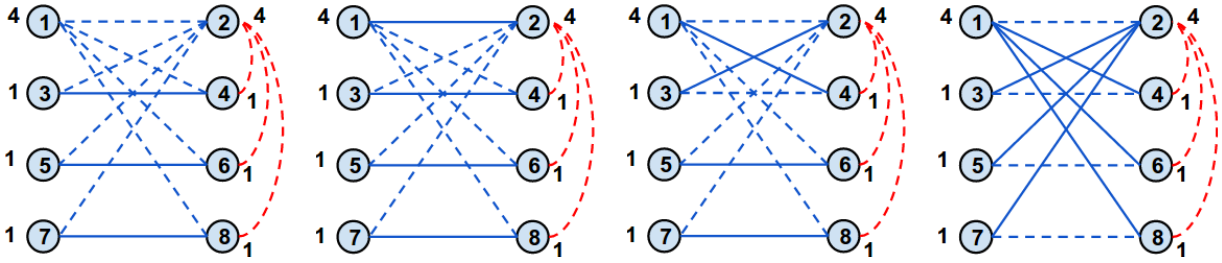
Theorem 6 (Berge [16])**.** *A b-matching M is maximum if and only if it has no augmenting path in G.*

Algorithm 2: ***Augmenting Path Algorithm***

Input: An undirected and unweighted bipartite graph $G = \langle (U \cup V), E \rangle$, each vertex v is associated with a degree constraint $B(v)$

Output: A maximum b-matching M^*

```
1 M = ∅, M* = ∅ ;
2 while ∃ an augmenting path P do
3 |   M = M ⊕ P;
4 M* = M ;
5 return M*;
```



(a) M_a (b) M_b (c) M_c (d) M_d

Figure 3.11: An example illustrating the need for flipping paths. Solid blue lines represent the current bipartite b-matching M. Dashed blue lines are unmatched edges. Red dashed lines on the right side are conflict constraints between buyer nodes

Algorithm 2 outlines the brief steps of the basic augmenting path algorithm. The operator $\oplus$ represents the symmetric difference of two sets, $A \oplus B = (A \setminus B) \cup (B \setminus A)$. Based on Algorithm 2, the Hopcroft-Karp algorithm uses breadth-first-search (BFS) to partition the vertices of the graph into layers. It finds a maximal set of shortest augmenting paths in each iteration. The blossom algorithm uses a new idea, when finding augmenting paths in a general graph, that an odd-length alternating cycle (blossom) can be shrunk into a single vertex. Note that the Hopcroft-Karp algorithm and the blossom algorithm are simple extensions of Algorithm 2. Therefore, we mainly discuss why Algorithm 2 that can be used to solve BM needs substantial modifications to be applicable to CA-BM.

Generally speaking, the conflict constraints may result in pseudo-augmenting paths (PAPs), i.e., augmenting paths that do not satisfy the conflict constraints. PAPs lead to two major difficulties that will require careful design and implementation in order for them to be useful towards computing the maximum matching:

1. The need for flipping paths (FP)

2. The stopping criteria for the algorithm

We show the difficulties using three examples in the following subsections.

The Need for Flipping Paths (FP)

In Figure 3.11, seller vertices are $\{1, 3, 5, 7\}$ and buyer vertices are $\{2, 4, 6, 8\}$. The number beside each node shows its degree constraint. Suppose vertex 2 is in conflict with all other buyers. Let the conflict constraint threshold be $t = 0$ for each seller.

A trivial extension to the augmenting path algorithm is to consider the conflict constraint when doing the operation $M \leftarrow M \oplus P$; it succeeds only if the new matching satisfies all conflict constraints. Due to the conflict constraints, however, an augmenting path may not be helpful to increase the size of a current matching M.

Suppose we start from a matching M_a in Figure 3.11a, and we look for the shortest augmenting path first (as in Hopcroft-Karp algorithm). Assume that we always search for augmenting paths from seller vertices. Vertex 1 is free and the only free buyer vertex is 2. Therefore the shortest path is $\{(1,2)\}$. After adding (1,2) to M_a, we obtain M_b in Figure 3.11b. Now an augmenting path from 1 is $\{(1,4),(4,3),(3,2)\}$. However, adding the edge $(1,4)$ violates the conflict constraint of 1 because vertices 2 and 4 are in conflict. So this path is invalid. Similarly, other augmenting paths (e.g., $\{(1,6),(6,5),(5,2)\}$) are also invalid. This creates a dilemma where existing augmenting paths cannot be used due to conflict constraints. In other words, only PAPs exist.

Since we cannot find a valid AP, we should stop according to Algorithm 2. M_b, however, is not maximum. As easily verified, M_d shown in Figure 3.11d is the maximum matching in this graph.

In order to expand M_b, the matched edge $(1,2)$ in M_b has to be unmatched. We have to find a path P (not necessarily augmenting) that contains $(1,2)$, so that the operation $M \oplus P$ results in a matching where $(1,2)$ is unmatched. A possible path is $\{(1,4),(4,3),(3,2),(2,1)\}$. Then $M_c = M_b \oplus P$. Obviously, using this path does not increase the size of a matching, but flips matched and unmatched edges along the path. After flipping these edges, we obtain new APs immediately, $\{(1,6),(6,5),(5,2)\}$ and $\{(1,8),(8,7),(7,2)\}$. Using the two APs, we can finally obtain the maximum matching M_d, where there are no more PAPs or APs. Therefore, flipping paths is a necessary step to amend the current maximal matching.

In a bigger graph with millions of vertices and edges, flipping paths will not be so obvious and not all flipping paths will work due to the conflict constraints. If the algorithm tries all flipping paths, it will simply lead to an exhaustive search approach (as flipping paths changes the current matching). Finding valid flipping paths in an efficient way turns out to be hard.

The Stopping Criteria

When should we stop? In M_b, we do not stop because PAPs exist and we want to try FPs. In M_d, we stop because neither PAP nor AP can be found, and M_d is indeed maximum. However, it is difficult to tell if a b-matching is maximum in a big graph under conflict constraints. We need to extend Theorem 6 to provide us the stopping criteria for CA-BM. In order to get a better understanding, we ask the following questions. Should we keep searching and flipping until we do not find PAP or AP? Can there be a situation where PAPs always exist? Can there be a situation where PAPs or APs do not exist but M is still far from maximum? We discuss these questions using two examples shown in Figure 3.12 and Figure 3.13. The answers to these questions imply that adapting Theorem 6 to CA-BM requires new ideas.

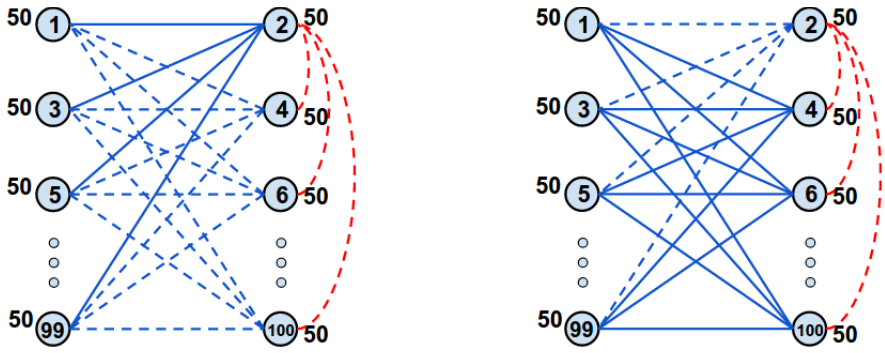
(a) M_a (b) M_b

Figure 3.12: An example illustrating the difficulty in determining when to stop the algorithm if there exists a PAP. Solid blue lines represent the current matching M. Dashed blue lines are unmatched edges. Red dashed lines on the right side are conflict constraints between buyer nodes.

In Figure 3.12a, there exists a PAP, $\{(1,2),(2,3),(3,4)\}$. This path is invalid because vertices 2 and 4 are in conflict. Can we further increase the size of M_a? A long FP can be found, $\{(1,2),(2,3),(3,4),(4,1),(1,6),(6,3)\}$. Applying this FP creates M_b which has the same size of M_a. Now, we realize that the flipping operation does not find a new AP. In fact, both M_a and M_b are already maximum. Although PAP exists in M_b ($\{(3,6),(6,1),(1,4)\}$), the algorithm should stop.

In addition, we also note that termination upon the non-existence of PAPs or APs does not necessarily imply a good, approximate solution. For example, in Figures 3.13a and 3.13b, the size of a maximal b-matching M is 50, while the maximum b-matching M^* contains $49 * 50 = 2450$ edges, i.e., $|M| = \frac{1}{49}|M^*|$, implying a large gap.

Due to these difficulties, it is not clear how to extend Algorithm 2, or the Hopcroft-Karp algorithm, or the blossom algorithm to obtain the maximum b-matching under



(a) A maximal CA-BM (b) A maximum CA-BM

Figure 3.13: An example illustrating the termination upon non-existence of PAPs or APs does not necessarily imply a good, approximate solution. Solid blue lines represent the current bipartite b-matching M. Dashed blue lines are unmatched edges. Red dashed lines on the right side are conflict constraints between buyer nodes.

conflict constraints even in the approximate sense. The situation becomes more complicated in the weighted version.

3.8 Conclusions

We initiated the study of a novel extension of classic weighted bipartite b-matching (WBM). The question we addressed is how to maximize the total weight when matching vertices under both degree and conflict constraints (CA-WBM). The CA-WBM problem is general and can find many applications in the domain of E-Commerce, such as Internet advertising and personalized recommendation.

We provided a formal definition of the central problem, CA-WBM, that directly models both the degree constraint on each vertex and conflict relationship between vertices on the same side. We showed that by considering the conflict constraints, the complexity of WBM increases significantly. We proved that CA-WBM is NP-hard. We modelled it using semidefinite programming (SDP) and integer linear programming (ILP). Then we proposed a SDP algorithm with rounding, LP relaxation with rounding, and a greedy algorithm to solve CA-WBM. We also proposed a randomized algorithm to solve online CA-WBM. We showed that they achieve close to optimal solutions via comprehensive experiments using synthetic datasets. We derived a theoretical bound on the approximation ratio of the greedy algorithm and showed that it is scalable on a large-scale real-world dataset.

Chapter 4

Group-Aware Weighted Bipartite b-Matching

4.1 Introduction

In Chapter 3, we propose CA-WBM which incorporates diversity constraints into the classic WBM for the purpose of diversified recommendation with utility maximization. CA-WBM is a general extension of WBM and vastly expands the expressiveness of WBM, but it has two notable shortcomings. Firstly, while WBM can be solved efficiently in polynomial time with the Hungarian Algorithm [86], CA-WBM is instead NP-hard [31]. In fact, our first contribution in this chapter is a new hardness result (Section 4.2) for CA-WBM, reducing from the Maximum Weight Independent Set problem, which proves that it is hard *even to approximate CA-WBM*. Secondly, arbitrarily adding conflict edges between pairs of vertices is often unnecessarily general, because in many applications the conflicts are transitive within *groups* (e.g., households, genres, topics, temporal ranges), i.e., the conflicts arrange in cliques.

In this chapter, motivated by the intractability, even inapproximability, of CA-WBM, but the need for more expressive models than WBM, and the organization of conflicts into disjoint groups, we introduce a novel generalization of WBM: *group-aware* WBM (GA-WBM). We study two variants of the problem: the first generalizes the constraints of WBM by constraining the degree for each group (Section 4.3); the other generalizes the optimization function of WBM by imposing a ceiling on the budget/payoff for each group (Section 4.4). The former captures scenarios when one wishes to limit the overall *number* of products from one group matched to a user, e.g.,

no more than three news articles on a single topic; the latter captures scenarios when one wishes to limit the overall *weight* of products matched to a user, e.g., no more than $100 is to be spent on online advertising for keywords of a certain category.

For the degree constraint variant, we present an exact, linear programming algorithm, and a scalable, greedy algorithm with approximation guarantees. For the budgeted variant, we prove NP-hardness, but again give a greedy algorithm with a constant-factor approximation guarantee, precisely what we prove cannot be done for CA-WBM. As such, *group-aware weighted bipartite B-matching* can model a broader range of problems than WBM while still admitting efficient algorithms with good performance guarantees.

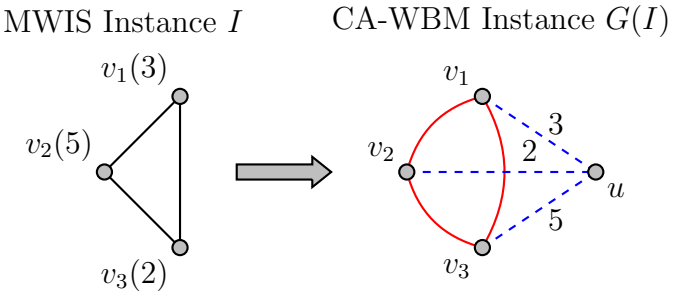


Figure 4.1: The reduction from MWIS to CA-WBM. Vertex weights are in parentheses and edge weights are above the edge. Edges in the original graph become (red) conflicts, while all vertices are connected to a new vertex u with a dashed blue edge. Degree constraints (not shown) are 1 for v_1, v_2, v_3 and 3 for u (i.e., the degrees of the vertices).

4.2 Stronger Hardness for CA-WBM

CA-WBM is already proven to be NP-Hard in Chapter 3 (corresponding to [31]), based on a reduction from the NP-hard Revenue Maximisation in Interval Scheduling problem [12, 19, 83]. Here we give a simpler reduction from the Maximum Weight Independent Set (MWIS) problem (defined below), which, due to Håstad [59], has the stronger implication of being hard to approximate.

Maximum Weight Independent Set (MWIS)
Given a (not generally bipartite) graph $G = (V, L, R)$*, with vertex set* $V = \{v_1, \ldots, v_n\}$*, edge set* $L = \{l_1, \ldots, l_m\}$*, and vertex-labelling function* $R : V \to \mathbb{R}_+$*, find the independent set* $V' \subseteq V$ *that maximises* $\sum_{v \in V'} R(v)$*.*

Theorem 7. *Unless P=NP, there is no approximation algorithm for CA-WBM with an approximation ratio $n^{1-\epsilon}$, for any fixed $\epsilon > 0$, where $n = |U \cup V|$.*

Proof. We give a polynomial-time reduction from the NP-hard Maximum Weight Independent Set (MWIS) problem. Let I be an instance of MWIS. We construct a graph $G(I) = ((U, V), E, W)$, which is an instance of CA-WBM, as follows:

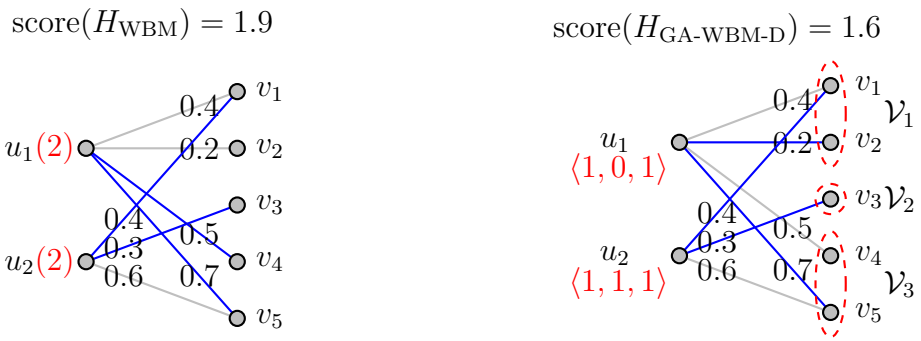
- $U = \{u\}$ and V is the same as in G,
- $W(v_k, u) = R(v_k)$ for $1 \leq k \leq n$,
- $C = L$,
- $B(v_k) = 1$ for $1 \leq k \leq n$ and $B(u) = n$, and
- $\tau = 0$.

We construct a graph with all vertices V of I isolated, and connect them to a new vertex u. Any adjacent vertices $(u, v) \in L$ are added to the set of conflicts C. Denote E as the set of edges connecting u and vertices of V, i.e., $E = \{u\} \times V$ and assign the weight of vertex v_i to edge (u, v_i). Figure 4.1 illustrates an example of such a reduction.

For an optimal solution O of the CA-WBM instance $G(I)$, any two matched edges in O must not have the same end point in V, since $\tau = 0$. This implies that O corresponds to an independent set in V for I, an instance of MWIS. In addition, since $W(v_k, u) = R(v_k)$ for $1 \leq k \leq n$, the total weight of edges in O also corresponds to the total weight of vertices in the independent set in V. Hence, a maximum weight subgraph of an instance of CA-WBM satisfying the degree constraints and conflict constraints exactly corresponds to a maximum weight independent set for an instance of MWIS. Furthermore, we observe that the reduction above is a polynomial-time reduction, gap-preserving reduction. Therefore, using the hardness of approximation result for MWIS [59], we conclude that CA-WBM is NP-hard and is not approximable with $n^{1-\epsilon}$ for any $\epsilon > 0$. □

4.3 GA-WBM + Degree Constraints

In this section, we introduce our first GA-WBM variant (Section 4.3.1), wherein each group can be adjacent to a limited number of edges. We present an exact, efficient



(a) Solution H_{WBM}, no group constraints

(b) Solution $H_{\text{GA-WBM-D}}$ satisfying group *degree* constraints

Figure 4.2: Contrasting WBM (a) and GA-WBM-D (b). All right-hand vertices $v \in V$ have degree constraint 1. In WBM, the only modellable diversity constraint is that u_1, u_2 cannot both choose v_1 nor v_5. In GA-WBM-D, we explicitly model three equivalence classes of similar vertices in V. We constrain u_1 and u_2 to matching at most $\langle 1, 0, 1\rangle$ and $\langle 1, 1, 1\rangle$ vertices, respectively, from $\langle \mathcal{V}_1, \mathcal{V}_2, \mathcal{V}_3\rangle$, producing a more diversified assignment.

linear programming algorithm (Section 4.3.2), demonstrating that GA-WBM-D $\in \mathbf{P}$ (Theorem 8), and a scalable, greedy algorithm (Section 4.3.3) that is 2-approximate (Theorem 9).

4.3.1 Problem Formulation

In many applications, items cluster into groups of mutual similarity, and users should be matched to a limited number of items from the same group. For example, books can be grouped by genre, topic, or author, and diverse matchings for a user should span the groups. GA-WBM partitions the items into equivalence classes (groups) and imposes constraints on how much each user can be matched to each class. In comparison to CA-WBM [31], partitioning into equivalence classes can be viewed as all conflicts being transitive (i.e., the existence of conflicts (a, b) and (b, c) implies (a, c) is a conflict). For genres, topics, and authors of books, transitivity clearly holds.

In the degree-constraints version of GA-WBM, we limit the *number of edges* that match each user u to each equivalence class $\mathcal{V}_i$. The problem is illustrated and contrasted to WBM in Figure 4.2: here, V is partitioned into 3 groups, $\mathcal{V} = \langle \mathcal{V}_1 = \{v_1, v_2\}, \mathcal{V}_2 = \{v_3\}, \mathcal{V}_3 = \{v_4, v_5\}\rangle$. Each vertex of U has now a sequence of degree constraints, one for each $\mathcal{V}_i$. Thus GA-WBM can model that u_1 should match to at most one of v_4, v_5, a diversity constraint that is inexpressible in WBM. We obtain

(a) Original graph

(b) Split graph

Figure 4.3: Setup of the linear program for GA-WBM-D. The original vertices of U (a) are split, one for each equivalence class (b). The red ellipses indicate equivalence classes and the integers on the edges show their sequence numbers.

an instance of WBM if we create only one equivalence class, in which case the degree constraint sequence reduces to a single degree constraint for each $u \in U$.

Formally, we represent the degree constraints as a mapping from user-class pairs to integers. The number of edges from a user u to vertices in an equivalence class $\mathcal{V}_i$ must not exceed the limit specified in the mapping.

Group-Aware Weighted Bipartite B-Matching
Subject to Degree Constraints (GA-WBM-D)
Given G, a vertex-labelling function $B : V \to \mathbb{N}$, a partitioning of V into k equivalence classes $\mathcal{V} = \langle \mathcal{V}_1 \ldots, \mathcal{V}_k \rangle$, and a U-degree-constraint mapping $D : (U \times \mathcal{V}) \to \mathbb{N}$, find the subgraph $H = ((U, V), E', W)$ maximising $\sum_{e \in E'} W(e)$ with every vertex $v \in V$ adjacent to at most $B(v)$ edges and for all vertex-class pairs $(u \in U, \mathcal{V}_i \in \mathcal{V})$:

$$|\{e = (u, v_j) \in E' : v_j \in \mathcal{V}_i\}| \leq D(u, \mathcal{V}_i).$$

4.3.2 A Linear Program for GA-WBM-D

Here we present a linear programming formulation to solve GA-WBM-D, given inputs $G = ((U, V), E, W), \mathcal{V}, B, D$, with $m = |U|$ and $n = |V|$. We denote by $X = [x_{ij}]^T$ the mn-dimensional column vector of 0-1 variables, by $x_{ij} = 1$ that item i is matched to user j and by $x_{ij} = 0$ otherwise. Then GA-WBM-D *finds the set of matches such that the total profit is maximized under the degree constraints*, i.e.,

$$\begin{aligned} \max_{X} \quad & WX \\ s.t. \quad & \mathbb{A}X(i) \leq Q(i), \forall i, 1 \leq i \leq km + n \\ & x_{ij} \in \{0, 1\}, \forall i, j, 1 \leq i \leq m, 1 \leq j \leq n, \end{aligned} \tag{4.1}$$

where matrix $\mathbb{A}$ is an $(km+n) \times mn$ matrix and Q is a vector of values of B and D. That is, the degree constraints are given by $\mathbb{A}X(i) \leq Q(i)$, where $\mathbb{A}X(i)$ denotes the i-th element in (vector) $\mathbb{A}X$ and $Q(i)$ the i-th element in Q.

Below, we show that GA-WBM-D is in **P** by proving that $\mathbb{A}$ is totally unimodular (TU). A matrix A is said to be totally unimodular if the determinant of each square submatrix of A is 0, -1 or 1. As [124, 119] show, if $\mathbb{A}$ is TU, the polyhedron $P = \{X : \mathbb{A}X \leq Q\}$ is integral and an integral optimal solution can be found in polynomial time using an LP algorithm.

Lemma 3. *In the LP formulation of GA-WBM-D, the coefficient matrix $\mathbb{A}$ is totally unimodular (TU).*

Proof. We need to prove that the matrix $\mathbb{A}$ in Problem 4.1 is totally unimodular. Note that $\mathbb{A}$ captures two types of degree constraints; 1) a degree constraint of each item, and 2) a sub-degree constraint for each user with regard to each item group. There are n constraints of type 1 and km constraints of type 2 for a total of $km+n$ constraints. In other words, A is a $(km+n) \times mn$ matrix.

In order to prove the lemma, we would like to make use to well-known result that the incidence matrix of a bipartite graph is totally unimodular [13, 138]. Our main observation is that considering each users' sub-degree constraints for k conflict groups is equivalent to replacing each user vertex with k copies, with each only connecting to a single item group. Figure 4.3 shows such a transformation. The degree constraint of each copy is the original vertex's sub-degree constraint for the corresponding item group. This transformation only increases the number of user vertices and the transformed graph is still a bipartite graph. We call the resulting graph as the split graph.

From the transformation in Figure 4.3, we can see that $\mathbb{A}$ is the incidence matrix of the split graph in which original user vertices have been replaced by the copies and is therefore totally unimodular. As an example, in Figure 4.3b, $\mathbb{A}$ is defined as

$$\mathbb{A}_s = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ u_1^1 \\ u_1^2 \\ u_1^3 \\ u_2^1 \\ u_2^2 \\ u_2^3 \end{array} \begin{array}{c} \begin{array}{cccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{array} \\ \left(\begin{array}{cccccccccc} 1 & 1 & & & & & & & & \\ & & 1 & 1 & & & & & & \\ & & & & 1 & 1 & & & & \\ & & & & & & 1 & 1 & & \\ & & & & & & & & 1 & 1 \\ 1 & & 1 & & & & & & & \\ & & & & 1 & & 1 & & & \\ & & & & & & & & 1 & \\ & 1 & & 1 & & & & & & \\ & & & & & 1 & & 1 & & \\ & & & & & & & & & 1 \end{array} \right) \end{array}$$

Therefore, we conclude that in the LP formulation of GA-WBM-D, the coefficient matrix $\mathbb{A}$ is totally unimodular and hence solving the LP always yields integral solutions. □

From [124, 119], we immediately have the following theorem:

Theorem 8. *There exists an algorithm that solves GA-WBM-D in polynomial time.*

4.3.3 A Greedy Algorithm for GA-WBM-D

Although Program 1 in Section 4.3.2 can be solved in polynomial time and is exact, it scales poorly (see Section 4.5.3). Therefore, we also introduce GREEDY-D (Algorithm 3), a simple but provably approximate greedy algorithm. The idea is to start with an empty edge set E' (Line 1) and repeatedly add the next highest-weighted edge e (Lines 2–3) if $E' \bigcup \{e\}$ does not violate the degree constraints (Line 4). After trying all $e \in E$, we return $H = ((U, V), E', W)$, a new graph constructed with the greedily built edge set E'.

Although we will later evaluate the efficiency, scalability, and accuracy of GREEDY-D empirically (Section 4.5.3), we prove here a theoretical *guarantee* on its performance:

Theorem 9. *Algorithm 3 is a* 2*-approximation algorithm.*

Proof. Recall from Section 3.4.3 that we use the concept of a *k-extendible system* to performance guarantees for greedy algorithms. To apply this result to our problem,

Algorithm 3: GREEDY-D

Input: $G = ((U, V), E, W), \mathcal{V}, B, D$
Output: A subgraph $H = ((U, V), E', W)$ satisfying constraints B, D with a greedily-maximised score, $\sum_{e \in E'} W(e)$

```
E' = ∅
Sort E by descending W(e)
for e ∈ E do
    if H = ((U, V), E' ⋃ {e}, W) does not violate B, D then
        E' = E' ⋃ {e}
return A new graph H = ((U, V), E', W)
```

we will check that the set of all feasible solutions to GA-WBM-D forms a 2-extendible system. For the GA-WBM-D problem, let $U = E$ and $\mathcal{F}$ be the set of all subgraphs of G satisfying the degree constraints on the items and the sub-degree constraints on the users. Then it is easy to see that $(U, \mathcal{F})$ is downward closed. That is, removing an edge from a feasible solution H will always result in a feasible solution as this will not cause any violation of constraints.

For the exchange property, consider the case when a new edge $e = (u, v)$ is added to a feasible solution H. Assume that $v \in \mathcal{V}_i$. We observe that adding e could result in a violation of the degree constraint at v, and sub-degree constraint at u. However, this can be rectified by removing two other edges, one incident on u and other incident on v. Therefore, we obtain a 2-extendible system. □

4.4 GA-WBM + Budget Ceilings

This section presents a variant of GA-WBM that introduces diversity through the optimisation function rather than the constraints: it introduces a ceiling on the sum of edge weights each $u \in U$ can amass for each group, $\mathcal{V}_i$. We formally define the problem and prove it is hard (Section 4.4.1). We then give an ILP formulation along with a tractable relaxation (Section 4.4.2) and, again, a greedy algorithm with a constant-factor approximation guarantee (Section 4.4.3).

4.4.1 Problem Formulation

Section 4.3.1 formally modelled diversity in V by partitioning V into groups and constraining each user $u \in U$ to a limited number of edges per group. With GA-

WBM-B, we cap the score each user can derive from each group by mapping user-group pairs onto a real-valued budget ceiling. Figure 4.4 illustrates how capping budgets can produce the same diverse matchings as the group degree constraints in Figure 4.2.

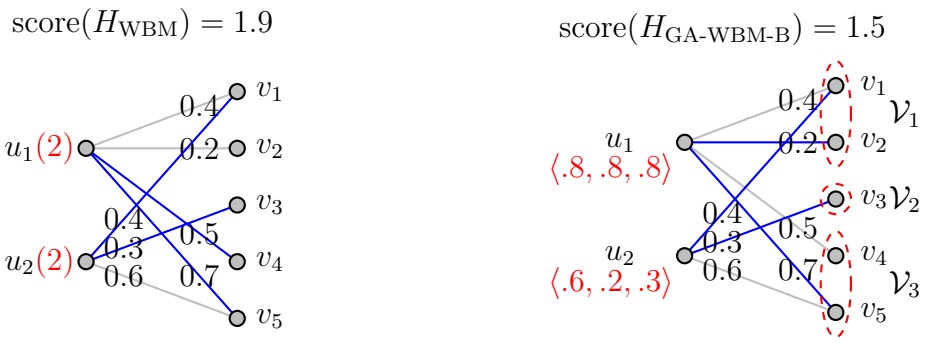
Group-Aware Weighted Bipartite B-Matching Subject to Budget Ceilings (GA-WBM-B)
Given G, $B : U \bigcup V \to \mathbb{N}$, a partitioning of V into $\mathcal{V} = \langle \mathcal{V}_1 \ldots, \mathcal{V}_k \rangle$, and a mapping $C : (U \times \mathcal{V}) \to \mathbb{R}_+$, find the subgraph $H = ((U, V), E', W)$ with every vertex $u \in U \bigcup V$ adjacent to $\leq B(u)$ edges that maximises the score function:

$$\sum_{u,\mathcal{V}_i} \min \left\{ C(u, \mathcal{V}_i), \sum_{(u,v_j) \in E', v_j \in \mathcal{V}_i} W(u, v_j) \right\}.$$

This formulation is more natural for some applications. For example, sponsored search auctions hosted by search engines (e.g., Google, Bing, and Yahoo) include budget specification as a feature. To achieve better coverage, advertisers who bid on keywords can specify budgets for different categories of keywords. The constraints are identical to WBM, but a subgraph will no longer profit from matching a specific user and item group once the ceiling has been hit; so, maximising the objective function requires diversifying across groups. We obtain an instance of WBM if the ceilings are sufficiently high, e.g., $\sum_{e \in E} W(e)$ for all user-group pairs.

GA-WBM-B generalizes the *maximum budgeted allocation* (MBA) problem [11, 52, 89, 123]. Given n items and m users, where each user i with budget C_i is willing to pay w_{ij} on item j, MBA finds an allocation which maximises the total revenue. In MBA, each user has an overall budget on all items and there is no constraint on the number of allocated items. Each item, however, can be allocated only once. Therefore, MBA can be regarded as a special case of GA-WBM-B in which k is 1, $B(u) = n$ for $u \in U$ and $B(v) = 1$ for $v \in V$. Since MBA is known to be NP-hard and it is a special case of GA-WBM-B, GA-WBM-B is also NP-hard. Therefore we have Theorem 10:

Theorem 10. *GA-WBM-B is NP-hard.*



(a) Solution H_{WBM}, repeated from Figure 4.2a

(b) Solution $H_{\text{GA-WBM-B}}$ satisfying *budget ceilings* for user-group pairs

Figure 4.4: Contrasting WBM (a) and GA-WBM-B (b) (c.f., Figure 4.2b). $B(v) = 1$ $\forall v \in V$ and $B(u) = 3$ $\forall u \in U$. The $\mathcal{V}_3$ budget for u_1 is nearly saturated by the edge (u_1, v_5); so, a higher score can be obtained by matching (u_1, v_2) rather than (u_1, v_4). The $\mathcal{V}_2$ budget for u_2 is over-satured by (u_2, v_3), but no alternative produces a higher score; so, the score only obtains the ceiling of 0.2 from the pair $u_2, \mathcal{V}_2$.

4.4.2 Integer LP for GA-WBM-B

If one considers the sum of edge weights (e.g., users' payments) in E' to be the total revenue, then GA-WBM-B finds the allocation which maximises the total revenue while satisfying degree constraints. Due to the existence of budget ceilings, users will have to receive items from diverse groups. GA-WBM-B can be formulated as a linear program as follows by adapting the formulation in (4.1):

$$\begin{aligned} \max_{X} \quad & \sum\nolimits_{i \in U} \sum\nolimits_{p=1}^{k} \min \left\{ C(i, \mathcal{V}_p), \sum\nolimits_{j \in \mathcal{V}_p} w_{ij} x_{ij} \right\} \\ s.t. \quad & \mathbb{A}X(i) \leq B(i), \forall i, 1 \leq i \leq m+n \\ & x_{ij} \in \{0, 1\}, \forall i, j, 1 \leq i \leq m, 1 \leq j \leq n, \end{aligned} \tag{4.2}$$

where $X = [x_{ij}]^T$.

Since obtaining an integer solution from the LP in GA-WBM-B is NP-hard, we use a rounding procedure to further improve efficiency after solving the LP relaxation. Our LP-based algorithm for GA-WBM-B is as follows:

1. Solve the linear program relaxation to obtain an optimal solution S.

2. Sort the first mn elements of S from largest to smallest. We round each non-zero value to 1 provided doing so does not violate the degree constraints or the budget ceilings. Otherwise, we set it to 0.

We refer to this as the *LP relaxation with rounding.*

4.4.3 A Greedy Algorithm for GA-WBM-B

Given the hardness of GA-WBM-B, we introduce a greedy algorithm with a theoretical guarantee by again establishing connections to the problem of maximizing a non-negative monotone submodular function subject to a k-extendible system constraint.

Submodular set functions are used to capture a natural diminishing returns property:

Definition 3 (monotone submodular function)**.**
Let U be a finite set of elements. A positive set function $f : 2^U \to \mathbb{R}_+$ is monotone if for any two sets $A \subseteq B \subseteq U$, we have that $f(A) \leq f(B)$. Set function f is submodular if for every $A, B \subseteq U$ with $A \subseteq B$ and every $x \in U \setminus B$, we have that $f(A \cup \{x\}) - f(A) \geq f(B \cup \{x\}) - f(B)$.

Fisher *et. al* [107] showed that if the set of feasible solutions forms a k-extendible system and the objective function is positive, monotone submodular, a natural greedy algorithm, that incrementally adds an element that most improves the current solution, is guaranteed to produce a $(k+1)$-approximate solution.

Theorem 11 (Fisher, Nemhauser, Wolsey [107])**.** *Let $(U, \mathcal{F})$ be a k-extendible system for some k. Let $W : 2^U \to \mathbb{R}_+$ be positive, monotone submodular function. The greedy algorithm gives a $(k+1)$-approximation algorithm for the optimization problem that asks to determine $\max_{F \in \mathcal{F}} W(F)$.*

In Problem 4.2, the objective function,

$$g(X) = \sum\nolimits_{i \in U} \sum\nolimits_{p=1}^{k} \min \left\{ C(i, \mathcal{V}_p), \sum\nolimits_{j \in \mathcal{V}_p} w_{ij} x_{ij} \right\},$$

is the sum of a series of budget-additive functions,
$\min(C(i, \mathcal{V}_p), \sum_{j \in \mathcal{V}_p} w_{ij} x_{ij})$, each of which is a monotone submodular function. Since the sum of submodular functions retains submodularity [50, 85], $g(X)$ is also submodular. w_{ij} is non-negative, thus it is easy to see that $g(X)$ is monotone. In addition, as discussed in Section 4.3.3, the degree constraints form a 2-extendible system.

Algorithm 4 outlines a natural greedy algorithm, GREEDY-B, for GA-WBM-B. Denote $\delta_l = \Delta_{E'_{l-1}}(e_l) = g(E'_l) - g(E'_{l-1})$ as the value of increment (marginal

Algorithm 4: GREEDY-B

Input: $G = ((U, V), E, W), \mathcal{V}, B, C$

Output: A subgraph $H = ((U, V), E', W)$ satisfying constraints $\mathcal{V}, B, C$ with a greedily-maximised score, $\sum_{e \in E'} W(e)$

1 $E' = \emptyset$, $A = \emptyset$

2 $\mathcal{F}$ is a universal set that contains all candidate edge sets satisfying the degree constraints

3 $A = \{e | E' \cup e \in \mathcal{F}\}$

4 **while** $A \neq \emptyset$ **do**

5 $\quad e^* = \arg\max_{e \in A} \Delta_{E'}(e)$

6 $\quad E' = E' \cup \{e^*\}$

7 $\quad A = \{e | E' \cup \{e\} \in \mathcal{F}\}$

8 **return** *A new graph* $H = ((U, V), E', W)$

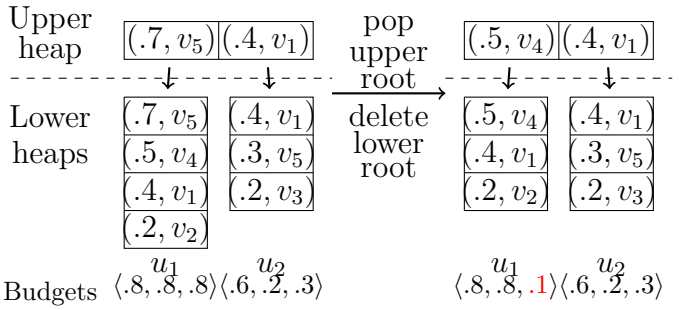


Figure 4.5: The two-level heap and the lazy forward technique. This example corresponds to Figure 4.4b.

revenue) in the objective function when the l^{th} edge e_l is added. The algorithm starts with an empty solution edge set E' and in each iteration adds to E' an edge that provides the maximum marginal revenue without violating degree constraints. To ensure the newly added edge is always valid, Algorithm 4 keeps track of a set of all valid edges (A in Algorithm 4) with regard to the current E' (Lines 3 and 7). For any $e \in A$, it can be safely added to E' and the increased edge set still satisfies degree constraints, i.e., $E' \cup \{e\} \in \mathcal{F}$, where $\mathcal{F}$ represents a collection of subsets of E, each of which satisfies degree constraints. Fisher, Nemhauser and Wolsey [107] showed that this greedy algorithm gives a $(k+1)$-approximation algorithm to the problem of maximising a non-negative monotone submodular function subject to a k-extendible system constraint. Therefore, GREEDY-B is 3-approximate for GA-WBM-B.

Algorithm 5: GREEDY-B-LF

Input: $G = ((U, V), E, W), \mathcal{V}, B, C$
Output: A subgraph $H = ((U, V), E', W)$ satisfying constraints $\mathcal{V}, B, C$ with a greedily-maximised score, $\sum_{e \in E'} W(e)$

$E' = \emptyset$
Set the upper heap ($\mathcal{H}^{\text{upper}}$) as an empty max heap
for $u \in U$ **do**
    Set the lower heap ($\mathcal{H}^{\text{lower}}_{(u)}$) as an empty max heap
    for *u's neighbouring vertices* $v \in V$ **do**
        $w = W((u, v))$; $\delta = \min(w, C_{u,p})$, $v \in \mathcal{V}_p$
        $\mathcal{H}^{\text{lower}}_{(u)}$.Append($(\delta, (u, v))$);
    Heapify($\mathcal{H}^{\text{lower}}_{(u)}$); $\mathcal{H}^{\text{upper}}$.Append($\mathcal{H}^{\text{lower}}_{(u)}[0]$)
Heapify($\mathcal{H}^{\text{upper}}$)
while $\mathcal{H}^{\text{upper}} \neq \emptyset$ **do**
    $(\delta, e) = \mathcal{H}^{\text{upper}}[0]$, $e = (u, v)$, $v \in \mathcal{V}_p$
    if $\delta \leq 0$ **then break**
    if $E' \cup \{e\}$ *violates degree constraints on* u *or* v **then**
        Heappop($\mathcal{H}^{\text{upper}}$); Heappop($\mathcal{H}^{\text{lower}}_{(u)}$)
        if $\mathcal{H}^{\text{lower}}_{(u)} \neq \emptyset$ **then** Heappush($\mathcal{H}^{\text{upper}}$, $\mathcal{H}^{\text{lower}}_{(u)}[0]$)
    else if $\delta \leq C_{u,p}$ **then**
        $E' = E' \cup \{e\}$; Update $C_{u,p}$ and degree constraints on u and v
        Heappop($\mathcal{H}^{\text{upper}}$); Heappop($\mathcal{H}^{\text{lower}}_{(u)}$)
        if $\mathcal{H}^{\text{lower}}_{(u)} \neq \emptyset$ **then** Heappush($\mathcal{H}^{\text{upper}}$, $\mathcal{H}^{\text{lower}}_{(u)}[0]$)
    else
        $(\delta_{\text{root}}, e_{\text{root}}) = \mathcal{H}^{\text{lower}}_{(u)}[0], e_{\text{root}} = (u_{\text{root}}, v_{\text{root}})$, $v_{\text{root}} \in \mathcal{V}_p$
        while $\delta_{\text{root}} > C_{u,p}$ **do**
            Update δ_{root} and maintain the heap invariant of $\mathcal{H}^{\text{lower}}_{(u)}$
            $(\delta_{\text{root}}, e_{\text{root}}) = \mathcal{H}^{\text{lower}}_{(u)}[0], e_{\text{root}} = (u_{\text{root}}, v_{\text{root}})$, $v_{\text{root}} \in \mathcal{V}_p$
        Heappop($\mathcal{H}^{\text{upper}}$)
        Heappush($\mathcal{H}^{\text{upper}}$, $\mathcal{H}^{\text{lower}}_{(u)}[0]$)
return *A new graph* $H = ((U, V), E', W)$

Algorithm 4 provides an intuitive high-level description of the greedy idea. This simple approach, however, has poor scalability in practice. For example, in each iteration, the reconstruction of A must check every edge that is not in E', which is expensive if the original edge set E is large. In fact, this is often the case in the real-world, where graph datasets can have millions of edges. Therefore, the imple-

mentation of the greedy approach requires non-trivial optimization to be scalable to real-world datasets which are often large in size.

GREEDY-B-LF (Algorithm 5) improves the scalability of GREEDY-B by employing similar techniques to those proposed in [103] and [93]. The main idea is to efficiently identify a valid edge with maximum marginal revenue and keep the marginal revenue of an edge updated only when necessary. Specifically, we implement a two-level heap (e.g., Figure 4.5) to store all valid edges and apply the lazy forward (LF) technique for updating the marginal value when necessary.

Using the heap structure improves the efficiency of finding an edge with maximum marginal value. For large graphs, however, maintaining the heap invariant in a single, large heap of marginal revenues for all edges is too expensive. So, we exploit the observation that distinct item vertices in V outnumber those user vertices in U (Table 4.1), and for any two vertices $u_1, u_2 \in U$, marginal revenue updates to edges of u_1 does not affect edges of u_2. Therefore we use a two-level heap data structure as shown in Figure 4.5. In Figure 4.5, an entry of each lower heap of $u \in U$ is a tuple of u's neighbouring vertex $v \in V$ and v's marginal revenue with regard to u's current budget (Lines 4-7 in Algorithm 5). The numbers below u's lower heap show u's current budget for each class. Since each $u \in U$ maintains its own heap, the size of each heap is reduced from $|E|$ to $\approx |E|/|U|$, and the corresponding maintenance cost is much less. This approach also applies to many other scenarios, such as budgeted resource allocation and online advertising, where GA-WBM-B comes in naturally, and where the bipartite graphs often have imbalanced vertex sets. Ergo, we can use the smaller set's entities to distinguish the lower-level heaps, i.e., in our case, each distinct $u \in U$ maintains a heap structure where nodes consist of all neighbouring $v \in V$ and the corresponding marginal values. The upper-level heap is constructed by the roots of the lower-level heap (Line 8) and it will be used for obtaining the edge with maximum marginal revenue (Line 11).

		25%	50%	75%	100%
eBay Canada	# Sellers ($\|U\|$)	471	942	1.4K	1,884
	# Buyers ($\|V\|$)	4.7K	9.4K	14K	18,742
	# Edges ($\|E\|$)	14K	28K	42K	56,520
eBay US	# Sellers ($\|U\|$)	66,751	90,925	109,511	126,101
	# Buyers ($\|V\|$)	1,574,114	2,988,717	4,300,322	5,751,334
	# Edges ($\|E\|$)	2,846,880	5,693,759	8,540,638	11,387,517

Table 4.1: Statistics of the semi-synthetic eBay datasets.

	25%	50%	75%	100%
LP	2.98	11.50	22.64	41.88
GREEDY-D	0.06	0.13	0.23	0.29

Table 4.2: GA-WBM-D run times (s) on eBay Canada

	25%	50%	75%	100%
ILP	0.91	2.50	2.66	3.96
LPR	0.55	1.00	1.52	2.10
GREEDY-B-LF	0.06	0.13	0.20	0.28

Table 4.3: GA-WBM-B run times (s) on eBay Canada

In addition, the lazy forward (LF) technique [103] can significantly reduce unnecessary computation, because it updates a stale marginal value of a node of the lower-level heap only when the node becomes the root of the upper heap (Lines 21-24 in Algorithm 5). Figure 4.5 shows an example of LF. At the current iteration, after popping the current upper root which is the root of u_1's heap, u_1's budget for $\mathcal{V}_3$ is reduced from 0.8 to 0.1. Since $v_4, v_5 \in \mathcal{V}_3$, the marginal revenue of v_4 to u_1 should be reduced from 0.5 to 0.1. However, LF suppresses the update and hence v_4 of u_1 still has a larger marginal revenue and becomes the new upper root. In the next iteration, the upper root will be considered stale and will be updated.

4.5 Experimental Evaluation

In Sections 4.3–4.4 we proved worst-case guarantees for the greedy algorithms. Here, we experimentally evaluate how closely the greedy algorithms perform to the worst-case guarantees and how all proposed algorithms scale with input size.

4.5.1 Methodology

All experiments are run on a 64-bit Ubuntu 14.04 desktop of 3.40 GHz * 8 Intel Core i7 CPU with 12 GB memory.

GA-WBM-D Both algorithms (LP and greedy) are polynomial time, but the size of the LP grows quickly. We use small datasets to evaluate the accuracy of the greedy algorithm relative to the (exact) LP and the scalability of the LP. We use large graphs to test the greedy algorithm's scalability.

GA-WBM-B As GA-WBM-B is NP-hard, our LP relaxation with rounding (LPR) is also approximate. We use small graphs to evaluate the accuracy of LPR and the greedy method and the scalability of the ILP/LPR methods. Large graphs again test the greedy algorithm's scalability.

For LP, ILP and LPR, we use the general mathematical programming solver, Gurobi[①], on Matlab. When we measure the running time of an algorithm, we exclude the time spent on loading data from the disk to the memory.

4.5.2 Datasets

We use two semi-synthetic eBay transaction datasets provided by the authors of [31] and described in Table 4.1. Each transaction dataset consists of seller vertices (U), buyer vertices (V), and edges (E) representing buyer-seller interactions, such as purchases. While the graph structure of eBay US reflects the true interaction, the eBay Canada dataset has imputed edges: according to [31], each seller is connected to 30 buyers after imputation. For each dataset, there are four subsets of different sizes from the full graph, i.e., using 25%, 50%, 75%, and 100% of the total number of edges (the number of buyer and seller nodes decreases accordingly).

To accommodate the respective problem scenarios, we assign random integer weights from the range $[1, 1000]$ to edges in both datasets. Each $v \in V$ is uniformly, randomly classified into one of 20 groups. For GA-WBM-D, the group-degree constraint of each $u \in U$ (i.e. $D(u, \mathcal{V}_j)$) is uniformly, randomly chosen from $\{0.1, 0.2, 0.3, 0.4, 0.5\} * |\{v : v \in \mathcal{V}_j \ \& \ (u, v) \in E\}|$. For GA-WBM-B, the degree constraint of each $u \in U \bigcup V$ is set to 0.3 of the degree of u. The budget ceiling of each $u \in U$ (i.e., $C(u, \mathcal{V}_j)$) is $\left\lfloor 0.8 \sum_{(u,v)\in E, v\in\mathcal{V}_j} W(v) \right\rfloor$.

4.5.3 Results and Discussion

GA-WBM-D Table 4.2 gives the running times of the LP and the greedy algorithm (GREEDY-D) for the degree-constrained problem on the small eBay Canada dataset. On the 25% sample, GREEDY-D is already $\approx 50\times$ faster than the LP method. As the graph size doubles to 50% and then 100%, the run time of GREEDY-D increases linearly. In contrast, the LP method quadruples run times for every doubling of the graph. At $56,520$ edges, GREEDY-D is $\approx 144\times$ faster. The memory consumption for

[①]http://www.gurobi.com/

this experiment is shown in Figure 4.6b. Given the large number of variables in the LP formulation, the solver consumes $\approx 3\times$ more memory than GREEDY-D ($\approx 15\times$ more memory if one includes the constant overhead of loading Matlab, shown by the "init" segment of the LP bar). The memory consumption and quadratic scalability thus limit the graphs that LP can handle.

Figure 4.6a compares the quality of the solution produced by GREEDY-D to the optimal solution produced by the LP, measured as the score of (i.e., sum of edge weights in) the output subgraph H. GREEDY-D consistently achieves a score that is $\geq 97.5\%$ of the LP score with low variability, a range of 0.0044, in that ratio. This indicates that although GREEDY-D is a 2-approximation, it is able to vastly outperform its theoretical guarantee in practice.

Finally, Figure 4.6c evaluates GREEDY-D on the large eBay US graph. The input instances—even the 25% sample of this graph which is $50\times$ larger than the 100% sample of eBay Canada used in the previous experiments—are too large for the LP. Observe that the greedy algorithm scales linearly with the dataset in terms of both run time and memory consumption and can handle the full graph, containing 5.8 million vertices and 11 million edges, in roughly one minute.

To summarise the GA-WBM-D results, the LP is reasonably efficient on small graphs and provides an exact solution. For larger graphs where the LP encounters scalability issues both in terms of efficiency and memory consumption, GREEDY-D appears to obtain a $\geq 97.5\%$ quality solution with very fast run times and good, linear scalability.

GA-WBM-B Table 4.3 gives the running times of the three algorithms for the budget-capped problem on the smaller eBay Canada dataset. Observe that the type of constraints does not influence much the efficiency of our greedy algorithms: GREEDY-B-LF (Algorithm 5) obtains near identical run times, and hence scalability, as GREEDY-D (c.f., Table 4.2). The approximate linear program using rounding (LPR) is about $1.9\times$ faster than the ILP, albeit still $\approx 8\times$ slower than the greedy algorithm.

Figure 4.7b shows the peak memory consumption of the algorithms for the previous experiment. As in Figure 4.6b, the LP-based approaches incur a 400 MB Matlab overhead (the "init" component on the bars), accounting here for half the memory consumption. LPR generally requires slightly more memory than ILP, except on the 100% sample. The better run time scalability of LP on GA-WBM-B, relative to GA-WBM-D, is offset by escalated memory costs: both ILP and LPR require, even

excluding the "init" overhead, an order of magnitude more memory than GREEDY-B-LF.

Figure 4.7a contrasts the solution quality of the two approximate algorithms to the exact solution produced by the ILP on the eBay Canada datasets. Both GREEDY-B-LF and LPR obtain 98% of the ILP score on all samples. LPR outperforms the greedy algorithm by 0.01-0.03%, except on the 25% sample, where the greedy algorithm actually provides the best approximate solution.

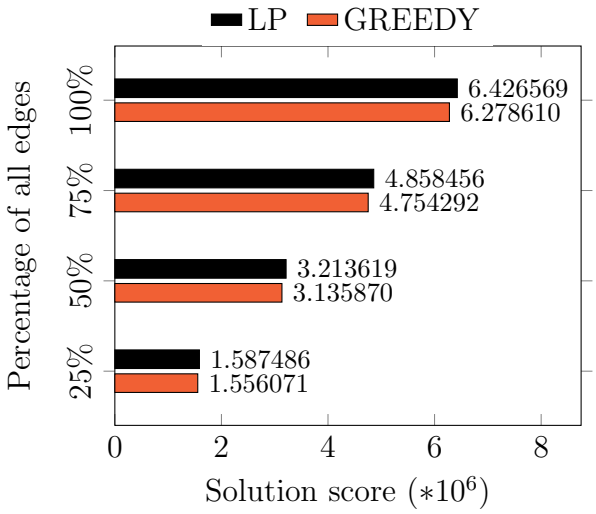
Considering the large eBay US dataset in Figure 4.7c, on which the memory consumption is prohibitive for the LP-based algorithms, we observe that GREEDY-B-LF again achieves linear scalability with respect to both run time and memory consumption, albeit with a gentler slope than in the degree-constrained problem (c.f., Figure 4.6c). On the full graph, the greedy algorithm requires less than 5 GB of memory, indicating that the two-level heap data structure in the lazy forwarding scheme is quite compact.

To summarise the GA-WBM-B results, the ILP is very efficient on small graphs, producing optimal solutions, but it also demands a lot of memory relative to the greedy algorithm. This impedes its scalability to larger graphs. The greedy algorithm achieves excellent scalability while sacrificing $< 2\%$ of the solution quality and easily scales to 10^7 edges. LPR provides a modest improvement in scalability over ILP, coupled with a modest improvement in solution quality over the greedy algorithm, so is poised to handle boundary instances that are slightly too large for ILP.

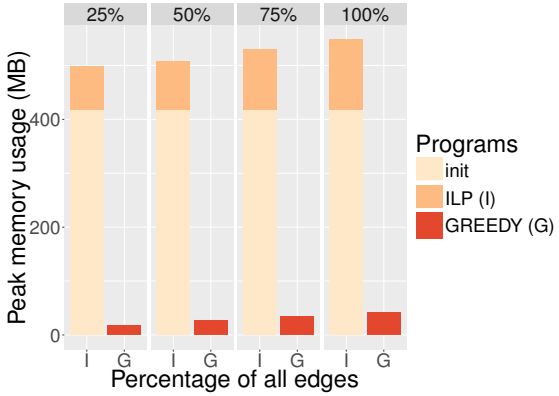
4.6 Conclusions

In this chapter, we investigated generalizations of weighted bipartite B-matching. We gave a new hardness proof for CA-WBM by a reduction from Maximum Weight Independent Set, yielding the stronger result that CA-WBM is hard to approximate. We proposed GA-WBM-D and GA-WBM-B, which partition the right-hand vertex set plus constrain the degree and cap the budget, respectively, of the partitions. While GA-WBM-D can be solved efficiently with linear programming, the number of variables in the linear program creates scalability challenges. GA-WBM-B, on the other hand, is a generalization of the NP-hard maximum budgeted allocation problem so cannot be solved efficiently. Nonetheless, for both problems we introduced intuitive greedy algorithms with approximation guarantees that, in practice, are nearly as

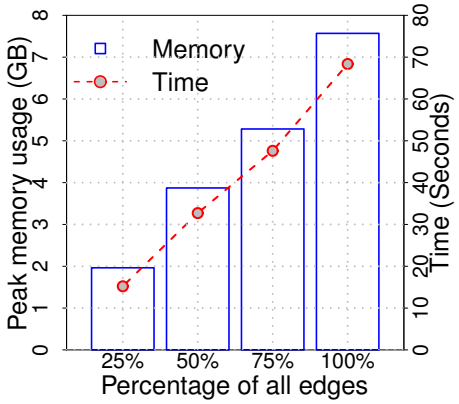
accurate as the linear programs, and moreover can process a dataset with 11.3 million edges in about one minute.



(a) Solution score on $25, 50, 75, 100$ % samples of the eBay Canada dataset

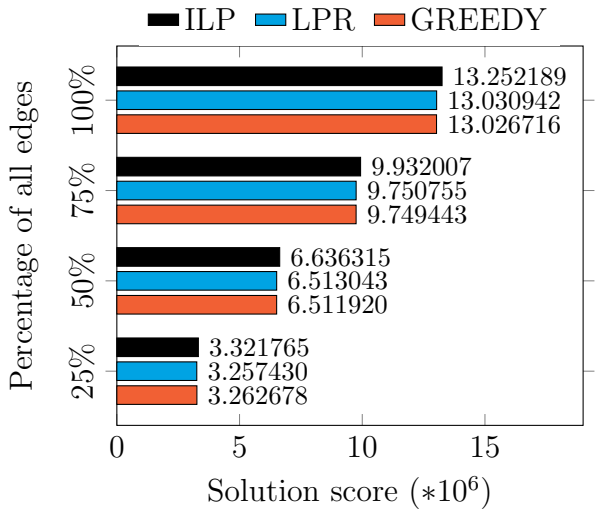


(b) Peak memory usage on $25, 50, 75, 100$ % samples of the eBay Canada dataset

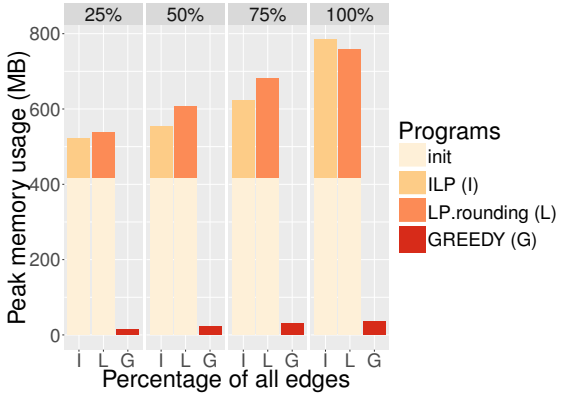


(c) Scalability of GREEDY-D on $25, 50, 75, 100$ % samples of eBay US

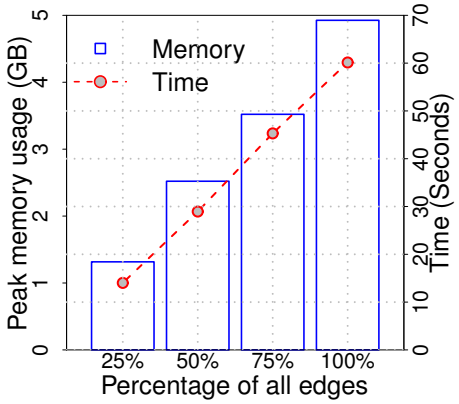
Figure 4.6: Experiment plots for the LP and GREEDY-D algorithms on the degree-constrained problem (GA-WBM-D)



(a) Solution score on $25, 50, 75, 100$ % samples of the eBay Canada dataset



(b) Peak memory usage on $25, 50, 75, 100$ % samples of the eBay Canada dataset



(c) Scalability of GREEDY-B-LF on $25, 50, 75, 100$ % samples of eBay US

Figure 4.7: Experiment plots for the ILP, LPR, and GREEDY-B algorithms on the budget-capped problem (GA-WBM-B)

Chapter 5

From Recommendation to Profile Inference

5.1 Introduction

In this chapter, we investigate the user profile inference problem in an unique business market where users' purchase behaviors are often limited but can be augmented using data collected from a new source. We first introduce the background and discuss how the emerging technologies and application scenarios motivate our general idea of approaching this problem.

In the current era of mobile technology, smart devices (e.g., smartphones, tablets, and smart wearables) have become more prevalent than ever before. Smart devices have provided people with ubiquitous access to the Internet, leading to an ever-growing ecosystem of Mobile Internet. Recent mobile marketing statistics shows that mobile users have outnumbered the desktop users worldwide and over 80% of mobile users access the Internet via smartphones [3]. Following the trend of Mobile Internet, retailers with physical stores (the so-called brick-and-mortar retailers) are building their own wireless access points for smart devices to improve the user experience. Currently, free Wi-Fi services are offered in many places, including cafes, airports, hotels, restaurants, cinemas, and shopping malls.

Considering that retailers were reluctant to invest in Wi-Fi not so long ago [5], it is surprising to see that retailers are now embracing the in-store Wi-Fi. This opens up the opportunity that, in addition to improving customer satisfaction, the retailers could actually obtain a "goldmine" of customer data. With the free Wi-Fi

services provided, retailers can collect useful information about their customers such as their geographic data and dwell times at different locations. The data, which the customers opt to share, offers retailers a better understanding of customers' behavior and demographics and helps them make informed marketing decisions.

Currently, analytics based on Wi-Fi data has become the focus of many Wi-Fi provider companies, such as AirTight Wi-Fi① and Purple Wi-Fi②. They help retailers not only deploy Wi-Fi, but also launch analytics engines for in-store business intelligence and customer engagement. This rapidly evolving market has also attracted the attention of the government agencies. Recently, New York is transforming old phone booths into city-wide free Wi-Fi hotspots, that can collect pertinent information for the purpose of targeted advertising [2].

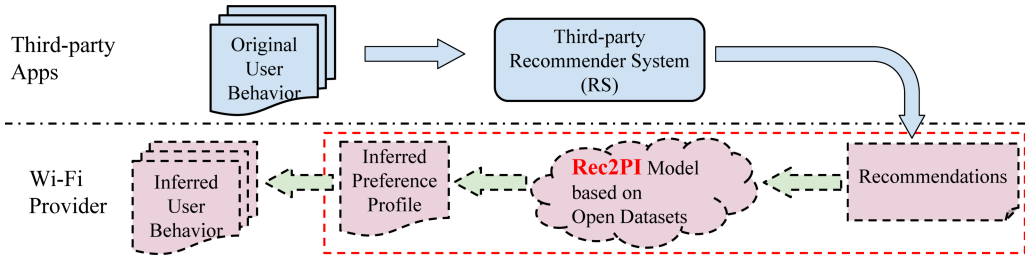


Figure 5.1: The general framework of Rec2PI model (red dashed box) and the work flow.

Compared to major e-commerce companies such as Amazon and eBay that have a great amount of data to learn user behaviors, a brick-and-mortar retailer generally can only collect a limited amount of information. The problem of providing competitive value-added services based on Wi-Fi collected data remains open and has not been well studied in the literature. Most existing industry solutions mine the collected data for basic customer demographics, presence analytics, Wi-Fi usage, and loyalty and engagement [4]. A natural question is: can we gain more knowledge on the customer preference profile for products of interest using a very limited amount of data collected by the Wi-Fi service provider?

In fact, abundant information is hidden in the small amount of Wi-Fi data. The mobile apps offer the customers a wide range of services such as social networking, shopping, and entertainment. After connecting to the in-store Wi-Fi, apps with integrated recommender systems normally push recommendations to the user based

①`http://www.mojonetworks.com/`, Accessed Jan. 2016
②`http://purple.ai/enterprise/`, Accessed Jan. 2016

on the user's past behavior or preference (e.g., purchase history, ratings) and current environment. For example, a customer who is visiting a book store may receive recommended books that the customer might purchase. By exploring the hidden knowledge behind the recommendations, the Wi-Fi provider can practically learn more information regarding this customer.

Recommendation data can be obtained by tracking the user's browsing behavior. In the service agreement (or the privacy policy) of free Wi-Fi services, such as RetailNext, CityBridge and Target, the providers are allowed to collect users' browsing behavior (e.g., URLs, pages visited, etc.) if the users choose to use the free service. In this chapter, we assume that the recommendations pushed by the mobile-apps are (partially) available to the free Wi-Fi service provider, who aims to infer the customer's preference profile based on the collected recommendations. While service agreements of real-world free Wi-Fi providers allow them to collect the above information, the debate on privacy and ethical concerns is out of the scope of the thesis. We believe that being able to infer the preference profile of a customer is beneficial for: (a) the customer because he will be offered a better range of products, enabling him to obtain the best product cheaper, and (b) the brick-and-mortar retailer enabling it to participate in the global advertising business and increase its revenues.

In this chapter, we introduce and study a novel value-added service, Recommendation to Profile Inference (Rec2PI), for Wi-Fi data mining. Rec2PI utilizes a new source of data, i.e., recommendations pushed to a user in a certain domain (e.g., books or movies), to infer the user's preference profile in that domain. Figure 5.1 depicts a general framework, where the red dashed box contains the input, the inference model, and the output of Rec2PI. When a target user logs into the in-store Wi-Fi for Internet access, third-party mobile apps may push recommendations to the user. After collecting the pushed recommendations, Rec2PI infers the user's preference. Once the inferred preference profile is obtained, the Wi-Fi provider can further estimate the customer's behavior, which is valuable for retailers to deliver more personalized services.

Rec2PI is significantly different from any existing work of recommender systems (RS). Traditional RS has universal access to a user's ground truth (e.g., past purchase behavior and item ratings). On the contrary, Rec2PI may not possess this information. As such, Rec2PI can be viewed as a reverse of the learning procedure in RS. The main challenge behind Rec2PI is: without knowing the algorithm(s) and the dataset behind the third-party RS, how can we effectively infer the user's preference profile?

This chapter answers the above challenge. To make the reverse learning possible, we make use of open datasets in the same domain as the recommendations. For example, Epinions is well-known for customer reviews about products. IMDB and MovieLens are popular sources for movie average ratings, meta information, and individual ratings. With the knowledge from the open datasets, we can learn the most likely user factors that would result in the recommendations.

5.2 Assumptions and Preliminaries

5.2.1 Assumptions

To introduce Rec2PI, we begin with a high-level model shown within the red dashed box in Figure 5.1. It consists of the recommendations from third-party RS (the input), the inference model, and inferred user preference profile (the output). Based on the inferred user profile, Rec2PI further estimates the user behavior. Our goal is to make the inferred user behavior as close as possible to the original.

Notation	Explanation	Accessible to Wi-Fi Provider?
t	a target user t	Yes
r_t	t's original behavior	No
r'_t	t's inferred behavior	Yes
$\mathcal{F}$	recommendation model of the third-party RS	No
Λ_t	t's inferred preference profile	Yes
$\hat{\mathcal{R}}_t$	recommendations by third-party RS to t	Yes
R	the open dataset	Yes
U	the user set of the open dataset	Yes
I	the item set of the open dataset	Yes
M_r	the highest rate value	Yes

Table 5.1: Notations of General Rec2PI

Following the notations defined in Table 5.1, we make the following assumptions.

- Let $\mathcal{F} : r_t \rightarrow \hat{\mathcal{R}}_t$ denote the recommendation model of the third-party app's RS. We will assume that $\mathcal{F}$, as well as the dataset it uses, are hidden from the Wi-Fi provider.
- The Wi-Fi provider has the access to an open dataset (called open dataset hereafter), which belongs to the same category as the dataset that the third-

party RS uses (called hidden dataset hereafter). For example, both are movie datasets or both are book datasets. Nevertheless, there is no guarantee that the two datasets are identical.

- Rec2PI does not rely on any hypothesis regarding the recommendation algorithm used by $\mathcal{F}$.

In Rec2PI, we first need to determine a method to represent users' preference profile, Λ_t. There are different ways to represent preference profiles, including the vector representation [28], ontology representation [102], and multidimensional representation [87]. Among these, the latent factor model (LFM) [84], a well-known variant of the vector representation, has been the most popular one. We adopt LFM in the chapter for users' preference profile. We stress that LFM is only used in Rec2PI. The third-party RS does not necessarily use LFM inside $\mathcal{F}$.

5.2.2 Background of Latent Factor Models

Latent factor models (LFMs) assume that a user's behavior (e.g., purchases and ratings) is influenced by a set of latent factors. The term "latent" implies that these factors do not necessarily correspond to physical meanings. LFMs serve as one of the most popular collaborative filtering (CF) techniques in rating-based item recommendation [84].

Definition 4 (User Behavior)**.** *The behavior of a user u is captured by a vector of rating, denoted by $\mathbf{r}_u$ as follows:*

$$\mathbf{r}_u = [r_{u,i_1}, r_{u,i_2}, \ldots, r_{u,i_n}]^T, \tag{5.1}$$

where $i_j (j = 1, \ldots, n)$ denote the items that user u has rated and r_{u,i_j} denotes the rating that the user gave on item i_j.

Definition 5 (User Preference Profile)**.** *The (latent) preference profile of user u, denoted as Λ_u, is a D-dimensional vector,*

$$\Lambda_u = [u^{(1)}, u^{(2)}, \ldots, u^{(D)}]^T, \tag{5.2}$$

where each $u^{(d)}, d = 1, \ldots, D$, is called a latent factor of user u, and D is the total number of latent factors.

Definition 6 (Item Latent Profile)**.** *The (latent) profile of item i, denoted as Γ_i, is also a D-dimensional vector,*

$$\Gamma_i = [i^{(1)}, i^{(2)}, \ldots, i^{(D)}]^T, \tag{5.3}$$

where each $i^{(d)}, d = 1, \ldots, D$, is called a latent factor of item i, and D is the total number of latent factors.

The latent profile Λ_u represents the user's preference in the D-dimensional latent factor space, and Γ_i captures item i's feature in the D-dimensional latent factor space. A LFM produces predicted rating that user u gives to i, $r'_{u,i}$, as

$$r'_{u,i} = \Lambda_u^T \cdot \Gamma_i. \tag{5.4}$$

A LFM thus tries to learn the latent factors for all users and all items, by minimizing the regularized squared error [84]:

$$\underset{\Lambda_u, \Gamma_i, \theta_1, \theta_2}{\operatorname{argmin}} \sum_{r_{u,i} \in \mathbf{R}} (r_{u,i} - \Lambda_u^T \cdot \Gamma_i)^2 + \theta_1 \cdot ||\Lambda_u||^2 + \theta_2 \cdot ||\Gamma_i||^2, \tag{5.5}$$

where $\mathbf{R} = (r_{u,i})_{|U| \times |I|}$ represents a (sparse) rating matrix which contains the ground truth ratings for $u \in U$ and $i \in I$, U and I denote the set of users and the set of items, respectively. This can be seen as factorizing a rating matrix $\mathbf{R}$ into a user factor matrix and an item factor matrix.

5.3 Problem Formulation

5.3.1 The Goal of Rec2PI

Definition 7 (User's Recommendations)**.** *The recommendations from the third-party RS to a target user t, $\hat{\mathcal{R}}_t$, is a vector of ratings*[③] *as follows:*

$$\hat{\mathcal{R}}_t = [\hat{r}_{t,i_1}, \hat{r}_{t,i_2}, \ldots, \hat{r}_{t,i_{\hat{n}}}]^T, \tag{5.6}$$

where $i_j \in \hat{I}_t$ is an item in t's recommended item set $\hat{I}_t$ with rating $\hat{r}_{t,i_j}$, and $\hat{n} = |\hat{I}_t|$.

[③]A prominent RS that sends predicted ratings along with recommended items is Netflix (refer to Section 5.5.1 as well).

Denote $\mathcal{G} : \hat{\mathcal{R}}_t \rightarrow \Lambda_t$ as the inference function of Rec2PI. **The goal of Rec2PI is thus to estimate Λ_t by applying $\mathcal{G}$ into an open dataset.** Once Λ_t is available, Rec2PI can use LFM to produce the inferred user behavior r'_t.

5.3.2 Why Does Traditional RS Not Work for Rec2PI?

Estimating Λ_t based on $\hat{\mathcal{R}}_t$ is fundamentally different from traditional RS, due to the fact that (1) the open dataset is not the same as the hidden dataset and the target user may not exist in the open dataset, (2) we may not have any ground truth rating[④] from the target user. Consequently, Rec2PI needs to tackle two challenges:

1. Recommended item ratings $\hat{\mathcal{R}}_t$ cannot be used in the same way as the ground truth ratings in traditional RS that uses the ground truth ratings to obtain latent factors by solving (5.5). Recommended item ratings are determined by a specific third-party RS and do not necessarily correspond to the user's true ratings.

2. The details of $\mathcal{F}$ is hidden from the Wi-Fi provider, meaning that Rec2PI has no knowledge on how recommended ratings are generated. Since there are many recommendation algorithms even in the same domain (e.g., movie recommendation), we cannot assume a specific algorithm for $\mathcal{F}$. Guessing the algorithms behind a *proprietary* RS is still an open challenge.

5.3.3 Intuition and Discussion

One may wonder about two obstacles we need to overcome in Rec2PI: why is it possible to infer a user's profile based on the recommendations generated with some unknown algorithm over a hidden dataset? how can the accuracy of the inference results be evaluated without knowing the original user behavior (ground truth)?

To answer the first question, we give an intuitive explanation before we dive into technical details. In principle, an RS makes recommendations based on the *dependence* between the features of items and the flavor of the user. This dependence structure reflects the statistical patterns *inherent* in the real-world phenomenon, e.g., males tend to love action movies more than females. These statistical patterns can be found in different datasets. In other words, it is the statistical patterns instead

[④]Ground truth ratings of the user in consideration are part of the input in traditional RS, because RS will not be able to create a personalized recommendation if the user has not rated any item.

of the uniqueness of dataset that determine the recommendation results. Two different datasets in the same domain (e.g., user-movie ratings), even if they are not identical, should exhibit similar statistical patterns as long as both are large enough to reflect the real-world phenomenon. This explains why we can infer a user's profile based on the recommendations generated from a hidden dataset using an unknown recommendation algorithm.[5]

To answer the second question, we design a special evaluation method to avoid the ground-truth problem in the evaluation of Rec2PI. Our idea is to randomly select users from the open dataset so that we know their ground-truth behavior. For each selected user, we manually create an "agent" user in the third-party service and manually set the ratings of the "agent" user the same as the ones in the corresponding user in the open dataset. Note that since the third-party datasets and open datasets belong to the same category (e.g., books or movies), it is reasonable to assume that the items contained in both datasets have overlaps and that we can find corresponding items in the third-party datasets for rating assignments. The recommendations to the "agent" user by the third-party RS will be used as the input to Rec2PI. The inferred behavior by Rec2PI for the "agent" user is compared to the ground-truth behavior in the corresponding user in the open dataset. More details will be disclosed in Section 5.5.

5.3.4 A New Approach

To overcome the difficulties raised in Section 5.3.2, we model Rec2PI in a new approach different from traditional RS. Specifically, we only assume limited prior knowledge about $\mathcal{F}$, i.e., $\mathcal{F}$ is trained on a certain type of hidden datasets (e.g., user-movie ratings) in a particular domain. Although Rec2PI does not have access to the hidden datasets, it can make use of similar types of open datasets in the same domain and learn similar patterns via popular methods. The open datasets do not necessarily contain the target user's records, and therefore we do not attempt to infer t's profile with Equation (5.5). Instead, we consider each recommended item $i \in \hat{I}_t$ individually and explore its association with the user, i.e., what type of users are likely to be interested in the recommended item. In other words, given a recommended item, we first obtain its latent factors using the open dataset and then calculate the most-likely

[5]The unknown algorithm is assumed to make reasonable recommendations that reflect the statistical patterns.

latent factors that a user should have such that the item would be recommended to the user.

This task requires us to study the dependence between user latent factors and item latent factors. To model the relationship in a probabilistic approach, we first denote $\mathcal{U} = \{\mathcal{U}^{(1)}, \ldots, \mathcal{U}^{(D)}\}$ as a set of continuous random variables for each dimension of user factors, and $\mathcal{I} = \{\mathcal{I}^{(1)}, \ldots, \mathcal{I}^{(D)}\}$ as a set of continuous random variables for each dimension of item factors. Let f denote the probability density function (PDF) and F the cumulative density function (CDF). In addition, let $\mathbf{E}$ denote the set of observed evidence variables (i.e., item factors and the ratings in the open dataset). Given a set of recommended items $\hat{I}_t$ with ratings $\hat{\mathcal{R}}_t$, Rec2PI solves a series of the maximum a posteriori probability estimation problems (MAP) as follows:

$$\text{MAP}(\mathcal{U}|\mathbf{E}) = \underset{\mathbf{u}}{\text{argmax}}\, f(\mathbf{u}|\mathbf{i}, \hat{r}_{t,i}), \forall i \in \hat{I}_t, \tag{5.7}$$

where $\mathbf{u}$ and $\mathbf{i}$ represent values for random variables in $\mathcal{U}$ and $\mathcal{I}$, respectively.

5.4 Copula-based Probabilistic Profile Inference

5.4.1 Outline of Solution

We introduce our proposed method to solve the problem formulated in Equation (5.7) by converting the equation to a more detailed form in our context. Without loss of generality, we assume that $\hat{r}_{t,i}$ has an integer range[6]. Denote the rating matrix in the open dataset as $\mathbf{R}$. Denote the range of ratings as $[1, M_r]$. It is reasonable to assume that the item sets (e.g., movies) in the hidden dataset (used in the third-party RS) and the open dataset (explored by Rec2PI) are close, because we have the freedom to use any open dataset that sufficiently covers all pushed items to the target user.

Ratings of recommended items, predicted by the (unknown) recommendation algorithm $\mathcal{F}$, reflect the statistical patterns in user-item association. Since we do not attempt to minimize the distance metric such as that defined in Equation (5.5), we solve the problem based on the fact that the ratings reflect the dependence structure between item factors and user factors. As such, for each rating value $x \in [1, M_r]$, we

[6]Decimal ratings can be converted into integers.

associate x with $\mathcal{U}_x = \{\mathcal{U}_x^{(1)}, \ldots, \mathcal{U}_x^{(D)}\}$, the set of continuous random variables for each dimension of user factors, and with $\mathcal{I}_x = \{\mathcal{I}_x^{(1)}, \ldots, \mathcal{I}_x^{(D)}\}$, the set of continuous random variables for each dimension of item factors. The details on how to obtain sample values of $\mathcal{U}_x$ and $\mathcal{I}_x$ from $\mathbf{R}$ will be given in Subsection 5.4.3.

A common assumption in LFM is that the latent user factors are independent of each other [104, 94]. This assumption suggests that we can infer the D-dimensional user factors one at a time. Therefore, corresponding to a recommended item $i \in \hat{I}_t$ with rating $\hat{r}_{t,i}$, we can infer user t's d-th latent factor, conditioning on i, denoted as $\Lambda_{t,i}^{(d)}$, as

$$\Lambda_{t,i}^{(d)} = \underset{u_x^{(d)}}{\operatorname{argmax}} f(u_x^{(d)} | \mathbf{i}_x), \forall i \in \hat{I}_t, \tag{5.8}$$

where to simplify notation $x = \hat{r}_{t,i}$ and $\mathbf{i}_x = \{i_x^{(1)}, \ldots, i_x^{(D)}\}$. Note that $\mathbf{i}_x = \{i_x^{(1)}, \ldots, i_x^{(D)}\}$ represent the values for random variables in $\mathcal{I}_x = \{\mathcal{I}_x^{(1)}, \ldots, \mathcal{I}_x^{(D)}\}$.

We then take the average value over all $\Lambda_{t,i}^{(d)}, \forall i \in \hat{I}_t$ as the final value of inferred d-th latent factor, i.e.,

$$\Lambda_t^{(d)} = \frac{1}{|\hat{I}_t|} \sum_{i \in \hat{I}_t} \Lambda_{t,i}^{(d)}. \tag{5.9}$$

Nevertheless, $f(u_x^{(d)} | \mathbf{i}_x)$ in Equation (5.8) is non-trivial to model because it involves the dependence structure between user factors and item factors. In the next subsection, we will introduce a powerful statistical method, copula modeling, that can characterize this dependence information.

5.4.2 Why Copula-based Inference?

We start the introduction to copulas with the definition of 2-dimensional (bivariate) copulas and core theorems.

Definition 8. *A 2-dimensional copula is a function C having the following properties [106]:*

1. *Its domain is $[0,1]^2$;*
2. *For every $u_1, u_2, v_1, v_2 \in [0,1]$ and $u_1 \leq u_2, v_1 \leq v_2$, we have $C(u_2, v_2) - C(u_2, v_1) - C(u_1, v_2) + C(u_1, v_1) \geq 0$.*
3. *$C(u, 0) = C(0, v) = 0$, $C(u, 1) = u$, $C(1, v) = v$, for every $u, v \in [0,1]$.*

The second property describes the 2-dimensional version of an increasing function of one variable. A function with this property is known to be "2-increasing".

Theorem 12. ***(Sklar's theorem)*** *[106] Consider two random variables X and Y, with $F(x, y)$ as the joint CDF. Denote the marginal CDF of X and the marginal CDF of Y as $F_X(x)$ and $F_Y(y)$, respectively. Then there exists a copula C such that for all x and y, $F(x, y) = Pr(X \leq x, Y \leq y) = C(F_X(x), F_Y(y))$.*

Sklar's theorem shows how the copula models the dependence between univariates. The copula is a function that links univariate marginals to their joint distributions. This property is especially useful since the joint distribution of random variables is difficult to find directly in many applications. For example, in our problem, when considering a feature of users and a feature of items as two random variables, their joint distribution is not easy to obtain. By using copulas to model the dependence, their joint behavior can be revealed based on Theorem 12 by integrating marginal distribution into a copula model.

Copulas are efficient and well known for dependence modeling and have been widely used in finance and risk management [40, 106]. There are a variety of copula families which are well studied, such as Archimedean copulas, Gaussian copula, student's t copula, and etc. These known copulas are powerful tools to capture different types of dependence structure [40], so that we will have enough choices when using copulas for dependence modelling in our problems. In addition, compared with other dependence measurements, such as covariance and correlation, copulas not only capture the linear dependence, but also model the functional dependence between random variables [40]. In the context of Rec2PI, there is no evidence that the dependence structure between users and items is linear. In this case, copulas can be used to model this dependence well even if the dependence goes beyond the linear scope.

The following theorem is also useful in our copula-based model.

Theorem 13. *[14] Assume X and Y have the copula C between them, the conditional density function $f(Y = y|X = x)$ is*

$$f(Y = y|X = x) = c(F_X(x), F_Y(y))f_Y(y) \tag{5.10}$$

where $c(u, v) = \frac{\partial}{\partial u}\frac{\partial}{\partial v}C(u, v)$ is called the copula density function, and $f_Y(y)$ is the probability density function (PDF) of the marginal.

5.4.3 Copula-based Probabilistic Model (CPM)

To ease notation, in the rest of the chapter, we use f to denote the PDF for univariate, bivariate and multivariate and F to denote the CDF for univariate, bivariate and multivariate, without using subscripts. For instance, $F(i_x^{(1)})F(i_x^{(2)})\cdots F(i_x^{(D)})$ should be read as $F_{\mathcal{I}_x^{(1)}}(i_x^{(1)})F_{\mathcal{I}_x^{(2)}}(i_x^{(2)})\cdots F_{\mathcal{I}_x^{(D)}}(i_x^{(D)})$.

We use the copula method to solve Equation (5.8). To simplify calculation, we in this section assume that the item factors are independent of each other. In the next section, we will relax this assumption and introduce an improved algorithm that can handle the dependence between item factors.

Assume $\hat{r}_{t,i} = x$. Due to the independence assumption on item latent factors, $F(\mathbf{i}_x)$ can be computed as:

$$\begin{aligned} F(\mathbf{i}_x) &= F(i_x^{(1)}, i_x^{(2)}, \ldots, i_x^{(D)}) \\ &= F(i_x^{(1)})F(i_x^{(2)})\cdots F(i_x^{(D)}) \end{aligned} \tag{5.11}$$

Based on Theorem 13, Equation (5.8) can be rewritten as:

$$\begin{aligned} \Lambda_{t,i}^{(d)} &= \underset{u_x^{(d)}}{\operatorname{argmax}}\, f(u_x^{(d)}|\mathbf{i}_x) \\ &= \underset{u_x^{(d)}}{\operatorname{argmax}}\, c_x(F(u_x^{(d)}), F(\mathbf{i}_x))f(u_x^{(d)}), \quad \forall i \in \hat{I}_t, \end{aligned} \tag{5.12}$$

where c_x is the bivariate copula density function of $\mathcal{U}_x^{(d)}$ and $\mathcal{I}_x$. Note that we transform $\mathcal{I}_x$ as one random variable with Equation (5.11). From Equation (5.12), the inference on d-th latent factor will be made by modelling its dependence with recommended items and its own marginals.

Next, we present an algorithm that, for each rate value $x \in [1, M_r]$, constructs the sample values for $\mathcal{U}_x = \{\mathcal{U}_x^{(1)}, \ldots, \mathcal{U}_x^{(D)}\}$ and $\mathcal{I}_x = \{\mathcal{I}_x^{(1)}, \ldots, \mathcal{I}_x^{(D)}\}$, using the open dataset $\mathbf{R}$.

Let N_x denote the total times of occurrence for a rating value x in $\mathbf{R}$. For a rating value x, let a $(N_x \times D)$-dimensional matrix $\mathbf{L}_U^{(x)}$ denote the user-side matrix where each row is a vector of user latent factors, and let a $(N_x \times D)$-dimensional matrix $\mathbf{L}_I^{(x)}$ denote an item-side matrix where each row is a vector of item latent factors. Algorithm 6 presents the details of constructing $\mathbf{L}_U^{(x)}$ and $\mathbf{L}_I^{(x)}$, for all $x \in \{1, \ldots, M_r\}$.

Algorithm 6: ***Preprocessing for Latent Factors***

Input: A rating matrix $\mathbf{R}$, user set U and item set I
Output: $\mathbf{L}_U^{(x)}$ and $\mathbf{L}_I^{(x)}$, $\forall x \in \{1, \ldots, M_r\}$
1 Learn a LFM model by factorizing $\mathbf{R}$ with Equation (5.5) to obtain D-dimensional latent factors Λ_u for $u \in U$ and Γ_i for $i \in I$;
2 For all $r_{u,i} \in \mathbf{R}$, insert Λ_u^T into $\mathbf{L}_U^{(r_{u,i})}$, and insert Γ_i^T into $\mathbf{L}_I^{(r_{u,i})}$;

As an example, consider the case where any rating value $x \in \{1, 2, 3, 4, 5\}$. Algorithm 6 outputs 5 user-side matrices $\{\mathbf{L}_U^{(1)}, \cdots, \mathbf{L}_U^{(5)}\}$, and 5 item-side matrices $\{\mathbf{L}_I^{(1)}, \cdots, \mathbf{L}_I^{(5)}\}$.

In Algorithm 7, we describe the training and inference procedures. We place Gaussian priors over $\mathcal{U}_x^{(d)}$ and $\mathcal{I}_x^{(d)}$, i.e., $\mathcal{U}_x^{(d)} \sim \mathcal{N}(\mu_{\mathcal{U}_x^{(d)}}, \sigma_{\mathcal{U}_x^{(d)}})$ and $\mathcal{I}_x^{(d)} \sim \mathcal{N}(\mu_{\mathcal{I}_x^{(d)}}, \sigma_{\mathcal{I}_x^{(d)}})$. The samples for $\mathcal{U}_x^{(d)}$ is the d-th column of $\mathbf{L}_U^{(x)}$. Similarly, the samples for $\mathcal{I}_x^{(d)}$ is the d-th column of $\mathbf{L}_I^{(x)}$.

5.4.4 Vine-copula Probabilistic Model (VPM)

In **Line 7** of Algorithm 7, we assume that the D random variables of item latent factors are independent, i.e., $\mathcal{I}_x^{(j)} \perp\!\!\!\perp \mathcal{I}_x^{(k)}, j \neq k$, so that we can transform $\mathcal{I}_x$ as one random variable with Equation (5.11) and apply bi-variate copula modeling. In this section, we relax this assumption and propose a vine-copula probabilistic model (VPM) to explore the dependence between item factors for further improvement.

Figure 5.2 shows the correlation coefficient values (i.e., linear dependence[7]) among 10 user/item factors learned from LFM on a movie ratings dataset, the details of which will be described in Section 5.5. While the dependence between user factors is weak, we can easily observe the strong correlations between item factors (e.g., $V2$-$V9$, $V5$-$V8$). Therefore, by capturing the dependence structure among the D random variables for item factors, we may be able to obtain better inference results in Rec2PI.

While so far we introduced copulas for the bivariate case, all definitions and theorems in Subsection 5.4.2 can be extended to multivariate copulas [106]. In this section, multivariate copulas are used to capture the dependence between item factors, i.e., to calculate the joint CDF $F(\mathbf{i}_x) = F(i_x^{(1)}, i_x^{(2)}, \ldots, i_x^{(D)})$.

A general approach to construct multivariate copulas is to extend a bivariate copula into the high dimensional version. Such extension has been made for multivariate

[7]Note that the linear dependence between item factors does not suggest/imply the linear dependence between users and items.

Algorithm 7: ***Bivariate-CPM***

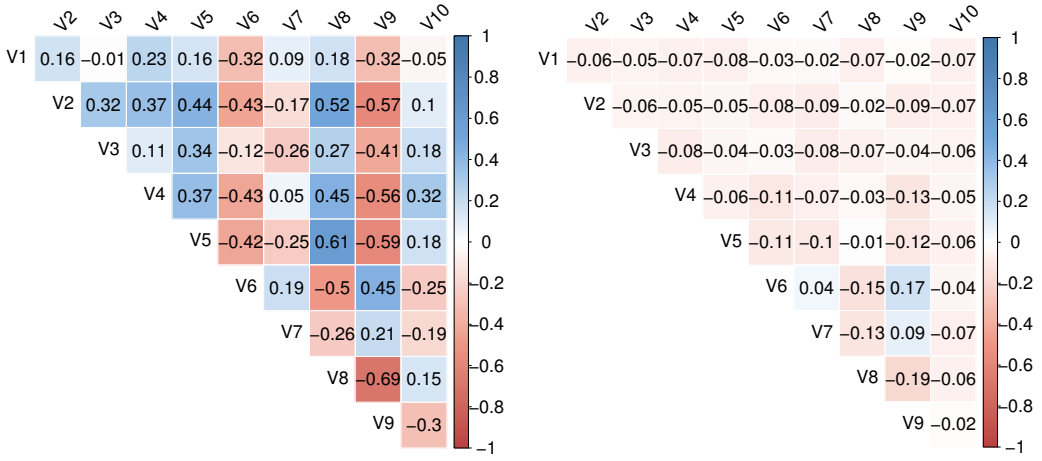
Input: $\mathbf{L}_U^{(x)}$ and $\mathbf{L}_I^{(x)}$, $x \in \{1, \ldots, M_r\}$, and a set of recommended items $\hat{I}_t$ with ratings $\hat{\mathcal{R}}_t$

Output: The target user t's inferred profile $\Lambda_t = [t^{(1)}, t^{(2)}, \ldots, t^{(D)}]^T$

foreach x *in* $\{1, \ldots, M_r\}$ **do**
  foreach d *in* $\{1, \ldots, D\}$ **do**
    Fit a Gaussian distribution $\mathcal{N}(\mu_{\mathcal{U}_x^{(d)}}, \sigma_{\mathcal{U}_x^{(d)}})$ to the d-th column of $\mathbf{L}_U^{(x)}$, i.e., $\mathbf{L}_U^{(x)}[:, d]$;
    Fit a Gaussian distribution $\mathcal{N}(\mu_{\mathcal{I}_x^{(d)}}, \sigma_{\mathcal{I}_x^{(d)}})$ to the d-th column of $\mathbf{L}_I^{(x)}$, i.e., $\mathbf{L}_I^{(x)}[:, d]$;
    Set $\mathcal{U}_x^{(d)} \sim \mathcal{N}(\mu_{\mathcal{U}_x^{(d)}}, \sigma_{\mathcal{U}_x^{(d)}})$ and $\mathcal{I}_x^{(d)} \sim \mathcal{N}(\mu_{\mathcal{I}_x^{(d)}}, \sigma_{\mathcal{I}_x^{(d)}})$;
    Compute a CDF value for each entry in the d-th column of $\mathbf{L}_U^{(x)}$ as $F(u_x^{(d)})$;
    Compute a CDF value for each row of $\mathbf{L}_I^{(x)}$ as $F(\mathbf{i}_x)$ by Equation (5.11) ;
    Fit a bivariate copula $c_x(F(u_x^{(d)}), F(\mathbf{i}_x))$ to the CDF values of $F(u_x^{(d)})$ and $F(\mathbf{i}_x)$;
foreach d *in* $\{1, \ldots, D\}$ **do**
  $\Lambda_t^{(d)} = \frac{1}{|\hat{I}_t|} \sum_{i \in \hat{I}_t} \underset{u_x^{(d)}}{\operatorname{argmax}}\, c_x(F(u_x^{(d)}), F(\mathbf{i}_x)) f(u_x^{(d)})$;
return $\Lambda_t = [t^{(1)}, t^{(2)}, \ldots, t^{(D)}]^T$;

Gaussian copula and multivariate student's t copula. Taking multivariate Gaussian copula as an example, it essentially models the dependence between any pairs of the multivariates with a bivariate Gaussian copula. This approach, however, is inflexible in high dimensions. In addition, since it constrains all pairs of random variables with the same dependence structure, the modelling power is limited.

Another construction method for multivariate copulas is to decompose the multivariate distribution into products of marginal PDFs and bivariate copulas PDFs. This method is called pair-copula construction (PCC). With PCC, each pair-copula can be chosen independently from the others, making the model more flexible. For high dimensional distributions, PCC often results in a great number of possible pair-copula constructions. Brechmann et al. [24] proposed to organize these constructions using a graphical model involving a sequence of nested trees, which are denoted as *regular vines*. There are two popular special cases of regular vines: the canonical (C-) vine and the D-vine. Each vine decomposes a multivariate distribution in a



(a) Item factors. (b) User factors.

Figure 5.2: Correlation coefficients between 10-dimensional of latent factors (rating $x = 3$). Correlation coefficient values correspond to the colors on right color bar.

specific structure. In this chapter, we adopt D-vine in our algorithm. The D-vine [24] decomposes a D-dimensional multivariate PDF $f(\mathbf{x})$ as

$$f(\mathbf{x}) = \prod_{d=1}^{D} f(x_d) \times \prod_{i=1}^{D-1} \prod_{j=1}^{D-i} c_{j,j+i|j+1,\ldots,(j+i-1)}(F(x_j|x_{j+1},\ldots,x_{j+i-1}), F(x_{j+i}|x_{j+1},\ldots,x_{j+i-1})), \tag{5.13}$$

where $f(x_d), d = 1, \cdots, D$, denote the marginal PDFs and $c_{j,j+i|j+1,\ldots,(j+i-1)}$ denote bivariate copula densities. Joe [72] showed that marginal conditional distributions of the form $F(x|\mathbf{v})$ in Equation (5.13) can be computed as:

$$F(x|\mathbf{v}) = \frac{\partial C_{x|v_j}(F(x|\mathbf{v}_{-j}), F(v_j|\mathbf{v}_{-j}))}{\partial F(v_j|\mathbf{v}_{-j})} \quad \forall v_j \in \mathbf{v}, \tag{5.14}$$

where v_j is any element of $\mathbf{v}$ and $\mathbf{v}_{-j}$ denotes the vector without this element. Once $f(\mathbf{x})$ is calculated, we use sampling to estimate the joint CDF $F(\mathbf{x})$. Given any instance $\mathbf{i}$ (values of an item's factors), the idea is to generate S samples from $f(\mathbf{x})$, and count the number of samples s satisfying $x^{(d)} \leq i^{(d)}, \forall d \in \{1,\ldots,D\}$. Then $F(\mathbf{i}_x)$ in Line 7 of Algorithm 7 can be calculated as $F(\mathbf{i}_x) \approx s/S$.

To summarize, VPM is also based on Algorithm 7, except that it uses Equations (5.13), (5.14) and a sampling technique to calculate $F(\mathbf{i}_x)$ in Line 7 of Algorithm 7.

5.5 Experimental Evaluation

We conduct experiments to evaluate our proposed methods, CPM and VPM, in the scenario of movie preference inference. To avoid the ground-truth problem raised in Section 5.3.3, we design an evaluation method shown in Figure 5.3, whose details will be given in the next subsection. Note that the experimental steps shown in Figure 5.3 is for purpose of evaluation only. The true work flow of Rec2PI in real world should follow Figure 5.1. In addition, we perform evaluation in the movie domain not only because it is an important source of preference information, but also because it allows us to conveniently collect item recommendations generated by third-party RS (details explained in the next subsection), which are the input to Rec2PI but are often not included in publicly available datasets. Rec2PI can be applied to data in other domains as well, such as transactional datasets (purchases as binary ratings) in e-commerce.

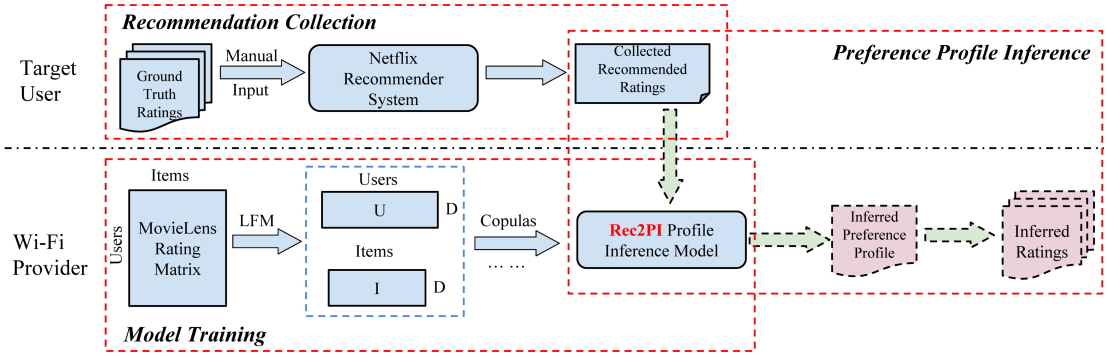


Figure 5.3: Experimental setup for the evaluation of Rec2PI (Refer to Subsection 5.5.1 for details).

5.5.1 Data Preparation and Evaluation Steps

In the experiments, we use MovieLens as the open dataset to train Rec2PI for the Wi-Fi provider side and use Netflix's recommender system to create recommendations for target users. We describe how we select target users and collect recommended movie ratings in the following steps:

TID	1	2	3	4	5	6	7	8	9	10
$\|I_t\|$	55	54	57	49	56	51	44	53	46	46
$\|\hat{I}_t\|$	49	47	49	49	49	47	48	47	39	41
TID	11	12	13	14	15	16	17	18	19	20
$\|I_t\|$	22	93	22	60	23	21	76	26	73	21
$\|\hat{I}_t\|$	50	50	50	50	50	50	50	50	50	50

Table 5.2: Statistics of Target Users

1. We use the latest (released in April, 2015) MovieLens-20m dataset[⑧] as the open dataset. We first filter movies in MovieLens to retain those appearing in both MovieLens and Netflix.

2. We randomly select target users from MovieLens (so that ground truth ratings can be obtained) and exclude them as well as the associated ratings from the dataset used for evaluation afterwards.

3. For each target user, we randomly choose a set of titles of his/her unrated movies from MovieLens. Movies in this set serve as the recommended ones whose predicted ratings will be determined by Netflix's RS.

4. For each target user, we create an individual "agent" account on Netflix and manually assign the ground truth ratings to the corresponding movies for the account.

5. On each target user's account, we search the titles of the selected unrated movies on Netflix's RS. We then record Netflix's predicted ratings.

We summarize the key aspects in data preparation:

- **Open Dataset:** The filtered MovieLens 20m dataset contains 137055 users, 1231 movies and 2606513 movie ratings.

- **Target Users:** In total, we select 20 distinct target users from the Movielens dataset. This simulates a real-world scenario where 20 customers are in the store and accessing the Internet via Wi-Fi. In reality, a target user t's number of rated movies $|I_t|$ (with ground truth ratings) may or may not be near the number of recommended movies $|\hat{I}_t|$ (which is set to 50 in the experiments). To

[⑧] `http://grouplens.org/datasets/movielens/20m/`

conduct experiments on different users, we randomly select 10 targets (T1 to T10) from the set of users whose $|I_t|$ is within the range from 50 to 60. We also randomly select 10 targets (T11-T20), whose $|I_t|$ is in a wider range from 20 to 100. Table 5.2 summarizes $|I_t|$ of each target user[⑨].

- **Ground Truth Ratings:** Each target user's ground truth ratings are all ratings he/she assigns in MovieLens. We round decimal ratings to integers based on IEEE 754 standard for arithmetic operations.

- **Movie Titles of Recommendations:** We sort movie titles in the filtered MovieLens by the number of ratings in descending order. For each target user, we randomly select 50 movies from the top-500 as the recommended movie set whose ratings will be predicted by Netflix. Table 5.2 summarizes $|\hat{I}_t|$ of each target user[9].

- **Predicted Ratings by Netflix:** Netflix displays the predicted rating via a feature shown under the movie title as "Our best guess for ... is ...". We record such ratings for all movies in the recommended movie set. We round decimal ratings to integers based on IEEE 754 standard for arithmetic operations.

5.5.2 Metrics

In the experiments, Rec2PI infers the target user's preference profile about movies. To evaluate the effectiveness, we need to first obtain the user's inferred behavior (inferred ratings) based on the inferred profile and then compare it to the original behavior (ground truth ratings). For a number of repeated runs (10 in the experiments), the average distance between inferred ratings and ground truth ratings, from retailers' viewpoint, should not only be small (accuracy), but also remain stable (stability). Therefore, we use the following two metrics:

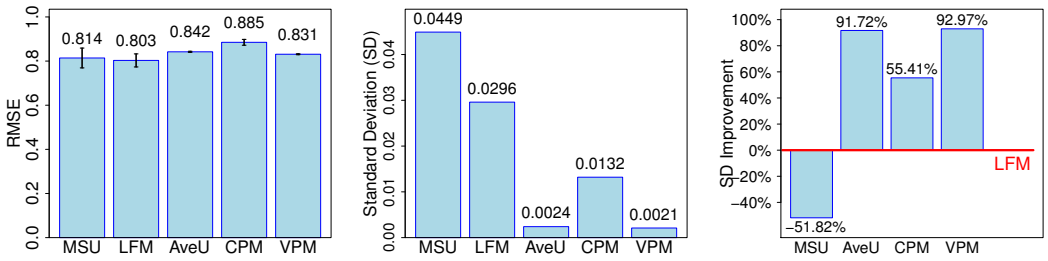
1. Root mean squared error (RMSE) to measure the distance in each run. RMSE is defined as:

$$\text{RMSE} = \sqrt{(\sum_{n=1}^{N}(r_n - r'_n)^2)/N}, \tag{5.15}$$

[⑨]For some target users, $|I_t|$ or $|\hat{I}_t|$ differs from the range or value setting because Netflix is constantly updating their streaming movies and some pre-selected movies have become inaccessible when we collect the ratings (Dec. 2015).

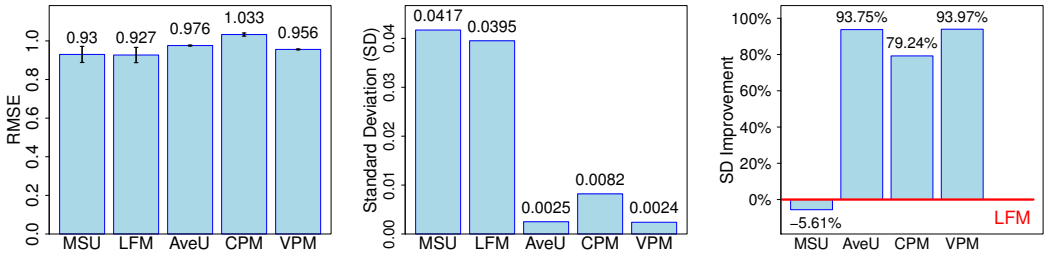
where N is the number of ground truth ratings, r_n is the ground truth rating and r'_n is the inferred rating. Denote $\text{MEAN}_{\text{RMSE}}$ as the average RMSE value of all repeated runs on each target user.

2. Standard deviation of RMSE values (SD_{RMSE}) to measure the stability.



(a) Ave($\text{MEAN}_{\text{RMSE}}$). (b) Ave($\text{SD}_{\text{RMSE}}$). (c) SD IMP over **LFM**.

Figure 5.4: Average metrics and SD improvements against LFM of users from T1 to T10.

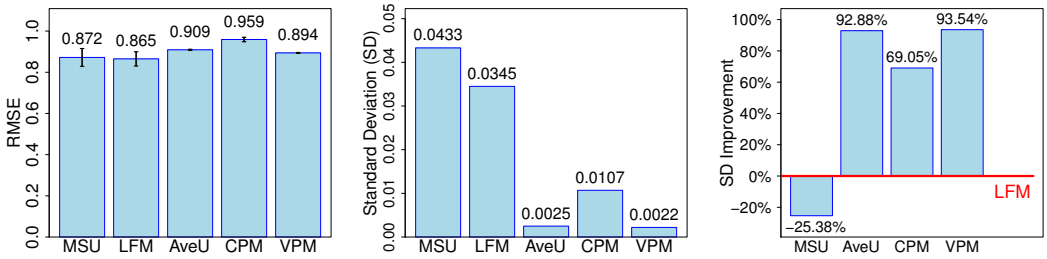


(a) Ave($\text{MEAN}_{\text{RMSE}}$). (b) Ave($\text{SD}_{\text{RMSE}}$). (c) SD IMP over **LFM**

Figure 5.5: Average metrics and SD improvements against LFM of users from T11 to T20.

5.5.3 Algorithm Settings and Baselines

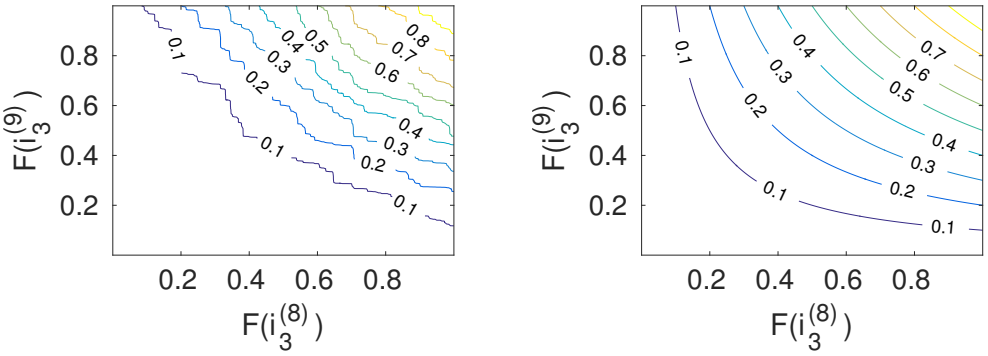
The types of copulas that are used to fit the data should be specified for the proposed algorithms. For CPM, we specify Gaussian copula in Line 8 of Algorithm 7. For VPM, we specify Clayton copula for modeling the dependence structures for pair copulas constructed by item factors (Line 7 of Algorithm 7). To demonstrate how we specify these copula types, we show the empirical copula result for a certain pair of item factors with a specific rating value. In Figure 5.7(a), we plot the empirical copula contour of the pair of 8-th and 9-th item factors for 3-valued ratings. There



(a) Ave($\text{MEAN}_{\text{RMSE}}$). (b) Ave($\text{SD}_{\text{RMSE}}$). (c) SD IMP over **LFM**

Figure 5.6: Average metrics and SD improvements against LFM of all target users.

are no well-known copula families that well fit the empirical copula contour. The closest fit that we can find so far is Clayton copula. Therefore, we approximate the empirical copula with Clayton copula. Figure 5.7(b) depicts the fitted Clayton copula contour. In Subsection 5.5.4, we show that with the best approximation we have found, we can obtain performance results better than that of existing methods with regard to stability. If in the future, a new parametric copula type can be found to fit the empirical copula better, we expect even better performance results.



(a) Empirical contour. (b) Fitted contour.

Figure 5.7: Empirical and fitted Clayton copula contour comparison of the pair of 8-th and 9-th item factors for rating $x = 3$.

To illustrate the effectiveness of the proposed algorithms, we compare them to three baseline methods that estimate a target user t's profile Λ_t in an intuitive way:

1. **Most Similar User (MSU):** We first factorize $\mathbf{R}$ to get $\Lambda_u, \forall u \in U$ and $\Gamma_i, \forall i \in I$. For each $u \in U$, we construct a vector $\mathbf{v}_u$, containing ground truth and predicted (if u has not rated i) ratings for $i \in \hat{I}_t$. Then we find a user $u^* = \underset{u \in U}{\text{argmin}} \, ||\hat{\mathcal{R}}_t - \mathbf{v}_u||$. Set $\Lambda_t = \Lambda_{u^*}$.

2. **Average User Profile (AveU):** We first factorize $\mathbf{R}$ to get $\Lambda_u, \forall u \in U$ and $\Gamma_i, \forall i \in I$. For each element $t^{(d)} \in \Lambda_t$ (for $1 \leq d \leq D$), set $t^{(d)} = \frac{1}{|U|} \sum_{u \in U} u^{(d)}$, for $1 \leq d \leq D$.

3. **LFM:** Treat the recommended ratings as t's ground truth ratings. Add ratings in $\hat{\mathcal{R}}_t$ to $\mathbf{R}$. Then factorize the rating matrix $\mathbf{R} \cap \hat{\mathcal{R}}_t$ to get Λ_t.

The dimension of user/item factors is set as $D = 10$. We use the LFM implementation provided by the authors of [32].

5.5.4 Performance Comparison

For each target user, we run all algorithms 10 times. We then take the average values of the two metrics, $\text{MEAN}_{\text{RMSE}}$ and SD_{RMSE}, of each group of target users and of all 20 target users, as specified in Subsection 5.5.1. The results are shown in Figures 5.4, 5.5, and 5.6. Each figure consists of three subfigures, including (a) the average of $\text{MEAN}_{\text{RMSE}}$ over the users, (b) the average of SD_{RMSE} over the users and (c) the SD improvement of each algorithm against LFM, which has the lowest average $\text{MEAN}_{\text{RMSE}}$ value. Figures 5.4 and 5.5 depict the results of the two target user groups. From the figures, we can see that the average RMSE values do not indicate significant difference for the 5 algorithms, with LFM only slightly better than others. The average SD values and the SD improvements, however, clearly show the advantages of our proposed methods, especially VPM. While CPM and VPM maintain close performance to LFM in accuracy (similar average RMSE values), they achieve much better stability (less variance), with average SD improvements of all target users at 69.05% and 93.54%, respectively, as shown in Figure 5.6. This implies that VPM is more robust and can generate much stabler profiles. Although AveU is also stable (because it takes the average of all users' factors), it is less accurate than VPM.

In addition, the superiority of our proposed vine-based modelling of dependence structure among item factors is verified due to the fact that VPM improves the performance over CPM (which applies independence assumption for item factors) with regard to both accuracy and stability. The promising performance implies that the inferred target users' ratings by our proposed methods, especially VPM, are consistently close to their ground truth ratings in different runs. This characteristic has an important practical meaning because the user preference inferred by Rec2PI is not

only accurate, but also stable. Retailers thus can take advantage of it to maintain reliable and valuable customer management.

5.6 Conclusions

In this chapter, we initiated the study of a novel value-added service, Rec2PI, to Wi-Fi data mining. The goal is to infer a user's preference profile given a set of recommendations from third-party RS, whose algorithms and the used dataset in the recommendation are unknown.

We formulated the inference task as a marginal maximum a posteriori probability (MAP) estimation problem. We provided a formal definition of Rec2PI in the context of rating-based item recommendation, including the modelling of user/item profile, user behavior and the interaction between users and items. We proposed a novel approach incorporating copulas into the modelling procedure to capture the dependence structure between any user feature and a set of item features. Vines for multivariate copulas capture the dependence structures among item features. We learned the inference model on an open dataset and evaluated the performance of Rec2PI using recommendations generated by a real-world RS. Evaluation results showed that our proposed algorithms, especially VPM, are accurate and stable.

Chapter 6

Related Work

In this chapter we review the state-of-the-art research of RSs, with regard to the topics of this thesis, i.e., trustworthiness in CQA websites, diversity requirement of recommendations, weighted b-matching and user profile inference.

6.1 Trustworthiness

Our work of identifying malicious content in CQA websites is mostly related to work on spam detection and recognizing experts or authoritative users and trustworthy content in the social media. These topics have become crucial to many online services, especially the question and answer communities, whose contents are generated by millions of users. We discuss related work on two aspects.

6.1.1 Retrieving High-quality Answers in CQA Sites

A lot of research has been done on finding high quality content in CQA sites. However, we haven't seen any paper which explicitly solved the credibility problem introduced in our work. Usually, researchers treated the best answers as the high-quality answers which has the risk of being defeated by the paid posters. In our work, we explicitly consider the credibility issues about the best answers.

Jeon *et al.* [70] attempted to predict the quality of answers in a community based question answering service with only non-textual features, such as *Answerer's Acceptance Ration*, *Answer Length* and *User's Recommendation*. They assumed the user feedback was a reliable source for the evaluation. Jurczyk and Agichtein [73] presented a study of link structure of Yahoo! Answers. They adopted a variant of the

HITS algorithm [81] for finding experts in the Q&A portal. Their research was also based on the assumption that the user feedback could be used to assign weights on the edges of their graph representing user relationships.

Liu *et al.* [92] applied their automated summary technique to summarize answers for questions which ask for opinions. They used cosine similarity to cluster topic-oriented answers and eliminated irrelevant ones. Bian *et al.* [20] tried to use both relevance between questions and answers and the quality of answers to retrieve good answers for a user query. Both textual features and statistical features such as user ratings are used in their approach. Later, in another work by Bian *et al.* [21], they explicitly considered the effect of several vote spam attacks. Such activities involve malicious voting for specific answers to improve their ranking and to decrease the ranking of competitors at the same time.

Agichtein *et al.* [9] studied the basic elements of social media and combined three features of the social media (Yahoo! Answers) to facilitate the task of identifying high quality content, namely intrinsic content quality, interactions between users and content usage statistics. Traditional link analysis algorithms are used to calculate the hubs and authorities scores (as in HITS algorithm [81]), and PageRank scores [109]. In addition, usage statistics such as the number of clicks of the Q&A session are used to complement the link-based analysis.

Pera and Ng [113] developed a CQA refinement system that could retrieve top-ranked answers for a user query based on similarity scores and the length of the answers.

Fichman [47] conducted a comparative study of answer quality on multiple Q&A websites, Yahoo! Answers, Wiki Answers①, Askville② and the Wikipedia Reference Desk③. Accuracy, completeness and verifiability were used as the quality measures for cross platform comparison. Fichman found that the quality of answers was significantly improved only in terms of answer completeness and verifiability, rather than the answer accuracy.

Sakai *et al.* [122] proposed system evaluation methods for the task of selecting or ranking answers for a given question. They noticed that the asker-selected best answers might be biased and even if they were not, there might be other good answers besides the best ones. In order to overcome the bias problems of BA-based evaluation,

①`http://wiki.answers.com/`, March 2015

②`http://askville.amazon.com/Index.do`, March 2015

③`http://en.wikipedia.org/wiki/Wikipedia:Reference_desk`, March 2015

they hired four assessors to independently assess every answer for the Q&A answers. Their experiments showed that their methods found substantial difference between systems that would have been overlooked by BA-based evaluation. In our point of view, we announce that the best answers not only are biased, but also could be unreliable or fake commercial campaign.

6.1.2 Other Research Work about Crowd-sourcing Spams in Different Realms

Previous research has also investigated the crowd-sourcing spam in other areas. Jindal and Liu [71], Ott *et al.* [108] and Arjun *et al.* [105] attempted to detect fake review or opinion spam in the online shopping stores, like Amazon's online store. Similar to research in CQA websites, they also used textual similarity features and user-oriented features, like ratings and history records. Huang *et al.* [66] developed a regression model with features suggesting quality-biased short text in Microblogging service, Twitter. They judged the quality of tweets based on relevance, informativeness, readability, and politeness of the short content and assigned different scores from 1 to 5. However, they didn't explicitly present how they define a spam-like tweet. Huang [65] conducted a similar study of commercial spam on blogging sites. They showed that the propaganda of some products in the comment of a blog post was crucial in detecting the malicious comments. The propaganda appeared in the form of URL, phone number, E-mail address, MSN numbers etc.

6.2 Diversity

6.2.1 Conflict-Aware Weighted b-Matching

Conflict-Aware Weighted b-Matching (CA-WBM) is a non-trivial extension of the classical weighted bipartite b-matching (WBM), a fundamental problem in computer science.

WBM finds applications in different scientific fields, including resource allocation [17, 100], scheduling [44], Internet advertising [38, 99, 34, 98, 18] and recommender systems [7, 93]. In addition, weighted b-matching in general graphs has been shown to be useful for a wide range of machine learning tasks, including classifi-

cation [63, 64], structured prediction models [130], spectral clustering [67], graph embedding [125], semi-supervised learning [68], and manifold learning [126].

Since WBM can be reduced to the transportation problem (or the maximum flow problem) in operations research [10, 124], it can be solved in polynomial time by a number of classical algorithms. Denoting the number of vertices as n and the number of edges as m, WBM can solved in $O(m\sqrt{n})$ time using Dinic's algorithm [36, 37]. Similarly, Hopcroft-Karp algorithm [62] finds a maximum matching in bipartite graphs in $O(n^{2.5})$ time.

Approximation algorithms for WBM also exist to improve the efficiency on large-scale real-world datasets, which often have millions of vertices and edges. In a centralized setting, Mestre [101] showed that a greedy algorithm can achieve 2-approximation in $O(m\log n)$ time. The author also provided two linear time approximation algorithms: a 2-approximation algorithm that runs in O(bm) time and a $(\frac{2}{3} - \epsilon)$-approximation algorithm that runs in expected $O(bm\log\frac{1}{\epsilon})$ time. In a distributed setting, Hoepman [61] showed that the greedy algorithm can be easily distributed and achieves 2-approximation in $O(m)$ time. Morales et al. [33] showed how to implement the similar greedy algorithm in MapReduce paradigm. None of the above algorithms includes conflict constraints. Recently, Manshadi et al. [95] modelled the generalized matching problem as linear program and proposed a distributed algorithm to cope with large-scale datasets. The algorithm allows for a small violation of lower- and upper-bound constraints in the optimization. Since their focus was on the scalability issue of linear program, they did not consider and analyze the impact of conflict constraints either.

From the point of view of practical applications, CA-WBM is related to the task of constraint-based recommendation ([46, 139, 140, 77, 78, 137, 112]). The first three works considered constraints on item features (attributes) and user preference relaxation [140], thus they did not study the same type of constraints as we do. The rest of works are more closely related to our problem but they did not consider conflict constraints either. Karimzadehgan et al. [79, 77, 78] studied the problem of optimizing the review assignments of scientific papers. They employed constraints on the quota of papers each reviewer is assigned. However, differently from our approach, in their optimization setup, matching of reviewers with a paper was done based on matching of multiple aspects of expertise. Xie, Lakshmanan, and Wood in [137] studied the problem of composite recommendations, where each recommendation comprises a set of items. They also considered constraints including the number of items that can

be recommended to a user. Their objective, however, was to minimize the cost of a recommended set of items when each item has a price to be paid. Parameswaran, Venetis, and Garcia-Molina in [112] studied the problem of course recommendations with course requirement constraints. Similarly as [137], the goal of [112] was to come up with set recommendations. However, the challenge they addressed was the modeling of complex academic requirements (e.g., take 2 out of a set of 5 math courses to meet the degree requirement). Such constraints are different from those that we consider in Chapter 3. Diversity is also a relevant topic in recommender systems. Adomavicius and Kwon [8] proposed a number of different ranking approaches for improving recommendation diversity. Their approach was mostly based on item popularity and they focused on controlling accuracy-diversity trade-off. They did not consider the conflict constraints between similar items.

CA-WBM is also related to graph-based recommendations, such as [56, 141, 57]. Guan et al. [56] studied resource recommendation based on tagging data. The authors proposed a graph-based representation learning algorithm to investigate the relationship between users, tags and documents. Zhao et al. [141] tackled the problem of personalized tag recommendation. They modelled the complex relationships in tagging data as a heterogeneous graph. A novel ranking algorithmic framework was proposed to deal with multi-type interrelated objects. Guan et al. [57] analyzed members' web surfing data and utilized a probabilistic graphical model to investigate fine-grained knowledge acquiring and sharing in collaborative environments.

Additionally, CA-WBM can be applied to facilitate design of negotiation-based trade mechanisms in the context of bilateral markets [43]. A bilateral market consists of sellers and buyers who wish to exchange goods. The market's main objective is to compute the optimal allocation that maximizes gain from trade. Naturally, bilateral automated negotiation among rational agents (market participants) with one-shot protocol (where one participant proposes a deal and the other one may only accept or refuse it) can be conceptualized as WBM [121, 118] and can be solved efficiently. Typical research work in this field includes exploring the agent's utility spaces [15, 96] and designing negotiation strategies that achieve Pareto-efficient agreements [69, 117]. By incorporating conflicts, CA-WBM can be used to capture multiple compatibility issues among requests or offers [116, 117], and could lead to novel negotiation strategy design.

The online version of CA-WBM is related to online WBM. Karp et al. [80] first introduced the online bipartite matching problem, in which $b = 1$ and edges weights

are all 1. They showed a greedy algorithm *GREEDY* with competitive ratio of $\frac{1}{2}$ and a randomized algorithm *RANKING* (under the assumption of adversarial order) with the optimal competitive ratio of $1 - \frac{1}{e} \approx 0.632$. Later, simpler proofs for the competitive ratio of *RANKING* were given in [53, 23, 35]. Kalyanasundaram and Pruhs [75] studied online unweighted b-matching and presented a deterministic algorithm *BALANCE* which achieves an optimal competitive ratio of $1-\frac{1}{(1+\frac{1}{b})^b}$. For online WBM, Ting and Xiang [131] proposed a randomized algorithm and a deterministic algorithm. Both algorithms were proven to be near optimal. Online WBM has also been intensely studied in the emerging domain of Internet advertising [98]. Typical practical scenarios include online adwords [99, 34] and display advertising [45, 18]. Mehta [98] summarized various results in different arrival models (adversarial order, random order and known IID) for online matching problems, including bipartite matching, vertex-weighted bipartite matching (online adwords) and edge-weighted and capacitated bipartite matching (display advertising). Typically, these problems do not consider conflict between entities. Therefore, online CA-WBM is valuable to provide more flexible service to online advertising.

6.2.2 GA-WBM

GA-WBM-D is a special case of CA-WBM, first proposed by Chen et al. [31] where the authors presented a generalized formulation of CA-WBM in the context of E-commerce, where diverse matching results are often desired (e.g., movies of different genres and merchants selling products of different categories). They showed that CA-WBM is NP-hard and proposed approximate and randomized algorithms to solve CA-WBM. Since CA-WBM generalizes the classic WBM problem, it has extended applicability in related scientific fields, including resource allocation [17, 100], scheduling [44], Internet advertising [38, 99, 34, 98, 18] and recommender systems [7, 93]. GA-WBM-D is a special case of CA-WBM useful for studying transitive conflicts.

GA-WBM-B generalizes the maximum budgeted allocation (MBA) problem and therefore is NP-hard [89, 52, 11, 123]. For MBA, the integrality gap is 3/4 [11]. Chakrabarty and Goel [27] studied the approximability of MBA and achieved 3/4-approximation ratio by a linear programming based
(Assignment-LP) iterative rounding algorithm. They also used hardness reductions to get better hardness results for other allocation problems such as submodular welfare maximization (SWM), generalized assignment problem (GAP) and maximum span-

ning star-forest (MSSF). Kalaitzis [74] obtained an improved $(3/4+c)$-approximation ratio, for some constant $c > 0$, for MBA by by rounding solutions to an LP called the Configuration-LP. They showed that the Configuration-LP is strictly stronger than the Assignment-LP for MBA. GA-WBM-B is also related to monotone submodular set function maximization subject to the k-system constraint. Fisher, Nemhauser and Wolsey [107] showed that the natural greedy algorithm has a tight approximation ratio of $1/(k+1)$. Călinescu et al. [25] proposed a randomized $(1-1/e)$- approximation for any monotone submodular function and an arbitrary matroid. They also provided performance analysis of the greedy algorithm under the k-system constraint. To improve the efficiency of the greedy approach, Minoux [103] proposed accelerated greedy algorithms. Those techniques were recently used in [90, 93].

6.3 Inference

Rec2PI solves a user profile inference problem arising in Wi-Fi data mining. Wi-Fi providers of brick-and-mortar stores can take advantage of Rec2PI to infer customers' preference based on Wi-Fi collected information so as to obtain a better understanding of in-store customers. Data mining on Wi-Fi collected information has not been fully investigated. Most relevant research works about mining the interaction between users and items focus on online recommender systems (RS) in a variety of domains, such as Ecommerce ([91]), online social networks ([26]) and Internet streaming media ([55]). As Rec2PI does not assume access to a user's ground truth behavior, it is fundamentally different from these existing works on RS. Rec2PI is also related to transfer learning. Pan et al. [110] categorized and reviewed the progress on transfer learning for classification, regression, and clustering problems. Pan and Yang [111] proposed to model both the numerical ratings and binary auxiliary preference in a principled way, and therefore to alleviate data sparsity (cold start) for collaborative filtering domains expressed in numerical ratings. Wongchokprasitti et al. [136] transferred user models built by one system to another to address the cold start problem. Indeed, transfer learning is part of Rec2PI: Rec2PI learns the pattern of user behavior from open datasets and apply it to the novel recommendation-based user profiling (the middle step in the red dashed box in Figure 5.1). However, Rec2PI goes beyond transfer learning because inferring a target user's profile from recommended content is different from the inference from his/her ground truth behavior, and requires novel

and dedicated approaches. To the best of our knowledge, the research problem posed in Rec2PI for Wi-Fi data mining is completely new.

The core of Rec2PI involves a copula-based probabilistic model, which is vital to profile inference. As a powerful statistical tool, copulas (including vines) are quite mature and have been broadly used in the domain of financial analysis for multivariate dependence modelling [6, 41]. The properties of copula theory are described in detail in [106]. With copula theory, we can capture the dependence structure between user factors and item factors, as well as those among item factors.

Chapter 7

Conclusions and Future Research

In this thesis, we have initiated studies of novel problems with regard to three important aspects of recommendation systems, trustworthiness, diversity and inference. The research problems we have tackled stem from emerging trends of the latest technologies and application scenario of recommendation systems. The work conducted in this thesis will facilitate further advancement of recommendation systems in many promising domains. In the following, we summarize the main contributions and discuss future research directions.

7.1 An Adaptive Detection System for Filtering Untrustworthy Content in CQA websites

In Chapter 2, we tackled the trustworthiness challenge in community question and answer (CQA) websites, where paid posters can post deceptive answers for commercial campaigns. To understand the hidden pattern of commercial campaigns, we collected Q&A sessions from a popular CQA website and campaign templates from a crowdsourcing website. By analyzing users' behaviors and similarities between templates and Q&A sessions, we proposed a set of features that can help distinguish deceptive answers from normal ones. We trained a machine learning-based detection model and conducted performance evaluation using different methods of data preprocessing. The prototype of an adaptive detection system we implemented can indicate the trustworthiness of an answer in real time.

Along the line of this direction, we also identify promising future directions. Since crowdsourcing has become an cost-effective approach for commercial campaigns, we

need to take advantage of the publicly available campaign templates (if available) to find corresponding campaigns. The crowdsourcing task may require campaigns conducted in different channels, such as CQA websites and social media. How do we efficiently determine campaigns in different channels stem from the same template and therefore actually refer to the same crowdsourcing task? In addition, more efficient and scalable feature extraction is needed to help the detection system accommodate multidimensional data sources.

7.2 Generalizations of WBM for Explicit Diversity Requirements

In Chapter 3, we proposed a novel graph-theoretic framework, conflict-aware WBM (CA-WBM), that naturally extends the classic WBM to explicitly model diversity requirements in the e-commerce matching scenario, where recommended items can be in conflict when being recommended to a user and the number of such conflicts should be below the user's tolerance threshold. CA-WBM is general in that one can arbitrarily assign conflicts between vertices. While the generalization offers great expressive power, it significantly increases the complexity of CA-WBM. We showed that CA-WBM is NP-hard by providing a polynomial-time reduction from the NP-hard problem Revenue Maximization in Interval Scheduling (RMIS). We developed SDP-based algorithms, LP/ILP-based algorithms and a greedy algorithm to solve CA-WBM. By showing that CA-WBM forms a k-extendible system, we proved a theoretical guarantee on the performance of the greedy algorithm. Comprehensive experiments using synthetic and real-world bipartite graph data showed that the greedy algorithm is scalable on large-scale datasets. For the online version of CA-WBM where vertices arrive one at a time, we proposed a randomized algorithm with theoretical performance guarantee.

Apart from recommendation systems of e-commerce, CA-WBM is general enough to model diversity requirements in a variety of practical applications such as online advertising, negotiation-based trade mechanisms, resource allocation and scheduling.

In Chapter 4, we introduced another novel generalization of WBM, *group-aware* WBM (GA-WBM), for diversity modelling. It is motivated by the fact that CA-WBM could be unnecessarily general in some applications, where the conflicts are not arbitrarily assigned but are transitive within groups. While CA-WBM has more

expressive power, it makes problem solving considerably harder. In fact, CA-WBM is even hard to approximate. We proved a stronger result for hardness of CA-WBM by a reduction from Maximum Weight Independent Set (MWIS).

We studied two variants of GA-WBM for modelling diversity in different approaches, GA-WBM-D and GA-WBM-B. The first generalizes the constraints of WBM by constraining the degree for each group; the other generalizes the optimization function of WBM by imposing a ceiling on the budget/payoff for each group. We proved that GA-WBM-D can be solved in polynomial time with linear programming. GA-WBM-B, on the other hand, is a generalization of the NP-hard Maximum Budgeted Allocation problem and cannot be solved efficiently. However, the transitivity assumption on conflicts helps improve the theoretical performance guarantee of approximate algorithms for GA-WBM-B. We showed that the greedy algorithm guarantees 3-approximate solutions. In addition, scaling the greedy algorithm for GA-WBM-Bto large-scale datasets is non-trivial: obtaining the top element with the maximum marginal revenue involves large amount of updating and sorting when the revenue keeps changing. By implementing a two-level heap structure to keep track of marginal revenues, we enabled the greedy algorithm to process a large bipartite graph with over eleven million edges in about one minute.

Along the line of this direction, the algorithmic aspect of both CA-WBM and GA-WBM deserves further investigation. Currently, the big data situation poses great challenges on algorithm development and analysis. Distributed algorithms and processing frameworks are often employed to alleviate computation difficulties. Adapting the algorithms proposed in thesis for CA-WBM and GA-WBM to distributed computational architectures, such as Hadoop Giraph and Spark GraphX, is promising and it is also interesting to study how performance guarantee changes along the adaption. For example, in a distributed environment, allowing a small violation to the constraints may reduce the complexity of implementation as well as the computational overheads. How does the performance change if we allow such violations for the algorithms proposed in this thesis?

7.3 A General Framework for Recommendations-based Profile Inference

In Chapter 5, we introduced and studied a novel value-added service, "From Recommendation to Profile Inference" (Rec2PI), for Wi-Fi data mining. We discussed a typical application scenario of brick-and-mortar retailers, where users' ground truth behavior data is often limited. Compared to general recommendation tasks, Rec2PI utilizes a new source of data and can be regarded as a reversed procedure: it infer users' preference profiles from recommended content. Recommendations become the input of Rec2PI and are assumed to be generated by third-party recommendation systems, whose algorithms and the used datasets are unknown. To infer a user's profile given recommended items, we investigated the statistical dependence between user features and item features, and formulated the inference task as a marginal maximum a posteriori probability (MAP) estimation problem. We proposed a novel approach, copula-based probabilistic model (CPM), incorporating copulas into the modelling procedure to capture the dependence structure between any user feature and a set of item features. In addition, we used vines for multivariate copulas that can capture the dependence structures among item features and proposed a vine-copula probabilistic model (VPM). We learned the inference models on an open dataset and evaluated the performance of Rec2PI using recommendations generated by a real-world RS. Evaluation results showed that our proposed algorithms, especially VPM, are accurate and stable.

Along the line of this direction, the methodologies of Rec2PI deserves further investigation. First, the input to Rec2PI can come from different categories in real-world, such as books, movies and music. Each category has a different set of features and therefore may requires dedicated modelling. How do we reasonably combine all the information to produce accurate and consistent user profiles? Second, better parametric copula models lead to further improvement on dependency structure characterization. Finding more accurate copula models is a promising future work direction.

Bibliography

[1] Internet users. http://www.internetlivestats.com/internet-users/. Accessed: 2016-06-30.

[2] Linknyc begins rolling out free gigabit wi-fi across new york city. http://www.zdnet.com/article/linknyc-begins-rolling-out-free-gigabit-wi-fi-across-new-york-city/. Accessed: 2016-01-03.

[3] Mobile marketing statistics 2015. http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/. Accessed: 2016-01-03.

[4] Retail analytics: Who owns the data? http://blog.mojonetworks.com/retail-analytics-who-owns-the-data/. Accessed: 2016-01-13.

[5] Why stores are finally turning on to wifi. http://fortune.com/2012/12/14/why-stores-are-finally-turning-on-to-wifi/. Accessed: 2016-01-03.

[6] Kjersti Aas, Claudia Czado, Arnoldo Frigessi, and Henrik Bakken. Pair-copula constructions of multiple dependence. *Insurance: Mathematics and economics*, 44(2):182–198, 2009.

[7] Gediminas Adomavicius and YoungOk Kwon. Maximizing aggregate recommendation diversity: A graph-theoretic approach. In *Proc. 1st International Workshop on Novelty and Diversity in Recommender Systems (DiveRS 2011)*, pages 3–10, 2011.

[8] Gediminas Adomavicius and YoungOk Kwon. Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Trans. Knowl. Data Eng.*, 24(5):896–911, 2012.

[9] Eugene Agichtein, Carlos Castillo, Debora Donato, Aristides Gionis, and Gilad Mishne. Finding high-quality content in social media. In *Proc. 2008 International Conference on Web Search and Web Data Mining*, pages 183–194, February 2008.

[10] Omid Amini, David Peleg, Stéphane Pérennes, Ignasi Sau, and Saket Saurabh. On the approximability of some degree-constrained subgraph problems. *Discrete Applied Mathematics*, 160(12):1661–1679, 2012.

[11] Nir Andelman and Yishay Mansour. Auctions with budget constraints. In *Proc. Scandinavian Workshop on Algorithm Theory*, pages 26–38, 2004.

[12] E. M. Arkin and E. B. Silverberg. Scheduling jobs with fixed start and end times. *Discrete Appl. Math.*, 18(1):1–8, November 1987.

[13] Armen S. Asratian, Tristan M. J. Denley, and Roland Häggkvist. *Bipartite Graphs and Their Applications*. Cambridge University Press, New York, NY, USA, 1998.

[14] Kazim Azam and Michael K Pitt. *Bayesian inference for a semi-parametric copula-based Markov chain*. University of Warwick. Dept of Economics, 2014.

[15] Yair Bartal, Rica Gonen, and Pierfrancesco La Mura. Negotiation-range mechanisms: Exploring the limits of truthful efficient markets. In *Proc. 5th ACM Conference on Electronic Commerce*, pages 1–8, 2004.

[16] C. Berge. *Graphs and Hypergraphs*. Elsevier Science Ltd., Oxford, UK, UK, 1985.

[17] Dimitri P Bertsekas. A new algorithm for the assignment problem. *Mathematical Programming*, 21(1):152–171, 1981.

[18] Anand Bhalgat, Nitish Korula, Hennadiy Leontyev, Max Lin, and Vahab S. Mirrokni. Partner tiering in display advertising. In *Proc. 7th ACM International Conference on Web Search and Data Mining*, pages 133–142, February 2014.

[19] Randeep Bhatia, Julia Chuzhoy, Ari Freund, and Joseph (Seffi) Naor. Algorithmic aspects of bandwidth trading. *ACM Trans. Algorithms*, 3(1):10:1–10:19, February 2007.

[20] Jiang Bian, Yandong Liu, Eugene Agichtein, and Hongyuan Zha. Finding the right facts in the crowd: factoid question answering over social media. In *Proc. 17th International Conference on World Wide Web*, pages 467–476, April 2008.

[21] Jiang Bian, Yandong Liu, Eugene Agichtein, and Hongyuan Zha. A few bad votes too many?: towards robust ranking in social media. In *Proc. 4th International Workshop on Adversarial Information Retrieval on the Web*, AIRWeb '08, pages 53–60, April 2008.

[22] Jiang Bian, Yandong Liu, Ding Zhou, Eugene Agichtein, and Hongyuan Zha. Learning to recognize reliable users and content in social media with coupled mutual reinforcement. In *Proc. 18th International Conference on World Wide Web*, pages 51–60, April 2009.

[23] Benjamin Birnbaum and Claire Mathieu. On-line bipartite matching made simple. *ACM SIGACT News*, 39(1):80–87, 2008.

[24] Eike Christian Brechmann and Ulf Schepsmeier. Modeling dependence with c- and d-vine copulas: The r-package cdvine. *J. of Statistical Software*, 52(3):1–27, 2013.

[25] Gruia Călinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM J. Comput.*, 40(6):1740–1766, 2011.

[26] Giuliana Carullo, Aniello Castiglione, Alfredo De Santis, and Francesco Palmieri. A triadic closure and homophily-based recommendation system for online social networks. *J. of World Wide Web*, 18(6):1579–1601, 2015.

[27] Deeparnab Chakrabarty and Gagan Goel. On the approximability of budgeted allocations and improved lower bounds for submodular welfare maximization and gap. *SIAM J. Comp.*, 39(6):2189–2211, 2010.

[28] Nguyen Ngoc Chan, Walid Gaaloul, and Samir Tata. A web service recommender system using vector space model and latent semantic indexing. In *Proc. 2011 IEEE International Conference on Advanced Information Networking and Applications*, pages 602–609, 2011.

[29] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[30] Cheng Chen, Kui Wu, Venkatesh Srinivasan, and Xudong Zhang. Battling the Internet water army: Detection of hidden paid posters. In *Proc. 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 116–120, August 2013.

[31] Cheng Chen, Lan Zheng, Venkatesh Srinivasan, Alex Thomo, Kui Wu, and Anthony Sukow. Conflict-aware weighted bipartite b-matching and its application to e-commerce. *IEEE Trans. on Knowl. and Data Eng.*, 28(6):1475–1488, June 2016.

[32] Cheng Chen, Lan Zheng, Alex Thomo, Kui Wu, and Venkatesh Srinivasan. Comparing the staples in latent factor models for recommender systems. In *Proc. 29th Annual ACM Symposium on Applied Computing*, pages 91–96, 2014.

[33] Gianmarco De Francisci Morales, Aristides Gionis, and Mauro Sozio. Social content matching in mapreduce. *Proc. VLDB Endow.*, 4(7):460–469, April 2011.

[34] Nikhil R. Devanur and Thomas P. Hayes. The adwords problem: online keyword matching with budgeted bidders under random permutations. In *Proc. 10th ACM Conference on Electronic Commerce (EC-2009)*, pages 71–78, July 2009.

[35] Nikhil R Devanur, Kamal Jain, and Robert D Kleinberg. Randomized primal-dual analysis of ranking for online bipartite matching. In *Proc. 24th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 101–107. SIAM, 2013.

[36] EA Dinic. An algorithm for the solution of the max-flow problem with the polynomial estimation. *Doklady Akademii Nauk SSSR*, 194(4):1277–1280, 1970.

[37] Yefim Dinitz. Dinitzalgorithm: The original version and evens version. In *Theoretical Computer Science*, pages 218–240. Springer, 2006.

[38] Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *The American economic review*, 97(1):242–259, 2007.

[39] Jack Edmonds. Paths, trees, and flowers. *Canadian Journal of mathematics*, 17(3):449–467, 1965.

[40] Paul Embrechts, Filip Lindskog, and Alexander McNeil. Modelling dependence with copulas. *Rapport technique, Département de mathématiques, Institut Fédéral de Technologie de Zurich, Zurich*, 2001.

[41] Paul Embrechts, Filip Lindskog, and Alexander McNeil. Modelling dependence with copulas and applications to risk management. *Handbook of heavy tailed distributions in finance*, 8(1):329–384, 2003.

[42] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[43] Shaheen Fatima, Sarit Kraus, and Michael Wooldridge. *Principles of Automated Negotiation*. Cambridge University Press, 2014.

[44] Morteza Fayyazi, David Kaeli, and Waleed Meleis. Parallel maximum weight bipartite matching algorithms for scheduling in input-queued switches. In *Proc. 18th International Symposium on Parallel and Distributed Processing*, page 4. IEEE, 2004.

[45] Jon Feldman, Monika Henzinger, Nitish Korula, Vahab S. Mirrokni, and Cliff Stein. Online stochastic packing applied to display ad allocation. In *Proceedings of the 18th Annual European Conference on Algorithms: Part I*, ESA'10, pages 182–194, 2010.

[46] Alexander Felfernig and Robin D. Burke. Constraint-based recommender systems: technologies and research issues. In *Proc. 10th International Conference on Electronic Commerce*, pages 3:1–3:10, 2008.

[47] Pnina Fichman. A comparative assessment of answer quality on four question answering sites. *Journal of Information Science*, 37(5):476–486, 2011.

[48] Daniel Fleder and Kartik Hosanagar. Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity. *Manage. Sci.*, 55(5):697–712, 2009.

[49] George Forman. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3:1289–1305, March 2003.

[50] Satoru Fujishige. *Submodular functions and optimization.* Annals of discrete mathematics. Elsevier, Amsterdam, Boston, Paris, 2005.

[51] Harold N. Gabow. An efficient reduction technique for degree-constrained subgraph and bidirected network flow problems. In *Proc. 15th Annual ACM Symposium on Theory of Computing*, STOC '83, pages 448–456, New York, NY, USA, 1983. ACM.

[52] Rahul Garg, Vijay Kumar, and Vinayaka Pandit. Approximation algorithms for budget-constrained auctions. In *Proc. 4th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems and 5th International Workshop on Randomization and Approximation Techniques in Computer Science: Approximation, Randomization and Combinatorial Optimization*, APPROX '01/RANDOM '01, pages 102–113, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[53] Gagan Goel and Aranyak Mehta. Online budgeted matching in random input models with applications to adwords. In *Proc. 19th annual ACM-SIAM symposium on Discrete algorithms*, pages 982–991. Society for Industrial and Applied Mathematics, 2008.

[54] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, November 1995.

[55] Carlos A. Gomez-Uribe and Neil Hunt. The netflix recommender system: Algorithms, business value, and innovation. *ACM Trans. Manage. Inf. Syst.*, 6(4):13:1–13:19, December 2015.

[56] Ziyu Guan, Can Wang, Jiajun Bu, Chun Chen, Kun Yang, Deng Cai, and Xiaofei He. Document recommendation in social tagging services. In *Proc. 19th International Conference on World Wide Web*, pages 391–400, April 2010.

[57] Ziyu Guan, Shengqi Yang, Huan Sun, Mudhakar Srivatsa, and Xifeng Yan. Fine-grained knowledge sharing in collaborative environments. *IEEE Trans. Knowl. Data Eng.*, 27(8):2163–2174, 2015.

[58] Inc. Gurobi Optimization. Gurobi optimizer reference manual, 2015.

[59] Johan Håstad. Clique is hard to approximate withinn 1- ε. *Acta Mathematica*, 182(1):105–142, 1999.

[60] Will Hill, Larry Stead, Mark Rosenstein, and George Furnas. Recommending and evaluating choices in a virtual community of use. In *Proc. SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, pages 194–201, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[61] Jaap-Henk Hoepman. Simple distributed weighted matchings. *arXiv preprint cs/0410047*, 2004.

[62] John E Hopcroft and Richard M Karp. An n^5/2 algorithm for maximum matchings in bipartite graphs. *SIAM Journal on computing*, 2(4):225–231, 1973.

[63] Bert C Huang and Tony Jebara. Loopy belief propagation for bipartite maximum weight b-matching. In *Proc. 11th International Conference on Artificial Intelligence and Statistics*, pages 195–202, March 2007.

[64] Bert C Huang and Tony Jebara. Fast b-matching via sufficient selection belief propagation. In *Proc. 14th International Conference on Artificial Intelligence and Statistics*, pages 361–369, April 2011.

[65] Congrui Huang, Qiancheng Jiang, and Yan Zhang. Detecting comment spam through content analysis. In *Proc. 2010 International Conference on Web-age Information Management*, pages 222–233, July 2010.

[66] Minlie Huang, Yi Yang, and Xiaoyan Zhu. Quality-biased ranking of short texts in microblogging services. In *Proc. 5th International Joint Conference on Natural Language Processing*, pages 373–382, Chiang Mai, Thailand, November 2011. Asian Federation of Natural Language Processing.

[67] Tony Jebara and Vlad Shchogolev. B-matching for spectral clustering. In *Proc. 17th European Conference on Machine Learning*, pages 679–686. Springer, 2006.

[68] Tony Jebara, Jun Wang, and Shih-Fu Chang. Graph construction and b-matching for semi-supervised learning. In *Proc. 26th Annual International Conference on Machine Learning*, pages 441–448. ACM, 2009.

[69] Nicholas R Jennings, Peyman Faratin, Alessio R Lomuscio, Simon Parsons, Michael J Wooldridge, and Carles Sierra. Automated negotiation: prospects, methods and challenges. *Group Decision and Negotiation*, 10(2):199–215, 2001.

[70] Jiwoon Jeon, W. Bruce Croft, Joon Ho Lee, and Soyeon Park. A framework to predict the quality of answers with non-textual features. In *Proc. 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 228–235, August 2006.

[71] Nitin Jindal and Bing Liu. Opinion spam and analysis. In *Proc. 2008 International Conference on Web Search and Web Data Mining*, pages 219–230, February 2008.

[72] Harry Joe. Families of m-variate distributions with given margins and m (m-1)/2 bivariate dependence parameters. *Lecture Notes-Monograph Series*, pages 120–141, 1996.

[73] Pawel Jurczyk and Eugene Agichtein. Discovering authorities in question answer communities by using link analysis. In *Proc. 16th ACM Conference on Information and Knowledge Management*, pages 919–922, November 2007.

[74] Christos Kalaitzis. An improved approximation guarantee for the maximum budgeted allocation problem. In *Proc. 27th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1048–1066, 2016.

[75] Bala Kalyanasundaram and Kirk R Pruhs. An optimal deterministic algorithm for online b-matching. *Theoretical Computer Science*, 233(1):319–325, 2000.

[76] J.N. Kapur and H.K. Kesavan. *Entropy Optimization Principles with Applications*. Academic Press Inc., 1992.

[77] Maryam Karimzadehgan and ChengXiang Zhai. Constrained multi-aspect expertise matching for committee review assignment. In *Proc. 18th ACM Conference on Information and Knowledge Management*, pages 1697–1700, 2009.

[78] Maryam Karimzadehgan and ChengXiang Zhai. Integer linear programming for constrained multi-aspect committee review assignment. *Information processing & management*, 48(4):725–740, 2012.

[79] Maryam Karimzadehgan, ChengXiang Zhai, and Geneva Belford. Multi-aspect expertise matching for review assignment. In *Proc. 17th ACM Conference on Information and Knowledge Management*, pages 1113–1122. ACM, 2008.

[80] Richard M. Karp, Umesh V. Vazirani, and Vijay V. Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990*, pages 352–358, 1990.

[81] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[82] Michal Kocvara and Michael Stingl. On the solution of large-scale sdp problems by the modified barrier method using iterative solvers. *Math. Program.*, 109(2-3):413–444, 2007.

[83] Antoon WJ Kolen, Jan Karel Lenstra, Christos H Papadimitriou, and Frits CR Spieksma. Interval scheduling: A survey. *Naval Research Logistics (NRL)*, 54(5):530–543, 2007.

[84] Yehuda Koren and Robert Bell. Advances in collaborative filtering. In *Recommender systems handbook*, pages 145–186. Springer, 2011.

[85] Andreas Krause and Carlos Guestrin. Beyond convexity: Submodularity in machine learning. *ICML Tutorials*, 2008.

[86] Harold W Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

[87] Kleanthi Lakiotaki, Nikolaos F. Matsatsinis, and Alexis Tsoukias. Multicriteria user modeling in recommender systems. *IEEE Intelligent Systems*, 26(2):64–76, March 2011.

[88] Neal Lathia, Stephen Hailes, Licia Capra, and Xavier Amatriain. Temporal diversity in recommender systems. In *Proc. 33rd International ACM SIGIR*

Conference on Research and Development in Information Retrieval, pages 210–217, 2010.

[89] Benny Lehmann, Daniel Lehmann, and Noam Nisan. Combinatorial auctions with decreasing marginal utilities. In *Proc. 3rd ACM Conference on Electronic Commerce*, pages 18–28, 2001.

[90] Jure Leskovec, Lada A Adamic, and Bernardo A Huberman. The dynamics of viral marketing. *TWEB*, 1(1):5, 2007.

[91] Greg Linden, Brent Smith, and Jeremy York. Amazon. com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1):76–80, 2003.

[92] Yuanjie Liu, Shasha Li, Yunbo Cao, Chin-Yew Lin, Dingyi Han, and Yong Yu. Understanding and summarizing answers in community-based question answering services. In *Proc. 22nd International Conference on Computational Linguistics - Volume 1*, pages 497–504, Stroudsburg, PA, USA, August 2008. Association for Computational Linguistics.

[93] Wei Lu, Shanshan Chen, Keqian Li, and Laks V. S. Lakshmanan. Show me the money: Dynamic recommendations for revenue maximization. *Proc. VLDB Endow.*, 7(14):1785–1796, October 2014.

[94] Hao Ma, Haixuan Yang, Michael R. Lyu, and Irwin King. Sorec: Social recommendation using probabilistic matrix factorization. In *Proc. 17th ACM Conference on Information and Knowledge Management*, pages 931–940, 2008.

[95] Faraz Makari Manshadi, Baruch Awerbuch, Rainer Gemula, Rohit Khandekar, Julián Mestre, and Mauro Sozio. A distributed algorithm for large-scale generalized matching. *Proc. VLDB Endow.*, 6(9):613–624, July 2013.

[96] Ivan Marsa-Maestre, Miguel A. Lopez-Carmona, Juan R. Velasco, Takayuki Ito, Mark Klein, and Katsuhide Fujita. Balancing utility and deal probability for auction-based negotiations in highly nonlinear utility spaces. In *Proc. 21st International Joint Conference on Artifical Intelligence*, pages 214–219, 2009.

[97] Daniel Mcfadden. Conditional Logit Analysis of Qualitative Choice Behavior. In P. Zarembka, editor, *Frontiers in econometrics*, pages 105–142. Academic Press, New York, 1974.

[98] Aranyak Mehta. Online matching and ad allocation. *Foundations and Trends in Theoretical Computer Science*, 8(4):265–368, 2013.

[99] Aranyak Mehta, Amin Saberi, Umesh V. Vazirani, and Vijay V. Vazirani. Adwords and generalized online matching. *J. ACM*, 54(5), 2007.

[100] Reshef Meir, Yiling Chen, and Michal Feldman. Efficient parking allocation as online bipartite matching with posted prices. In *Proc. 2013 International Conference on Autonomous Agents and Multi-agent Systems*, pages 303–310. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

[101] Julián Mestre. Greedy in approximation algorithms. In *European Symposium on Algorithms*, pages 528–539. Springer, 2006.

[102] Stuart E Middleton, Nigel R Shadbolt, and David C De Roure. Ontological user profiling in recommender systems. *ACM Trans. on Info. Sys.*, 22(1):54–88, 2004.

[103] Michel Minoux. Accelerated greedy algorithms for maximizing submodular set functions. In *Proc. Optimization Techniques*, pages 234–243, 1978.

[104] Andriy Mnih and Ruslan R Salakhutdinov. Probabilistic matrix factorization. In *Proc. Advances in Neural Information Processing Systems 20 (NIPS 07)*, pages 1257–1264. Curran Associates, Inc., 2008.

[105] Arjun Mukherjee, Bing Liu, and Natalie S. Glance. Spotting fake reviewer groups in consumer reviews. In *Proc. 21st International Conference on World Wide Web*, pages 191–200, April 2012.

[106] Roger B. Nelsen. *An Introduction to Copulas (Springer Series in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[107] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—I. *Math. Program.*, 14(1):265–294, 1978.

[108] Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T. Hancock. Finding deceptive opinion spam by any stretch of the imagination. In *Proc. 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, pages 309–319, June 2011.

[109] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[110] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Trans. on Knowl. and Data Eng.*, 22(10):1345–1359, October 2010.

[111] Weike Pan and Qiang Yang. Transfer learning in heterogeneous collaborative filtering domains. *Artificial intelligence*, 197:39–55, 2013.

[112] Aditya G. Parameswaran, Petros Venetis, and Hector Garcia-Molina. Recommendation systems with complex constraints: A course recommendation perspective. *ACM Trans. Inf. Syst.*, 29(4):20, 2011.

[113] Maria Soledad Pera and Yiu-Kai Ng. A community question-answering refinement system. In *Proc. 22nd ACM Conference on Hypertext and Hypermedia*, pages 251–260, June 2011.

[114] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical recipes in C: The art of scientific computing (second edition)*. Cambridge University Press, 1992.

[115] Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. Learning diverse rankings with multi-armed bandits. In *Proc. 25th International Conference on Machine Learning*, pages 784–791. ACM, 2008.

[116] Azzurra Ragone, Tommaso Di Noia, Eugenio Di Sciascio, and Francesco M Donini. Dl-based alternating-offers protocol for automated multi-issue bilateral negotiation. In *Proc. 20th International Workshop on Description Logics*, DL07, pages 443–450, 2007.

[117] Azzurra Ragone, Tommaso Di Noia, Eugenio Di Sciascio, and Francesco M Donini. Logic-based automated multi-issue bilateral negotiation in peer-to-peer e-marketplaces. *Autonomous Agents and Multi-Agent Systems*, 16(3):249–270, 2008.

[118] Howard Raiffa. *The art and science of negotiation*. Harvard University Press, 1982.

[119] Kenneth R. Rebman. Total unimodularity and the transportation problem: a generalization. *Linear Algebra and its Applications*, 8(1):11 – 24, 1974.

[120] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proc. 1994 ACM Conference on Computer Supported Cooperative Work*, CSCW '94, pages 175–186. ACM, 1994.

[121] Jeffrey S. Rosenschein and Gilad Zlotkin. *Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers*. MIT Press, Cambridge, MA, USA, 1994.

[122] Tetsuya Sakai, Daisuke Ishikawa, Noriko Kando, Yohei Seki, Kazuko Kuriyama, and Chin-Yew Lin. Using graded-relevance metrics for evaluating community QA answer selection. In *Proc. 4th International Conference on Web Search and Web Data Mining*, pages 187–196, February 2011.

[123] Tuomas Sandholm and Subhash Suri. Side constraints and non-price attributes in markets. *Games and Economic Behavior*, 55(2):321–330, 2006.

[124] Alexander Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1986.

[125] Blake Shaw and Tony Jebara. Minimum volume embedding. In *Proc. 11th International Conference on Artificial Intelligence and Statistics*, pages 460–467, March 2007.

[126] Blake Shaw and Tony Jebara. Structure preserving embedding. In *Proc. 26th Annual International Conference on Machine Learning*, pages 937–944. ACM, 2009.

[127] Yue Shi, Martha Larson, and Alan Hanjalic. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Comput. Surv.*, 47(1):3:1–3:45, May 2014.

[128] Daniel Dominic Sleator and Robert Endre Tarjan. Amortized efficiency of list update and paging rules. *Commun. ACM*, 28(2):202–208, 1985.

[129] Xiaoyuan Su and Taghi M. Khoshgoftaar. A survey of collaborative filtering techniques. *Adv. in Artif. Intell.*, 2009:4:2–4:2, January 2009.

[130] Ben Taskar, Vassil Chatalbashev, Daphne Koller, and Carlos Guestrin. Learning structured prediction models: A large margin approach. In *Proc. 22nd international conference on Machine learning*, pages 896–903. ACM, 2005.

[131] Hing-Fung Ting and Xiangzhong Xiang. Near optimal algorithms for online maximum weighted b-matching. In *Proc. 8th International Workshop on Frontiers in Algorithmics (FAW)*, pages 240–251, June 2014.

[132] K. C. Toh, M.J. Todd, and R. H. Tütüncü. Sdpt3 – a matlab software package for semidefinite programming. *Optimization Methods and Software*, 11:545–581, 1999.

[133] Reha H. Tütüncü, K. C. Toh, and Michael J. Todd. Solving semidefinite-quadratic-linear programs using sdpt3. *Math. Program.*, 95(2):189–217, 2003.

[134] Erik Vee, Utkarsh Srivastava, Jayavel Shanmugasundaram, Prashant Bhat, and Sihem Amer Yahia. Efficient computation of diverse query results. In *Proc. IEEE 24th International Conference on Data Engineering*, pages 228–236, 2008.

[135] Gang Wang, Christo Wilson, Xiaohan Zhao, Yibo Zhu, Manish Mohanlal, Haitao Zheng, and Ben Y. Zhao. Serf and turf: crowdturfing for fun and profit. In *Proc. 21st International Conference on World Wide Web*, pages 679–688, April 2012.

[136] Chirayu Wongchokprasitti, Jaakko Peltonen, Tuukka Ruotsalo, Payel Bandyopadhyay, Giulio Jacucci, and Peter Brusilovsky. User model in a box: Cross-system user model transfer for resolving cold start problems. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 289–301. Springer, 2015.

[137] Min Xie, Laks V. S. Lakshmanan, and Peter T. Wood. Breaking out of the box of recommendations: from items to packages. In *Proc. 4th ACM Conference on Recommender Systems*, RecSys '10, pages 151–158, 2010.

[138] Mihalis Yannakakis. On a class of totally unimodular matrices. In *Proc. 21st Annual Symposium on Foundations of Computer Science*, pages 10–16, Oct 1980.

[139] Markus Zanker and Markus Jessenitschnig. Case-studies on exploiting explicit customer requirements in recommender systems. *User Model. User-Adapt. Interact.*, 19(1-2):133–166, 2009.

[140] Markus Zanker, Markus Jessenitschnig, and Wolfgang Schmid. Preference reasoning with soft constraints in constraint-based recommender systems. *Constraints*, 15(4):574–595, 2010.

[141] Wei Zhao, Ziyu Guan, and Zheng Liu. Ranking on heterogeneous manifolds for tag recommendation in social tagging services. *Neurocomputing*, 148:521–534, 2015.

[142] Zhaohui Zheng, Xiaoyun Wu, and Rohini Srihari. Feature selection for text categorization on imbalanced data. *ACM SIGKDD Explorations Newsletter*, 6(1):80–89, June 2004.

[143] Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. Improving recommendation lists through topic diversification. In *Proc. 14th International Conference on World Wide Web*, pages 22–32. ACM, 2005.