COLLABORATIVE FILTERING APPROACHES FOR SINGLE-DOMAIN AND CROSS-DOMAIN RECOMMENDER SYSTEMS

by

ROHIT PARIMI

MS, Kansas State University, 2010

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2015

Abstract

Increasing amounts of content on the Web means that users can select from a wide variety of items (i.e., items that concur with their tastes and requirements). The generation of personalized item suggestions to users has become a crucial functionality for many web applications as users benefit from being shown only items of potential interest to them. One popular solution to creating personalized item suggestions to users is recommender systems. Recommender systems can address the item recommendation task by utilizing past user preferences for items captured as either *explicit* or *implicit* user feedback.

Numerous collaborative filtering (CF) approaches have been proposed in the literature to address the recommendation problem in the single-domain setting (user preferences from only one domain are used to recommend items). However, increasingly large datasets often prevent experimentation of every approach in order to choose the one that best fits an application domain. The work in this dissertation on the single-domain setting studies two CF algorithms, Adsorption and Matrix Factorization (MF), considered to be state-of-the-art approaches for implicit feedback and suggests that characteristics of a domain (e.g., close connections versus loose connections among users) or characteristics of data available (e.g., density of the feedback matrix) can be useful in selecting the most suitable CF approach to use for a particular recommendation problem. Furthermore, for Adsorption, a neighborhood-based approach, this work studies several ways to construct user neighborhoods based on similarity functions and on community detection approaches, and suggests that domain and data characteristics can also be useful in selecting the neighborhood approach to use for Adsorption. Finally, motivated by the need to decrease computational costs of recommendation algorithms, this work studies the effectiveness of using short-user

histories and suggests that short-user histories can successfully replace long-user histories for recommendation tasks.

Although most approaches for recommender systems use user preferences from only one domain, in many applications, user interests span items of various types (e.g., artists and tags). Each recommendation problem (e.g., recommending artists to users or recommending tags to users) can be considered unique domains, and user preferences from several domains can be used to improve accuracy in one domain, an area of research known as *cross-domain recommender systems*. The work in this dissertation on cross-domain recommender systems investigates several limitations of existing approaches and proposes three novel approaches (two Adsorption-based and one MF-based) to improve recommendation accuracy in one domain by leveraging knowledge from multiple domains with implicit feedback.

The first approach performs aggregation of neighborhoods (WAN) from the source and target domains, and the neighborhoods are used with Adsorption to recommend target items. The second approach performs aggregation of target recommendations (WAR) from Adsorption computed using neighborhoods from the source and target domains. The third approach integrates latent user factors from source domains into the target through a regularized latent factor model (CIMF). Experimental results on six target recommendation tasks from two real-world applications suggest that the proposed approaches effectively improve target recommendation accuracy as compared to single-domain CF approaches and successfully utilize varying amounts of user overlap between source and target domains. Furthermore, under the assumption that tuning may not be possible for large recommendation problems, this work proposes an approach to calculate knowledge aggregation weights based on network alignment for WAN and WAR approaches, and results show the usefulness of the proposed solution. The results also suggest that the WAN and WAR approaches effectively address the cold-start user problem in the target domain.

COLLABORATIVE FILTERING APPROACHES FOR SINGLE-DOMAIN AND CROSS-DOMAIN RECOMMENDER SYSTEMS

by

Rohit Parimi

MS, Kansas State University, 2010

A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2015

Approved by:

Major Professor
Doina Caragea

Copyright

Abstract

Increasing amounts of content on the Web means that users can select from a wide variety of items (i.e., items that concur with their tastes and requirements). The generation of personalized item suggestions to users has become a crucial functionality for many web applications as users benefit from being shown only items of potential interest to them. One popular solution to creating personalized item suggestions to users is recommender systems. Recommender systems can address the item recommendation task by utilizing past user preferences for items captured as either *explicit* or *implicit* user feedback.

Numerous collaborative filtering (CF) approaches have been proposed in the literature to address the recommendation problem in the single-domain setting (user preferences from only one domain are used to recommend items). However, increasingly large datasets often prevent experimentation of every approach in order to choose the one that best fits an application domain. The work in this dissertation on the single-domain setting studies two CF algorithms, Adsorption and Matrix Factorization (MF), considered to be state-of-the-art approaches for implicit feedback and suggests that characteristics of a domain (e.g., close connections versus loose connections among users) or characteristics of data available (e.g., density of the feedback matrix) can be useful in selecting the most suitable CF approach to use for a particular recommendation problem. Furthermore, for Adsorption, a neighborhood-based approach, this work studies several ways to construct user neighborhoods based on similarity functions and on community detection approaches, and suggests that domain and data characteristics can also be useful in selecting the neighborhood approach to use for Adsorption. Finally, motivated by the need to decrease computational costs of recommendation algorithms, this work studies the effectiveness of using short-user

histories and suggests that short-user histories can successfully replace long-user histories for recommendation tasks.

Although most approaches for recommender systems use user preferences from only one domain, in many applications, user interests span items of various types (e.g., artists and tags). Each recommendation problem (e.g., recommending artists to users or recommending tags to users) can be considered unique domains, and user preferences from several domains can be used to improve accuracy in one domain, an area of research known as *cross-domain recommender systems*. The work in this dissertation on cross-domain recommender systems investigates several limitations of existing approaches and proposes three novel approaches (two Adsorption-based and one MF-based) to improve recommendation accuracy in one domain by leveraging knowledge from multiple domains with implicit feedback.

The first approach performs aggregation of neighborhoods (WAN) from the source and target domains, and the neighborhoods are used with Adsorption to recommend target items. The second approach performs aggregation of target recommendations (WAR) from Adsorption computed using neighborhoods from the source and target domains. The third approach integrates latent user factors from source domains into the target through a regularized latent factor model (CIMF). Experimental results on six target recommendation tasks from two real-world applications suggest that the proposed approaches effectively improve target recommendation accuracy as compared to single-domain CF approaches and successfully utilize varying amounts of user overlap between source and target domains. Furthermore, under the assumption that tuning may not be possible for large recommendation problems, this work proposes an approach to calculate knowledge aggregation weights based on network alignment for WAN and WAR approaches, and results show the usefulness of the proposed solution. The results also suggest that the WAN and WAR approaches effectively address the cold-start user problem in the target domain.

Table of Contents

List of Figures

List of Tables

Acknowledgments

Last nine months of my PhD work has been a testing period. I thank God for comforting me during the difficult times, and for giving me help, wisdom, and guidance to complete this work. The verse "Whom have I in heaven but thee? and there is none upon earth that I desire beside thee." (Psalm 73:25, King James Bible) has touched me, helped me get past some painful moments, and made me realign my priorities in life. Forever I am grateful for God's love and compassion.

I offer my sincerest gratitude to my advisor, Dr. Doina Caragea, for her guidance, countless hours of reflecting, reading, and most of all patience throughout the entire process. I also thank my dissertation committee of Dr. Dan Andresen, Dr. Torben Amtoft, and Dr. Reo Song for serving in my committee, and for their support over the past two years as I moved from an idea to a completed study.

A special thanks to Dr. Andresen for giving me (unlimited) resources on Beocat, and to Adam Tygart for his help in setting up the infrastructure required to complete this work.

I thank my friends, Ashok, Sam, Sandeep, Roshan, Bhushan, and Vijaya, for being there, listening to my worries, and sharing the joy with me. I thoroughly enjoyed the time we spent on our trips, and you guys are an important reason for all the fun time I had at KSU.

It has been a long and exciting journey for me, with many ups and downs, some painful but many cherishing moments. At the end of it, I have learned a lot, met great people, and am ready for another exciting journey!

Dedication

I dedicate my dissertation work to my family. A special feeling of gratitude to my parents, Saradhi and Nirmala who have always encouraged and believed in me. Words cannot describe their love and affection for me; I am forever thankful to them and will always try to make them proud.

I also dedicate this dissertation to my loving wife Kavya. Over the past six years, she cheered me for my success, stood by my side in my failures, and has always motivated me in this journey. I will never be able to find enough words to express how lucky and grateful I am to have Kavya in my life. All these years would not have been so wonderful without Kavya in my life.

Finally, I dedicate this work to my brother Ravi and sister-in-law Keerthi. You guys always shared your happiness with me and advised me at times of difficulty. Ravi, you have been my idol since childhood; you are one of the reasons why I pursued PhD. I am always thankful for your love and support.

Chapter 1

Introduction

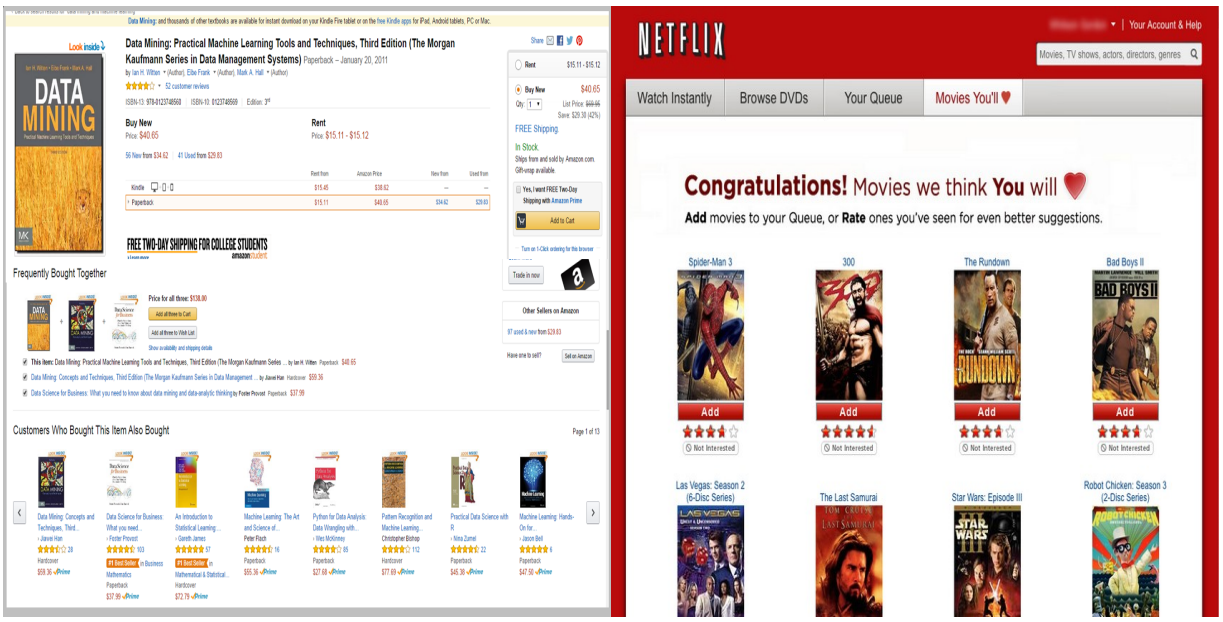
1.1 Basics of Recommender Systems

The growth of the Internet over the past decade has resulted in an exponential growth of online content and the growth of of online markets such as e-Commerce applications. Consequently, an increased variety of products and product information is available to customers. With such large volumes of online data, a Web user could struggle to find content that correlates to his or her interests, a challenge popularly known as the *information overload problem*. For example, a user may be interested in satisfying a current information need or finding a product of interest (e.g., a movie, a book, news, etc.) among the many products available online. Although search engines partially alleviate the problem of finding desired Web content, especially when an information need can be expressed as a query, in many cases, a user often may not be aware of what to look for. Recommender systems can be used to address the information overload problem by suggesting to users interesting movies to watch, news to read, music to listen to, people to collaborate with, etc. These systems are considered to play an important role in the e-Commerce domain. Popular e-Commerce applications such as Amazon and Netflix, where recommender systems are critical to retain users, claim that majority of product sales result from recommendations. For example, Greg

Linden, the implementer of Amazon's first recommendation engine stated[1]:

"Amazon.com recommendations generated a couple orders of magnitude more sales than just showing top sellers."

In general, a recommender system provides personalized item suggestions to users by identifying patterns in user's *explicit* opinions or *implicit* behavioral history. An example of book recommendations and movie recommendations from Amazon.com and Netflix.com are shown in Figures 1.1a and 1.1b, respectively.



(a) Amazon.com Site (b) Netflix.com

Figure 1.1: Recommendations in (a) Amazon.com product site and (b) Netflix.com movie site.

The success of recommender systems in e-Commerce websites such as Amazon.com and Netflix.com has promoted the application of these systems to other domains, such as friendship recommendation in social networks [Konstas et al., 2009], recommendation of scientific articles [Wang and Blei, 2011], ad recommendation via behavioral targeting [Yan et al.,

[1]http://glinden.blogspot.com/2007/05/google-news-personalization-paper.html

2009], and co-author recommendation [Yang et al., 2012]. Although recommender systems have been widely studied in academia and industry in the last decade, interest in this area remains high because of abundant application domains, increasingly large datasets, and significant room for improvement in user personalization accuracy.

1.1.1 Recommendation Problems

A recommendation problem can be defined as follows: given a set of users U and a set of items I, the objective is to estimate the preference of a user $u \epsilon U$ for new items $i \epsilon I$ using historical data stored in the system, and consequently recommend novel items (items with which the user has not previously interacted) for which the estimated preference is high. The preference type for an item by a user varies from one application to another and can be categorized as: *explicit* and *implicit* [Desrosiers and Karypis, 2011]. *Explicit* preference has multiple values, generally numerical (e.g., 1-5 stars) or nominal (love, like, neutral, dislike) values by which a user can show a specific interest level for an item. *Implicit* preference, which captures how often a user interacts with an item, is expressed as a numerical value (e.g., 5 clicks, 2 views, etc.). The underlying assumption of implicit preferences is that users tend to interact with items that they find interesting. Because implicit user preferences are more common in real-world applications as compared to explicit user preferences and this type of data is less explored in the literature, the work described in this dissertation focuses only on problems and approaches with implicit user preferences.

1.1.2 Popular Solutions

Approaches to recommender systems are usually classified into the following categories, based on how recommendations are generated [Desrosiers and Karypis, 2011; Adomavicius and Tuzhilin, 2005; Herlocker et al., 2004; Sarwar et al., 2001]:

- **Content-based Approaches:** Content-based techniques first construct a profile for

each item by identifying defining characteristics for items in the dataset and then associate users with items based on similarity of item profiles [Balabanović and Shoham, 1997; Mooney and Roy, 2000; Pazzani and Billsus, 1997]. For example, in order to recommend movies to a user, movies could be represented using features such as genre, actors, storyline, and cast, and similarities could be computed between movies using these features. A user can receive recommendations for movies similar to movies previously highly preferred. The main drawback of content-based techniques is that they require acquisition of additional information about users and/or items in order to generate recommendations.

- **Collaborative Filtering Approaches:** Contrary to content-based approaches that use item contents previously preferred by a user, collaborative (or social) filtering (CF) approaches rely on the user preferences stored in the system [Koren, 2008; Hu et al., 2008; Baluja et al., 2008; Koren et al., 2009; Ma et al., 2012; Anastasakos et al., 2009; Wang and Blei, 2011; Sarwar et al., 2001; Bell et al., 2007; Hofmann, 2003; Yuan et al., 2011; Konstas et al., 2009; Sarwar et al., 2002; Bell and Koren]. In addition to avoiding the need for acquisition of additional data, another advantage of CF techniques is that they are domain independent. Unlike content-based techniques that require domain knowledge to extract features for user and/or item profiles, CF techniques can be applied to any data domain with information about user-item interactions [Desrosiers and Karypis, 2011; Breese et al., 1998; Herlocker et al., 1999; Bell and Koren; Sarwar et al., 2001].

  CF techniques can be further classified into **neighborhood-based techniques** and **latent factor model-based techniques**. Neighborhood-based techniques predict the preference of a user-item pair based on preferences given to that item by neighbor users. Neighborhood-based approaches have the following merits: ability to explain recommendations to a user, easy and incremental addition of new data, a small number of parameters to tune, and intuitiveness about how items are recom-

mended to users [Bell and Koren]. In contrast to neighborhood-based approaches, latent factor model-based techniques learn a predictive model from user-item preference information in order to predict unknown user-item preferences [Wang and Blei, 2011; Desrosiers and Karypis, 2011; Adomavicius and Tuzhilin, 2005; Cremonesi et al., 2010]. Latent factor model-based techniques also have the advantage of better recommendation accuracy compared to neighborhood-based approaches at the expense of high computational complexity [Sarwar et al., 2001; Hu et al., 2008; Koren et al., 2009; Sarwar et al., 2002].

Given the domain independent nature of CF techniques and their effectiveness in identifying hidden relationships and interdependencies between users and items to predict new and unknown user-item interactions, this dissertation focuses on CF approaches in order to address recommendation problems in various application domains.

1.2 Single-Domain Study

The success of recommender systems to effectively identify content of interest from a potentially overwhelming set of choices, and the ability of these approaches to handle problems in diverse application domains, such as, movies, music, news, and fashion, has made recommender systems a popular and prominent research domain in the commercial industry and the academic community. As a result, many approaches for recommender systems, especially CF techniques, with unique characteristics and strengths have been proposed in the literature. Most of these approaches primarily focus on the utilization of user preferences (explicit or implicit) from one domain to recommend items from the same domain, referred to as single-domain recommender systems. For example, Netflix suggests movies by analyzing existing explicit user ratings for movies, and Last.fm recommends artists by analyzing existing implicit user preferences for artists.

The availability of numerous CF approaches to generate personalized item suggestions

is beneficial to the designers and developers of a recommender system. However, once a need for recommender systems is identified for a data domain, the system designer must determine which approach to implement from the available options. With the tremendous volume of user-item interactions captured and stored daily on the Web, it is impractical to deploy and test every approach in order to choose the most accurate approach for the current application domain. Therefore, the primary objective of the single-domain study was to investigate the effectiveness of two state-of-the-art single-domain CF approaches capable of handling implicit user preferences, specifically Adsorption [Baluja et al., 2008] and Matrix Factorization (MF) [Hu et al., 2008], in order to address the item recommendation problem. Furthermore, the goal was to determine if knowledge about a data domain, specifically knowledge about how links are formed in the domain (based on close connections - resulting in strong local neighborhoods, or based on loose connections - resulting in useful global information), or characteristics of the dataset (e.g., dense or sparse feedback matrices) can effectively advice the choice of CF technique to be used for the current recommendation problem [Parimi and Caragea, 2015d].

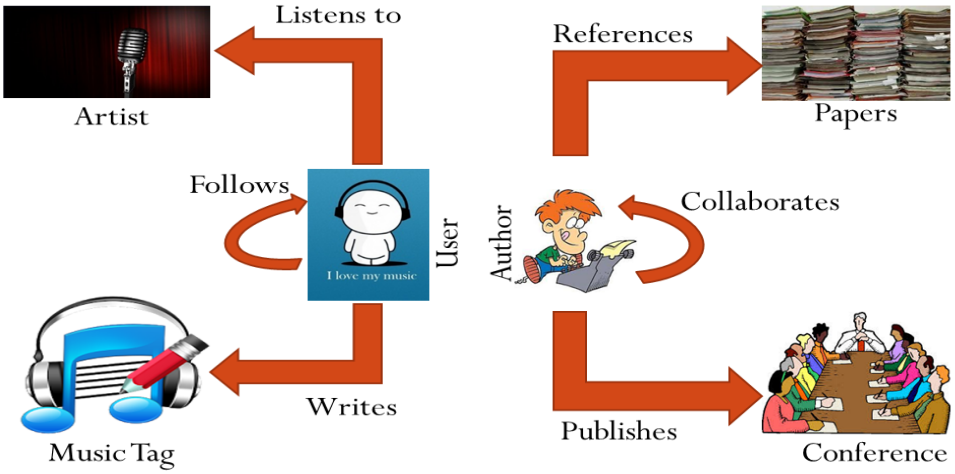
In addition, because the Adsorption algorithm is a neighborhood-based approach, another goal of this work was to research various ways to construct user neighborhoods for Adsorption. Specifically, the focus included devising similarity functions that utilize counts for items preferred by a pair of users and understanding if domain knowledge and dataset characteristics (e.g., median number of items clicked) can be used to choose the most appropriate neighborhood construction approach for an application domain [Parimi and Caragea, 2015d]. Furthermore, because many real-world datasets have an underlying community structure, applicability of community detection approaches to construct neighborhoods for Adsorption were also studied in this work [Parimi and Caragea, 2014].

Finally, motivated by the need to decrease computational costs and runtime of recommendation algorithms, a study was conducted to determine if short user histories can successfully replace long user histories for recommender systems [Parimi and Caragea, 2015d].

1.3 Cross-Domain Study

For many real-world applications, single-domain CF approaches (which utilize user preferences from only one domain) have been the preferred approaches because of their popularity, availability of many open-sourced implementations, and large volumes of information available on the Web regarding implementation, configuration, and deployment of these approaches into production. However, in some real-world scenarios, users often interact with items of multiple types. Advancements in web and BigData technologies have enabled web applications to store user preferences for various types of items. For example, in the music site *Last.FM*, summarized in Figure 1.2a, users click on several *artists* to listen to tracks, connect with other users to make *friends* and share information, and *tag* music tracks with words of their choice. Similarly, in a scientometric site such as *DBLP*, summarized in Figure 1.2b, authors collaborate with other *authors* to publish research work in *conferences* around the world. Furthermore, in each publication, authors *refer* other published articles. Instead of treating each type of item independently and creating a recommendation model for each task/domain, user knowledge gained in one domain can be used in other domains to enhance user personalization experience by improving accuracy of recommendations in one domain, alleviating the cold-start problem, or offering novel, diverse, and serendipitous recommendations [Winoto and Tang, 2008; Berkovsky et al., 2007a; Shapira et al., 2013; Cantador and Cremonesi, 2014; Li et al., 2009a; Pan et al., 2010]. This research area is known as *cross-domain* recommender systems. Approaches proposed in this dissertation for cross-domain recommendation problems are primarily focused on improving recommendation accuracy in one domain, the target domain, using knowledge from auxiliary domains, or source domains.

In the literature, several CF approaches have been proposed to enhance target personalization accuracy using knowledge from source domains. However, these approaches make certain assumptions regarding the cross-domain recommendation problem addressed, assumptions that are not met for some real-world cross-domain datasets. For example, Li



(a) Last.FM Music Site (b) DBLP Scientometric Dataset

Figure 1.2: User interactions with various types of items in (a) the Last.FM music site and (b) the DBLP scientometric dataset.

et al. [2009a,b] proposed two approaches to transfer knowledge from a dense source domain to a target domain under the assumption that multiple domains have the same cluster-level rating pattern. However, in practice, the assumption regarding the existence of a common rating pattern in related domains may not always be valid, and diversity among related domains may outweigh the advantages of knowledge transferring [Gao et al., 2008]. Pan et al. [2010] assumed the existence of two source domains, one with similar users to the target domain and the other with similar items to the target domain, and proposed an approach to transfer knowledge to the target domain using the two source domains. Singh and Gordon [2008] assumed that the same users (or items) in source and target domains can be associated with the same latent factors, and they proposed an approach to collectively factorize data from multiple domains. However, because of the differences in related domains, using the same user (or item) factors may lead to performance degradation. Furthermore, most existing approaches for cross-domain recommender systems require *explicit* (Boolean or numerical) user preferences in both source and target domains. For many problems, however, only implicit feedback is readily available.

In order to address cross-domain recommendation problems that do not satisfy the aforementioned assumptions, two knowledge aggregation approaches based on the neighborhood-based approach for implicit feedback, Adsorption [Baluja et al., 2008], were proposed [Parimi and Caragea, 2015e,b]. Specifically, one approach performs aggregation of neighborhoods (WAN), while the other performs aggregation of recommendations (WAR). Both WAN and WAR approaches can handle cross-domain problems with one or more source domains, implicit user feedback in all domains, and items that share no similarity across domains, while making no assumptions about dataset density. The amount of knowledge transferred from each source domain to the target is controlled through the use of weights. Furthermore, an approach based on MF was also proposed to address the cross-domain recommendation problem mentioned above [Parimi and Caragea, 2015a], given the superior performance of latent factor model-based approaches compared to neighborhood-based approaches. Specifically, the proposed approach identifies latent user and item factors in the source domains and transfers user factors to the target while controlling the amount of knowledge transferred through regularization parameters. Given the proposed approaches, one objective of this work in the cross-domain setting was to experimentally validate the effectiveness of these approaches in improving target recommendation.

1.4 Summary of Contributions

To summarize, the main contributions of this dissertation are as follows:

1. Studied the effectiveness of Adsorption and MF approaches for large-scale user personalization on implicit feedback datasets from different domains and performed an analysis to understand the usefulness of knowledge about domain and data characteristics in order to select the CF approach to use for the domain at hand [Parimi and Caragea, 2015d].
2. Studied and compared several approaches to construct neighborhoods for Adsorption

algorithm. More specifically:

- Devised three ways to compute similarity scores for implicit feedback datasets and used the similarity scores to construct and compare nearest neighborhoods [Parimi and Caragea, 2015d].
- Studied the effectiveness of neighborhoods constructed using modularity-based community detection approaches for large-scale user personalization [Parimi and Caragea, 2014].
- Studied the usefulness of domain knowledge and knowledge about data characteristics to select the neighborhood construction approach for Adsorption algorithm [Parimi and Caragea, 2015d].

3. Proposed two novel approaches based on Adsorption algorithm to handle cross-domain recommendation problems with the goal of improving target recommendation accuracy [Parimi and Caragea, 2015e,b]. In addition:
   - Studied the effectiveness of the proposed approaches to address the cold-start user problem in target domains.
   - Proposed and experimentally validated a novel way to automatically compute weights to be used for source and target domains when aggregating knowledge.
4. Proposed a novel approach based on MF to integrate user latent factors from multiple implicit feedback datasets in order to improve target recommendation accuracy [Parimi and Caragea, 2015a].
5. Experimentally validated the effectiveness of the proposed Adsorption-based and MF-based cross-domain approaches to handle varying amounts of user overlap between the source and target domains [Parimi and Caragea, 2015a,b].

1.5 Thesis Outline

This dissertation is structured as follows: Chapter 2 introduces the Adsorption and Matrix Factorization approaches and describes the intuition and assumptions of these approaches to handle implicit feedback data. Chapter 3 reviews the literature by focusing on the categorization of CF approaches, and on problems and approaches proposed in the single-domain and cross-domain settings. Chapter 4 details several goals of the study in the single-domain setting and presents different ways to construct user neighborhoods for the Adsorption algorithm. Similarly, Chapter 5 introduces the cross-domain recommendation problem, identifies several limitations of existing approaches proposed in the literature, and describes in detail Adsorption-based and MF-based approaches proposed in order to address the cross-domain recommendation problem. Chapter 6 describes datasets used and preprocessing steps required to prepare datasets and evaluation metrics used to measure the performance of recommendation algorithms. Chapters 7 and 8 present experimental results and analysis of results in order to address research questions in the single-domain and cross-domain settings, respectively. Finally, Chapter 9 draws conclusions and discusses open issues and future work.

Chapter 2

Background

Section 2.1 of this chapter describes the basics of collaborative filtering (CF) and provides details regarding two state-of-the-art CF approaches capable of handling *implicit* user preferences that were studied and extended for this dissertation: Matrix Factorization (MF), a latent factor model-based approach and Adsorption, a neighborhood-based approach. Section 2.2 focuses on MF for *explicit* feedback data and describes how MF can be extended for *implicit* feedback datasets. Section 2.3 explains the working of the Adsorption algorithm.

2.1 Types of Collaborative Filtering Approaches

CF is a popular and widely used approach for recommender systems, regardless of the application domain. The basic objective of CF-based algorithms is to provide item recommendations based on item preferences of other like-minded users [Desrosiers and Karypis, 2011; Sarwar et al., 2001; Adomavicius and Tuzhilin, 2005]. CF approaches can be grouped into neighborhood-based approaches and latent factor model-based approaches [Adomavicius and Tuzhilin, 2005; Koren, 2008; Bell and Koren; Cremonesi et al., 2010; Wang and Blei, 2011].

2.1.1 Neighborhood-based Approaches

Neighborhood-based approaches directly utilize user preferences stored as a user-item preference matrix in order to predict user preferences for new items. This can be accomplished in two ways: user-based or item-based recommendation algorithms [Desrosiers and Karypis, 2011; Adomavicius and Tuzhilin, 2005]. User-based recommender systems [Baluja et al., 2008; Konstan et al., 1997; Sarwar et al., 2002; Ma et al., 2012] predict the preference of user u for an item i using preferences of other users v, called neighbors, who have similar preferences as user u. Item-based recommender systems [Sarwar et al., 2001; Linden et al., 2003], on the other hand, predict the preference of user u for an item i based on the preferences of u for items similar to i. In such systems, two items are considered similar if users in the system have similarly preferred these items.

2.1.2 Latent Factor Model-based Approaches

In contrast to neighborhood-based approaches, latent factor model-based techniques [Bell et al., 2007; Hu et al., 2008; Koren et al., 2009] make use of the user-item preference matrix to learn a predictive model. The intuition behind these approaches is to model user preferences for items with factors that represent latent characteristics of users and items in the system. The model, which can be trained using available data in the system, can be used to predict user preference for new items. Many latent factor model-based approaches for recommender systems have been proposed in the literature, including Latent Semantic Analysis [Hofmann, 2003], Latent Dirichlet Allocation [Blei et al., 2003], and Singular Value Decomposition [Bell et al., 2007; Koren, 2008; Koren et al., 2009; Takács et al., 2009].

2.1.3 CF Approaches for Implicit and Explicit User Feedback

In the recommender systems domain, user preferences for items are primarily categorized into *explicit* and *implicit* user feedback [Desrosiers and Karypis, 2011; Sarwar et al., 2002].

Explicit user feedback corresponds to preferences a user provides on items, such as liking or disliking an article or evaluating a movie on a scale of 1 through 5. In contrast to explicit feedback, implicit user feedback captures how often a user uses an item (e.g., clicking on links, purchasing fashion items, collaborating with people, etc.). Most CF approaches proposed in the literature are designed to handle explicit user feedback in the data domains [Bell et al., 2007; Desrosiers and Karypis, 2011; Koren, 2008; Bell and Koren; Koren et al., 2009; Sarwar et al., 2001, 2002], with the exception of a few approaches [Baluja et al., 2008; Hu et al., 2008; Ma et al., 2012; Linden et al., 2003] that assume implicit user feedback.

2.2 Matrix Factorization

In the recent past, MF techniques have garnered extensive attention and popularity because of their accuracy and ability to scale to large datasets. The general idea is to model users and items into a joint latent space so that new user-item preferences can be computed using inner products of these latent features. This section discusses how MF can be used with *explicit* feedback data [Koren et al., 2009; Koren, 2008; Bell et al., 2007] and how it can be adapted to handle *implicit* feedback data [Hu et al., 2008].

2.2.1 Matrix Factorization for Explicit Feedback

In the presence of explicit ratings for items by users, the MF model associates each user with a user-factors vector $x_u \in \Re^f$ and each item with an item-factors vector $y_i \in \Re^f$, where f is dimensionality of the latent space. New user-item ratings are computed as the inner product of these user-factor and item-factor vectors. For example, a rating score for item i by user u is given by $\widehat{r_{ui}} = x_u^T y_i$. The user and item factors can be learned, iteratively, by minimizing the squared error between observed ratings and computed ratings. Regularization can be used to avoid overfitting the model. Equation (2.1) represents regularized squared-error minimization for the MF model.

$$\min_{x_*,y_*} \sum_{(u,i)\epsilon k} \left(r_{ui} - x_u^T y_i\right)^2 + \lambda \left(\sum_u \|x_u^2\| + \sum_i \|y_i^2\| \right) \tag{2.1}$$

In Equation (2.1), k is the set of (u, i) pairs for which r_{ui} is available apriori; λ is the regularization parameter. Previous work on MF relied on techniques such as imputation to fill in missing values in the initial matrix and use all ratings to learn the model, but recent work has suggested the use of only prior observed ratings to learn the model [Koren, 2008]. This is because of: high computational complexity of imputation and improper imputation that may mislead the model. In order to optimize the cost function given by Equation (2.1) and learn the factors, approaches such as stochastic gradient descent are widely used. An example of factorization of a user-item rating matrix can be seen in Figure 2.1.

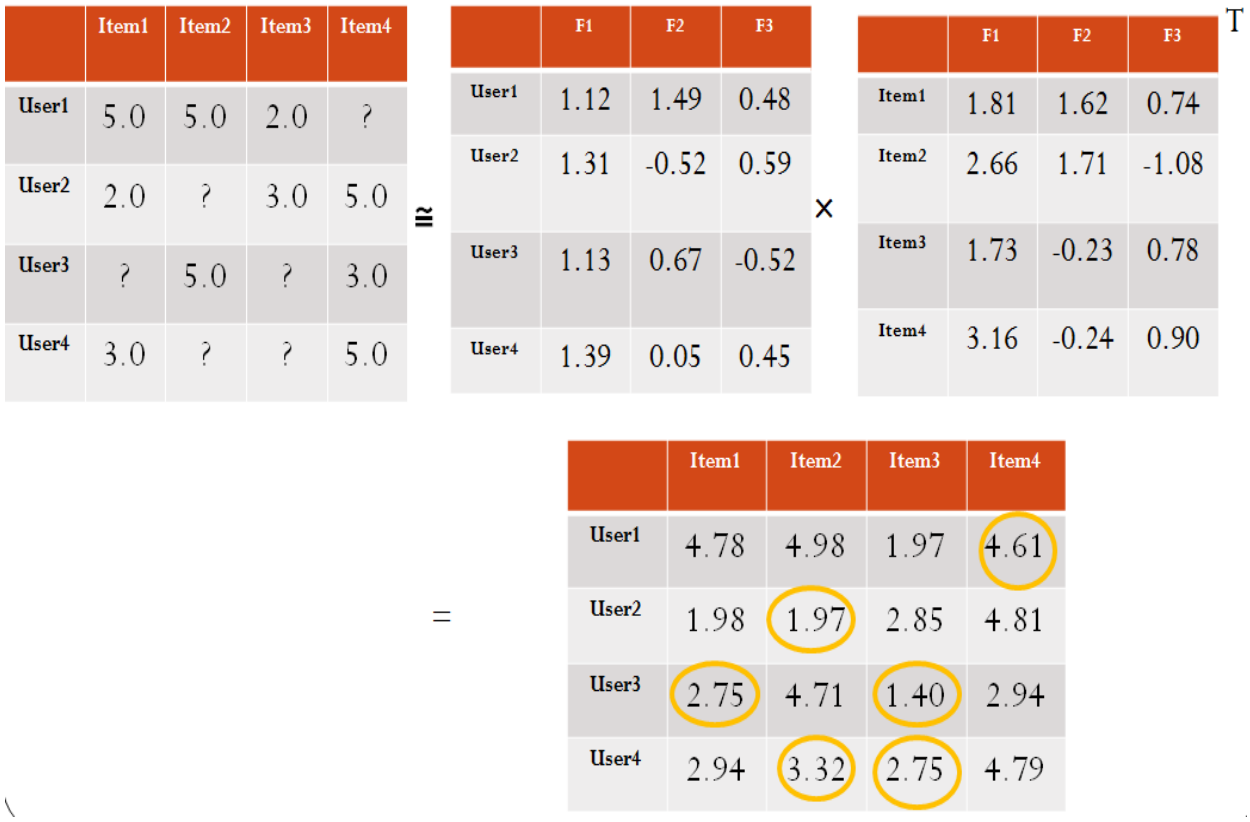


Figure 2.1: Factorizing a user-item rating matrix into a user-factor and an item-factor matrices.

In Figure 2.1, we can see a user-item rating matrix in which each cell in the matrix corresponds to an explicit rating for an item by a user; question mark ("?") in a cell of the user-item rating matrix indicates that the item corresponding to the cell column is not rated by the user corresponding to the cell row, and is referred to as a missing value. The MF approach factorizes the user-item rating matrix into a user-factor matrix and an item-factor matrix as shown in Figure 2.1. The product of user-factor and item-factor matrices fills the missing values in the original rating matrix, as indicated by the yellow colored circle in Figure 2.1, and correspond to predicted ratings for an item by a user. Top n items with the highest predicted ratings are then recommended to the user.

2.2.2 Matrix Factorization for Implicit Feedback

Hu et al. [2008] identified unique characteristics of *implicit* feedback that restrict usage of the traditional MF approach, and they proposed a new approach, a variation to traditional MF, that can handle *implicit* user preferences. In their approach, user feedback for user u and item i, denoted by r_{ui}, is represented by two new variables: preference and confidence. While preference, denoted by p_{ui}, indicates the preference of user u for item i, the confidence variable, denoted by c_{ui}, associates confidence values to items preferred by the user. The intuition is to associate small confidence values even if the user did not use (click, listen, buy, read, etc.) the item, since not using an item may stem from various reasons such as lack of knowledge about the item or limited availability in addition to dislike of the item. Equations (2.2) and (2.3) depict computation of p_{ui} and c_{ui} from user feedback r_{ui}, respectively.

$$p_{ui} = \begin{cases} 1 & \text{if } r_{ui} > 0 \\ 0 & \text{if } r_{ui} = 0 \end{cases} \tag{2.2}$$

$$c_{ui} = 1 + \alpha r_{ui} \tag{2.3}$$

	Item1	Item2	Item3	Item4
User1	50	20	2	0
User2	10	0	5	31
User3	0	1	0	38
User4	3	0	0	41

r_{ui}

	Item1	Item2	Item3	Item4
User1	1	1	1	0
User2	1	0	1	1
User3	0	1	0	1
User4	1	0	0	1

p_{ui}

Figure 2.2: Computing a preference matrix (p_{ui}) from a user-item implicit feedback matrix (r_{ui}) according to Equation 2.2.

According to Equation (2.2), user u prefers item i (i.e., p_{ui} takes a value 1 when the user uses the item: i.e., $r_{ui} > 0$). Figure 2.2 shows an example of computing preference values (p_{ui}) from user feedback values (r_{ui}). The preference variable is associated with widely varying confidence levels: zero values for p_{ui} are associated with low confidence values (c_{ui}) and confidence increases as r_{ui} values increase, according to Equation (2.3). The rate of increase for the c_{ui} value is controlled by parameter α, which can be determined by cross-validation. Preferences for unobserved user-item interactions can be computed as the inner product of user latent factors and item latent factors (i.e., $p_{ui} = x_u^T y_i$). Equation (2.4) represents the objective function for minimization of the MF model with implicit feedback data:

$$\min_{x_*, y_*} \sum_{(u,i)} c_{ui} \left(p_{ui} - x_u^T y_i\right)^2 + \lambda \left(\sum_u \|x_u^2\| + \sum_i \|y_i^2\| \right) \tag{2.4}$$

In Equation (2.4), λ denotes the regularization parameter. The objective function can be efficiently solved using alternating least squares (ALS), and analytic expressions for user and item factors that minimize Equation (2.4) can be obtained by differentiation.

2.2.3 Implementation Details

The MF approach for implicit feedback data used in this work was implemented using the Map-Reduce framework [Dean and Ghemawat, 2008] and is part of the Apache Mahout software[1]. An overview of the computation of user factors x_u for user u and item factors y_i for item i at iteration t given the user and item factors at iteration $t-1$ is as follows:

- **User Factor Map Procedure:** User factors x_u for user u are computed using preference and confidence variables and item factors according to the update Equation (4) in [Hu et al., 2008].
- **Item Factor Map Procedure:** Item factors y_i for item i are computed using preference and confidence variables and user factors according to the update Equation (5) in [Hu et al., 2008].

2.3 Adsorption Algorithm

The Adsorption algorithm proposed by Baluja et al. [2008] was originally designed to analyze a user-video graph and generate personalized video suggestions to a user. This algorithm is a very general semi-supervised framework for classification that works by propagating preference information through the rich graph structure. The algorithm was successfully used for various tasks such as recommending YouTube videos [Baluja et al., 2008], classification, and sentiment analysis [Talukdar and Crammer, 2009]. In order to generate recommendations for a user, the algorithm selects items commonly co-viewed with the user's watched videos. This is achieved by identifying items with multiple short paths starting from the user node in the user-item bipartite graph. While identifying these paths, the algorithm ignores paths passing through high-degree nodes in order to avoid drifting away from user's interest. The following properties are ensured by the Adsorption algorithm:

[1] https://mahout.apache.org/

1. Several paths exist in the graph from user u to item i.
2. The path chosen between user u and item i in the user-item graph is short.
3. Paths from user u and item i avoid high-degree nodes.

2.3.1 Basic Terminology

Let $G = (V, E, w)$ be an undirected graph, where V is the set of users, E is the set of edges between users, and $w : E \rightarrow \mathbb{R}^+$ is a function that assigns a positive weight to edges. Let L be the set of possible labels and let m be the size of the set L (i.e., $|L|$). In a classification setting, labels correspond to classes; in a recommendation setting, labels correspond to items preferred by users in the dataset.

Each user in the graph is associated with two row-vectors, Y_v, $\widehat{Y}_v \, \epsilon \, \mathbb{R}^+_m$. Vector Y_v denotes initial label distribution for user v (i.e., Y_{vx} represents the probability that user v prefers label x). Vector $\widehat{Y}_v$ indicates predictions made by the algorithm for user v and encodes a distribution over the m labels. The higher the value of Y_{vx}, the stronger the belief that user v has a high preference for label x; $Y_{vy} = 0$ implies that no prior knowledge exists regarding the label y for user v ($x \neq y$). Similarly, the higher the value of $\widehat{Y}_{vy}$, the stronger the a posteriori belief that y corresponds to a good label for user v, assuming that y is a label that was not preferred by user v a priori (i.e., $Y_{vy} = 0$).

Using the above definitions, the Adsorption algorithm can be expressed using three distinct, yet equivalent, views: "Random-walk View", "Averaging View", and "Adsorption via Linear Systems". We explain the algorithm using the "Random-walk View" (see [Baluja et al., 2008; Talukdar and Crammer, 2009] for more details).

2.3.2 Adsorption via Random-walk

The Adsorption algorithm can be described as a random-walk on the user-user graph G. At each node, the algorithm is presented with three options: stop and return, or *inject* the

initial label distribution Y_v of the node; *terminate* or abandon the walk and return an all-zero vector, $\mathbf{0}_m$; or *continue* the walk to neighbor node u chosen according to the probability $Pr[u|v]$, given by Equation (2.5), and emit predicted labels $\widehat{Y}_u$, given by Equation (2.6). The injection, termination, and continuation steps have probabilities p_{inj}, p_{term}, and p_{cont}, respectively, and the sum of these probabilities should be 1. For a particular problem, these probability values can be selected using cross-validation.

The probability distribution over the neighbors u of a user v is estimated using the following equation:

$$Pr[u|v] = \begin{cases} \dfrac{w_{uv}}{\sum\limits_{u:(u,v)\in E} w_{uv}}, & \text{if } (u,v) \in E \\ 0, & otherwise \end{cases} \tag{2.5}$$

Furthermore, new labels $\widehat{Y}_v$ for a user v can be computed using the following equation:

$$\widehat{\mathbf{Y}}_v = p_{inj} \times \mathbf{Y}_v \ + \ p_{cont} \times \sum_{u:(u,v)\in E} Pr[u|v]\widehat{\mathbf{Y}}_u \ + \ p_{term} \times \mathbf{0}_m \tag{2.6}$$

The random-walk process is initiated at every node v in the graph G and is repeated until the algorithm converges (i.e., values in $\widehat{Y}_v$ cease to change). In practice, the algorithm is run for a fixed number of iterations or until the change in $\widehat{Y}_v$ is below a small threshold. The final values in $\widehat{Y}_v$ are used to make recommendations to user v. Specifically, items y that have high probability in $\widehat{Y}_v$ and have not yet been preferred by user v are recommended.

Figure 2.3 shows a user-user graph with user-user similarity as edge weights. User preference for items are converted into a distribution and are shown by different colors on the nodes of the graph as shown in Figure 2.3. In order to compute recommendations for the node colored with white, shown in Figure 2.3, the Adsorption algorithm utilizes item distributions of the three neighboring nodes and the user-user similarity on edges and predicts the item preferences as a distribution over all items in the dataset. From the item

distribution computed by the Adsorption algorithm, top n items with the highest predicted preferences are recommended to the user.

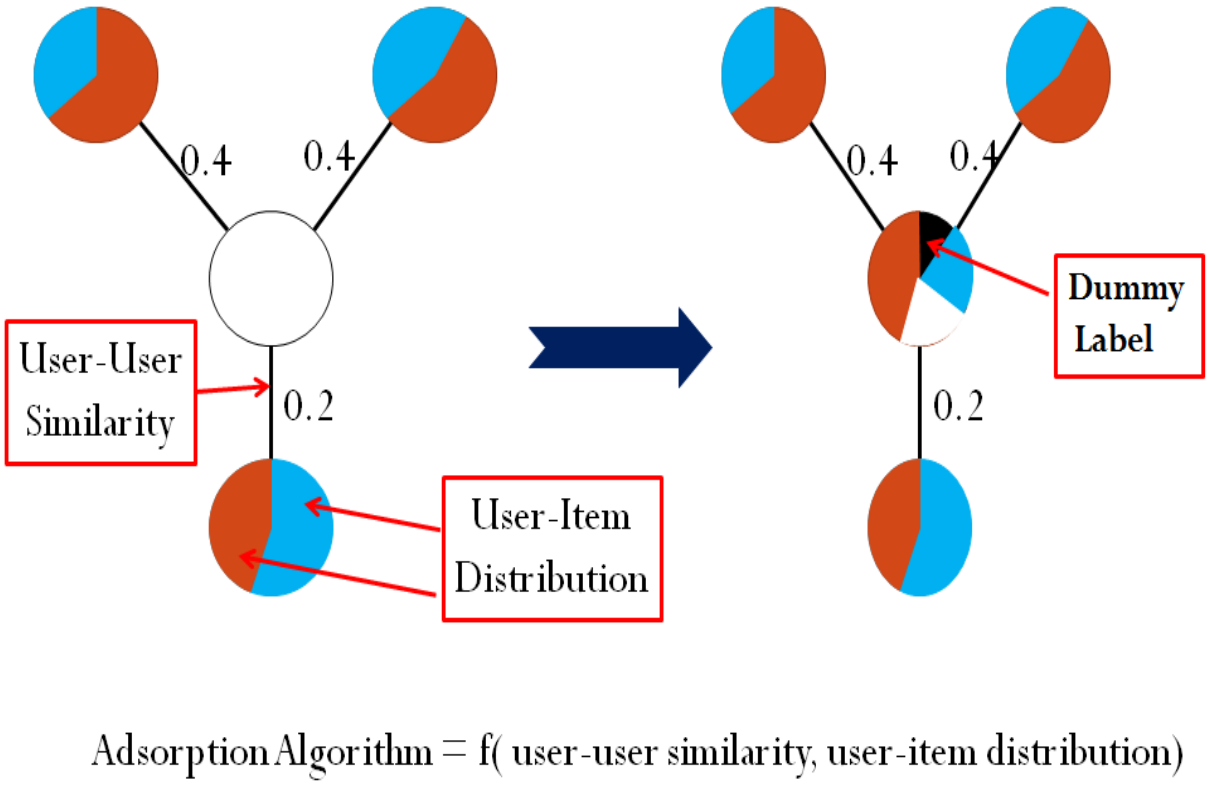


Figure 2.3: Illustration of the Adsorption algorithm.

2.3.3 Implementation Details

The Adsorption algorithm was implemented using Map-Reduce framework [Dean and Ghemawat, 2008], specifically, its open source implementation, Hadoop. The algorithm consists of two phases. In the first phase, $\langle userID, itemID, preferenceCount \rangle$ information was used to compute user-user similarity and item distribution for each user (i.e., preference counts for all items clicked by that user are converted into a distribution). Computation of user-user similarity involves a sequence of Map-Reduce jobs, and final output from these sequence of jobs consists of records in the format: $(\langle v \rangle, \{\langle u_1, w_1 \rangle, \langle u_2, w_2 \rangle, \cdots, \langle u_n, w_n \rangle\})$, where v

is a user, $u_1, u_2 \cdots, u_n$ are neighbor users of v, and $w_1, w_2, \cdots, w_n$ are weights between v and users $u_1, u_2, \cdots$, and u_n, respectively. The final user neighborhood corresponds to the k nearest neighbors (kNN) selected for each user according to computed user-user similarity. Similarly, user-item distribution computations consist of a sequence of Map-Reduce jobs, and the final output from these sequence of jobs consists of records in the format: $(\langle v \rangle, \{\langle i_1, s_1 \rangle, \langle i_2, s_2 \rangle, \cdots, \langle i_m, s_m \rangle\})$, where v is a user, $i_1, i_2 \cdots, i_m$ are items preferred by user v, and $s_1, s_2, \cdots, s_m$ are preference scores for items $i_1, i_2, \cdots$, and i_m, respectively, such that $\sum_{j=1}^{m} s_j = 1$. The user-item distribution corresponds to the input to the first iteration of the Adsorption algorithm.

In the second phase, iterative computation of estimated item preferences for users was performed. The kNN result files generated in the first phase are distributed across all nodes in the platform and are loaded into memory in a HashMap as follows: suppose there is a record $(\langle v \rangle, \{\langle u_1, w_1 \rangle\})$ in the kNN of a user, the contents of the HashMap will be $\{\langle u_1 \rangle, \langle v, w_1 \rangle\}$. This indicates that the estimated item preferences of user u_1 at iteration $(t-1)$ should be propagated to user v at iteration t. Each iteration starts a Map-Reduce job, and input to the mapper at iteration t corresponds to output from the job at iteration $(t-1)$. Details of the iteration computation are described below.

- **Map Procedure:** The k nearest neighborhood of each user, created in the first phase, is stored across all nodes as described above. Assuming that the estimated item distribution for a user u_1 at iteration $(t-1)$ is recorded as $(\langle u_1 \rangle, \{\langle i_1, s_1 \rangle, \langle i_2, s_2 \rangle\})$, and with record $\{\langle u_1 \rangle, \langle v, w_1 \rangle\}$ in memory, records $(\langle v \rangle, \langle i_1, s_1 \times w_1 \rangle)$ and $(\langle v \rangle, \langle i_2, s_2 \times w_1 \rangle)$ are emitted in the mapper.

- **Reduce Procedure:** At the reducer, all item preferences emitted from neighbor users for user v are collected, and scores for each item i_j are aggregated, where $1 \leq j \leq m$. Aggregated scores are then converted into a distribution (i.e., sum of scores of all items is 1). The output from the reducer corresponds to the estimated item preference for user v at iteration t.

Chapter 3

Literature Review

Because this work focuses on collaborative filtering (CF) approaches designed to address two distinct but related tasks for recommender systems (single-domain and cross-domain recommendation tasks), this chapter is primarily organized into two sections. Section 3.1 focuses on several recommendation applications addressed using CF approaches in the single-domain setting (Section 3.1.1) and on approaches that study clustering users for neighborhood-based CF approaches (Section 3.1.2). Section 3.2 focuses on CF approaches that have the goal of improving target recommendation accuracy by leveraging knowledge from multiple source domains (Section 3.2.2) and on approaches that propose ways to compute weights between a pair of domains for knowledge aggregation (Section 3.2.3).

3.1 Single-Domain Setting

This section first reviews collaborative filtering approaches based on Adsorption and Matrix Factorization, and relevant recommendation applications in the single-domain setting in Section 3.1.1, given that the Adsorption and the Matrix Factorization approaches (described in Chapter 2) are used in this work. Later, Section 3.1.2 describes existing approaches and applications that cluster users for recommender systems, since one goal of this study is to

understand the usefulness of community detection for the Adsorption algorithm.

3.1.1 Single-Domain CF Approaches and Relevant Applications

Neighborhood-based Approaches

As mentioned, recommender systems, particularly neighborhood-based approaches, have been used to tackle the information overload problem in many application domains. For example, Konstan et al. [1997] discussed several challenges associated with recommending Usenet news, such as infrequent occurrences and longer lifetimes of news articles, in addition to the rating sparsity problem common in real-world recommender systems, and they demonstrated that neighborhood-based CF approaches could be used to recommend Usenet news. Data used in their work has explicit user feedback.

Sarwar et al. [2001] suggested the existence of a trade-off between the amount of time a user-based CF approach spends on searching for neighbors and the quality of recommendations generated. In order to design a highly scalable CF approach without compromising recommendation quality, as compared to user-based approaches, the authors proposed item-based collaborative filtering approaches. They also studied the applicability of several similarity computation techniques for computing item-item similarities (e.g., item-item correlation vs. cosine similarities between item vectors) and techniques for obtaining recommendations from the similarity techniques (e.g., weighted sum vs. regression model), and they evaluated the effectiveness of the studied approaches for recommending movies. Data used in their work had explicit user feedback.

Linden et al. [2003] also suggested that user-based CF approaches for recommender systems may be computationally expensive and proposed an item-to-item CF technique to recommend items in *Amazon.com*. The idea was to match a user's purchased and rated items to similar items, and then combine those similar items into a recommendation list. In order to determine the most-similar match for a given item, their algorithm builds a similar-items table by finding items that customers tend to purchase together (co-occurrence). Although

data used in their work had explicit user feedback, the ratings were ignored when computing item similarities.

Baluja et al. [2008] proposed the Adsorption algorithm to recommend YouTube videos to users by analyzing the user-video graph. The algorithm propagates preference information to neighboring nodes in the graph structure, as explained in Section 2.3. Data used in their work had implicit user feedback. Adsorption was successfully used for other tasks such as classification and sentiment analysis of textual data [Talukdar and Crammer, 2009].

Ma et al. [2012] proposed a general framework to mine web graphs for recommendations. The proposed approach was based on the concept of heat diffusion in Physics and used random-walks to propagate similarity information to neighboring nodes in a user-item bipartite graph. The proposed approach was used to suggest queries from a query-URL bipartite graph and images from an image-tag bipartite graph. Data used in this work had implicit user feedback.

Latent Factor Model-based Approaches

Because MF approaches have significantly increased in popularity due to their impressive recommendation accuracy and scalability, several approaches based on MF have been proposed to address many recommendation applications. Among the many authors that have used MF Koren et al. [2009] notably used MF in the Netflix prize competition where the task was to improve movie recommendation accuracy using ratings (explicit feedback). The proposed approach was successfully improved the Netflix Root Mean Squared Error (RMSE) score (at that time) on movie ratings by 10%.

Hu et al. [2008] extended the MF approach proposed by Koren et al. [2009] in order to accommodate implicit user feedback. The proposed approach, described in Section 2.2.2, was used to recommend TV series to a user based on his or her watching history for TV series (implicit feedback).

Wang and Blei [2011] proposed an approach that combined the merits of two traditional

CF approaches, specifically MF and probabilistic topic modeling, in order to recommend scientific articles to users of online communities, such as *Mendeley*[1], *CiteULike*[2]. The primary objective of this work was to overcome the item cold-start problem (recommending items that are not yet preferred by any users in the dataset) by utilizing additional knowledge. The proposed approach used textual data from scientific articles and the list of papers in user libraries (implicit feedback) to generate recommendations of existing and newly published articles.

In the literature, a small number of works have compared the performance of neighborhood-based and MF approaches for applications, such as recommending movies and TV series [Takács et al., 2008; Hu et al., 2008], and suggested that MF approaches have better recommendation accuracy compared to neighborhood-based approaches. However, no prior work was found that compared the performance of Adsorption [Baluja et al., 2008] and MF [Hu et al., 2008] approaches for datasets with implicit user feedback. Also, no prior work was found that studied, by comparison, the influence of data and domain characteristics on the performance of Adsorption and MF. For Adsorption, no prior work has studied the importance of different neighborhood construction approaches for implicit feedback datasets or the usefulness of data and domain characteristics for selecting user neighborhoods. This dissertation specifically focuses on the aforementioned tasks that have not been studied in the literature.

Other Applications

Yan et al. [2009] provided an empirical study of the click-through log of advertisements collected from a commercial search engine in order to validate and compare various behavioral targeting strategies for online advertising. The primary objective of this study was to verify whether behavioral targeting is able to help online advertising. From experimental results,

[1] https://www.mendeley.com/
[2] http://www.citeulike.org/

they concluded that a) users who clicked the same ad demonstrate similar behaviors on the Web; b) click-through rate (CTR) of an ad can be improved by behavioral targeting in sponsored search; (c) use of short-term user behaviors to represent users is more effective than using long term user behaviors for behavioral targeting. The proposed approach first represented users as vectors of clicked URLs and as vectors of utilized search query words. Later, users are segmented into groups based on the similarity of the two vectors using two common clustering algorithms: k-means [Kanungo et al., 2002] and CLUTO[3].

3.1.2 Community Detection for Recommender Systems

Many researchers have recently addressed the item recommendation problem by clustering users into groups. For example, Sahebi and Cohen [2011] used user communities extracted from different dimensions of social networks, such as friendship networks, item similarity networks, and commenting networks, in order to capture user similarities. Extracted user communities were used with k nearest neighborhood-based CF approaches to improve the quality of recommendations and to provide a solution to the user cold-start problem. In their approach, the authors optimized a modularity metric to detect communities from multidimensional networks, and they showed that performing CF within community members is more effective than running CF on all users. They also showed that the proposed approach can be used to overcome the cold-start user problem.

Sarwar et al. [2002] discussed various limitations of CF techniques, including scalability and sparsity problems, and proposed a clustering-based CF algorithm in order to address these limitations for large datasets, such as datasets found in e-commerce applications. The objective was to group users into a fixed number of clusters using k-means clustering algorithm. The users in a cluster represent the nearest neighbors for each user in that cluster. These nearest neighborhoods were then used with user-based CF approaches to

[3] http://glaros.dtc.umn.edu/gkhome/views/cluto

predict movie ratings.

Ying et al. [2013] proposed a preference-aware community detection approach that integrates social factor information and user preference information into the recommendation model. The core idea of their approach was to first use the social network structure existing between users to create initial communities and then use the user preference information to filter the communities into clusters. Users in clusters represent the nearest neighborhoods for each user in that cluster. The nearest neighborhoods were then used to predict item ratings for users, similar to CF approaches.

Xin et al. [2014] used three different metrics to measure the importance of nodes in a weighted user-user similarity network, specifically, average node weight, betweenness centrality [Leydesdorff, 2007], and node degrees, in order to construct user communities. These user communities were used to select the nearest neighbors for a user, and neighborhoods were used with a neighborhood-based CF approach in order to recommend books.

Although some work has been conducted on clustering users and use of clusters to construct neighborhoods for user-based CF approaches, to the best of the author's knowledge, no prior work has studied the application of modularity-based community detection to construct neighborhoods for use with the Adsorption algorithm for large-scale recommendation applications in a homogeneous setting.

3.2 Cross-Domain Setting

This section describes well-known classifications of cross-domain approaches in the literature and reviews several cross-domain neighborhood-based and latent factor model-based approaches, whose goal is to improve recommendation accuracy in a target domain by leveraging knowledge from auxiliary domains and corresponding recommendation applications. This section also describes work on approaches that quantitatively represent the correlation between domains.

3.2.1 Classification of Cross-Domain Approaches

Cross-domain recommender systems are gaining popularity because many applications, such as social networks and e-commerce sites, have begun to collect user histories for items from many domains. However, no unified perception of the cross-domain recommendation problem exists. According to work in [Cremonesi et al., 2011; Fernández-Tobías et al., 2012], given two domains $\mathbb{A}$ and $\mathbb{B}$, the objective of a cross-domain recommendation task can be a) to improve recommendation accuracy in a target domain, for example, $\mathbb{A}$, using knowledge from both domains $\mathbb{A}$ and $\mathbb{B}$, b) to recommend items in both $\mathbb{A}$ and $\mathbb{B}$ to users in $\mathbb{A}$ and $\mathbb{B}$, respectively, or c) to recommend items in $\mathbb{B}$ to users in $\mathbb{A}$ and vice versa. The cross-domain recommendation problem described in this dissertation is focused on the task of improving target recommendation accuracy using knowledge from two or more domains.

The survey by Fernández-Tobías et al. [2012] and the tutorial by Cantador and Cremonesi [2014] present another classification of cross-domain approaches, extending previous classification schemes [Li, 2011; Pan et al., 2011]. Two main groups of cross-domain recommendation approaches are distinguished: *Collective* and *Adaptive*. Collective models simultaneously exploit information from several domains and use this information to improve recommendation accuracy in a target domain [Winoto and Tang, 2008; Singh and Gordon, 2008; Gao et al., 2008; Hu et al., 2013]. In contrast to collective approaches, adaptive approaches capture information such as similar users, similar items, or latent user and item factors in each source domain separately and then transfer this information to the target domain in order to improve recommendation accuracy [Berkovsky et al., 2007a,b; Li et al., 2009a,b; Pan et al., 2010; Shapira et al., 2013; Burke et al., 2014; Vahedian, 2014]. According to this classification, approaches proposed in this dissertation for cross-domain recommendations tasks can be categorized as Adaptive approaches.

3.2.2 Cross-Domain CF Approaches and Relevant Applications

This section reviews several neighborhood-based and latent factor model-based cross-domain approaches whose objective is to improve target recommendation accuracy. This section also includes discussion of works that have studied the cold-start user problem and user overlap scenarios.

Neighborhood-based Cross-Domain Approaches

Berkovsky et al. [2007a,b] proposed approaches to integrate four types of information captured from multiple sources into a target domain: *a*) standard, where ratings from source domains are integrated into the target domain to enrich ratings, *b*) heuristic, where nearest-neighborhoods (similar items or similar users) are computed in the source domains and shared with the target domain, *c*) cross-domain, where domain-dependent similarity scores (user or item) are computed in the source domains and shared with the target domain, *d*) remote-average, where recommendations are computed in the source domains and transferred into the target domain. Experiments were conducted on a movie rating dataset with explicit user feedback. In order to simulate the existence of multiple domains, the movie dataset was partitioned into domains based on genre.

Winoto and Tang [2008] aggregated user preference matrices from several domains into a single preference matrix and used a standard neighborhood-based approach on the aggregated matrix. The authors experimented with several combinations for source and target domains to verify if knowledge from multiple domains can improve target recommendation accuracy. They concluded that cross-domain recommendations tend to be less precise than single-domain recommendations. Experiments were conducted on a manually collected dataset with explicit user feedback for items belonging to 12 distinct domains (e.g., movies, TV series, books, songs, games, etc.). Four combinations of source and target domains were considered in this work.

Shapira et al. [2013] conducted a study to investigate a) if user preferences in the Facebook social network can be used to replace or complement the user rating information for items from a domain (Facebook mentions about different types of items are considered as different domains), and b) if user preferences from several domains, such as movies, music, and TV shows, from Facebook can be used to alleviate the data sparsity problem in a single target domain for CF. In their approach, they aggregated user preferences for various item types in order to enrich ratings because ratings were too sparse to determine accurate similarities in a single domain. The enriched dataset was used to determine similar users and recommend items from the target domain. The dataset used in this study had implicit user preferences (number of times an item is mentioned on Facebook social networking site).

Latent Factor Model-based Cross-Domain Approaches

Several MF approaches have also been proposed in the literature to improve target accuracy by collectively factorizing preference matrices from multiple domains or by transferring user and/or item latent factors from source domains to the target. For example, Singh and Gordon [2008] proposed the Collective Matrix Factorization (CMF) approach that collectively factorizes a user-item rating matrix and an item-context matrix by sharing item-specific latent features. The underlying assumption of their approach was that latent factors of the same users (or items) can be shared between source and target domains. The proposed approach was used to predict movie ratings (explicit feedback) using user ratings for movies and genre information for movies. User ratings were sampled from the Netflix Prize data [Bennett et al., 2007], and genre information was collected from the Internet Movie Database[4].

Li et al. [2009a] proposed Codebook Transfer (CBT) in order to transfer knowledge from a dense source domain and reduce sparsity in a target domain. The CBT approach, which does not require an overlap between users or items in the two domains, has two steps: first, it

[4] http://www.imdb.com/interfaces

computes a cluster level rating pattern referred to as the *codebook* in the source domain, and later, it uses the *codebook* to cluster users and items in the target domain. A later extension, known as rating matrix generative model (RMGM) [Li et al., 2009b], integrated the two steps in CBT into a single step with soft clustering for users and items. The two approaches were tested using two datasets: a movie dataset sampled from MovieLens dataset (target domain) with explicit user preferences (ratings) in order to recommend movies, and a book dataset sampled from BookCrossing dataset (target domain) with explicit user preferences (ratings) in order to recommend books. For both target recommendation tasks, the source domain was sampled from EachMovie dataset with explicit ratings for movies.

Gao et al. [2008] suggested that diversity in related domains may restrict the existence of a cluster-level rating pattern, and they proposed the Cluster-Level based Latent Factor Model (CLFM) approach using a joint nonnegative matrix tri-factorization framework. The main idea of the proposed approach was to learn the common rating pattern shared across domains. Furthermore, the approach had the flexibility to control the optimal level of sharing and the ability to capture domain-specific rating patterns of users in each domain. The proposed approach was used to improve the accuracy of movie and book recommendations. Datasets used in [Gao et al., 2008] were identical to datasets used in [Li et al., 2009a].

The Coordinate System Transfer (CST) proposed by Pan et al. [2010] is another transfer learning approach that utilized two auxiliary source domains: one domain with user overlap (user source domain) and the other domain with item overlap (item source domain) with the target domain. The idea was to first learn the user and item latent factors using the user source domain and the item source domain, respectively, and later integrate the user and item latent factors from the source domains into the target domain through regularization. The proposed approach was evaluated to recommend movies using two movie rating datasets with explicit user feedback.

Loni et al. [2014] proposed a cross-domain approach based on merging user ratings in order to improve recommendation accuracy in a target domain. In their approach, the

target user ratings and the user ratings from several source domains were merged into a single user-item rating matrix using a domain-dependent real valued function to control the amount of knowledge transferred from a source domain to the target. The merged user-item rating matrix was given as input to a Factorization Machine model that can predict user preferences for items. The proposed approach was tested on a dataset from Amazon with explicit user ratings for books, music CDs, DVDs, and VHS videotapes.

Tensor Factorization-based Cross-Domain Approaches

The use of tensor factorization for recommender systems has been studied in several works. For example, Karatzoglou et al. [2010] used tensor factorization to generate context-aware recommendations from user-item-context tensors, where a context can be time or gender. Hu et al. [2013] factorized the user-item-domain cubical tensor to address the data sparsity and cold-start problems in target domains. The proposed approach was evaluated on the Amazon dataset with explicit user feedback used in the work by [Loni et al., 2014] and a social network dataset in which users follow items (implicit user feedback). However, these approaches do not distinguish auxiliary domains from a target domain, and they jointly optimize a tensor by sharing the user latent factors across domains, similar to the approach in [Singh and Gordon, 2008]. Consequently, these approaches may be less effective for a target domain, especially when the auxiliary domain is dense [Pan et al., 2010]. Furthermore, tensor decomposition can be computationally expensive (the cost function in [Hu et al., 2013] has four independent variables to be minimized individually) and additional care must be taken in order to address the inconsistent number of items across domains.

Meta-Path-based Cross-Domain Approaches

Approaches based on identifying and using meta-paths [Sun and Han, 2012] in a heterogeneous network to improve recommendation accuracy have also been proposed in the literature. A meta-path is defined as "an abstraction of a network path in a heterogeneous

network into a sequence of edge types" [Burke et al., 2014]. Navigation of a meta-path from a node includes collection of all destination nodes reachable by following edges of the appropriate type. Examples of applications that used meta-paths in heterogeneous networks to improve target recommendation accuracy include work by Burke et al. [2014]. The authors extended the Weighted Hybrid of Low-Dimensional Recommenders (WHyLDR) proposed for recommending tags in social tagging systems [Gemmell et al., 2012] to more complex networks using the concept of meta-paths. The key insight of the WHyLDR design is that a complex network structure can be viewed as a set of two-dimensional (2-D) projections from one type of nodes to nodes of another type, and that a recommendation can be made by combining results of recommendation components built from these low-dimensional projections. Although any CF approach can be used on the 2-D projections of meta-paths, the work in [Burke et al., 2014] used neighborhood-based CF approaches. The authors evaluated their approach on a Bibsonomy dataset with implicit user preferences and reported that the meta-path-based approach to recommendation in heterogeneous networks yields improvements in accuracy and diversity. In a later extension, Vahedian [2014] studied the effect of the use of successively longer meta-paths on the performance of the WHyLDR approach. The author concluded that longer meta-paths may have better precision and recall. Furthermore, the authors generalized the approach into a component-based hybrid model whose components can be reused for multiple recommendation tasks in order to reduce the overall computational complexity of the recommendation algorithm.

Link Prediction-based Approaches

Yang et al. [2012] identified the co-author recommendation problem in the *DBLP* scientometric dataset as a link prediction problem in a heterogeneous network. They suggested that many real-world networks are heterogeneous and can, therefore, be leveraged to improve prediction performance as compared to the homogeneous network. The authors proposed an approach to quantitatively represent the flow of information in a heterogeneous network;

flow value was used as one feature in the link prediction problem.

Cold-Start User Problem

The literature does not contain an extensive study of cross-domain approaches in a cold-start setting. The work by Hu et al. [2013] was one of the first attempts to study the usefulness of cross-domain approaches in order to address the cold-start user problem, also referred as the unacquainted world scenario. In another study, conducted by Sahebi and Brusilovsky [2013], the authors aggregated user profiles from multiple domains and used traditional neighborhood-based approach to study the usefulness of cross-domain approaches in order to address the cold-start user problem.

User Overlap

Most approaches for cross-domain recommender systems proposed in the literature require some overlap (either between users, or between items, or both) in order to aggregate/transfer knowledge between domains [Winoto and Tang, 2008; Berkovsky et al., 2007a; Cremonesi et al., 2011; Pan et al., 2010; Shapira et al., 2013; Singh and Gordon, 2008; Hu et al., 2013; Berkovsky et al., 2007b], with the exception of works in [Li et al., 2009a,b; Gao et al., 2008] that require no overlap between users or items. Cremonesi et al. [2011] studied the effect that the degree of user and item overlap, between source and target domains, has on the performance of the cross-domain approaches proposed in [Cremonesi et al., 2011], whose goal was to recommend items in the source domain to users in the target and vice versa. Cremonesi and Quadrana [2014] showed that the *CodeBook* constructed by randomly generating ratings demonstrates better recommendation performance as compared to the *CodeBook* constructed according to work in [Li et al., 2009a]. They suggested that knowledge transfer may not be possible without overlapping users or items between source and target domains.

Given this literature review of related approaches and applications, the readers should note that a drawback of the cross-domain approaches in the literature is that these approaches make certain assumptions that restrict their usage for various real-world problems. For example, some studies in the literature were conducted with "simulated" cross-domain data: Berkovsky et al. [2007a] simulated a cross-domain framework by partitioning a movie dataset based on genre; Pan et al. [2010] used various movie rating datasets as different domains, and other researchers transferred user rating knowledge between movies and books because they have similarity in genre and many movies are based on books. [Li et al., 2009a,b]. Furthermore, a majority of cross-domain approaches proposed in the literature assumed explicit (Boolean or numerical) user preferences in the source and target domains [Berkovsky et al., 2007a; Li et al., 2009a,b; Pan et al., 2010; Singh and Gordon, 2008; Winoto and Tang, 2008; Loni et al., 2014; Gao et al., 2008]. To the best of this author's knowledge, no prior work has extended Adsorption and MF approaches to address a cross-domain recommendation problem with various types of dissimilar items and implicit user feedback in all domains. Furthermore, no prior work was found that experimentally evaluates the effectiveness of a cross-domain approach whose goal is to improve target recommendation accuracy under various user overlap scenarios.

3.2.3 Determining Weights between Domains

One important aspect of a cross-domain recommender system is the choice of weights used to control the amount of knowledge aggregated from source to target. However, determining relatedness between domains, and consequently estimating the weights, is a challenging research problem. Berkovsky et al. [2007b] proposed two ways, based on textual descriptions of items or on rating information, respectively, to compute inter-domain correlations. However, item descriptions are not always available and rating correlation between domains with implicit user feedback cannot be computed. Hu et al. [2013] optimized weights corresponding to each domain using Genetic Algorithms. However, given the large search space

and high computational complexity of the algorithms, these approaches are not suitable for large-scale datasets.

Yang et al. [2012] studied information propagation in a heterogeneous network in order to predict author collaborations. They suggested that information flow is asymmetric, and they quantitatively represented the correlation between two networks i and j using the conditional probability (CP) $Pr(i|j)$ (equivalent to the edge correctness metric). The CP value was used to compute the information flow in heterogeneous networks, which was used as a feature to predict collaborations. Motivated by work in Yang et al. [2012], this study used the CP value of user neighborhoods computed between source and target domains as the relative weight between source and target domains. To the best of this author's knowledge, the applicability of neighborhood alignment score as knowledge aggregation parameters has not previously been studied in the literature.

Chapter 4

Problems and Proposed Approaches in Single-Domain Setting

Section 4.1 in this chapter motivates the need for study in the single-domain setting and identifies several goals of this work. Section 4.2 details several ways to construct user neighborhoods for Adsorption algorithm and introduces application of community detection for neighborhood construction. Finally, Section 4.3 summarizes main contributions of this work in the single-domain setting.

4.1 Motivation for the Single-Domain Study

Numerous single-domain collaborative filtering (CF) approaches with unique characteristics have been proposed in the literature in order to generate item suggestions to users. However, with the ever increasing size of datasets (in terms of users, items, and user-item interactions), it is impractical to try every CF approach and choose the approach that offers optimum accuracy for the application domain at hand. *Therefore, one goal of the study in the single-domain setting was to investigate if knowledge about a data domain, specifically knowledge about how links are formed in the domain (based on close connections -*

resulting in strong local neighborhoods or based on loose connections - resulting in useful global information), or characteristics of the data (e.g., dense or sparse feedback matrices) can be useful in identifying the appropriate CF approach to use, Adsorption or MF, in that particular domain [Parimi and Caragea, 2015d].

Among CF approaches for recommender systems, neighborhood-based techniques have advantages such as small number of model parameters, low computational complexity, and intuitiveness of the recommendations [Bell and Koren]. These advantages make neighborhood-based approaches a suitable choice for large-scale recommender systems. However, it has been suggested in the literature that the neighborhood-based approaches have three important requirements that can significantly impact the accuracy of a recommender system: *a*) normalization of data, *b*) computation of similarity weights, and *c*) selection of neighbors [Desrosiers and Karypis, 2011; Bell and Koren; Su and Khoshgoftaar, 2009]. Because the work in this dissertation on single-domain approaches use implicit user preferences, data normalization does not apply; however, based on prior experience, the other two requirements namely, computing similarity score and neighborhood selection, also known as neighborhood construction, have proved pivotal in improving the recommendation accuracy. *Thus, a second goal of this work was to investigate if domain and data characteristics (e.g., frequent versus infrequent users) can influence the choice of the neighborhood construction approach* [Parimi and Caragea, 2015d]. To gain insights into this, four custom similarity functions (or equivalently, four ways to construct neighborhoods) were used, as explained in Section 4.2.1.

One problem with the custom similarity functions used to construct user neighborhoods is that these approaches do not consider the underlying community structure that inherently exists among users. Intuitively, the community structure (a principled way of organizing vertices in a graph into densely connected clusters) in the user-user graph may be helpful in constructing better user neighborhoods compared to the custom similarity functions. *In order to understand the usefulness of community detection, this work studied the application*

of modularity-based community detection [Blondel et al., 2008] to select user neighborhoods for use with the Adsorption algorithm [Parimi and Caragea, 2014]. This corresponds to the third goal of the work in single-domain setting.

Finally, dataset sizes are continuously increasing for recommender systems because more users and items are added to the application domain and because web applications collect a lot of information about user preferences for items, each day. One challenge associated with such large datasets is the computational complexity of recommendation algorithms. Although the algorithms are expected to scale and handle the large number of users and items, the question about how much user information should be used while computing recommendations (without degrading accuracy) is intriguing because the runtime of an algorithm increases with the number of user-item preferences in the dataset. *Motivated by the need to decrease computational costs, and studies that have indicated that short user histories are more effective than longer histories when recommending ads [Yan et al., 2009], the fourth goal of this study was to investigate how the size of the user history influences the performance of the algorithms for domains with timestamp information* [Parimi and Caragea, 2015d].

In order to address the aforementioned goals (i.e., to compare the Adsorption performance with MF performance and to understand the influence of data and domain characteristics in order to select user neighborhoods), four weight-based approaches, described in Section 4.2.1, and a community detection-based approach, described in Section 4.2.2, were studied to construct neighborhoods for Adsorption algorithm.

4.2 Neighborhood Construction for Adsorption

The Adsorption algorithm proposed by Baluja et al. [2008], described in Section 2.3, can be used in a recommendation framework to propagate user preferences for items on the user-user graph. Conceptually, in the Adsorption algorithm, the transition can be made to all neighbor

users u of a user v when propagating preference information. However, in practice, it is often computationally intractable to use all neighbors, because of the large neighborhood sizes, in addition to the large number of users and items in the dataset. Furthermore, it has been suggested that using the entire user neighborhood, as opposed to only nearest neighbors, does not always result in great improvements in recommendation accuracy [Sarwar et al., 2001, 2002; Desrosiers and Karypis, 2011]. Therefore, a popular solution to the problem of large user neighborhoods has been to use the k-nearest neighbors. The nearest neighbors for a user can be selected using similarity or weight between users [Sarwar et al., 2001, 2002; Desrosiers and Karypis, 2011; Adomavicius and Tuzhilin, 2005].

Similar to this line of work in neighborhood approaches, this study restricts the Adsorption random-walk from a user v to only its nearest neighbors u. In general, the k-nearest neighbors u of a user v can be selected based on the strength of the edge between users v and u in the graph G. Therefore, the problem of finding informative neighbors reduces to the problem of defining informative edge weights. In addition to cosine-similarity, three weighting schemes were used in order to understand which scheme works the best in each of the three domains.

For implicit feedback datasets, the most straightforward way to define edge weights between two users v and u is based on the number of common items preferred by users v and u. However, this approach does not take into account the fact that some users are generally frequent users. Intuitively, this characteristic of the users in the dataset (i.e., frequent or infrequent users) can be important for some recommendation applications. In order to capture this intuition, the straightforward approach to computing edge weights (i.e., *Variant 1*), was compared, to two other variants, *Variant 2* and *Variant 3*, that take into account frequency information. The three variants were also compared to cosine-similarity (*Variant 4*). The four variants are described in detail in the following section.

4.2.1 Weight-based Neighborhood Construction Variants

- **Variant 1 (Baseline):** The weight between two users v and u was defined as the number of common items (i.e., $w_{uv} = \#common\ items(u, v)$), preferred by the two users. The k-nearest neighbors were selected according to the weights.

- **Variant 2:** The weight between two users was defined as the number of common items preferred by two users v and u, normalized by the sum of total items each user preferred (i.e., $w_{uv} = \frac{\#common\ items(u,v)}{\#items(u) + \#items(v)}$). Intuitively, two users that have many preferred items in common, but are both very frequent users, should be deemed less similar than two users that have many items in common, but are not very frequent users. Likewise, two infrequent users that have a significant number of common preferred items should be considered similar and given a higher edge weight than the weight associated with the edge between frequent users with a similar number of common preferred items. As for baseline, the neighbors were selected based on weights.

- **Variant 3:** This variant was motivated by the use of clustering to create neighborhoods in CF approaches [Sarwar et al., 2002]. In this variant, the users were explicitly categorized into two groups (clusters), specifically *frequent* and *infrequent* users, based on the median of the number of items that each user preferred because this approach was computationally less expensive as compared to grouping all users into clusters using a clustering algorithm. As in the baseline, the weight between two users in the graph was defined as the number of common items (i.e., $w_{uv} = \#common\ items(u, v)$), preferred by the two users u and v. However, as opposed to the baseline, here the k-nearest neighbors were chosen from the *frequent* users group, according to the weight values. The intuition behind this variant was that it is difficult to generate recommendations to *infrequent* users because of their relatively small behavioral history. By categorizing users into two groups and using the *frequent* group to select neighbors, more items will be recommended to an *infrequent* user, thanks to the relatively large browsing history of their neighboring *frequent* users.

- **Variant 4:** Each user was represented as a vector of preference counts, and the weight between two users v and u was defined as the *cosine* of the angle between their corresponding vectors i.e., $w_{uv} = \frac{\vec{u}.\vec{v}}{\|u\| \|v\|}$.

Note that in all the above variants, it is possible that many of the neighbors of a user have the same weight. To resolve ties, users that have identical weights were randomly selected to fill in the k-nearest neighbor positions.

4.2.2 Community Detection-based Neighborhood Variant

The interest in community detection techniques is ever increasing because many real-world networks such as World Wide Web (WWW), social networks, and citation networks, display a community structure. Communities, also called *clusters*, are groups of vertices in a graph that have strong connections with other vertices in the same group as compared to vertices in other groups. The vertices in a community generally share common properties and play similar roles in the graph [Fortunato, 2010]. For example, in a collaboration network, communities may reveal groups related to similar research; social network communities may reveal real social groupings. Algorithms for community detection aim to find clusters in a graph such that the amount of interaction within a cluster is more than the interaction outside the cluster. Intuitively, the community structure in a network may be helpful in constructing better user neighborhoods as compared to the neighborhoods constructed with custom similarity functions that use only user-user similarity.

Consider the example of a community structure in a co-author network shown in Figure 4.1. As shown in the figure, two communities exist in the graph. The first community corresponds to author collaborations in the research area of *Data Mining and Machine Learning*, while the second community corresponds to collaborations in the research area of *Algorithms*. For the author circled by red dots (referred to as **X**), in Figure 4.1, the custom similarity functions may choose the author circled by green dots (referred to as **Y**) as neighbor. In such a scenario, the Adsorption algorithm recommends the co-authors

of **Y** to author **X**. However, the co-authors of **Y** have shown interest in the *Algorithms* research area according to the community structure. Therefore, the co-authors of **Y** may not represent good recommendations for author **X**, given his interest in the research area of *Data Mining and Machine Learning*. Using community detection, authors who are in the same community as the author X will be selected as his neighbors. This might avoid recommending to author **X**, co-authors who have dissimilar research interests compared to the co-authors of **X**, a phenomenon known as *concept drift*.

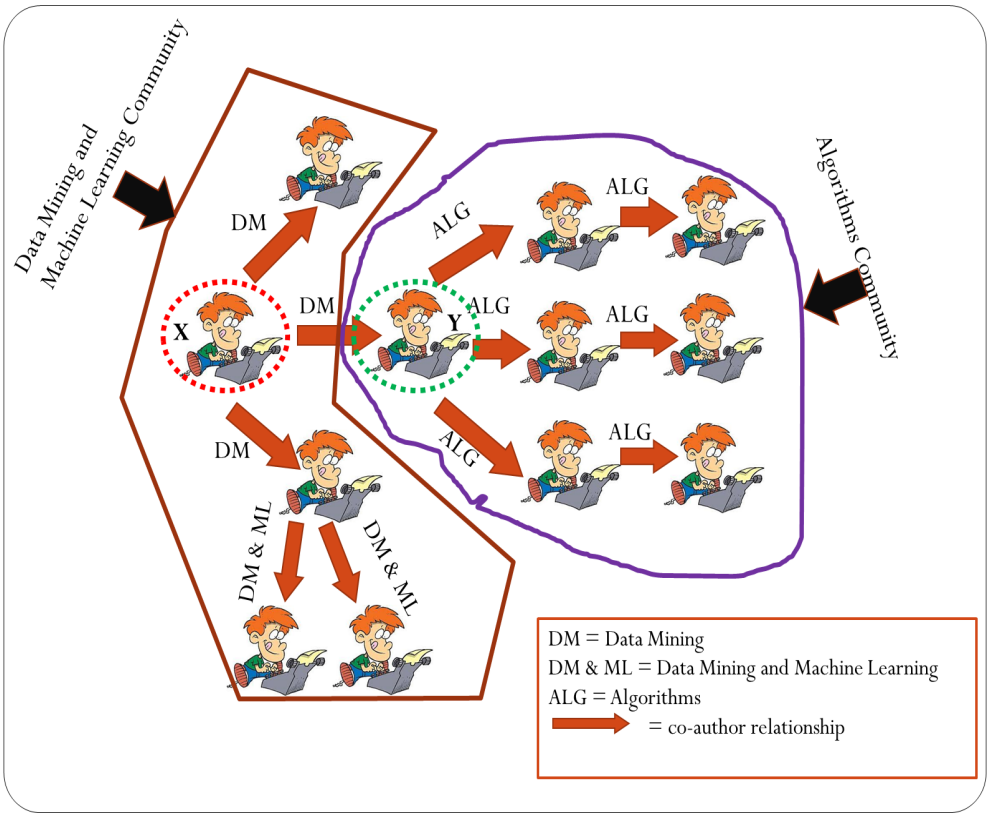


Figure 4.1: Community structure in the co-author network.

Given this motivation, the rest of this section discusses the background of discovering communities from graphs and construction of nearest neighborhoods for the Adsorption algorithm using user communities.

Community Discovery in Graphs

Detecting communities from networks is a well-known problem and several approaches have been proposed in the literature to address this problem [Fortunato, 2010]. The basics of community detection based on modularity optimization are first explained, and later the community detection algorithm used in this work is explained.

Modularity Optimization: The work by Newman [2004] is one of the first works aimed at detecting communities in graphs by optimizing modularity. Modularity measures the quality of partitions from a community detection algorithm; the modularity of a partition is a value between -1 and 1 and measures the density of edges within a community as compared to edges between different communities [Fortunato, 2010; Newman, 2004]. Given a graph $G = (V, E, w)$, where V is the set of nodes, E is the set of edges between two nodes, and $w : E \rightarrow \mathbb{R}^+$ is a function that assigns a positive weight to edges, Equation (4.1) shows how modularity is computed for a community structure:

$$Q = \frac{1}{2m} \sum_{i,j \epsilon V} \left[w_{ij} - \frac{k_i k_j}{2m} \right] \delta\left(c_i, c_j\right) \tag{4.1}$$

In the above equation, w_{ij} corresponds to the weight on the edge between nodes i and j in the network, $k_i = \sum_j w_{ij}$ is the sum of the weights of the edges that go through node i, c_i is the community to which the node i is assigned, the δ-function yields 1 if nodes i and j belong to the same community and 0 otherwise, and $m = \frac{1}{2} \sum_{ij} w_{ij}$.

Modularity is the most used and best known quality function for community detection in networks. Most algorithms for community detection work under the assumption that high values for modularity indicate good partitions and thus maximize Q [Fortunato, 2010]. Given the large real-world networks (with millions of nodes and edges) and numerous possibilities to partition a graph, an exhaustive optimization of Q becomes intractable. Also, it has been proved recently by Brandes et al. [2008] that modularity optimization is NP-complete (i.e., there probably does not exist a polynomial time algorithm that can find a global maximum

for Q). Therefore, algorithms focused on modularity optimization are based on approximate optimization methods such as greedy algorithms, simulated annealing. In this work, the Louvain method proposed by Blondel et al. [2008] was used to identify communities from a user-user graph G. The approach is described below.

Louvain Method for Community Detection: The Louvain method proposed by Blondel et al. [2008] is a heuristic method, based on modularity optimization, for community detection. This method, which can be applied to large weighted graphs, is known to outperform many other community detection algorithms in terms of computational time. The Louvain method is a two-phase iterative process. In the first phase, all nodes in the graph are assigned to different communities. At each node i, the algorithm computes the gain in modularity ΔQ, given by Equation (4.2), when node i is removed from its community and is placed in the community of a neighbor node j. Node i is assigned to the community for which the gain in modularity is positive and highest. At the end of this phase, the algorithm identifies the first-level partitions. In the second phase, a new network is constructed from the communities identified in the first phase; the nodes in the new network are the communities identified in the first phase. Two nodes in the new network are connected if there is at-least one edge between nodes in the corresponding communities from the first phase. These two steps are repeated iteratively until there is no change in modularity, yielding hierarchical levels of the original network.

$$\Delta Q = \left[\frac{\sum w_{in} + k_{i,in}}{2m} - \left(\frac{\sum w_{tot} + k_i}{2m}\right)^2\right] - \left[\frac{\sum w_{in}}{2m} - \left(\frac{\sum w_{tot}}{2m}\right)^2 - \frac{k_i}{2w}\right] \tag{4.2}$$

In Equation (4.2), $\sum w_{in}$ is the sum of the weights of links inside the community C where node i is placed, w_{tot} is the sum of the weights of the links incident to nodes in C, $k_{i,in}$ is the sum of the weights of the links from i to nodes in C, and m is the sum of the weights of all the links in the network. The reader is referred to the work in [Blondel et al.,

2008] for more information on the Louvian method.

Neighborhood Formation based on Community Detection

Given a dataset with $< userID,\ itemID,\ preferenceCount >$ information, the first task was to construct a graph $G = (V, E, w)$ where V is the set of users, E is the set of edges between users, and $w : E \rightarrow \mathbb{R}^+$ is a function that assigns a positive weight to edges. In order to compute weights on the edges of G, a similarity weighting scheme based on common item counts was used (specifically, Variant 2 described in Section 4.2.1). The graph G was then given as input to the community detection algorithm, and output from the algorithm was used to construct k nearest neighbors for users. Detailed steps to construct neighborhoods using the community detection approach, referred to as *Variant 5*, are given below.

Variant 5 (Community Detection): In this variant, users were clustered using a community detection algorithm proposed in [Blondel et al., 2008], and the clusters were used to compute neighborhoods. The following steps were conducted:

- Compute the weight between every pair of users (u, v) in the dataset according to the weight equation for Variant 2 (i.e., $w_{uv} = \dfrac{\#common\ items(u,v)}{\#items(u) + \#items(v)}$, $w_{uv}\ \epsilon\ [0, 0.5]$). In order to use the weights with the community detection algorithm, the range for w is changed from $[0, 0.5]$ to $[0, 1]$ by normalizing the weights between every user pair (u, v) by the highest weight value for the user u.
- Extract every user pair (u, v) for which the computed weight w_{uv} is above a threshold t $(0 \leq t \leq 1)$. The threshold was used to control the number of edges in the user-user graph. The number of edges are maximum when $t = 0$ (there will be an edge between every user pair (u, v) for which $w_{uv} > 0$) and the number of edges decreases as t increases.
- Construct a user-user graph from the user pairs extracted in the previous step. The weight w_{uv} represents the weight on the edge between users u and v.

- Apply the community detection algorithm proposed by Blondel et al. [2008] on the graph computed in the previous step.
- The community detection algorithm outputs a list of users for each identified community (output from the first phase). The number of communities identified by the algorithm varies from dataset to dataset and is determined at runtime by the algorithm based on modularity. In order to construct the k-nearest neighbors for a user v, top k users u were selected from the community of user v according to weights w_{uv}. For example, if $u1$, $u2$, $u3$, $u4$, $u5$ are the 5-nearest neighbors for user v, then users $u1$, $u2$, $u3$, $u4$, $u5$ should belong to the same community as user v and $w_{u1v} \geq w_{u2v} \geq w_{u3v} \geq w_{u4v} \geq w_{u5v}$.

Similar to other variants for constructing neighborhoods, it is possible that many of the neighbors of a user have the same weight. Ties were resolved by randomly selecting k users that have identical weights, following the constraints for this variant.

4.3 Contributions of the Single-Domain Study

This section summarizes the contributions of this work in the single-domain setting.

4.3.1 Comparison of Weight-based Neighborhoods

One goal of this study in the single-domain setting was to understand how the user neighborhoods constructed using custom similarity functions compare to each other. Towards this goal, the performance of the Adsorption algorithm with neighborhoods constructed using the weight-based neighborhood variants, described in Section 4.2.1, were compared for three implicit feedback datasets from different domains.

4.3.2 Evaluation of Community Detection-based Neighborhoods

Another goal of this study was to understand the usefulness of user communities to construct neighborhoods for the Adsorption algorithm. Towards this goal, the performance of the Adsorption algorithm with neighborhoods constructed using modularity-based community detection techniques was evaluated on two real-world implicit feedback datasets, one that is known to exhibit a strong community structure and another one that does not exhibit such strong community structure.

4.3.3 Comparison of Adsorption and Matrix Factorization

In order to understand which approach is better between Adsorption and Matrix Factorization (MF), two-state-of-the-art collaborative filtering approaches for implicit feedback data, the performance of the Adsorption algorithm was compared with the performance of MF on three implicit feedback datasets from different domains.

4.3.4 Influence of Length of User Histories on the Performance

Another goal of this study was to understand if short user histories can be successfully used to decrease computational time of recommendation algorithms without degrading the recommendation performance for large data domains with timestamp information. To accomplish this goal, the performance of Adsorption and MF with short user histories (the user feedback for items for a short period of time) was compared to the performance from these algorithms with long user histories (the user feedback for items for a longer period of time) for two datasets.

4.3.5 How Domain Knowledge Can Help

A final but important goal of this study in the single-domain setting was to understand the usefulness of knowledge regarding the data domain (e.g., close connections versus loose con-

nections among users) or characteristics of the dataset (e.g., density of the feedback matrix) in selecting the best approach to construct user neighborhoods for Adsorption. To accomplish this goal, the results from the experiments related to contributions in Sections 4.3.1 and 4.3.2 were analyzed in conjunction with the domain and dataset characteristics. Intuitively, the characteristics of a data domain, specifically, if user links result in strong local neighborhoods or are better captured as global user information, and the characteristics of users in the dataset, such as frequent or infrequent users based on the frequency of items preferred by the users, can be important for some recommendation applications and can be helpful to choose the neighborhood construction approach to use in each domain.

Similarly, usefulness of domain knowledge and dataset characteristics to select the most suitable CF approach for use in a particular recommendation problem was also studied by analyzing results from experiments related to contribution in Section 4.3.3 in conjunction with the domain and dataset characteristics. Because the performance of Adsorption increases as the amount of information about a user increases [Baluja et al., 2008] and also the relatively smaller computational complexity of neighborhood approaches compared to latent factor approaches [Hu et al., 2008; Koren et al., 2009; Sarwar et al., 2001, 2002], intuitively, Adsorption seems to be a suitable choice for dense datasets, assuming Adsorption's performance is also competitive with the performance of MF on such datasets. However, for sparser datasets, this study investigated if one approach is better than another, and attempted to understand if knowledge about the domain (specifically if user links result in strong local neighborhoods or are better captured as global user information) can be used to select one approach versus the other. Intuitively, Adsorption may perform better for datasets with strong neighborhood relation among users, given that the algorithm works by propagating preference information to nearest neighbors. However, MF may perform better for application domains in which items are more dynamic (new items are constantly added and items are outdated quickly), such as news recommendation and ad recommendation. In such domains, using item preferences from all the users and propagating information

globally may be a better approach. MF, a latent factor approach, captures user similarity globally by representing all users in a common latent space.

Chapter 5

Problems and Proposed Approaches in Cross-Domain Setting

Section 5.1 motivates the need for this study in the cross-domain setting and introduces several limitations of existing approaches in Section 5.1.1, and later provides an overview of proposed approaches and goals of this work in Sections 5.1.2 through 5.1.5. Section 5.2 formally defines the cross-domain recommendation problem addressed in this work. Section 5.3 introduces Adsorption-based approaches proposed in this work for aggregating knowledge from several domains. Section 5.4 describes the approach for transferring knowledge from multiple domains through the use of MF for implicit feedback. Finally, Section 5.5 summarizes the main contributions of this dissertation in the cross-domain study.

5.1 Motivation for the Cross-Domain Study

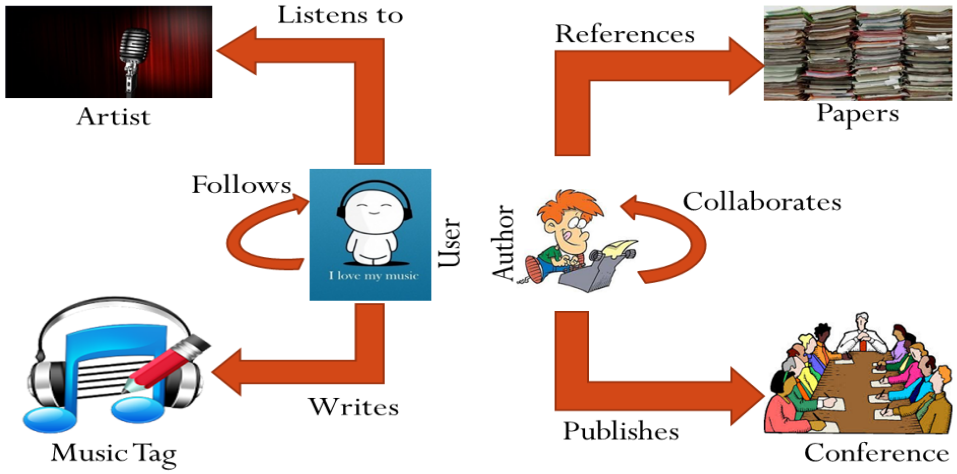
Most CF approaches use user preferences (*explicit*-such as ratings, like/dislike or *implicit*-such as, clicks, buys) from one domain and generate personalized item suggestions from the same domain. For example, Netflix suggests movies by analyzing existing user ratings for movies. However, in some real-world scenarios, user preferences for various types of

items from multiple domains are available. For example, in the music site *Last.FM*, summarized in Figure 5.1a, users listen to *artists*, make *friends* with other users, and annotate music with *tags*. In a scientometric datasets such as *DBLP*, summarized in Figure 5.1b, authors collaborate with other authors, publish in conferences, and reference other articles. Each recommendation problem (i.e., recommending *artists*, *friends*, and *tags* in the case of Last.FM and *authors*, *conferences*, and *articles* in the case of the DBLP) can be seen as a domain. User preferences from several domains represent valuable auxiliary information and can be used to enhance user personalization experience, such as, improving the accuracy of recommendations in one domain or offering novel, diverse, and serendipitous recommendations [Winoto and Tang, 2008; Berkovsky et al., 2007a; Shapira et al., 2013; Cantador and Cremonesi, 2014; Li et al., 2009a; Pan et al., 2010]. This research area is popularly known as *cross-domain* recommender systems.

This work on the cross-domain recommender systems in this dissertation focused on *approaches to aggregate/transfer knowledge from several auxiliary data domains, known as source domains, in order to improve recommendation accuracy in one domain, known as the target domain.*

5.1.1 Approaches Used in Prior Work and Limitations

One naive way to use knowledge about user preferences from multiple domains is to aggregate data about users and items from all domains into one user-item preference matrix and use any standard CF algorithm to generate recommendations [Winoto and Tang, 2008]. However, such models generally recommend items from the domain in which users have preferred many items [Cremonesi et al., 2011; Hu et al., 2013; Cantador and Cremonesi, 2014]. Furthermore, implicit user preferences across domains may mislead the model since implicit feedback can have a varied range [Hu et al., 2008]. In order to overcome these problems for cross-domain recommender systems, several neighborhood-based CF approaches have been proposed in the literature [Berkovsky et al., 2007a,b; Shapira et al., 2013]. The main idea



(a) Last.FM Music Site (b) DBLP Scientometric Dataset

Figure 5.1: A cross-domain recommendation problem in (a) the Last.FM music site and (b) the DBLP scientometric dataset.

of these approaches is to mediate (import and aggregate) user modeling information, such as user neighborhoods and user-user similarity scores, from source domains and the target. More recently, approaches based on MF have been proposed for cross-domain recommender systems because of the superior performance of latent factor model-based techniques as compared to neighborhood-based approaches [Adomavicius and Tuzhilin, 2005; Hu et al., 2008]. The underlying idea of these approaches is to use the latent factors from source domains as a bridge to transfer knowledge from source domains to target domain. Specifically, Li et al. [2009a,b] transferred the cluster level rating pattern from a dense source domain to a target. Pan et al. [2010] proposed an approach that addressed the "what to transfer and how to transfer" question in order to overcome negative transfer for transfer learning. Singh and Gordon [2008] collectively factorized data from multiple domains by sharing the user (or item) factors across domains. More details about the approaches in literature for cross-domain recommender systems are reviewed in Section 3.2.

However, many of these approaches make certain assumptions about the data domains that restrict their usage for some real-world applications. For example, many approaches as-

sume the same type or similar items in related domains [Berkovsky et al., 2007a,b; Singh and Gordon, 2008; Pan et al., 2010]; similar user preferences for items in related domains [Cremonesi et al., 2011; Li et al., 2009a,b]; the existence of a fixed number of source domains to transfer knowledge [Pan et al., 2010]; or dense source domains [Li et al., 2009a,b]. Furthermore, many of these works also assume that user preferences for items are given as explicit feedback in all domains [Winoto and Tang, 2008; Berkovsky et al., 2007a,b; Cremonesi et al., 2011; Li et al., 2009a,b; Pan et al., 2010; Singh and Gordon, 2008]. In practice, items across domains may be of different types and share no similarity, and user preferences may be different in related domains (two users who like similar movies may have entirely different interests in electronics). Furthermore, user preferences may correspond to implicit feedback in all domains (because this type of data is more common in real-world applications), the source domains may be sparse, similar to target domains, and, in general, one or more source domains can exist.

Therefore, the goal of this work was to design novel approaches to address cross-domain recommendation problems that do not meet the aforementioned assumptions in the literature. Specifically, this work on cross-domain approaches for recommender systems assumed one or more source domains with implicit user feedback in all domains. This work also assumed that items are of different types (e.g., artists and friends) across domains and share no similarity (a more general case that has not been addressed in prior work), and user preferences for items are different in various domains. Furthermore, no assumptions about the density of datasets were made. Specific goals of this research are described in the next section.

5.1.2 Neighborhood-based Cross-Domain Approaches

The first goal of this study was to design neighborhood-based approaches to address the aforementioned cross-domain recommendation problem. Specifically, two knowledge aggregation approaches based on a state-of-the-art neighborhood-based approach for *implicit* feedback, called Adsorption [Baluja et al., 2008], were proposed [Parimi and Caragea, 2015e]. One

approach performs aggregation of neighborhoods (WAN) from the source and target domains using weights, and then constructs nearest neighborhoods that are used with the Adsorption algorithm to recommend target items (Section 5.3.1). The second approach, aggregation of recommendations (WAR), first uses the nearest neighborhoods constructed in the source and target domains with the Adsorption algorithm to generate target recommendations and, later aggregates the recommendations computed from various nearest neighborhoods using weights (Section 5.3.2). The amount of knowledge transferred from each source domain to the target is controlled through the use of weights. *Given the two proposed aggregation approaches, another goal of this study was to evaluate the effectiveness of the proposed WAN and WAR approaches in improving recommendation accuracy as compared to single-domain approaches.*

Experiments with WAN and WAR approaches showed that the weights that control the amount of knowledge aggregated from source domains and the target are critical for the success (ability to overcome negative transfer and improve the target recommendation accuracy) of the algorithms. Although they can be carefully tuned using cross-validation techniques [Li et al., 2009a,b; Pan et al., 2010; Parimi and Caragea, 2015e], or optimization algorithms can be used to find the optimal weight assignment on the source domains [Hu et al., 2013], these approaches are not practical for large-scale recommender systems due to high computational costs. *Instead, an approach that can capture the relationship between two domains quantitatively and use that information to estimate the weights would be preferable, and this was another goal of this research.* Along these lines, a solution inspired by the use of the *edge correctness* metric that measures the percentage of correctly aligned edges to evaluate network alignment algorithms in bioinformatics [Yang et al., 2012; Singh et al., 2008] was proposed in this work. *Given the proposed approach to compute weights, another goal of this cross-domain study was to validate the proposed way to compute weights between a pair of source and target domains, and their usage for knowledge aggregation* [Parimi and Caragea, 2015b].

5.1.3 Cold-Start Problem

One common problem in real-world recommender systems is the cold-start user problem, in which no preferences are available for some users in a domain of interest (a.k.a., the unacquainted world scenario) [Hu et al., 2013]. However, in a cross-domain setting, those users may have expressed preferences in other domains. Intuitively, preferences of cold-start users in auxiliary domains can be used to recommend items from their unacquainted target domain. *Therefore, the fifth goal of this work was to investigate the effectiveness of the proposed WAN and WAR approaches in order to address the unacquainted world scenario* [Parimi and Caragea, 2015b].

5.1.4 Matrix Factorization-based Cross Domain Approaches

Another goal in this study was to design a cross-domain approach based on MF for the recommendation problem considered. Because MF approaches have gained significant of popularity for single-domain recommendation problems, a cross-domain approach based on MF to improve the recommendation accuracy in a target domain was also proposed. This approach, known as Cross-domain Implicit-feedback Matrix Factorization (CIMF) [Parimi and Caragea, 2015a], can handle the aforementioned cross-domain recommendation problem and is considered to be a knowledge transfer approach because it transfers the user latent factor information from source domains to the target domain. The underlying assumption is that although the items are of different types and user preferences for items vary across domains, there is some latent user information that is common for the source and target domains and can be shared between the two. Therefore, the objective was to discover domain independent semantic user concepts from the related source domains and transfer them to the target domain. Given the differences in items and user preferences across domains, only the user latent factors were transfered. Intuitively, the approach should be able to prevent negative transfer, to some extent, because it only requires the user latent factors in source and target domains to be similar through the regularization terms. *Given*

the proposed CIMF approach, another goal of this study in the cross-domain setting was in order to evaluate the effectiveness of the CIMF approach to address the cross-domain recommendation problem.

5.1.5 User Overlap Scenarios

Most cross-domain approaches in the literature require some overlap (either between users, or between items, or both) in order to aggregate/transfer knowledge between domains [Winoto and Tang, 2008; Berkovsky et al., 2007a; Cremonesi et al., 2011; Pan et al., 2010; Shapira et al., 2013; Singh and Gordon, 2008; Hu et al., 2013; Berkovsky et al., 2007b]. Similarly, the WAN and WAR approaches [Parimi and Caragea, 2015e] and the CIMF approach [Parimi and Caragea, 2015a] also require a partial user overlap between source and target domains (item overlap is not possible because items are assumed to be different across domains). Motivated by the study in [Cremonesi and Quadrana, 2014], which indicated that transfer of knowledge is not possible when no overlap exists between source and target domains, and the study in [Cremonesi et al., 2011], which indicated that the degree of overlap among domains strongly influences the accuracy of cross-domain recommendations, *the effect of user overlap between domains on the performance of the WAN and WAR approaches* [Parimi and Caragea, 2015b] *and the CIMF approach* [Parimi and Caragea, 2015a] *was studied. This corresponds to the another goal of this work in the cross-domain setting.*

The rest of this chapter primarily focuses on the specifics of the cross-domain recommendation problem addressed in this work (Section 5.2) and on the Adsorption-based (Section 5.3) and MF-based (Section 5.4) cross-domain approaches proposed to handle the aforementioned cross-domain problem.

5.2 Formal Problem Definition

The cross-domain recommendation problem addressed in this work is defined using a notation similar to the notation used in [Cremonesi et al., 2011] and [Fernández-Tobías et al., 2012]: in this problem setting, the existence of one target domain $\mathbf{T}$ with implicit feedback and m auxiliary source domains $\mathbf{S}^l$ (where $l \in [1, m]$), also with implicit feedback, was assumed. Let U_t, U_l be the sets of users such that $U_t \cap U_l \neq \emptyset$, I_t, I_l be the sets of items such that $I_t \cap I_l = \emptyset$ and are dissimilar, in domains $\mathbf{T}$ and $\mathbf{S}^l$, respectively. The objective is to improve the recommendation accuracy in the target domain $\mathbf{T}$ by exploiting user-item preferences from the m source domains.

When considering multiple domains for CF recommender systems, the set of users and items across multiple domains may or may not overlap. Cremonesi et al. [2011] identified four overlap situations in which a cross-domain recommendation task can be conducted: a) no overlap between users and items in target $\mathbf{T}$ and sources $\mathbf{S}^l$, i.e. $U_t \cap U_l = \emptyset \ \wedge \ I_t \cap I_l = \emptyset$ for all $i \in [1, m]$; b) some overlap between users in target $\mathbf{T}$ and sources $\mathbf{S}^l$, i.e. $U_t \cap U_l \neq \emptyset$ for all $i \in [1, m]$; c) some overlap between items in target $\mathbf{T}$ and sources $\mathbf{S}^l$, i.e. $I_t \cap I_l \neq \emptyset$ for all $i \in [1, m]$; and d) some overlap between users and items in target $\mathbf{T}$ and sources $\mathbf{S}^l$, i.e. $U_t \cap U_l \neq \emptyset \ \wedge \ I_t \cap I_l \neq \emptyset$ for all $i \in [1, m]$. This work considers the case in which some overlap exists between users, but no overlap exists between items for the target and source domains. Furthermore, this work assumes that items are of different types (and dissimilar) across domains (e.g., artists, tags, friends). This is considered to be a more general case, unlike domains such as movies and books, that share similarity in genre and many movies are based on books.

5.3 Adsorption-based Cross-Domain Approaches

In order to address the cross-domain problem described in Section 5.2, two approaches based on knowledge aggregation, specifically Weighted Aggregation of Neighborhoods (WAN), de-

scribed in Section 5.3.2, and Weighted Aggregation of Recommendations (WAR), described in Section 5.3.2, were proposed [Parimi and Caragea, 2015e]. Furthermore, an approach to compute weights between domains (Section 5.3.3), and its usability with the WAN and WAR approaches to aggregate knowledge was studied. Later, Section 5.3.3 focuses on the proposed way to compute weights between domains and usability of the proposed approach with the WAN and WAR approaches to aggregate knowledge.

5.3.1 Weighted Aggregation of Neighborhoods (WAN)

In this approach, the user-user similarities in the source domains $\mathbf{S}^l$, $l \in [1, m]$, and the target domain $\mathbf{T}$, respectively, were first computed. The computed user-user similarities represent the neighborhoods in the corresponding domains. The neighborhoods from the m source domains and the target domain are then linearly aggregated using weights, as shown in Equation (5.1), and for each target user, the set of k nearest neighbors (having highest similarity scores) are extracted. The selected neighbors (kNN_t) are used with the Adsorption algorithm to generate target recommendations [Parimi and Caragea, 2015b]. If WN_T denotes the weighted neighborhood for the target domain, then

$$WN_T = w_t \times N_T + \sum_{l=1}^{m} w_l \times N_{S^l} \tag{5.1}$$

where, $N_{S^l}, l \in [1, m]$, and N_T are the user neighborhoods for source and target domains, respectively. Parameters $w_l, l \in [1, m]$, and w_t are the weights for the m source domains and the target domain, respectively, and they control the amount of knowledge aggregated from sources and target. Intuitively, the item preferences of users who are similar to the current user may correspond to good recommendations although the similarity is derived from a different domain. However, because the items are of different types across domains, two users with similar item preferences in a source domain may have very different preferences in the target domain. Therefore, the information coming from different domains must be

weighed. Note that this approach is similar to the weighted k-NN approach proposed in [Shapira et al., 2013], with the distinction that the complete neighborhoods from all domains are aggregated as opposed to only nearest neighborhoods. In addition, the recommendation algorithm used is Adsorption as opposed to the standard neighborhood-based CF approach.

5.3.2 Weighted Aggregation of Recommendations (WAR)

In this approach, the nearest neighborhoods (kNN) in all domains are first computed and used with the Adsorption algorithm in the target domain. Intuitively, in the Adsorption algorithm, the kNN from the target domain can be replaced with the kNN from a source domain because the kNN from a source-domain also captures user-user similarities. However, for the neighborhood, kNN, only user preferences from the target domain are propagated to ensure that target items are recommended to users in the target. Final recommendations are computed by linearly aggregating the recommendations from Adsorption using the nearest neighborhoods from the target and the m sources, as shown in Equation (5.2) [Parimi and Caragea, 2015b]:

$$\widehat{\mathbf{Y}}_{\mathbf{t}} = w_t \times ADS(kNN_t, \mathbf{Y_t}) + \sum_{l=1}^{m} w_l \times ADS(kNN_{S^l}, \mathbf{Y_t}) \tag{5.2}$$

In the above equation, $\widehat{\mathbf{Y}}_{\mathbf{t}}$ corresponds to final recommendations in the target, $ADS(kNN_t, \mathbf{Y_t})$ is the set of recommendations from Adsorption using the nearest neighborhood and user preferences from target, $ADS(kNN_{S^l}, \mathbf{Y_t})$, $l \in [1, m]$ is the set of recommendations from Adsorption using kNN_{S^l} from source domain l and user preferences from target. The weights w_t and w_l ($l \in [1, m]$) control the amount of knowledge aggregated from the target and the m source domains, respectively. Note that the WAR approach is computationally more expensive than the WAN approach because the Adsorption algorithm must be executed m (number of source domains) times for the WAR approach as compared to just once for the WAN approach.

5.3.3 Determining Weights between Domains

In order to address the problem of choosing weights to aggregate knowledge, this work proposed a novel approach inspired by work in [Yang et al., 2012] to automatically determine the weights to be used for the target and source domains. Specifically, this thesis proposed to use conditional probabilities between the user-user graphs in the source and target domains, respectively, in order to determine target-source relative weights, and use the conditional probability weights (CP weights) in the WAN and WAR approaches [Parimi and Caragea, 2015b]. This way of choosing weights can circumvent the weight tuning process, as long as the accuracy of the approaches using these weights does not suggest negative transfer.

In order to compute CP weights between domains $\mathbb{A}$ and $\mathbb{B}$, a user-user graph based on user-user similarity scores was first constructed for each domain. Two users were connected in the graph if the similarity was greater than zero. The CP weight between $\mathbb{A}$ and $\mathbb{B}$, i.e., $Pr(\mathbb{A}|\mathbb{B})$, was then estimated as the number of common links in the user-user graphs of $\mathbb{A}$ and $\mathbb{B}$ (i.e., number of links that connect the same pair of users in the two graphs), divided by the total number of links in the user-user graph of $\mathbb{B}$. Accordingly, in the WAN and WAR approaches, w_t is replaced by $Pr(\mathbf{T}|\mathbf{T})$ (which is 1), and $w_l, l \in [1, m]$ is replaced by $Pr(\mathbf{T}|\mathbf{S}^l)$.

5.4 Cross-Domain Matrix Factorization Approaches

In order to address the cross-domain problem described in Section 5.2, an approach based on matrix factorization, referred to as cross-domain implicit-feedback matrix factorization (CIMF) [Parimi and Caragea, 2015b], was also proposed in this work. The proposed CIMF approach involves two steps: first, the framework proposed in [Hu et al., 2008] was used to extract the user and item latent factors from the m source domains with *implicit* user preferences; in the second step, a *novel* way to use the latent factors from the first step as a bridge between the source and the target domains was proposed thereby facilitating the

transfer of knowledge from source domains to the target. The two steps are discussed in detail in the rest of the section.

Step 1 - Computing Source Domain Latent Factors: In this step, the latent user and item factors from the user preference matrices for the auxiliary source domains are first computed. Typically, these latent factors correspond to semantic concepts and measure the extent to which a user and an item exhibit these concepts [Koren et al., 2009]. Given that the source preference matrices have *implicit* feedback data, the factorization technique proposed by Hu et al. [2008] was used as described in Section 2.2.2.

According to the work in Hu et al. [2008], for a domain S^l, the user and item factors for a user u and item i can be computed by finding x_{u_l} and y_{i_l} that minimizes the objective function $\mathcal{J}$ given by Equation (5.3):

$$\mathcal{J}(\mathbf{x_{u_l}}, \mathbf{y_{i_l}}) = \sum_{(u_l, i_l)} c_{u_l i_l} \left(p_{u_l i_l} - x_{u_l}^{\mathsf{T}} y_{i_l}\right)^2 + \lambda \left(\sum_{u_l} \left\| x_{u_l}^2 \right\| + \sum_{i_l} \left\| y_{i_l}^2 \right\| \right) \tag{5.3}$$

In Equation (5.3), λ denotes the regularization parameter. The objective function can be efficiently solved using alternating least squares (ALS), and analytic expressions for user and item factors that minimizes Equation (5.3) can be obtained by differentiation. The interested reader is referred to the work in [Hu et al., 2008] for more details about the optimizations.

Step 2 - Integrating Source Latent Factors into Target Domain: In the second step, information captured in Step 1 is integrated into the target domain through a regularization technique. To accomplish this, an approach similar to the approach in Step 1 is used because the target domain **T** also has *implicit* user feedback. Specifically, the *implicit* user feedback is represented using the preference (p) and confidence (c) variables given by Equations (2.2) and (2.3), respectively, and the preference matrix is reconstructed as the inner products of the user and item latent factors. However, the objective function in Step 1 given by Equation (5.3) is extended to incorporate the knowledge from source domains through regularization, as described below.

After obtaining the user and item latent factors from the source domains, m (number of source domains) regularization terms $\sum_{u_t} \|x_{u_t} - x_{u_l}\|^2$ are added to the objective function $\mathcal{J}$ for the target domain $\mathbf{T}$ as shown in Equation (5.4):

$$\mathcal{J}(\mathbf{x_{u_t}}, \mathbf{y_{i_t}}) = \sum_{(u_t,i_t)} c_{u_t i_t} \left(p_{u_t i_t} - x_{u_t}^{\mathsf{T}} y_{i_t}\right)^2 + \sum_{l \ \epsilon \ [1,m]} \lambda_l \sum_{u_t} \|x_{u_t} - x_{u_l}\|^2 \ + \ \lambda \sum_{i_t} \|y_{i_t}\|^2 \quad (5.4)$$

In Equation (5.4), x_{u_t} and x_{u_l} are the user latent factors for user u in the target domain $\mathbf{T}$ and the source domain $\mathbf{S}^l$, respectively, with dimensions $1 \times f$; y_{i_t} is the item latent factors in $\mathbf{T}$ also with dimensions $1 \times f$; f is the number of latent factors. It is possible to have a user v in $\mathbf{T}$ who does not have any preferences in source domain $\mathbf{S}^l$ because no complete user overlap exists between the source and the target domains. In such cases, no knowledge about v can be transferred from $\mathbf{S}^l$ to $\mathbf{T}$, and the user factors x_{v_l} will be zero vectors. Parameter λ_l is the regularization parameter for user factors and is used to control the amount of knowledge transferred from source domain $\mathbf{S}^l$, λ denotes the regularization for item factors, and n is the number of source domains.

When transferring source knowledge through the regularization terms, as shown in Equation (5.4), the user factors in the source and target domains were required to be similar instead of being identical. The reason for requiring source and target user latent factors to be identical is that, although the source domains are related to the target domain, the user factors in the source and target domains can only be similar given that the bi-factorization technique used in Step 1 integrates both domain dependent and independent semantic concepts into user and item latent factors [Hu et al., 2008; Koren et al., 2009; Pan et al., 2010]. By controlling the amount of knowledge transferred from source domains to the target domain through regularization parameters λ_l, the goal of this approach is to transfer only the domain independent part of a source $\mathbf{S}^l$ to the target $\mathbf{T}$. Note, also, that only the user latent factors from the m source domains are used when transferring knowledge to $\mathbf{T}$. This is

because the items across the domains are of different types and are dissimilar to one another according to the cross-domain problem definition. Therefore, they do not contribute to the recommendation problem in the target domain.

Minimization of the proposed model defined in Equation (5.4) can be performed by ALS algorithm, similar to the optimization in Step 1. Observe that, when either the user factors or item factors are assumed to be known and fixed, the cost function becomes quadratic and, by differentiation,an analytic expression can be computed for the user and the item factors, respectively. Wit the learning of the user factors as an example, the optimization of x_{u_t} is shown by deriving its updating rule while fixing the item factors. The derivative of $\mathcal{J}$ with respect to x_{u_t} is as follows:

$$\frac{\partial \mathcal{J}}{\partial x_{u_t}} = 2 \sum_{i_t} \left(c_{u_t i_t} y_{i_t}^{\mathsf{T}} x_{u_t} y_{i_t} \; - \; c_{u_t i_t} p_{u_t i_t} y_{i_t} \right) + 2 \sum_{l \; \epsilon \; [1,m]} \left(\lambda_l x_{u_t} \; - \; \lambda_k x_{u_l} \right) \tag{5.5}$$

The following updating rule for x_{u_t} is obtained when $\frac{\partial \mathcal{J}}{\partial x_{u_t}} = 0$:

$$x_{u_t} = \frac{\sum_{i_t} \left(c_{u_t i_t} p_{u_t i_t} y_{i_t} \right) \; + \; \sum_{l \; \epsilon \; [1,m]} \left(\lambda_l x_{u_l} \right)}{\sum_{i_t} \left(c_{u_t i_t} y_{i_t}^{\mathsf{T}} y_{i_t} \right) + \sum_{l \; \epsilon \; [1,m]} \left(\lambda_l I \right)} \tag{5.6}$$

Let $Y_{q \times f}$ be an item factor matrix, where q is the number of items, f is the number of factors. Let $C^u_{q \times q}$ be a diagonal matrix where $C^u_{ii} = c_{u_t i_t}$, and $p(u)$ be an $q \times 1$ vector. Using these notations, Equation (5.6) can be expressed as

$$x_{u_t} = \left(Y^{\mathsf{T}} C^u Y \; + \sum_{l \; \epsilon \; [1,m]} \lambda_l I \right)^{-1} \left(Y^{\mathsf{T}} C^u p(u) \; + \sum_{l \; \epsilon \; [1,m]} \lambda_l x_{u_l} \right) \tag{5.7}$$

Item factors can be learned similarly, and the updating rule for y_i is given by Equation (5.8):

$$y_{i_t} = \left(X^{\mathsf{T}} C^i X \; + \; \lambda I \right)^{-1} X^{\mathsf{T}} C^i p(i) \tag{5.8}$$

where $X_{w \times f}$ is a user factor matrix, w is the number of users, f is the number of factors, $C^i_{w \times w}$ is a diagonal matrix where $C^i_{uu} = c_{u_t i_t}$, and $p(i)$ is a $w \times 1$ vector. Note that the update rule for learning y_{i_t} for this cross-domain model is identical to the update rule for learning y_{i_l} in Step 1 (derivation not shown), because no knowledge about items between source and target domains is transferred.

5.5 Contributions of this Cross-Domain Study

This section summarizes contributions of this work in the cross-domain setting.

5.5.1 Proposal and Comparison of Two Adsorption-based Knowledge Aggregation Approaches

Proposed and Evaluated WAN and WAR Approaches

In order to address the cross-domain recommendation problem described in Section 5.2, two approaches, WAN and WAR, which aggregate the information about neighborhoods and recommendations, respectively, were proposed. One goal of this work was to evaluate the effectiveness of the proposed knowledge aggregation approaches in improving the target recommendation accuracy as compared to single-domain approaches. To accomplish this goal, the performance of the WAN and WAR approaches, described in Section 5.3, and the performance of single-domain approaches were compared for six target recommendation tasks from two datasets with implicit user feedback.

Proposed and Evaluated an Approach for Computing Weights between Domains

Anther goal of this study was to understand the effectiveness of the proposed approach to automatically determine weights (CP weights) to be used for target and source domains. To accomplish this goal, the performance of WAN and WAR approaches using CP weights

was compared with the performance of these approaches using manual weights and with the performance of single-domain approaches on six target recommendation tasks from two datasets with implicit user feedback.

Evaluation of WAN and WAR's Ability to Handle the Cold-Start User Problem

As discussed in Section 5.1.3, the cold-start user problem (make meaningful recommendations to users who have not expressed item preferences in the target domains) is a common problem in real-world recommender systems. Therefore, a goal of this study was to understand the usefulness of the proposed WAN and WAR approaches to address the cold-start user problem (a.k.a., the unacquainted world scenario). Towards this goal, the unacquainted world scenario was simulated for six target recommendation tasks from two datasets with implicit user feedback, and the effectiveness of the proposed approaches was experimentally validated.

5.5.2 Proposal and Evaluation of a Cross-Domain Matrix Factorization Approach

Because latent factor model-based approaches have better performance as compared to neighborhood-based approaches, a goal of this study was to propose an MF-based approach in order to address the cross-domain problem and to evaluate its effectiveness. Towards this goal, an approach based on MF for implicit user feedback, CIMF, was proposed, and the performance of the CIMF approach was compared to the performance of state-of-the-art single-domain CF approaches on six target recommendation tasks from two datasets with implicit user feedback.

5.5.3 Evaluation of the Proposed Approaches' Ability to Utilize Various Amounts of User Overlap

Another goal of this study was to understand the effectiveness of the proposed cross-domain approaches to handle varying amounts of user overlap between source and target domains. To gain better insights into this, performance of WAN and WAR approaches (using manual weights and the CP weights) and performance of the CIMF approach were studied by simulating two user overlap scenarios, as described in Sections 6.2.4 and 6.2.4, for six target recommendation tasks from two datasets with implicit user feedback.

5.5.4 Comparison of Adsorption and Matrix Factorization Approaches

Similar to the single-domain study, one objective in the cross-domain study was to understand which approach is better between Adsorption-based and Matrix Factorization-based cross-domain approaches. To accomplish this goal, results from the experiments on six target recommendation tasks from two datasets were analyzed (contributions in Sections 5.5.1, 5.5.2, 5.5.3) .

5.5.5 How Can Domain Knowledge Help?

A final goal of this study in the cross-domain setting was to understand the usefulness of knowledge about the data domain (e.g., close connections versus loose connections among users) or characteristics of the dataset (e.g., density of the feedback matrix) in selecting the most suitable cross-domain CF approach to use for a particular recommendation problem. To accomplish this goal, the results from the experiments on six target recommendation tasks from two datasets were analyzed in conjunction with the domain and dataset characteristics (contributions in Sections 5.5.1, 5.5.2, 5.5.3).

Intuitively, Adsorption-based cross-domain approaches may have better performance for

dense datasets and for sparse datasets with strong neighborhood relations between users. MF-based approaches may have better performance for sparse datasets with loose connections between neighbors, similar to the hypothesis in the single-domain setting described in Section 4.3.5.

Chapter 6

Datasets and Evaluation Metrics

In this chapter, Section 6.1 explains the datasets and preprocessing steps involved in creating the training and test sets used to evaluate approaches in the single-domain. Section 6.2 explains the datasets and preprocessing steps for creating training and test sets to verify the effectiveness of the proposed cross-domain approaches in order to improve target recommendation accuracy. This section also explains the creation of training and test sets for the cold-start user problem (unacquainted world scenario) (Section 6.2.3) and two user overlap scenarios (Section 6.2.4). Finally, Section 6.3 describes the metrics used to evaluate the performance of recommendation algorithms used in this work.

6.1 Single-Domain Setting

The following sections describe four datasets used in the single-domain experiments: *Audioscrobbler* music dataset, *DBLP* co-author dataset, *Adknowledge Inc.* URL dataset, and *BookCrossing* dataset, that were . Given that some of the datasets used in the experiments have timestamp information and some of the datasets do not have timestamp information, this section is organized as follows: Section 6.1.1 describes the data and preprocessing steps involved in constructing the training and test sets for datasets that do not have timespamps

(specifically, *Audioscrobbler* and *BookCrossing* datasets). Section 6.1.2 describes the data and preprocessing steps involved in constructing the training and test sets for datasets that have timespamps (*DBLP* and *Adknowledge* datasets).

6.1.1 Single-Domain Datasets without Timestamps

Audioscrobbler Music Dataset

The first dataset was the *Audioscrobbler* music dataset. The dataset had information about user-artist interactions and the task was to recommend unknown artists. The format of this dataset is: $\langle user, artist, plays \rangle$. The dataset contains approximately 150 thousand users, 1.5 million artists, and 24 million user-artist preferences[1]. The median number of items preferred by a user is 79; hence, this dataset was considered to be a dense dataset in this study. Furthermore, given the varied interest in artists and tracks from users, this dataset was assumed to have loose neighborhood relations among users.

BookCrossing Book Dataset

The second dataset used in the single-domain study was the *BookCrossing* dataset [2] [Ziegler et al., 2005] which had information about books read by members of the *BookCrossing* community. The task was to recommend books to users. The dataset consists of approximately 100 thousand users, 340 thousand books and 1.1 million user-book preferences, with three columns, specifically, *user*, *book*, *timesRead*. The median number of items preferred by a user is 7; hence, this dataset was considered to be a sparse dataset in this study. Furthermore, the book domain and the movie domain have many characteristics in common. For example, users may prefer a book based on genre, author (analogous to director in the movie domain), or lead characters. Therefore, the assumption was made that the users of this dataset do not exhibit a strong community structure (loose neighborhood relations)

[1] http://www-etud.iro.umontreal.ca/~bergstrj/audioscrobbler_data.html
[2] http://konect.uni-koblenz.de/networks/bookcrossing_full-rating

Table 6.1: Training and test subsets based on four CV folds (denoted, $0, \cdots, 3$) of the Audioscrobbler and BookCrossing datasets; $fold(...)$ indicates the CV folds used to create a particular subset, e.g., fold(0, 1, 2) gives a subset that is the union of CV folds 0, 1, and 2.

Subset	Training	Test
0	fold(0, 1, 2)	fold(3)
1	fold(1, 2, 3)	fold(0)
2	fold(2, 3, 0)	fold(1)
3	fold(3, 0, 1)	fold(2)

and may benefit from user information captured globally.

Training and Test Data Construction

In order to construct the training and test data for *Audioscrobbler* and *BookCrossing* datasets, users who interacted with less than four items (artists or books) were removed. Data was divided into four folds with approximately 25% of a user's preferences in each fold. This technique is referred to as *per-user CV* [Said and Bellogín, 2014] and ensures that every user had at least one preference in each fold. Next, three folds were used as training data to compute recommendations that were evaluated on the fourth fold (test data). This was repeated four times (four runs) to ensure that every fold was used as a test fold. Table 6.1 depicts the training and test data for each run of the recommendation algorithm.

Filtering the Test Set

Once the training and test sets for the two datasets were created as described above, two preprocessing steps on the test data were performed. First, from the test set, all users that did not appear in the corresponding training set were removed. This is because, if a user is not present in the training set (i.e., does not have some history available), there is no way to make recommendations for that user. Second, given that the objective was to recommend new unseen items to users, the items preferred by a user in the training set were removed from the user's test set. Note that this is a standard way of creating test datasets to evaluate

recommender systems and is widely used in literature [Cremonesi et al., 2010; Baluja et al., 2008; Sarwar et al., 2001; Said and Bellogín, 2014].

6.1.2 Single-Domain Datasets with Timestamps

DBLP Co-Author Dataset

The third dataset used was the *DBLP* co-author dataset [Ley, 2002; Kunegis, 2013]. The dataset had information about user-user collaborations between years 1940 and 2013, and the task was to recommend new collaborations. The dataset consists of approximately 1.3 million users (who can also be seen as items) and 18.9 million collaboration records with four columns, specifically, *From_id*, *To_id*, *weight*, and *timestamp*. A subset of this dataset with collaborations between years 1992 and 2012 was used in the experiments. This subset had approximately 1 million users and 18.3 million collaboration records. Given a small median for items preferred (3), this dataset was considered to be a sparse dataset in this study. Furthermore, given that collaborations between a pair of authors in real-world is based on whether the two authors are acquaintances (have close connections) and have similar research interests, this dataset was considered to have strong local neighborhood relations between users.

Adknowledge URL Dataset

The fourth dataset used in this study was provided by *Adknowledge Inc.*[3] This dataset contained information about anonymous users clicking various web pages between September 2011 and October 2012, and the task was to recommend new pages to users. The dataset was provided in an encrypted form; no raw data is available - users and web pages are encrypted using numeric IDs. Encryption was performed to prevent identification of users and web pages, or to associate any behavior, such as clicking, with a particular individual or page. The dataset contains approximately 1.5 million users, 475 thousand items, and 3.4 million

[3] http://www.adknowledge.com/

browsing records with five columns, three of which were used in this work: *user_id*, *url_id*, and *timestamp*. This dataset was considered to be a sparse dataset because the median for items preferred is 1. Furthermore, given the dynamic nature of the Web domain with frequent addition of new web pages containing various types of information (e.g., movies, news, sports, etc.), the assumption was made in this work that the users of this dataset would benefit from global information (loose neighborhood relations).

Training and Test Data Construction

For the *DBLP* and *Adknowledge* datasets, timestamps were used to generate training and test datasets. The data was divided into seven folds, and both short histories and increasingly long histories (with the goal of studying the influence of the length of the history on the performance of the algorithm) were created. Each of the seven folds in the *DBLP* co-author dataset had roughly data from three years. For the *Adknowledge* dataset, because the user clicks were highly skewed towards the last two months, the folds were created as follows: Fold 0 had user click data for the first six months, Fold 1 had user click data for the next six months, and the remaining five Folds were created from the last two months with approximately 12 days of user clicks in each fold. Using these seven folds, for each dataset, five subsets of training and test data were generated to capture long histories. Similarly five subsets of training and test data were generated to capture short histories. Table 6.2 depicts how the five subsets were generated from the folds.

Filtering the Test Set

After the training and test sets were created as described above, the test sets were filtered similar to the test sets for datasets without timestamp information, as described in Section 6.1.1.

Table 6.2: Training and test subsets for increasingly long and short histories, respectively, based on seven folds (denoted, $0, \cdots, 6$) of the DBLP and Adknowledge datasets; fold(...) indicates the folds used to create a particular subset. For example, fold(0,1) gives a subset that is the union of folds 0 and 1.

	Increasingly-long histories		**Short histories**	
Subset	Training Set	Test Set	Training Set	Test Set
0	fold(0,1)	fold(2)	fold(0,1)	fold(2)
1	fold(0,1,2)	fold(3)	fold(1,2)	fold(3)
2	fold(0,1,2,3)	fold(4)	fold(2,3)	fold(4)
3	fold(0,1,2,3,4)	fold(5)	fold(3,4)	fold(5)
4	fold(0,1,2,3,4,5)	fold(6)	fold(4,5)	fold(6)

6.2 Cross-Domain Setting

In order to evaluate the proposed cross-domain approaches, two datasets, specifically, a Last.FM music dataset and a DBLP citation dataset were used in this work. The following sections describe the two datasets and the preprocessing steps involved to create the training and test sets.

6.2.1 Cross-Domain Datasets without Timestamps

Last.FM Music Dataset

The first dataset used in this work was created by Cantador et al. [2011], and is a subset of the *Last.FM* dataset[4]. This dataset consists of the following three domains: artist domain in which each tuple has (*userID, artistID, #timesListened*) information, friend domain in which each tuple has (*userID, friendID,1*), tag domain in which each tuple has (*userID, tagID, #timesUsed*) information. The number of users in each domain is approximately $1,800$. The number of items in the artist, friend, and tag domains are approximately

[4] http://www.lastfm.com

17,000, 1,800, 11,000, respectively. The task was to recommend artists, friends, and tags to users.

Training and Test Data Construction

A cross-validation (CV) technique referred to as *per-user CV* [Said and Bellogín, 2014] was used to create the training and test sets for this dataset. This is because, timestamps are not available for this dataset. Specifically, users who preferred less than three items, such as three artists, were removed from each domain. This resulted in final datasets with approximately 1,800 users, 1,400 users, and 1,700 users in the artist, friend, and tag domains, respectively. The number of users who have preferences in all three domains (*common user set*) is approximately 1,400. The number of items in the artist, friend, and tag domains are approximately, 13,000, 17,000, and 7,000, respectively. The filtered data was then divided into three folds with approximately 33.3% of a user's preferences in each fold. This technique, referred to as the *per-user CV* [Said and Bellogín, 2014], ensures that every user had at least one preference in each fold. A 3-fold cross validation (CV) technique was then used to create the training and test subsets (two folds as training and one fold as test). Results reported in this work for this dataset were averaged over three runs of the recommendation algorithm. Table 6.3 depicts the training and test data for each run of the recommendation algorithm.

Table 6.3: Training and test subsets based on three CV folds (denoted, $0, \cdots, 2$) for the three domains of the Last.FM dataset; $fold(...)$ indicates CV folds used to create a particular subset, e.g., fold(0, 1) gives a subset that is the union of CV folds 0 and 1.

Subset	Training	Test
0	fold(0, 1)	fold(2)
1	fold(1, 2)	fold(0)
2	fold(2, 0)	fold(1)

For this dataset, the median number of items preferred by a user was 33, 6, 9 for the artist, friend, and tag domains, respectively. Furthermore, all three domains were considered

to exhibit loose neighborhood relations. Although this is intuitive for the artist and the tag domains, a common belief is that social networks, in particular the underlying network between users in the friend domain of this dataset, exhibit a strong community structure. However, it has to be noted that user links in a social network most often are based on *friend-of-a-friend* relationship. Therefore, two users who are friends may not have a regular line of communication (chat, message, etc.) and, therefore, considered to exhibit more loose neighborhood relations as compared to real-world friendships.

Filtering the Test Set

When creating the training and test data for the three domains of this dataset as described above, the following properties were ensured to hold: a user in the test set has some history available in the training set [Li et al., 2009a; Pan et al., 2010; Sarwar et al., 2001; Said and Bellogín, 2014], and the test set for a user does not contain items that are also in the corresponding training set for that user because the objective was to recommend only unknown items [Baluja et al., 2008; Said and Bellogín, 2014]. Items that are not preferred in the training data (cold-start items) were also filtered from the test data because those items could not be recommended.

6.2.2 Cross-Domain Dataset with Timestamps

DBLP Citation Dataset

The second dataset was a citation dataset extracted from *DBLP*, *ACM*, and other sources, and was downloaded from ArnetMiner[5] [Tang et al., 2008]. This dataset was used to construct a co-author domain in which each tuple had *(authorID, coauthorID, #papersCoauthored)* information, a conference domain in which each tuple had *(authorID, conferenceID, #papersPublished)* information, and a reference domain in which each tuple had *(authorID,*

[5] http://arnetminer.org/citation

referenceID, #papersReferenced) information. The objective was to recommend collaborators, conferences, and references to authors.

Training and Test Data Construction

The original dataset had approximately 2×10^7 publications and 4×10^7 citation relations. From this set, papers published between the years 1990 and 2006 were used to create a *training paper set* (papers from which information about authors, conferences, and references was extracted to create training data for the three domains). Papers published after the year 2007 were used to create a *test paper set* (papers from which information about authors, conferences, and references was extracted to create test data for the three domains). The following rules were used to decide if an author was included in the three domains:

1. The author had at least one paper in the *training paper set* and at least one paper in the *test paper set*.
2. The author co-authored with at least five different authors in the *training paper set* and co-authored with at least one author (different to the co-authors from the training paper set) in the *test paper set*.
3. Cardinality of the set of all citations from the papers published by the author that belong to the *training paper set*, should be at least five, and cardinality of the set of all citations from the papers published by the author that belong to the *test paper set* (different to the citations from the training paper set) should be at least one.
4. Cardinality of the set of all conferences in which the author had published papers belonging to the *training paper set* should be at least one, and cardinality of the set of all conferences in which the author has published papers belonging to the *test paper set* (different to the set of conferences from the training paper set) should be at least one.

After filtering the authors as described, the total number of authors selected were 29,189. For the selected authors, the publications in the *training paper set* and the *test paper set* were used to construct the training and the test data for the co-author, conference, and reference domains. Note that the co-author, conference, and reference domains constructed have the same users, approximately 29,000, in each domain (*common user set*). The training set for the co-author, conference, and reference domains had approximately 140,000 items (co-authors), 2,000 items (conferences), and 201,000 items (references), respectively. The median number of items preferred by a user was 5, 3, 19 for the co-author, conference, and reference domains, respectively. Furthermore, the co-author domain was considered to exhibit a strong neighborhood relation, whereas the reference and conference domains were considered to have a more loose neighborhood relations among authors.

For this dataset, for a train user, 50% of preferences from his/her training data were randomly picked, and only these preferences were used to generate recommendations. This was repeated five times, similar to cross-validation (CV), in order to account for variation in results from the algorithms, and the results were averaged over the five runs.

Filtering the Test Set

The two properties, described in Section 6.2.1, were ensured to hold for the test sets from the three domains of the DBLP dataset, similar to the Last.FM dataset.

6.2.3 Cold-Start User Problem

In order to understand if the proposed WAN and WAR cross-domain approaches can overcome the cold-start user problem in the target domain (i.e., check their ability to make meaningful recommendations to users who do not have any preferences), an unacquainted world scenario was simulated in this work for the six recommendation tasks from the Last.FM and the DBLP datasets. Specifically, to construct the training data for a target domain, 50% of the users were randomly selected from the *common user set* (users with preferences in all

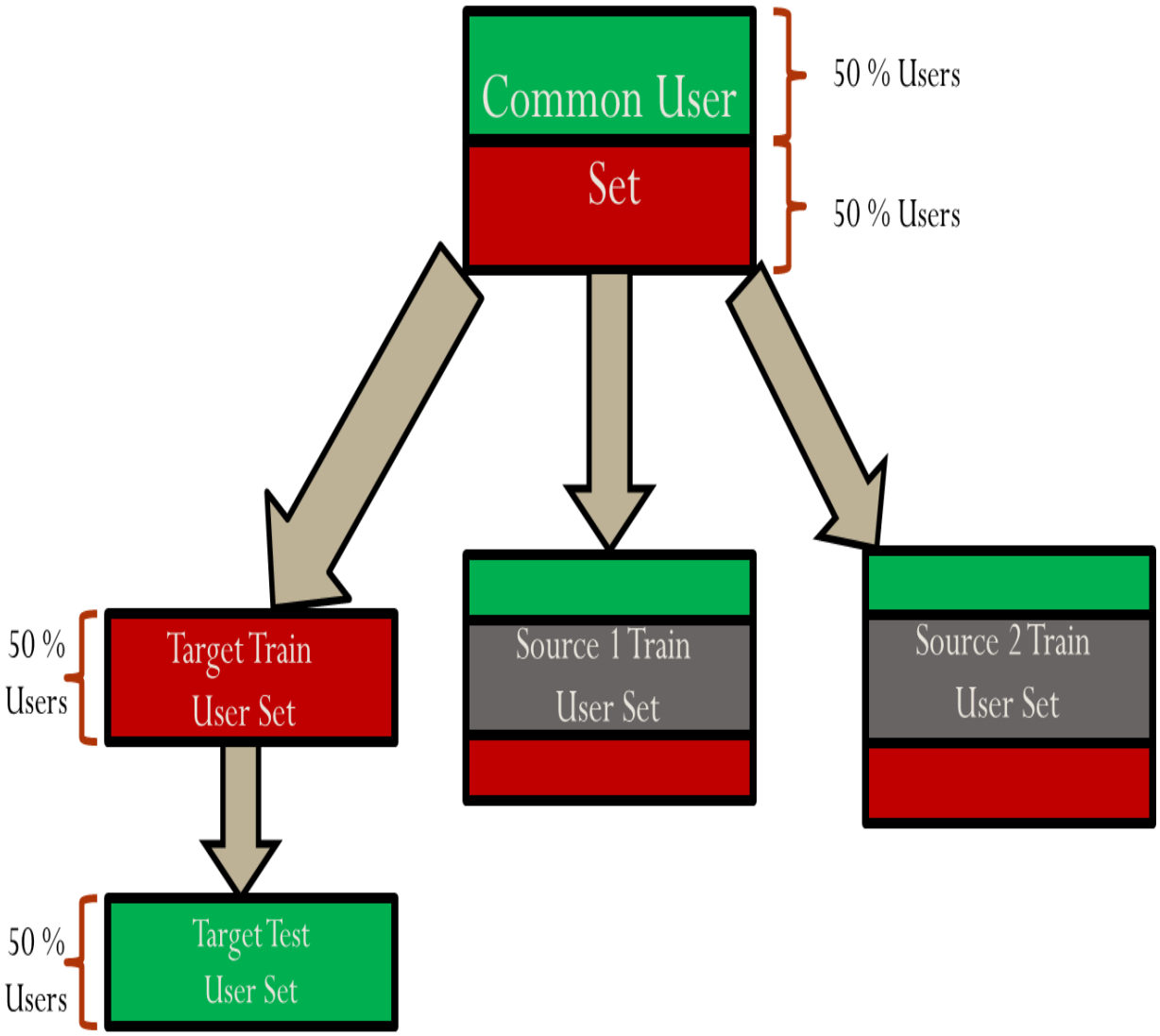


Figure 6.1: Creation of training and test user sets for the target domain and training user sets for the two source domains from the common user set in order to simulate cold-start user problem in the target domain.

three domains of the dataset). Then all preferences of selected users were held out from the CV fold of the original training data (described in the previous section) in the target domain in order to create the CV fold in the unacquainted world scenario. The corresponding test set consisted of only users selected above and their preferences. This was done by filtering the selected users and their preferences from the CV fold in the original test set. Further-

more, cold-start items were filtered out from the CV folds of the test set because these items could not be recommended. Note that no changes were made to the training data for the source domains, and CV folds from the original training set were used. Figure 6.1 summarizes the creation of training and test data in order to simulate the cold-start user problem in a target domain.

6.2.4 User Overlap Scenarios

User Overlap Scenario I

The goal of this set of experiments was to study the influence of user overlap between target and source domains on the performance of the proposed cross-domain approaches in order to understand their effectiveness in using various amounts of user overlap. To accomplish this goal the percentage of user overlap between source and target domains was varied from 25% to 50% and 75% for the six domains from the two datasets. For example, in order to create the training data for source and target domains for 25% user overlap, the following steps were used:

1. From the *common user set* (users with preferences in all domains of the dataset), 25% of users were randomly picked to be included in all domains (users will have preferences in all domains).
2. From the remaining users in the *common user set*, 50% were randomly picked to be included only in the target domain (users will have preferences only in the target) and the remaining 50% to be included only in source domains (users will have preferences only in the sources).

Figure 6.2 summarizes the steps to create user sets for the target and two source domains for 25% user overlap.

Test data for a target domain contained preferences only from the users in the corresponding training data (the rules described in Section 6.2.1 were followed). Note that when

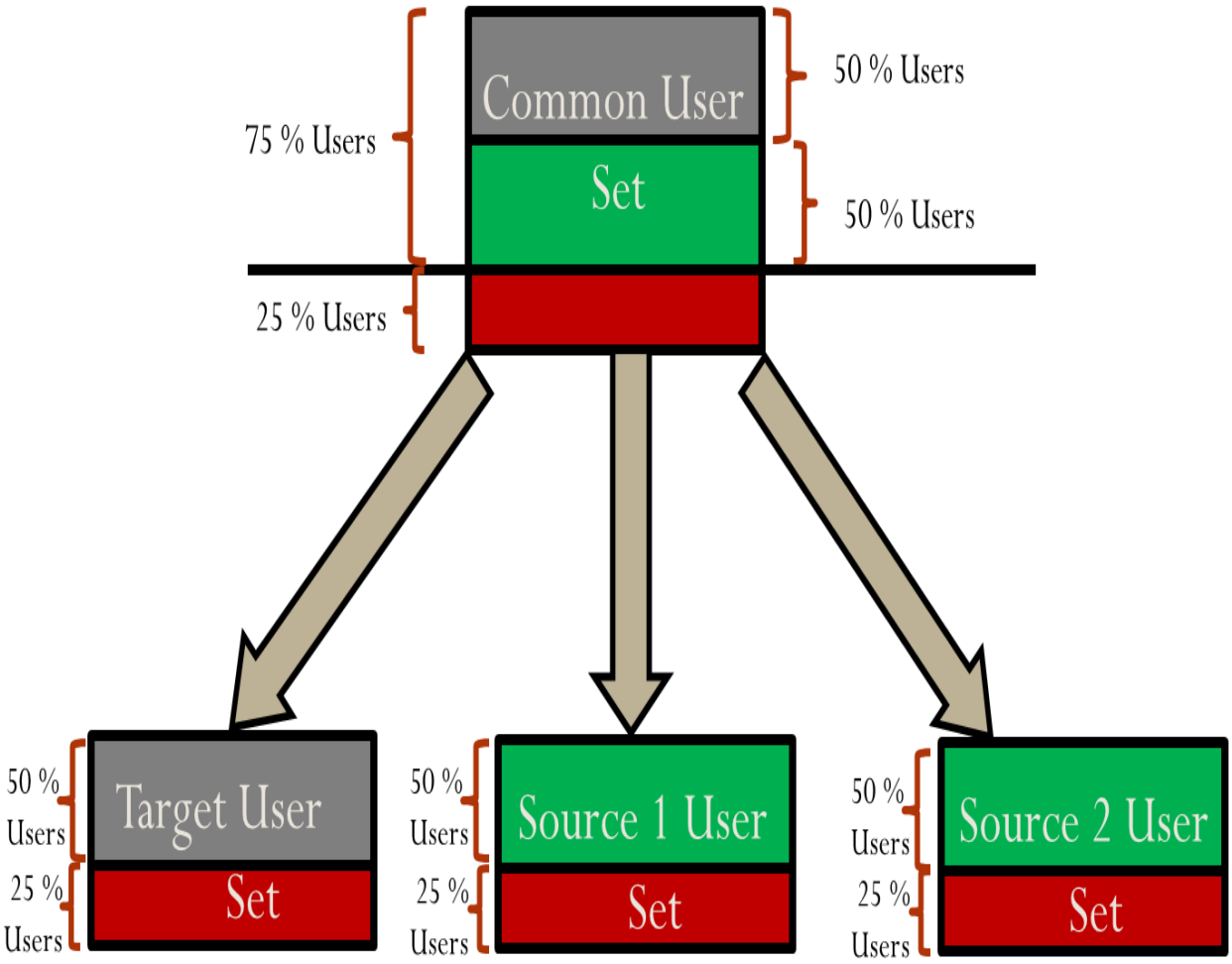


Figure 6.2: Creation of user sets for the target and the two source domains from the common user set for 25% user overlap.

creating the training and test sets for these experiments, the scenario in which the target and source domains have the specified percentage of overlap was ensured. The scenario in which the training set for one overlap percentage is built on top of the training set of the previous overlap percentage was not ensured because the main goal of this overlap scenario was to show that the proposed approaches can handle various percentages of overlap, as opposed to investigating if performance increases with the amount of overlap.

User Overlap Scenario II

The goal of this set of experiments was to: a) investigate if the performance of the proposed cross-domain approaches increases with an increase in the overlap percentage between the source domains and the target domain, and b) bring additional experimental evidence regarding the effectiveness of the proposed cross-domain approaches to handle varying amounts of user overlap. To accomplish this goal, the training set for one overlap percentage was built on top of the training set of the previous overlap percentage. The test data for a target domain has preferences only from the users in the corresponding training data. Similar to other experiments, cold-start items were filtered from the test set. The specific steps involved in the construction of the training and the test sets for the Last.FM and the DBLP datasets are described below, assuming two source domains and one target domain, given that each of the two datasets used in this work had three domains. However, these steps can be generalized to any number of source domains and one target domain.

Training and Test Data Creation for DBLP Dataset

1. For a target domain, the *common user set* (users with preferences in all domains of the dataset) was randomly divided into two sets with approximately 50% of users in each set.

   (a) Users in the first set (target user set) corresponded to users in the target domain $\mathbf{T}$, for the three overlap scenarios.

   (b) From the users in the second set (*source users*), 10% of users were randomly picked to be included only in source domain $\mathbf{S}^1$ and a different 10% of users were randomly picked to be included only in source domain $\mathbf{S}^2$. This was done to simulate a scenario in which few users have preferences only in one domain, similar to real-world datasets.

2. The users in the target domain $\mathbf{T}$ (50% of users randomly selected in Step 1 (a)) were

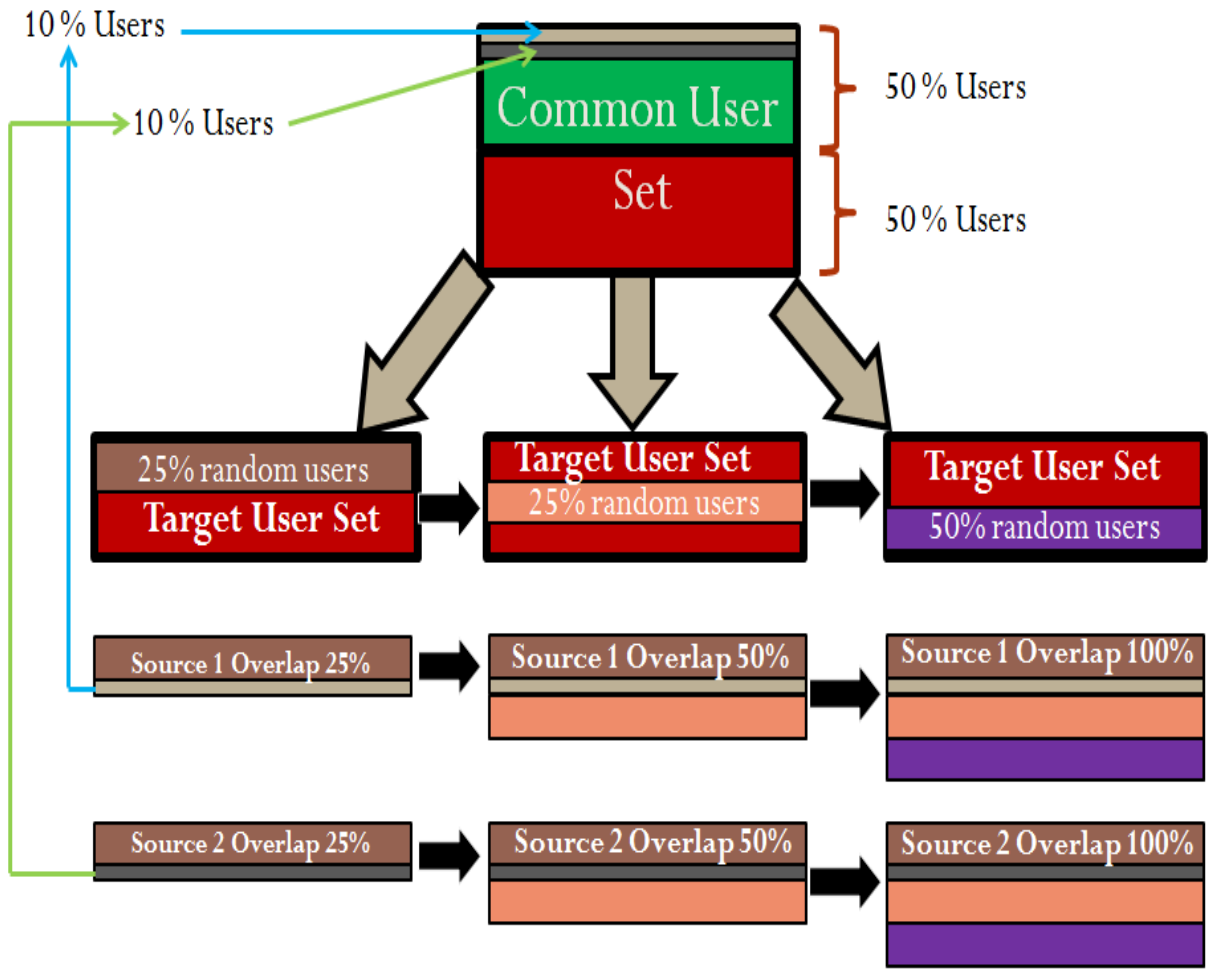


Figure 6.3: Creation of user sets for the target and the two source domains for 25%, 50%, and 100% user overlap between the target and the two source domains for the DBLP dataset.

used to create the three overlap scenarios as follows:

(a) First, 25% of users were randomly selected from the target user set and were added to the user sets of the two source domains, respectively. This scenario is referred to as 25% Overlap Scenario because the target domain had 25% of its users overlapping with each source domain.

(b) Next, 25% of users (different from the 25% selected in Step 2 (a)) were selected randomly from the target user set. The selected users were added to the user sets of the two source domains from the 25% Overlap Scenario. This scenario is

referred to as 50% Overlap Scenario because the target domain had 50% of its user overlapping with each source domain.

(c) Finally, the remaining users (users that are not selected in the 25% and 50% overlap scenarios) were selected from the target user set and were added to the user sets of the two source domains from 50% Overlap Scenario. This scenario is referred to as 100% Overlap Scenario because the target domain had all of its users overlapping with each source domain.

3. For each overlap scenario, the user sets for target and the two source domains were available. From these user sets, the training and test sets for each overlap scenario and each CV fold were constructed as follows: the training data for a target domain had preferences from only the users selected for the target domain. Similarly, the training data for a source domain had preferences from only the users selected for the source domain. In order to create corresponding test sets for each domain, the following two properties were ensured to hold:

   (a) The test set for a user does not contain items that are also in the corresponding training set for that user.

   (b) The set of all items in the test data is a subset of the set of all items in the corresponding training data (no cold-start items can be found in the test data).

Figure 6.3 summarizes the steps used to create user sets for the target and the two source domains for the three overlap percentages considered for the DBLP dataset.

Training and Test Data Creation for Last.FM Dataset

For the Last.FM dataset, the original dataset had a small number of users who did not have preferences in all three domains (there is a partial overlap between the users of the three domains). Therefore, the training and test sets for the three overlap scenarios were created, taking into account the partial user overlap between the three domains, as follows.

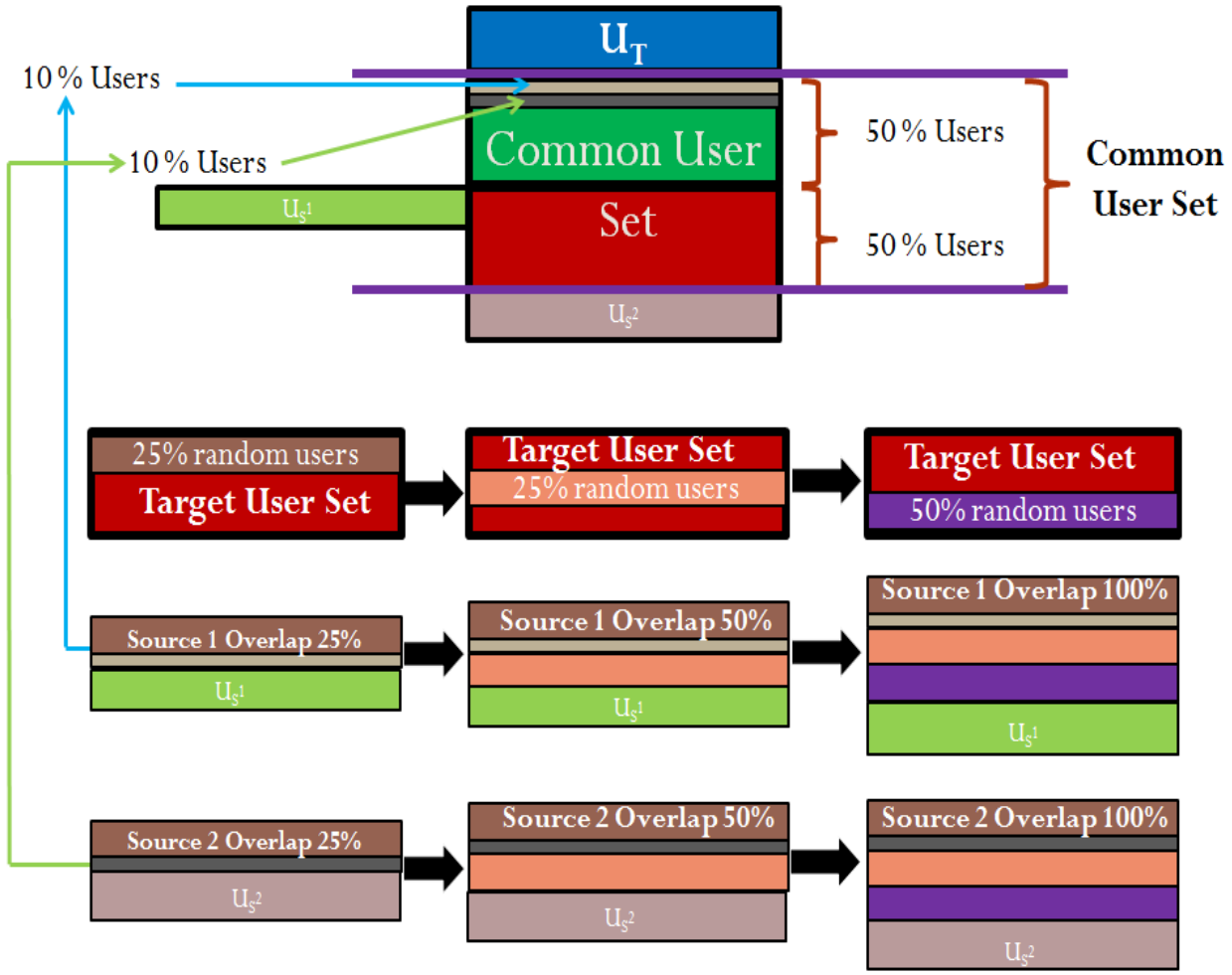


Figure 6.4: Creation of user sets for the target and the two source domains for 25%, 50%, and 100% user overlap between the target and the two source domains for the Last.FM dataset.

Let $\mathbf{T}$, $\mathbf{S}^1$, and $\mathbf{S}^2$ be the target and the two source domains, respectively, in the Last.FM dataset, and let $\mathbf{U}_T$, $\mathbf{U}_{S^1}$, and $\mathbf{U}_{S^2}$ be the set of users with preferences only in $\mathbf{T}$, $\mathbf{S}^1$, and $\mathbf{S}^2$, respectively. Given the target domain $\mathbf{T}$, Step 1 (a), described above, was followed in order to create the user set for the target domain. Next, Step 1 (b), described above, was followed, and 10% of source users were randomly selected to be included only in source domain $\mathbf{S}^1$. To this set of users, the users in $\mathbf{U}_{S^1}$ were added. Similarly, a different 10% of source users were randomly selected to be included only in source domain $\mathbf{S}^2$. To this set of users, the users in $\mathbf{U}_{S^2}$ were added. Finally, Steps 2 and 3, described above, were followed to

create the three user overlap scenarios for the target and the two source domains. Figure 6.4 summarizes the steps used to create user sets for the target and the two source domains for the three overlap percentages considered for the Last.FM dataset.

6.3 Evaluation Methodology

For each user, the Adsorption and MF algorithms generate a list of ($item$, $preferenceScore$) tuples as recommendations. From this list, tuples corresponding to items in the training set were removed because the goal was to recommend only new items to users. Next, an ordered list of n tuples (number of recommendations), sorted from the highest to the lowest $preferenceScore$, was generated. In order to evaluate the algorithms, standard metrics in the area of information retrieval [Manning et al., 2008] were used, specifically Mean Average Precision (MAP@n) and Mean Recall@n, described below.

6.3.1 Mean Average Precision

The MAP@n score, defined as the mean of Average Precision values (AP@n), takes the order of the recommendation into account. In order to compute the MAP score, one first computes Average Precision at n for each user u, according to Equation (6.1):

$$AveragePrecision_u@n = \frac{\sum_{i=1}^{n} P(i)}{\min(j, n)} \tag{6.1}$$

where $P(i)$ is the *Precision* at position i, and j is the number of relevant items from the test set. From the Average Precision scores, the MAP score is computed according to Equation (6.2):

$$MAP@n = \frac{\sum_{u=1}^{N} AveragePrecision_u@n}{N} \tag{6.2}$$

In Equation (6.2), N is the number of users in the test set.

6.3.2 Mean Recall

Recall is defined as the fraction of relevant items successfully recommended for a user. This metric, which does not take into account the ordering of the recommended items, can be computed using Equation (6.3):

$$Recall_u@n = \frac{\left|\{relevant\ items\} \bigcap \{recommended\ items\}\right|}{\min(j, n)} \tag{6.3}$$

where $\{relevant\ items\}$ are the items in the test set of u, $\{recommended\ items\}$ are the items recommended to u, j is the number of items in the relevant item set of u, and n is the number of items recommended to u. From the Recall scores, Mean Recall can be computed according to Equation (6.4):

$$MeanRecall@n = \frac{\sum_{u=1}^{N} Recall_u@n}{N} \tag{6.4}$$

In Equation (6.4), N is the number of users in the test set.

Chapter 7

Experiments in the Single-Domain Setting

This chapter describes research questions addressed, experiments conducted to address the research questions, and results for comparison of weight-based neighborhood variants for Adsorption variants and neighborhoods constructed using community detection for Adsorption algorithms in Sections 7.1 and 7.2, respectively. Section 7.3 focuses on the analysis of results for the Adsorption and MF approaches to understand how the two approaches compare with each other. Sections 7.4 focuses on analysis of the results to understand if short user histories are preferred over long user histories. Finally, Section 7.5 presents an analysis of results to understand if data and domain characteristics can influence the choice of neighborhood approach to use for Adsorption, and the collaborative filtering approach to use, between Adsorption and MF, for a recommendation application.

7.1 Comparison of Weight-based Neighborhoods for Adsorption

One goal of this study in the single-domain setting was to understand the effectiveness of the four weigh-based neighborhood variants for Adsorption, described in Section 4.2.1, for various recommendation tasks. To accomplish this goal, the performance of Adsorption algorithm from the four neighborhood variants on three implicit feedback datasets from different domains was compared [Parimi and Caragea, 2015d]. Analysis of results is presented in this section.

7.1.1 Datasets

Datasets used for this study were the dense *Audioscrobbler* music dataset, the sparse *DBLP* co-author dataset, and the sparse *Adknowledge Inc.*. More details about the datasets, the preprocessing steps, and the creation of training and test sets for these datasets are provided in Section 6.1.

7.1.2 Research Questions and Experiments

1. **Which weight-based neighborhood variant is better for each dataset?**

   In order to address this research question, four experiments were conducted in each dataset. In each experiment, one variant was used to construct the user neighborhood. Each neighborhood was used with the Adsorption algorithm to compute recommendations, and performance of the Adsorption algorithm using the four variants was compared.

2. **How do Boolean preferences compare to actual user preference counts for computing similarity?**

   In order to address this research question, four experiments similar to the exper-

iments conducted to address research Question 1 were conducted, and performance of Adsorption algorithms from neighborhood constructed using Variants 1, 2, and 3 was compared with the performance of the Adsorption algorithm from neighborhood constructed using Variant 4.

7.1.3 Hyper-parameter Values

Choosing hyper-parameters for recommender systems is an important task, but challenging because of the large datasets and high computational complexity of algorithms. The hyper-parameters for Adsorption are p_{inj}, p_{term}, p_{cont}, (random-walk probabilities). Several trail runs on the datasets indicated small differences in the MAP values for various p_{inj}, p_{term}, and p_{cont} values. Hence, their values were fixed to (0, .85, .15). The number of neighbors (k) and the number of recommendations (n) were fixed to 5 and 10, respectively.

7.1.4 Results and Discussion

MAP scores for the *Audioscrobbler* dataset, *DBLP* co-author dataset, and the *Adknowledge* URL dataset are presented in Tables 7.1, 7.2, and 7.3, respectively. A discussion of results is performed below.

Analysis of the *Audioscrobbler* Dataset

When comparing the four variants for the Adsorption algorithm, it can be seen from Table 7.1 that Variant 2 had the highest MAP score compared to all other variants. The high MAP score from Variant 2 can be attributed to its ability to find good neighbors (users interested in the same artists as the current user and yet having a small degree) based on the normalized similarity scores. In addition, Variants 2 and 4 demonstrated better MAP scores compared to Variants 1 and 3, suggesting that normalizing the similarity score results in better performance for this dataset. Furthermore, it can be seen that Variant 2

Table 7.1: MAP scores from the four Adsorption variants for the Audioscrobbler dataset. The neighborhood size (k) and recommendation size (n) are set to 5 and 10, respectively; results are averaged over four runs.

Variant 1	Variant 2	Variant 3	Variant 4
0.0254	**0.0728**	0.0274	0.0507

was significantly better than Variant 4, indicating that for implicit feedback, use of Boolean preferences as opposed to actual user preferences may be more relevant when computing similarity. Finally, the slight increase in the MAP score of the Variant 3 as compared to Variant 1 may be attributed to the fact that Variant 3 considers additional information (infrequent or frequent users) as compared to Variant 1, when constructing user neighborhoods. With Variant 3 for neighborhood construction, recommendations can be generated for infrequent users who prefer a small number of items by using the relatively long histories of the frequent users (selected as neighbors in this variant). However, given the high median value (79) for this dataset, one drawback of this variant is that, for a user who preferred a reasonable number of items (e.g., 20), the variant still selects a frequent user (who might have varied interest for artists) as neighbor, and this probably affected the overall performance.

Analysis of the *DBLP* Co-Author Dataset

For this dataset, Variant 2 for neighborhood generation outperformed all other weight-based variants considered for Adsorption, similar to results for the *Audioscrobbler* dataset. This result is consistent for both increasingly-long and short-term histories for all five subsets (Table 7.2). Furthermore, the performance of Adsorption with neighborhoods constructed using Variant 4 was slightly weaker yet comparable to the performance of Adsorption with neighborhoods constructed using Variant 2, together suggesting that Boolean preferences are better than the actual counts, and that normalizing the similarity score is important to compute user neighborhoods. However, results showed that Variant 3 is not better than any

Table 7.2: MAP scores from the four Adsorption variants for the subsets of increasingly-long and short histories for the DBLP co-author dataset. The neighborhood size (k) and recommendation size (n) are set to 5 and 10, respectively. In the table, Var indicates a neighborhood Variant of Adsorption.

	Increasingly-long Histories				Short Histories			
Subset	**Var 1**	**Var 2**	**Var 3**	**Var 4**	**Var 1**	**Var 2**	**Var 3**	**Var 4**
0	0.0157	**0.0172**	0.0127	0.0168	0.0156	**0.0173**	0.0127	0.0169
1	0.0149	**0.0167**	0.0111	0.0166	0.0163	**0.0182**	0.0128	0.0180
2	0.0142	**0.017**	0.0104	0.0165	0.0155	**0.018**	0.0118	0.0179
3	0.0135	**0.0167**	0.0098	0.016	0.015	**0.018**	0.011	0.0173
4	0.0129	**0.0165**	0.0091	0.0163	0.0147	**0.0174**	0.0107	0.017

variants considered, suggesting that choosing neighbors from the frequent user set is not a good strategy for the *DBLP* co-author dataset.

Analysis of the *Adknowledge* URL Dataset

When analyzing results for the *Adknowledge* dataset in Table 7.3, it can be observed that the best results for Adsorption were obtained with Variant 3, which outperformed other variants in almost all cases for both increasingly-long and short histories. Furthermore, Variants 1 and 3 were better than Variants 2 and 4 in almost all cases, suggesting that normalizing the similarity score may not be a good strategy to construct user neighborhoods for this dataset. Finally, Variant 2 was better than Variant 4 for almost all subsets, similar to *Audioscrobbler* and *DBLP* results, reinforcing the conclusion that using Boolean preferences may be more helpful for implicit feedback datasets.

Table 7.3: MAP scores from the four Adsorption variants for the subsets of increasingly-long and short histories for the Adknowledge URL dataset. The neighborhood size (k) and recommendation size (n) are set to 5 and 10, respectively. In the table, Var indicates a neighborhood Variant of Adsorption.

	Increasingly-long Histories				Short Histories			
Subset	**Var 1**	**Var 2**	**Var 3**	**Var 4**	**Var 1**	**Var 2**	**Var 3**	**Var 4**
0	0.0099	0.0040	**0.0105**	0.0039	0.0098	0.0047	**0.0106**	0.0040
1	0.0072	0.0019	**0.0097**	0.0020	0.0078	0.0021	**0.0101**	0.0019
2	0.0010	0.0003	**0.0013**	0.0007	0.0023	0.0009	**0.0034**	0.0010
3	**0.0171**	0.0131	**0.0171**	0.0125	0.0173	0.0139	**0.0176**	0.0123
4	0.0121	**0.0156**	0.0124	0.0140	0.0123	**0.0161**	0.0123	0.0128

7.2 Evaluation of Community Detection-based Neighborhoods for Adsorption

The second goal of this study was to understand the usefulness of user communities in constructing neighborhoods for Adsorption algorithm. To accomplish this goal, the performance of Adsorption algorithm from neighborhoods constructed using community detection was compared to Adsorption performance from neighborhoods constructed using Variant 2 (baseline) [Parimi and Caragea, 2014]. Analysis of results is presented in this section.

7.2.1 Datasets

Two datasets were used to study the usefulness of community detection for constructing neighborhoods for the Adsorption algorithm. The first dataset was the *DBLP* co-author dataset [Ley, 2002; Kunegis, 2013], which is known to exhibit a community structure between authors. The author communities may correspond to collaborations based on a research area, the institution of employment of the authors, or both. The second dataset was the *BookCrossing* dataset [Ziegler et al., 2005]. For this dataset, although users can be clustered

based on the genre or author, of the books that they preferred, books typiclly have more that one genre associated with them (similar to movies), and users generally read books from multiple authors. Therefore, this dataset is believed not to exhibit a strong community structure.

The creation of training and test sets for these datasets were described in Sections 6.1.2 and 6.1.1, respectively. For the *DBLP* co-author dataset, because the performance of Adsorption using short user histories was equal to or better than Adsorption performance using long user histories (as shown in Table 7.2), only short user histories were used in this study for this dataset. Furthermore, Variant 2 was used as the baseline for this study because this way of creating neighborhoods proved to be better for the Adsorption algorithm when comparing various neighborhood variants for the *DBLP* dataset.

7.2.2 Research Questions

This study addressed the following research questions:

1. **Which datasets benefit from neighborhoods constructed using community detection?**

   Intuitively, community detection approaches may result in better neighborhoods for the Adsorption algorithm as compared to the neighborhoods constructed using the custom similarity functions for datasets with strong neighborhood relation between users. To verify this intuition, the performance of Adsorption using neighborhoods constructed using Variant 2 (baseline), described in Section 4.2.1, was compared to Adsorption performance from neighborhoods constructed using community detection (Variant 5), described in Section 4.2.2. This comparison was performed on two real-world recommendation tasks, one which is known to exhibit a strong community structure between users (*DBLP* dataset) and the other one which does not exhibit a strong community structure (*BookCrossing* dataset) [Parimi and Caragea, 2014].

2. **How does the performance of Adsorption algorithm vary with the threshold t on the edge weights of user-user graph G (t is used to control the number of edges in G, the input to community detection algorithm)?** By varying t, the amount of information encoded in the user-user graph G can be controlled. Intuitively, the recommendation accuracy increases as t increases, reaching a maximum for some value of t and then decreasing when t is further increased. This is because of the difficulty associated with finding meaningful clusters from dense graphs (i.e., when t is close to 0). As t increases, the density of the graph G decreases, making it easier to find meaningful clusters. However, when t nears 1, the graph G becomes very spare and the clusters identified will have users only with high edge weights and does not capture novel information. This intuition is verified by comparing the performance of the Adsorption algorithm from Variant 5 at different values for t.

3. **What is the impact of the number of neighbors k on the recommendation accuracy?** The expectation was to see a trend in the results similar to results observed when the threshold t is varied. Results were expected to improve as the number of users k is increased. However, when k is increased beyond a threshold, i.e., when too many users were considered as neighbors for a user, the recommendations become less personalized, thereby decreasing the performance of the algorithm. This intuition was verified by comparing performance of the Adsorption algorithm at different values for k from Variants 2 and 5.

7.2.3 Hyper-parameter Values

The hyper-parameters for the Adsorption algorithm are p_{inj}, p_{term}, p_{cont}, (random-walk probabilities) and k (number of neighbors). Three combinations of p_{inj}, p_{term}, and p_{cont} were tried in several train runs, and their values were fixed to (0, .85, .15). For the number of neighbors (k), five values were considered: (5, 10, 15, 20, 25); for the weight threshold parameter (t), four values were considered: (0, 0.25, 0.5, 0.75). The number of algorithm

recommendations (i.e., n) was fixed to 10.

7.2.4 Results and Discussion

Results for the *DBLP* co-author and *BookCrossing* datasets are presented in tables 7.4 and 7.5, respectively. A discussion of results is included in the following sections.

Analysis of *DBLP* Co-Author Dataset

As expected, the recommendation performance of the Adsorption algorithm with neighbors constructed using Variant 5 (community detection) outperformed the algorithm with neighborhood constructed using Variant 2 (baseline). This result, shown in Table 7.4, was consistent across all neighborhood sizes considered ($n = 5, 10, 15, 20, 25$) and for three out of the four thresholds considered ($t = 0$, $t = 0.25$, $t = 0.5$). In general, the co-author domain can be seen as a social network. Similar to how users in a social network make friends based on *friend-of-a-friend* relationship, authors in the co-author network often collaborate with other authors who have similar research interests and are acquaintances. Thus, a natural community structure exists between the authors in the co-author domain. This underlying community structure between the authors was captured using Variant 5 and used to select the top k neighbors for authors, thus achieving good MAP score. In the case of Variant 2, the intuition is to pick users (authors) who have some items (publications) in common with the current user (author), but have small degree. Although such authors are good neighbors to an author in the co-author domain, the global information i.e., information about the co-authors of the potential neighbor author is ignored while constructing the neighborhood. For example, consider three authors A, B, and C, as shown in Figure 7.1. Assume that A authored two papers, one with B and one with C, both related to the *Data Mining* research area. Also assume that B co-authored with three other authors in the research field of *Algorithms* and C co-authored with four other authors in the research area of *Data Mining*. In order to select a neighbor for A from B and C, Variant 2 picks the author who authored

Table 7.4: MAP scores from Adsorption Variant 2 (baseline) and Variant 5 (community detection) for the DBLP co-author dataset. The threshold (t) on edge weights for Variant 5 is varied from 0 *to* 0.75 *with a step size of* 0.25*. The neighborhood size (k) is varied from* 5 *to* 25 *with a step size of* 5*, and number of recommendations (n) is set to* 10*. For each k value, variant with the best MAP score is highlighted.*

#Neighbors		Variant 2	Variant 5			
			$t = 0$	$t = 0.25$	$t = 0.5$	$t = 0.75$
$k = 5$	subset 0	0.0175	0.0185	**0.0188**	0.0172	0.0128
	subset 1	0.0182	0.0200	**0.0201**	0.0190	0.0147
	subset 2	0.0180	0.0196	**0.0199**	0.0182	0.0142
	subset 3	0.0180	0.0195	**0.0201**	0.0187	0.0141
	subset 4	0.0174	0.0185	**0.0193**	0.0182	0.0138
$k = 10$	subset 0	0.0164	0.0185	**0.0197**	0.0182	0.0135
	subset 1	0.0172	0.0206	**0.0211**	0.0199	0.0152
	subset 2	0.0167	0.0200	**0.0207**	0.0190	0.0148
	subset 3	0.0168	0.0198	**0.0207**	0.0195	0.0146
	subset 4	0.0165	0.0186	**0.0197**	0.019	0.0144
$k = 5$	subset 0	0.0150	0.0184	**0.0197**	0.0185	0.0136
	subset 1	0.0158	0.0204	**0.0212**	0.0201	0.0154
	subset 2	0.0153	0.0198	**0.0207**	0.0192	0.0149
	subset 3	0.0156	0.0198	**0.0209**	0.0198	0.0147
	subset 4	0.0153	0.0183	**0.0197**	0.0192	0.0145
$k = 20$	subset 0	0.0143	0.0181	**0.0197**	0.0185	0.0136
	subset 1	0.0148	0.0205	**0.0210**	0.0201	0.0154
	subset 2	0.0143	0.0196	**0.0207**	0.0193	0.0148
	subset 3	0.0147	0.0195	**0.0208**	0.0198	0.0148
	subset 4	0.0143	0.0180	**0.0196**	0.0192	0.0145
$k = 25$	subset 0	0.0135	0.0180	**0.0197**	0.0185	0.0136
	subset 1	0.0140	0.0201	**0.0210**	0.0201	0.0154
	subset 2	0.0134	0.0195	**0.0206**	0.0193	0.0149
	subset 3	0.0140	0.0194	**0.0204**	0.0198	0.0147
	subset 4	0.0137	0.0179	**0.0196**	0.0192	0.0145

the least number of papers among B and C, in this case, B. However, in this example, C may be more relevant as a neighbor to A considering their shared interests and interests of co-authors of C in the *Data Mining* research area. With Variant 5, the expectation is that authors A and C, along with the co-authors of C, will be grouped into a community in which a majority of authors will be interested in *Data Mining*. Author B and co-authors of B will be grouped into a different community in which a majority of authors will be interested in *Algorithms*,.

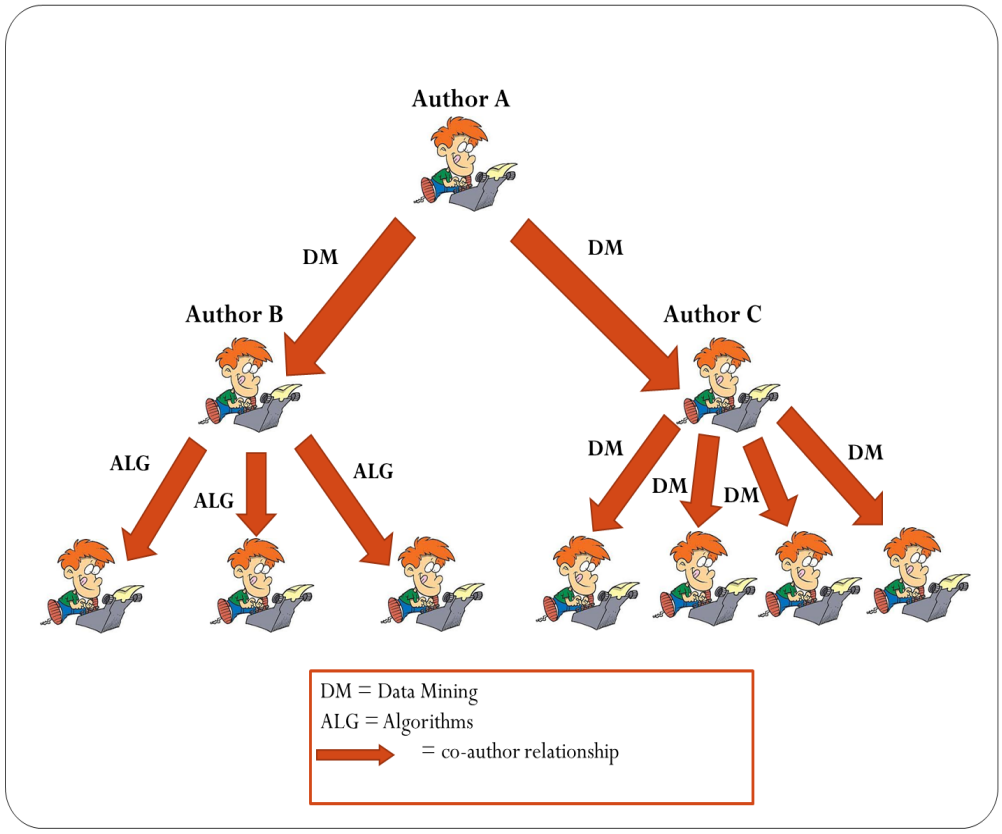


Figure 7.1: Collaboration relationships in the co-author network.

When varying threshold (t) on edge weights in the user-user graph G, results in Table 7.4 support the hypothesis that recommendation performance improves as t increases, reaching a maximum for some value of t, and then decreasing when t is further increased. In fact, this was true for all n values considered $(5, 10, 15, 20, 25)$. For example, consider the case when the number of neighbors $n = 5$ in Table 7.4, clearly the MAP scores for all subsets

increased when t was changed from 0 to 0.25, and decreased when t was further increased to 0.5 and then to 0.75. One possible explanation for this variation in the MAP scores from Adsorption relates to the difficulty associated with finding meaningful clusters from highly dense and highly sparse graphs. When a graph is the densest, i.e., each node is connected to every other node, there will only be one community for the entire graph. Similarly, when a graph is the sparsest, i.e., no node is connected to other nodes, each node will be a separate community. In both these cases, the community detection algorithm will not be helpful because it fails to identify meaningful communities. However, when density of the graph is varied, communities identified by the algorithm vary, resulting in changes in the MAP scores.

In order to study the effect of the number of neighbors (k) on recommendation performance, MAP scores of subsets 0 through 4 were compared to corresponding subsets for various k values, for both variants. It can be seen from the results in Table 7.4 that Adsorption performance decreased when k was varied from 5 to 25 for Variant 2. This suggests that Variant 2 can identify a small number of good neighbors, but as k increases, the recommendations become less personalized. Conversely, with Variant 5, the MAP scores increased when k increased, reached a maximum, and decreased as k was further increased for all subsets and t values considered, consistent with the hypothesis. For example, for $t = 0.25$, the MAP scores for all the subsets increased when k is varied from 5 to 10 and from 10 to 15. When the value of k was further increased, a decrease in the MAP scores could be observed in Table 7.4. Given these observations, this work suggests that Variant 5 is better suited for constructing neighborhood for the co-author domain.

Analysis of *BookCrossing* Dataset

When analyzing results for *BookCrossing* dataset in Table 7.5, it can be seen that the best results were achieved by Variant 2 (baseline), which outperformed Variant 5 (community

[1]Because the subsets for this dataset are created in a way similar to cross-validation, the average MAP score across the four subsets were reported.

Table 7.5: MAP scores from Adsorption Variant 2 (baseline) and Variant 5 (community detection) for the BookCrossing dataset. The threshold (t) on the edge weights for Variant 5 is varied from 0 *to* 0.75 *with a step size of* 0.25*. The neighborhood size (k) is varied from* 5 *to* 25 *with a step size of* 5*, and number of recommendations (n) is set to* 10*. For each k value, variant with the best MAP score is highlighted.* [1]

#Neighbors		Variant 2	Variant 5			
			$t = 0$	$t = 0.25$	$t = 0.5$	$t = 0.75$
$k = 5$	subset 0	0.0078	0.0052	0.0052	0.0057	0.0055
	subset 1	0.0085	0.0064	0.0066	0.0069	0.0074
	subset 2	0.0089	0.0065	0.0069	0.0077	0.0077
	subset 3	0.0088	0.0065	0.0067	0.0067	0.0079
	Average	**0.0085**	0.0062	0.0064	0.0068	0.0071
$k = 10$	subset 0	0.0082	0.0053	0.0052	0.0061	0.0061
	subset 1	0.0094	0.0061	0.0066	0.0067	0.0077
	subset 2	0.0083	0.0062	0.0060	0.0077	0.0075
	subset 3	0.0083	0.0061	0.0060	0.0067	0.0079
	Average	**0.0085**	0.0059	0.0060	0.0068	0.0073
$k = 15$	subset 0	0.0078	0.0053	0.0051	0.0062	0.0063
	subset 1	0.0086	0.0060	0.0067	0.0069	0.0079
	subset 2	0.0077	0.0059	0.0058	0.0076	0.0078
	subset 3	0.0074	0.0062	0.0060	0.0067	0.0082
	Average	**0.0079**	0.0059	0.0059	0.0069	0.0075
$k = 20$	subset 0	0.0078	0.0053	0.0052	0.0064	0.0064
	subset 1	0.0082	0.0057	0.0066	0.0069	0.0079
	subset 2	0.0072	0.0059	0.0057	0.0076	0.0079
	subset 3	0.0070	0.0063	0.0060	0.0065	0.0084
	Average	0.0075	0.0058	0.0059	0.0069	**0.0076**
$k = 25$	subset 0	0.0081	0.0052	0.0053	0.0065	0.0064
	subset 1	0.0081	0.0054	0.0064	0.0069	0.0080
	subset 2	0.0067	0.0059	0.0056	0.0077	0.0081
	subset 3	0.0068	0.0062	0.0060	0.0066	0.0084
	Average	0.0075	0.0057	0.0058	0.0069	**0.0077**

detection) in almost all cases. This result suggests that for the *BookCrossing* dataset, users with smaller degrees that have some items in common with the current user are more reliable neighbors than the neighbors from community detection. One explanation for this may be the high density of the user-user graph given as input to the community detection algorithm. In general, identification of meaningful clusters from dense graphs is difficult, and the number of communities identified from such graphs will be small, and each community has a large number of users. For this dataset, the average number of edges corresponding to each user in the user-user graph was approximately 550, 149, 38, and 12 when t was $0, 0.25, 0.5, 0.75$, respectively, compared to approximately 61, 19, 7, and 3 for the *DBLP* dataset. This reasoning was reinforced by the fact that the average number of users in each community ($35, 32, 12, 7$ for $t = 0, 0.25, 0.5, 0.75$, respectively) for *BookCrossing* dataset was greater than those in *DBLP* dataset ($10, 7, 6, 4$ for $t = 0, 0.25, 0.5, 0.75$, respectively).

As can be seen in Table 7.5, the MAP score increased when the threshold t on the edge weights was varied from 0.0 to 0.75. The increase in the MAP score justifies the hypothesis that the performance increases when t is increased. However, a maximum MAP score and then a decrease in the MAP score could not be observed for the t values considered, suggesting that the optimal value for t is domain dependent. This problem could be addressed by experimenting with t values greater than 0.75.

Finally, MAP scores for Variant 2 stayed the same or decreased when the number of neighbors k was varied from 5 to 25. This was consistent with the behavior observed for *DBLP* dataset with Variant 2 and reinforced the claim that Variant 2 can identify a small number of good neighbors. In the case of Variant 5, while the MAP scores decreased when k was increased from 5 to 25 for t values 0.0 and 0.25, a slight increase in MAP scores was observed for t values 0.5 and 0.75. This is because, at small values for t (dense user-user graphs), the communities identified are large and selecting too many users from such large communities makes the recommendations less personalized for the current user. However, at larger values for t (sparser user-user graphs), the identified communities are crisp and

compact, and the neighbors selected from such communities may be considered as good neighbors, leading to an improvement in MAP scores.

7.3 Adsorption versus Matrix Factorization

Another goal of this study in the single-domain setting was to understand how performance of the Adsorption algorithm, a neighborhood-based approach, compares with the performance of MF, a latent factor model-based approach for various recommendation tasks. This section discusses the datasets used, research questions addressed, and results to accomplish this goal.

7.3.1 Datasets

Datasets used to compare Adsorption variants (Section 7.1.1), i.e., the *Audioscrobbler*, *DBLP*, and *Adknowledge Inc.* datasets were also used for this study.

Variants 2 and 3 for Adsorption were considered for this study because Variant 2 proved to be better for the *Audioscrobbler* and *DBLP* datasets, and Variant 3 proved to be better for the A*Adknowledge* dataset according to the results described in Section 7.1.4

7.3.2 Research Questions

1. **Which type of collaborative filtering approach is better for each dataset?**

   In order to address this research question, performance of the Adsorption algorithm from Variants 2 and 3 for neighborhood construction were compared to the performance of MF on three implicit feedback datasets from various domains [Parimi and Caragea, 2015d].

Table 7.6: MAP scores from Variants 2 and 3 for the Adsorption algorithm and from MF for the Audioscrobbler dataset. The neighborhood size (k) and recommendation size (n) are set to 5 and 10, respectively. Results are averaged over four runs.

Variant 2	Variant 3	MF
0.0728	0.0274	0.0459

7.3.3 Hyper-parameter Values

For Adsorption, for all three datasets, the values for p_{inj}, p_{term}, and p_{cont} were fixed to (0, .85, .15). The number of neighbors (k) was fixed to 5. For MF, for parameters λ (regularization) and α (confidence rate), values (.1, 1) for (λ, α) were identified as best values for *Audioscrobbler* and *DBLP* datasets, and values (.01, 5) for (λ, α) were identified as best values for *Adknowledge* dataset based on several trial runs. The number of recommendations from the algorithms (n) was fixed to 10.

7.3.4 Results and Discussion

Results for the *Audioscrobbler* dataset, *DBLP* co-author dataset, and the *Adknowledge* URL dataset are presented in Tables 7.6, 7.7, and 7.8, respectively. A discussion of results is included in the following sections.

Analysis of the *Audioscrobbler* Dataset

When comparing Adsorption with MF, it can be seen from results in Table 7.6 that for the dense *Audioscrobbler* dataset, Adsorption using Variant 2 outperformed MF. One explanation for this is that, with a sufficiently large user history, Variant 2 identified good neighbors for users of this dataset. This explanation is also supported by the analysis of the Adsorption algorithm presented in [Baluja et al., 2008], which suggests that Adsorption performance increases as the amount of information about a user increases.

Analysis of the *DBLP* Co-Author Dataset

When comparing Adsorption with MF for the sparse *DBLP*, the results from Table 7.7 suggest that by carefully choosing neighbors using Variants 2 and 3, Adsorption can give better MAP scores compared to MF. In fact, MAP scores from both the variants of Adsorption were better than the MAP scores from MF for increasingly-long and short histories. One explanation for improved MAP scores from Adsorption when recommending co-authors relates to the way in which links are formed in a co-author network. Authors often collaborate with acquaintances as opposed to unknown authors. For a user, Adsorption recommends authors who co-authored with the k-nearest co-authors of a current user, and this is likely to result in good performance. This is not the case for MF, which recommends authors using latent factors computed from all users in the dataset.

Table 7.7: MAP scores from Variants 2 and 3 for the Adsorption algorithm and from MF for the subsets of increasingly-long and short histories for the DBLP co-author dataset . The neighborhood size (k) and recommendation size (n) are set to 5 and 10, respectively. In the table, Var indicates a neighborhood $Variant$ of Adsorption.

	Increasingly-long Histories			Short Histories		
Subset	**Var 2**	**Var 3**	**MF**	**Var 2**	**Var 3**	**MF**
0	**0.0172**	0.0127	0.0029	**0.0173**	0.0127	0.0029
1	**0.0167**	0.0111	0.003	**0.0182**	0.0128	0.003
2	**0.017**	0.0104	0.0023	**0.018**	0.0118	0.0024
3	**0.0167**	0.0098	0.0023	**0.018**	0.011	0.0025
4	**0.0165**	0.0091	0.0025	**0.0174**	0.0107	0.0025

Analysis of the *Adknowledge* URL Dataset

As can be seen in Table 7.8, for the *Adknowledge* dataset, MF results were significantly better than the Adsorption results. One reason for the weak performance of Adsorption compared to MF may be that the global user opinion may be very useful for dynamic applications

Table 7.8: MAP scores from Variants 2 and 3 for the Adsorption algorithm and from MF for the subsets of increasingly-long and short histories for the Adknowledge URL dataset. The neighborhood size (k) and recommendation size (n) are set to 5 and 10, respectively. In the table, Var indicates a neighborhood Variant of Adsorption.

	Increasingly-long Histories			Short Histories		
Subset	**Var 2**	**Var 3**	**MF**	**Var 2**	**Var 3**	**MF**
0	0.0040	0.0105	**0.0315**	0.0047	0.0106	**0.0307**
1	0.0019	0.0097	**0.0123**	0.0021	0.0101	**0.0132**
2	0.0003	0.0013	**0.0017**	0.0009	0.0034	**0.0044**
3	0.0131	0.0171	**0.0330**	0.0139	0.0176	**0.0303**
4	0.0156	0.0124	**0.0331**	0.0161	0.0123	**0.0307**

such as web pages which contain various types of information (e.g., news, movies, sports) and to which new information is added regularly. MF exploits the global user opinion by mapping all users to a common latent space. In contrast, Adsorption uses only the nearest neighbors and, as a consequence, does not produce accurate recommendations.

7.4 Short versus Long User Histories

The fourth goal of this study in the single-domain setting was to understand if short user histories can replace long user histories for recommender systems. To accomplish this goal, performance of the Adsorption algorithm and MF from short-user histories and long-user histories were compared. Analysis of results is presented in this section.

7.4.1 Datasets

Datasets used for this study were the *DBLP* and the *Adknowledge Inc.* datasets because these datasets contain timestamp information.

7.4.2 Research Questions

1. **Are short histories better than long histories for these application domains?**

   Intuitively, use of short user history may be more helpful because it reflects the most current interests of users. In order to verify this intuition, results from Adsorption and MF on the five subsets of long histories were compared to Adsorption and MF results on corresponding subsets of short histories for the *DBLP* and *Adknowledge* datasets [Parimi and Caragea, 2015d].

7.4.3 Results and Discussion

In order to study the effect of long-term versus short-term histories, MAP scores from increasingly-long histories were compared to MAP scores from short-histories for Subsets 1 through 4[2] as shown in Tables 7.7 and 7.8. Results suggest that the MAP scores from both Adsorption and MF for short-term histories are better than or very close to MAP scores for longer-term histories for both datasets. Furthermore, results showed that increasingly-longer histories do not give increased performance. Given these observations and the fact that shorter-term histories are computationally less expensive, it is suggested that short-term histories should be preferred to long-term histories. This conclusion is consistent with a similar conclusion reported in the literature [Yan et al., 2009].

7.5 Influence of Domain Knowledge

Another goal of this study was to analyze if domain knowledge, specifically about how links are formed in the domain (based on close connections - resulting in strong local neighborhoods, or based on loose connections - resulting in useful global information), or characteristics of the data (e.g., dense or sparse feedback matrices) can be useful to determine

[2]The training data for Subset 0 is identical in both histories. Hence, it was not used for this comparison. Also, the small variation in the results is due to randomly picking neighbors in case of ties for weight values.

the collaborative filtering approach to use and the neighborhood variant to use with the Adsorption algorithm was studied [Parimi and Caragea, 2015d]. The rest of this section discusses in detail the two research questions and the analysis of the experimental results in the single-domain setting in order to answer these research questions.

7.5.1 Research Questions and Analysis

1. **Does knowledge about data or domain characteristics help in choosing the neighborhood approach to use for Adsorption?**

   Intuitively, knowledge about the existence of strong neighborhood relation between users, density of the datasets, or characteristics of the users, e.g., *frequent* or *infrequent* users, may be useful in choosing the neighborhood approach to be used with Adsorption. This is verified by the performance comparison of the Adsorption algorithm from neighborhood construction variants on the *Audioscrobbler*, *DBLP*, and *Adknowledge* datasets.

   For the dense *Audioscrobbler* dataset (median number of items preferred is 79), the results in Table 7.1 suggest that information about *frequent* or *infrequent* users helped because the performance of Adsorption using Variants 2 and 3 was better than performance of Variant 1. Furthermore, results suggest that it is less beneficial to select frequent users as neighbors because the performance of Adsorption using Variants 1 and 4 is significantly better than using Variant 3. Intuitively, in a dense dataset, users have expressed a sufficient amount of item preferences to be able to build a good profile for them. Therefore, choosing a frequent user who is analogous to a popular item is not a good approach to select user neighborhood. Therefore, given the information about the dense nature of this dataset, the conclusion can be made that Variant 2 would be a more suitable choice for Adsorption.

   The second dataset considered for this study was the sparse *DBLP* co-author dataset in which the median number of items preferred by a user is 3. The small value

for median is intuitive because authors exhibit a strong neighborhood relation between them; authors often collaborate with acquaintances as opposed to unknown authors. For the co-author dataset, the results in Table 7.2 suggest that Variants 2 and 3 corresponded to the best and the worst approaches, respectively, in order to construct neighborhoods. Intuitively, using Variant 2, the algorithm tends to pick neighbors with some items in common with the current user, but having a small degree. In the co-author domain, such users (authors) are better neighbors compared to users who have interest in diverse fields (i.e., high degree). This reasoning is reinforced by the analysis of results in Table 7.4 for the co-author dataset, which suggests that community detection approaches (Variant 5) for neighborhood construction further improved the Adsorption performance as compared to Variant 2. However, Variant 3 gives priority to frequent users as neighbors for a current user. In the co-author domain, the frequent user neighbors may be very successful authors (possibly interdisciplinary) and there is no reason to believe that they would be interested in collaborating with an infrequent user (an author who does not have much publication history). Therefore, given the information about the sparse nature of this dataset and the knowledge that users in the co-author domain have strong neighborhood relation, the conclusion can be made that Variant 5 would be a more suitable choice for Adsorption.

Finally, for the sparse *Adknowledge* dataset (median number of items preferred is 1), results in Table 7.3 suggest that Variant 3 is the best approach to construct neighborhoods for Adsorption. This is intuitive given that global user opinion may be useful for recommending items in dynamic application domains, such as recommending web pages. Variant 3 captures this by prioritizing frequent clickers to generate neighborhood. Furthermore, given the sparse nature of this dataset, infrequent clickers are not as reliable neighbors as frequent clickers because they do not propagate a lot of information through the graph. This explanation is also supported by the MAP scores of Variant 2, which were surprisingly lower than MAP scores of Variant 1 for

almost all subsets. Therefore, given the information about the sparse nature of this dataset and the knowledge that global user information is more relevant in the web page recommendation task, the conclusion can be made that Variant 3 would be a more suitable choice for Adsorption.

The decision tree shown in Figure 7.2 summarizes the aforementioned analysis about choosing the neighborhood variant to use with Adsorption algorithm for a recommendation application.

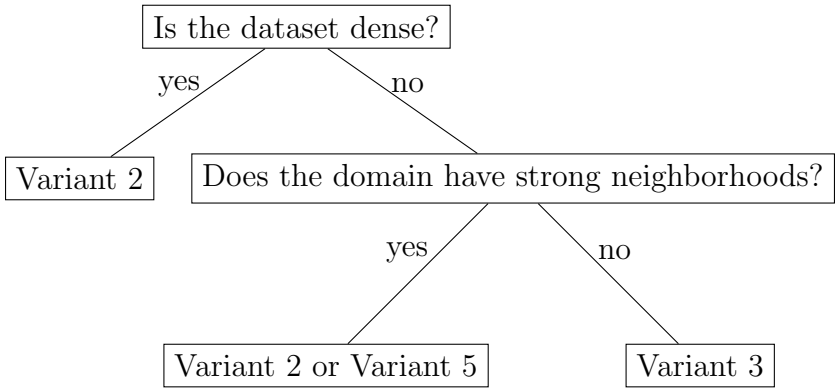


Figure 7.2: Decision tree based on domain knowledge and data characteristics to select a neighborhood variant for Adsorption algorithm.

2. **Does knowledge about data or domain characteristics help in choosing the collaborative filtering approach to use between Adsorption and Matrix Factorization?**

   Domain knowledge and knowledge about the density of the datasets may also be useful in choosing the CF approach to use in each application domain. This is verified by comparing the performances of the two CF approaches for the *Audioscrobbler*, *DBLP*, and *Adknowledge* datasets.

   Based on analysis of the Adsorption algorithm by Baluja et al. [2008] which suggests that Adsorption performance increases as the amount of information about users increases, and because of the dense nature of the *Audioscrobbler* dataset (neighbor-

hood approaches have smaller computational complexity [Hu et al., 2008; Koren et al., 2009; Sarwar et al., 2002]), it is intuitive that Adsorption may be preferable over MF for this dataset. The results reported in Table 7.6 support this intuition because Adsorption performance using Variant 2 was significantly better than the performance of MF.

For the sparse *DBLP* co-author dataset, Adsorption recommends authors who are acquaintances with the co-authors of a current user, while MF recommends authors using similarity of latent factors computed from all users in the dataset. Intuitively, given the strong neighborhood relation between authors, Adsorption is likely to result in good performance. This intuition is confirmed by from the results reported in Table 7.7. The performance of Adsorption using Variants 1, 2, and 3 was significantly better than the performance from MF.

Finally, for the sparse *Adknowledge* dataset, given the dynamic nature of the Web domain and the sparse nature of the dataset, user opinion captured globally (knowledge from all users) may be more relevant, and MF exploits the global user opinion by mapping all users to latent dimensions. This can be seen in the results reported in Table 7.7, in which MF showed better performance compared to Adsorption.

The decision tree shown in Figure 7.3 summarizes the aforementioned analysis about choosing the CF approach to use for a recommendation application.

7.6 Note about Results

Although the MAP scores reported in this study for the *Audioscrobbler*, *DBLP*, *Adknowledge*, and the *BookCrossing* datasets may seem low, it should be noted that the objective of this study in the single-domain setting was to recommend unknown items to users, as opposed to items that the user is already aware of (preferred in the past) but may prefer

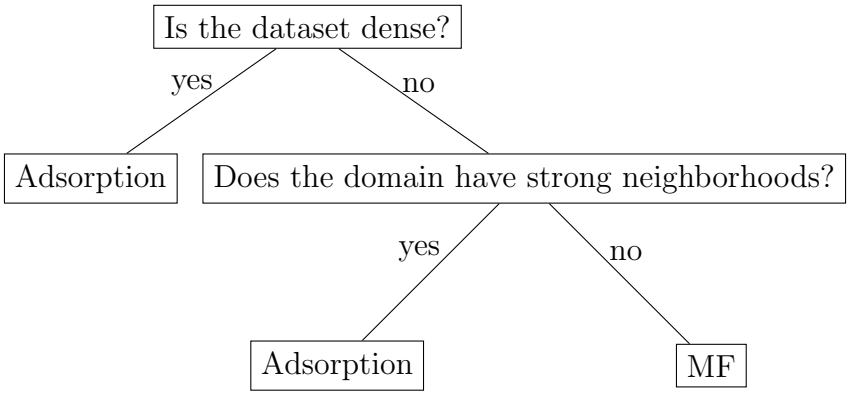


Figure 7.3: Decision tree based on domain knowledge and data characteristics to select the collaborative filtering approach to use between Adsorption and Matrix Factorization.

again in the future. Also, analysis of the datasets used in this study revealed that there are many users in the training data who preferred only one or two different items and there are many items in the training data which were preferred by only one or two different users, popularly known as the *Cold Start* problem. One common way to address this problem when evaluating the performance of a recommendation algorithm is to filter out users who preferred too few items and to filter out items that are preferred by too few users [Baluja et al., 2008; Ma et al., 2012; Yang et al., 2012]. However, given that one of the goals of this work was to study the influence of dataset characteristics such as, density of the dataset, information about frequent or infrequent users, on the performance of the recommendation algorithms users or items with small number of preferences were not filtered. This might be another reason for the small MAP scores reported for Adsorption and MF.

Chapter 8

Experimental Design and Results in the Cross-Domain Setting

This chapter is organized as follows: Sections 8.1 and 8.2 discuss the research questions, experiments designed to address the research questions, baseline approaches, evaluation metrics used, and results to evaluate the effectiveness of the two proposed Adsorption-based cross-domain approaches (WAN and WAR) and the MF-based cross-domain approach (CIMF), respectively. Sections 8.3 compares Adsorption-based and MF-based cross-domain approaches. Section 8.4explains results from Adsorption-based approaches and MF-based approaches for two overlap scenarios. Finally, Section 8.5 presents an analysis of results in order to understand if data and domain characteristics can influence the choice of cross-domain CF approach to use between Adsorption-based and MF approaches for a recommendation application.

8.1 Evaluation of Adsorption-based Cross-Domain Approaches

This study on Adsorption-based cross-domain approaches was primarily focused on the following three goals: *1*) understand the effectiveness of the proposed knowledge aggregation approaches, described in Section 5.3, in order to improve target recommendation accuracy, *2*) study the effectiveness of the proposed approach to compute conditional probability weights (CP weights) in order to address the weight selection problem for knowledge aggregation, and *3*) understand the usefulness of the proposed WAN and WAR approaches in order to address the cold-start user problem (make meaningful recommendations to users who have not expressed item preferences in the target domains). To accomplish these goals, the performance of WAN and WAR approaches (with manual and CP weights) and that of the baseline on six target recommendation tasks from two datasets, described in Section 6.2, were compared [Parimi and Caragea, 2015e,b]. Analysis of results is presented in this section.

8.1.1 Experimental Design

This section describes the research questions, experiments conducted in order to address the research questions, baseline used for the experiments to compare the proposed cross-domain knowledge aggregation approaches, the evaluation metric used, and hyper-parameters of the recommendation algorithms.

Baseline Approach

For this study of Adsorption-based knowledge aggregation approaches, the baseline corresponds to the performance of Adsorption using user preferences from only the target domain.

Research Questions and Experiments

1. **Are the proposed WAN and WAR approaches effective in improving target recommendation accuracy as compared to single-domain approaches?**

   In order to understand the effectiveness of the proposed WAN and WAR approaches, three experiments were conducted for each dataset. In each experiment, one domain was used as the target and the other two domains from the dataset were used as sources. For each target domain, performance of WAN and WAR approaches with five sets of manually chosen weights were compared to the baseline. Analysis of results is presented in Section 8.1.2.

2. **Are CP weights effective as target-source relative weights in the WAN and WAR approaches?**

   In order to address this question, six experiments, similar to the experiments for research Question 1 were conducted for the two datasets. For each target (other two domains correspond to source domains), the performance of WAN and WAR approaches with CP weights was compared to the performance of WAN and WAR approaches from the five sets of manually chosen weights (weights used in the experiments to address research Question 1) and the baseline. Analysis of results is presented in Section 8.1.3.

3. **Are the proposed WAN and WAR approaches effective in addressing the cold-start user problem (unacquainted-world scenario)?**

   In order to address this question, six experiments similar to experiments for research Question 1 were conducted. In each experiment, one domain was used as the target and the other two domains from the dataset were used as sources. However, additional processing steps were enforced, as described in Section 6.2.3, to ensure that training and test sets simulated the cold-start user problem in each target domain. For each target recommendation task, the highest performance of WAN and WAR from five sets of manually chosen weights and the CP weights were compared to the baseline. Analysis of results is presented in Section 8.1.4.

Evaluation Metrics

Performance of the recommendation algorithms was measured using Mean Average Precision (MAP@n) [Parimi and Caragea, 2015e; Shapira et al., 2013], as described in Section 6.3.

Parameter Settings

For the Adsorption algorithm, random-walk parameters (p_{inj}, p_{term}, p_{cont}) were fixed to (0, .85, .15) because these values proved good in the single-domain setting. For WAN and WAR approaches, five sets of (*target*, *source*) weights, (0.5, 0.25), (0.6, 0.2), (0.7, 0.15), (0.8, 0.1), (0.9, 0.05), were used. The number of neighbors (k) and the number of recommendations (n) were fixed to 5 and 10, respectively.

8.1.2 Results: WAN and WAR Approaches

Results for the Last.FM and DBLP datasets for the WAN and WAR approaches (using five sets of manually chosen weights and CP weights) are presented in Table 8.1. A discussion of results is included in the following sections.

Analysis of the Last.FM Dataset: It can be seen from the MAP scores in Table 8.1 (Rows I, II, and III) that, for all three domains of the *Last.FM* dataset, the WAN and WAR approaches outperformed the baseline approach. Furthermore, from Table 8.1, it can be seen that for almost all combinations of manual weights considered, the two cross-domain approaches demonstrated better MAP scores compared to the baseline, together indicating that the proposed cross-domain approaches were successful in improving target recommendation accuracy using knowledge from source domains.

Analysis of the DBLP Dataset: As can be seen from the results in Table 8.1 (Rows IV, V, and VI), in general, the WAN and WAR approaches outperformed the baseline approach for all target domains considered, similar to the *Last.FM* dataset, suggesting that these approaches are effective in various application domains. Furthermore, a considerable variation in the MAP scores from WAN and WAR approaches can be observed for the

Table 8.1: MAP@10 scores of the WAN and WAR approaches (for the five sets of manually chosen weights and CP weights) and of the baseline for the six target domains. The highest MAP value for a domain is highlighted in bold. Star (*) indicates the experiments in which the CP weights were better than the manual weights.

Target Network	Cross-Domain Setting							Baseline
	Approach	Manual Weights					CP Weights	$w_t = 1$ $w_i = 0$
		$w_t = 0.5$ $w_i = 0.25$	$w_t = 0.6$ $w_i = 0.2$	$w_t = 0.7$ $w_i = 0.15$	$w_t = 0.8$ $w_i = 0.1$	$w_t = 0.9$ $w_i = 0.05$		
I. Artist	WAN	0.0941	**0.0950**	0.0937	0.0938	0.0920	0.0935	0.0905
	WAR	0.0899	0.0923	0.0944	**0.0950**	0.0942	0.0885	
II. Friend	WAN	0.0568	0.0568	0.0563	0.0557	0.0532	**0.0574***	0.0517
	WAR	0.0517	0.0537	0.0548	0.0549	0.0546	0.0549*	
III. Tag	WAN	0.1066	0.1064	0.1068	0.1066	0.1070	0.1078*	0.1040
	WAR	0.1105	0.1105	0.1098	0.1084	0.1067	**0.1106***	
IV. Co-Author	WAN	0.0086	0.0101	0.0127	0.0169	0.0240	0.0320*	0.0331
	WAR	0.0342	**0.0349**	0.0348	0.0344	0.0338	0.0335	
V. Conference	WAN	0.0554	0.0513	0.0490	0.0481	0.0477	0.0522	0.0436
	WAR	**0.0806**	0.0789	0.0770	0.0748	0.0720	0.0745	
VI. Reference	WAN	0.0184	0.0239	0.0281	**0.0309**	0.0299	0.0272	0.0278
	WAR	0.0296	0.0297	0.0293	0.0288	0.0283	0.0295	

three domains of this dataset. This can be seen from the MAP scores in Table 8.1 (Rows IV, V, and VI), suggesting that choosing the right weights to be used for each domain for aggregation knowledge is essential in order to avoid negative transfer.

8.1.3 Results: WAN and WAR Approaches with CP Weights

It can be seen from the MAP scores for Last.FM tasks in Table 8.1 (Rows I, II, and III) that, in two of the three domains considered, the highest MAP score for cross-domain approaches was obtained using CP weights (experiments indicated by $*$). Furthermore in most cases, the MAP scores of the WAN and WAR approaches using CP weights were better than the baseline, together suggesting that the proposed approach to automatically compute weights

between two domains based on the alignment of the user-user networks is effective for this dataset.

As demonstrated by the MAP scores for DBLP tasks in Table 8.1 (Rows IV, V, and VI), in three of the six experiments, CP weights gave good results (comparable to the highest MAP score using manual weights). In four of the six experiments, the MAP scores using CP weights for WAN and WAR approaches were better than the MAP score of the baseline, together suggesting that CP weights are good also for this dataset.

8.1.4 Results: Ability of WAN and WAR Approaches to Handle Cold-Start Problem

Another goal of this work was to understand the usefulness of the proposed WAN and WAR approaches in order to overcome the cold-start user problem in the target domain. Towards this goal, the unacquainted world scenario, described in Section 6.2.3, was simulated for the six domains from the two datasets. Results for the Last.FM and DBLP datasets for the WAN and WAR approaches (using manual and CP weights) are presented in Table 8.2. A discussion of results is included in the following paragraphs.

For the six target recommendation tasks considered, the MAP scores in Table 8.2 indicate that the WAN and WAR approaches (with manual and CP weights) were successful in making recommendations to cold-start users in the target domains, whereas the baseline could not recommend any items to these users, indicated by the zero MAP score due to the absence of preferences from the cold-start users in the training data. This suggests the usefulness of the proposed approaches to address the cold-start user problem.

Furthermore, it can also be seen from the results in Table 8.2 that in most cases, MAP scores of WAN and WAR approaches using CP weights were comparable to (or better than) the highest MAP scores obtained from these approaches using the manual weights. This provides additional evidence regarding the effectiveness of the proposed CP weights as the target-source relative weights for knowledge aggregation, described in Section 8.1.3.

Table 8.2: The MAP@10 scores of WAN and WAR (shown are the highest MAP scores from the five sets of manual weights, and the MAP scores from the CP weights), and of the baseline for the six target domains in the unacquainted world scenario. Star (*) indicates experiments in which the CP weights were better than the manual weights.

Target Domain	Algorithm				
	WAN	WAN-CP	WAR	WAR-CP	Baseline
Artist	**0.0627**	0.0622	0.0563	0.0567*	0.0000
Friend	**0.0306**	0.0301	0.0126	0.0145*	0.0000
Tag	**0.0981**	0.0975	0.0912	0.0903	0.0000
Co-Author	0.0086	0.0101*	0.0098	**0.0109***	0.0000
Conference	0.0621	0.0596	**0.0640**	0.0625	0.0000
Reference	**0.0155**	0.0150	0.0111	0.0133*	0.0000

8.2 Evaluation of Matrix Factorization-based Cross-Domain Approach

An approach based on MF for implicit user feedback, CIMF, was also proposed to address the cross-domain recommendation problem described in Section 5.2. In order to understand the usefulness of the CIMF approach for improving target recommendation accuracy, performance of the CIMF approach was compared to performance of state-of-the-art single-domain CF approaches on six target recommendation tasks from two datasets, described in Section 6.2 [Parimi and Caragea, 2015a]. This corresponds to another goal of this work in the cross-domain setting.

8.2.1 Experimental Design

This section describes the research questions, experiments conducted to address the research questions, baselines used for the experiments, and hyper-parameters of the recommendation algorithms.

Baselines and Evaluation Metrics

Baselines for this study corresponded to three single-domain approaches that used user preferences only from the target: the MF approach for *explicit* feedback data proposed in [Zhou et al., 2008], the MF approach adapted for *implicit* feedback data (IMF) proposed in [Hu et al., 2008], and the item-based collaborative filtering approach (Item-CF) proposed in [Sarwar et al., 2001]. Implementations of MF, IMF, and Item-CF are part of the Apache Mahout software[1]. Note that when comparing different approaches, a controlled evaluation protocol was used as indicated in [Said and Bellogín, 2014], i.e., for all algorithms, the same training and test splits were used and the evaluation metrics, Mean Average Precision (MAP@n) [Parimi and Caragea, 2015e; Shapira et al., 2013], and Mean Recall@n described in Section 6.3 were computed in the same way.

Research Question and Experiment

1. **Is the proposed CIMF approach effective in improving recommendation accuracy in the target domains as compared to single-domain approaches?**

   In order to address this question, three experiments were conducted for each dataset. In each experiment, one domain was used as the target and the other two domains from the dataset were used as sources. For each target domain, performance of CIMF approach was compared to the performance of baseline approaches. Analysis of results is presented in Section 8.2.2.

Parameter Settings

Parameters of the algorithms were manually tuned, and best results obtained from combinations of various parameter settings were reported. For IMF, the following four combinations of (λ, α) were tried in the six domains: (.01, 1), (.01, 5), (.1, 1), and (.1, 5). Note that the values (.1, 5) for (λ, α) worked best for IMF in the three domains of the *Last.FM* dataset,

[1] http://mahout.apache.org/

and the values (.01, 5) for (λ, α) worked best for IMF in the three domains of the *DBLP* dataset and are reported in the results. For CIMF, the values .1 and .01 were used for λ (item regularization) for the *Last.FM* and the *DBLP* datasets, respectively, because these values proved to be best for IMF in the single-domain setting for corresponding datasets; for source user factors, different regularization parameter values (λ_k), specifically, $\{0.1, 0, 25, 0.5, 1, 5\}$, were tried. For MF, the values 0.1 and 0.01 for λ worked best for the three domains of the *Last.FM* and the *DBLP* datasets, respectively. Finally, the number of latent dimensions (f) for MF, IMF, and CIMF and the number of recommendations from the algorithm (n) were set to 50 and 10, respectively.

8.2.2 Results: CIMF Approach

Results for the Last.FM and DBLP datasets for the proposed CIMF cross-domain approach and three single-domain approaches (baselines) are presented in Table 8.3. A discussion of results is included in the following sections.

Analysis of the *Last.FM* Dataset

It can be seen from the MAP and Recall values in Rows I, II, and III in Table 8.3 for Artist, Friend, and Tag domains, respectively, that the CIMF approach of transferring user latent factors from source domains outperformed the baselines (single-domain approaches) in most cases considered. Although previous works on cross-domain approaches have suggested that cross-domain recommendations can be less precise compared to single-domain recommendations [Shapira et al., 2013; Winoto and Tang, 2008], the CIMF approach had better MAP scores in all three domains, with bigger improvements observed for the Friend and Tag domains and better Mean Recall values in two out of the three domains considered, for this dataset. These results suggest the effectiveness of the CIMF approach and confirm the intuition that information about related domains can help improve target recommendation accuracy. The CIMF approach captures the correlation between related domains through

latent user factors identified in each domain while controlling the amount of knowledge to use from each source domain through regularization parameters. Among the single-domain approaches, it can be seen from Table 8.3 (Rows I, II, and III) that performance of MF was significantly worse than performances of IMF and Item-CF for the three domains of the *Last.FM* dataset. This is consistent with a similar observation in literature that found that, factorizing a user-item preference matrix by assuming an implicit preference to be an explicit rating yields poor performance [Hu et al., 2008]. Between IMF and Item-CF, the results suggest that IMF is better than Item-CF for recommending friends and tags, but slightly worse than Item-CF for recommending artists, together suggesting that IMF is a better single-domain approach for the *Last.FM* dataset.

Table 8.3: The MAP@10 and Mean Recall@10 values of Item-CF, IMF, and MF (single-domain) and CIMF (proposed cross-domain approach) when the target domain is the Artist (I), Friend (II), Tag (III) in Last.FM, and Co-Author (IV), Conference (V), Reference (VI) in DBLP, respectively. For each target, the remaining two domains in the corresponding dataset are used as sources. The number of latent factors (f) is 50. Numbers in boldface correspond to the best results among the four methods.

Target Domain	Metrics	Without Transfer			With Transfer
		Item-CF	IMF	MF	CIMF
I. Artist	MAP	0.0658	0.0653	0.033	**0.0699**
	Recall	**0.1537**	0.1333	0.108	0.1380
II. Friend	MAP	0.0540	0.0771	0.0135	**0.0915**
	Recall	0.1249	0.1601	0.0318	**0.1925**
III. Tag	MAP	0.1038	0.1087	0.0134	**0.1459**
	Recall	0.2310	0.2003	0.0514	**0.2543**
IV. Co-Author	MAP	0.0340	0.0314	0.0251	**0.0357**
	Recall	**0.0824**	0.0702	0.0605	0.0799
V. Conference	MAP	0.0762	0.1017	0.0124	**0.1020**
	Recall	0.1674	**0.2014**	0.0373	**0.2014**
VI. Reference	MAP	0.0153	0.0470	0.0013	**0.0472**
	Recall	0.0347	0.0860	0.0041	**0.0866**

Analysis of the *DBLP* Dataset

As indicated from results in Table 8.3 (Rows IV, V, and VI), the CIMF approach was generally better than the single domain approaches for all three *DBLP* domains as well, with bigger improvements observed for the Conference and Reference domains, especially when comparing CIMF to Item-CF and MF approaches. When comparing CIMF with IMF, although good improvement was observed in the MAP and Mean Recall values for the Co-Author domain, a smaller increase was observed in these metrics for the Conference and Reference domains. Among the single-domain CF approaches, the IMF approach was slightly worse than the Item-CF approach when recommending co-authors, but was significantly better when the task was to recommend conferences and references. Weaker performance of IMF for the co-author domain probably relates to the way in which co-author relationships are formed in real-world. Authors often collaborate with acquaintances as opposed to unknown authors. For a user, Item-CF recommends authors who frequently co-authored with co-authors of the current user, while IMF recommends more global co-authors, potentially explaining why Item-CF may be a better choice between the two approaches for this domain. Finally, similar to what was observed for *Last.FM*, MF had the lowest MAP and Mean Recall values. This provides additional evidence to the observation in literature that, traditional factorization techniques that assume explicit user feedback yield poor performance when used to factorize a user-item matrix with implicit feedback [Hu et al., 2008].

8.3 Comparison of WAN, WAR, and CIMF Approaches

Another objective of this dissertation in the cross-domain study was to understand which approach is better between the two Adsorption-based knowledge aggregation approaches and the Matrix Factorization-based cross-domain approach. To accomplish this goal, the performance of these approaches on six target recommendation tasks from two datasets (including results from these approaches for the two overlap scenarios) were analyzed.

8.3.1 Research Questions and Analysis

1. **How does the performance of WAN approach compare to the performance of WAR approach?**

   In order to address this research question, the performance of these approaches for the six target recommendations tasks (including two overlap scenarios) was compared. An analysis of results is presented in Section 8.3.2.

2. **How does the performance of Adsorption-based cross-domain approaches compare to the performance of Matrix Factorization-based cross-domain approach?**

   In order to address this research question, the performance of WAN and WAR approaches for the six target recommendations tasks (including two overlap scenarios) was compared to the performance of the CIMF approach. An analysis of results is presented in Section 8.3.3.

8.3.2 Results: WAN versus WAR

When comparing WAN and WAR approaches, it can be seen from the results in Table 8.1, that for the three domains of the Last.FM dataset, MAP scores from both WAN and WAR approaches were comparable, indicating that either one of these approaches is a good choice to aggregate knowledge for the three domains of this dataset.

In the case of the DBLP dataset, it can be seen from the results in Tables 8.1 that for the Co-author and Conference domains, performance from WAR was significantly better as compared to WAN, and for the Reference domain, the performance of WAR was comparable to the performance from WAN. Together, these observations suggest that, for the three domains of the DBLP dataset, WAR may be a more suitable aggregation approach.

8.3.3 Results: Adsorption-based versus Matrix Factorization-based Approaches

When comparing WAN and WAR approaches with the CIMF approach for the three domains of the Last.FM dataset, it can be seen from the results in Tables 8.1 and 8.3 that, the CIMF approach was significantly better than the WAN and WAR approaches for the Friend and Tag domains. This is intuitive given that MF approaches demonstrate superior performance compared to neighborhood-based approaches. However, in contrast to the previous observation, when the target domain was the Artist domain, both WAN and WAR approaches had significantly better MAP scores compared to the MAP score from the CIMF approach. This is similar to results for the Audioscrobbler dataset in the single-domain setting (Section 7.3), suggesting that, for recommending artists, Adsorption may be a suitable approach.

In the case of the DBLP dataset, the results in Tables 8.1 and 8.3 indicate that the CIMF approach was significantly better than the WAN and WAR approaches for the Conference and Reference domains. However, for the Co-author domain, contrary to the intuition that neighborhood-based approaches have better performance for this domain, CIMF had slightly better MAP scores. This suggests that when additional information from auxiliary domains is available, CIMF may be a more suitable approach for the three domains of the DBLP dataset.

8.4 User Overlap Scenarios

As discussed in Sections 5.3 and 5.4, the two Adsorption-based cross-domain approaches (WAN and WAR) and the MF-based cross-domain approach (CIMF) require a partial user overlap between the source and target domains. Therefore, another goal of this work was to study the effectiveness of the proposed cross-domain approaches in order to utilize varying amounts of user overlap between source and target domains. To gain better insights into this,

performance of WAN and WAR approaches (using manual weights and the CP weights) and performance of the CIMF approach were studied by simulating two user overlap scenarios, as described in Sections 6.2.4 and 6.2.4, for six target recommendation tasks from two datasets with implicit user feedback.

8.4.1 Research Questions and Experiments

1. **Are the proposed WAN and WAR approaches effective in utilizing various amounts of user overlap between source and target domains?**

   In order to answer this research question, two user overlap scenarios, described in Sections 6.2.4 and 6.2.4, were simulated for the Last.FM and DBLP datasets. In each overlap scenario, for each target domain (the other two domains from the dataset were used as sources), three experiments were conducted; each experiment corresponded to a specific percentage of users overlapping between the target and the two source domains (25%, 50%, and 75% in Overlap Scenario I; 25%, 50%, and 100% in Overlap Scenario II). For each experiment, the performance of WAN and WAR (highest performance from the five sets of manually chosen weights, explained in Section 8.1.1, and CP weights) were compared to the baseline performance (Adsorption executed only on the target domain). Analysis of the results is presented in Section 8.4.2.

2. **Is the proposed CIMF approach effective in utilizing various amounts of user overlap between source and target domains?**

   In order to answer this research question, experiments similar to experiments described in research Question 1 were conducted for the two overlap scenarios. For each experiment, performance of the CIMF approach was compared to the performance of the IMF approach (baseline). Analysis of the results is presented in Section 8.4.3.

8.4.2 Results: Adsorption-based Cross-Domain Approaches

This section discusses results from the two Adsorption-based cross-domain approaches and the baseline, including analysis of results for Overlap Scenario I and Overlap Scenario II.

Performance Analysis for User Overlap Scenarios I

Table 8.4: The MAP@10 scores of WAN and WAR (highest MAP scores from the five sets of manual weights, and the MAP scores from the CP weights), and of the baseline, when user overlap between sources and target is varied from 25% to 50%, and to 75% according to Overlap Scenario I. Star (*) indicates experiments in which the CP weights were better than the manual weights.

Target Domain	Overlap%	Algorithm				
		WAN	WAN-CP	WAR	WAR-CP	Baseline
Artist	25%	0.0865	**0.0875***	0.0871	0.0870	0.0865
	50%	0.0876	0.0870	**0.0877**	0.0867	0.0867
	75%	0.0926	0.0911	**0.0937**	0.0905	0.0920
Friend	25%	0.0407	0.0408*	0.0429	**0.0430***	0.0429
	50%	**0.0514**	0.0508	0.0506	0.0499	0.0499
	75%	**0.0535**	0.0527	0.0516	0.0515	0.0511
Tag	25%	**0.1067**	0.1054	0.1054	0.1051	0.1039
	50%	0.1039	0.1048*	**0.1063**	0.1050	0.1025
	75%	0.1028	0.1028*	**0.1061**	0.1060	0.1027
Co-Author	25%	0.0275	0.0290*	**0.0292**	0.0291	0.0290
	50%	0.0288	0.0304*	**0.0307**	0.0304	0.0303
	75%	0.0304	0.0317*	**0.0322**	0.0315	0.0314
Conference	25%	0.0433	0.0439*	**0.0487**	0.0468	0.0413
	50%	0.0488	0.0453	**0.0615**	0.0578	0.0426
	75%	0.0522	0.0466	**0.0733**	0.0680	0.0427
Reference	25%	**0.0234**	0.0231	0.0233	0.0231	0.0230
	50%	0.0260	**0.0263***	0.0256	0.0253	0.0246
	75%	0.0293	**0.0304***	0.0277	0.0276	0.0262

Results for the Last.FM and DBLP datasets for the proposed WAN and WAR cross-domain approaches are presented in Table 8.4. A discussion of results is included below.

From the results reported in Table 8.4, it can be observed that for each target domain, in a majority of the overlap percentages, the WAN and WAR approaches with manual and CP weights had better performance as compared to the baseline (with the exception of WAN using manual weights for the co-author domain). However, when the overlap percentage was 25%, in many cases, the performance improvement of WAN and WAR as compared to the baseline was very small (or even negative), suggesting that the proposed approaches may not be able to transfer knowledge from a source when the overlap between source and target is small.

Performance Analysis for User Overlap Scenarios II

The motivation behind simulating this overlap scenario, as described in Section 6.2.4, was to study if the performance of the proposed cross-domain approaches increased with an increase in the overlap percentage between the source domains and the target (not studied in Overlap Scenario I), in addition to finding additional experimental evidence regarding the effectiveness of the proposed cross-domain approaches in order to utilize various amounts of user overlap. Results for the Last.FM and DBLP datasets for the WAN and WAR approaches, and the baseline are presented in Table 8.5. A discussion of results is included below.

From the results in Table 8.5, it can be seen that in four of the six target recommendation tasks (Artist, Tag, Conference, Reference), for a majority of the overlap percentages considered, the WAN and WAR approaches using manual and CP weights had better accuracy as compared to the baseline. However, for the Friend and Co-author target domains, MAP scores from both WAN and WAR approaches were only slightly better (or worse in some cases) for various overlap percentages. These observations, in addition to results for the Co-author and the Friend domains in Table 8.4, suggest that, for these target domains, knowledge from the source domains may not always be useful for improving recommendation accuracy. Furthermore, when the percentage user overlap between the source and

Table 8.5: The MAP@10 scores of WAN and WAR (highest MAP scores from the five sets of manual weights, and the MAP scores from the CP weights), and of the baseline, when user overlap between sources and target is varied from 25% to 50%, and to 100% according to Overlap Scenario II. Star (*) indicates experiments in which the CP weights were better than the manual weights.

Target Domain	Percentage Overlap	Algorithm				
		WAN	WAN-CP	WAR	WAR-CP	Baseline
Artist	25%	**0.0931**	0.0913	0.0927	0.0927*	0.0919
	50%	**0.0946**	0.0927	0.0937	0.0940*	
	100%	0.0985	**0.1013***	0.0964	0.0922	
Friend	25%	0.0449	**0.0453***	0.0448	0.0446	0.0447
	50%	**0.0453**	0.0441	0.0449	0.0450*	
	100%	**0.0459**	0.0446	0.0446	0.0447*	
Tag	25%	0.1280	0.1242	**0.1281**	**0.1281***	0.1279
	50%	0.1303	0.1281	**0.1317**	0.1312	
	100%	0.1246	0.1202	**0.1361**	0.1360	
Co-Author	25%	0.0264	0.0272*	**0.0273**	**0.0273***	**0.0273**
	50%	0.0253	0.0272*	**0.0273**	**0.0273***	
	100%	0.0240	0.0271*	**0.0280**	0.0275	
Conference	25%	0.0411	0.0418*	**0.0436**	**0.0436***	0.0410
	50%	0.0432	0.0428	**0.0495**	0.0484	
	100%	0.0480	0.0455	**0.0672**	0.0623	
Reference	25%	**0.0210**	0.0208	0.0206	0.0206*	0.0205
	50%	**0.0213**	0.0213*	0.0210	0.0210*	
	100%	0.0230	**0.0238***	0.0223	0.0222	

target was 25%, the performance improvement as compared to the baseline was small (or even negative) in many cases, similar to results reported in Table 8.4, thereby reinforcing the observation for Overlap Scenario I that the proposed approaches may not be able to transfer knowledge from a source when the overlap between source and target is small.

In order to understand if the performance of WAN and WAR approaches increases with an increase in the percentage of user overlap, for each target domain, the MAP scores from these approaches at 25%, 50%, and 100% user overlap were compared with each other. As can be seen from the results in Table 8.5, in only two target tasks (Conference and Ref-

erence), the MAP scores from both WAN and WAR approaches (using manual and CP weights) increased as the user overlap increased. In the remaining target recommendation tasks (Artist, Friend, Tag, Co-Author), the MAP score increased for either WAN or WAR approaches (with either manual or CP weights) with an increase in the user overlap percentage, suggesting that an increase in the performance of an approach with an increase in user overlap may not always be possible.

One reason for irregular MAP scores from the cross-domain approaches may be because of the differences in user preferences in source and target domains. When source and target domains are significantly different, at smaller overlap percentages, less noise may be transferred from source to target domains (because the number of users in source domains is small). However, at higher percentages of user overlap, the number of users in source domains is larger and more noise may be transferred from source to target domains. One way to test this hypothesis is to experiment with a larger set for manual weights and investigate if an increase in MAP score can be observed with an increase in overlap percentage for various combinations of weights.

8.4.3 Results: Matrix Factorization-based Cross-Domain Approaches

This section discusses results from the CIMF and baseline approaches, including analysis of results for Overlap Scenario I and Overlap Scenario II.

Performance Analysis for User Overlap Scenarios I

Results for the Last.FM and DBLP datasets for the proposed CIMF cross-domain approach are presented in Table 8.6. A discussion of results is included in the following paragraph.

From the results reported in Table 8.6, it can be seen that the CIMF approach had better performance compared to IMF across all percentages of overlap and for all target domains considered. This suggests that the CIMF approach can handle varying user overlap percentages between source and target domains and that it is effective in improving target

Table 8.6: MAP@10 scores of IMF and CIMF for the six target domains considered when user overlap between sources and target is varied from 25% to 50% and to 75%, as described in Section 6.2.4.

Target Domain	Algorithm	Percentage User Overlap		
		25%	50%	75%
Artist	IMF	0.0651	0.0660	0.0669
	CIMF	**0.0682**	**0.0693**	**0.0690**
Friend	IMF	0.0651	0.0756	0.0773
	CIMF	**0.0705**	**0.0791**	**0.0847**
Tag	IMF	0.1008	0.1065	0.1077
	CIMF	**0.1351**	**0.1381**	**0.1460**
Co-Author	IMF	0.0331	0.0324	0.0317
	CIMF	**0.0342**	**0.0351**	**0.0355**
Conference	IMF	0.0992	0.0994	0.1012
	CIMF	**0.0995**	**0.1001**	**0.1015**
Reference	IMF	0.0469	0.0470	0.0463
	CIMF	**0.0470**	**0.0472**	**0.0466**

recommendation accuracy as compared to IMF. Although the performance improvement of CIMF relative to IMF was considerable for the Artist, Friend, Tag, and Co-Author domains, the improvement was smaller for the Conference and Reference domains across all overlap percentages. This is similar to what was observed in Table 8.3, suggesting that, for Conference and Reference domains, the CIMF approach may not be as effective as it is for the other four domains.

Performance Analysis for User Overlap Scenarios II

Several experiments were conducted in order to evaluate the performance of the CIMF and IMF approaches at various amounts of user overlap for this scenario, similar to the experiments with the Adsorption-based cross-domain approaches. Results for the Last.FM and DBLP datasets for the CIMF and IMF approaches are presented in Table 8.7. A discussion of results is included below.

It can be seen from the results in Table 8.7 that for all target recommendation tasks and

Table 8.7: MAP@10 scores of IMF and CIMF for the six target domains considered when user overlap between sources and target is varied from 25% to 50% and to 100%, as described in Section 6.2.4.

Target Domain	% User Overlap	Algorithm	
		CIMF	IMF
Artist	25%	**0.0673**	0.0610
	50%	**0.0654**	
	100%	**0.0673**	
Friend	25%	**0.0728**	0.0718
	50%	**0.0765**	
	100%	**0.0794**	
Tag	25%	**0.1534**	0.117
	50%	**0.1504**	
	100%	**0.1524**	
Co-Author	25%	**0.0332**	0.0330
	50%	**0.0344**	
	100%	**0.0372**	
Conference	25%	**0.0962**	0.0960
	50%	**0.0961**	
	100%	**0.0964**	
Reference	25%	**0.0464**	0.0462
	50%	**0.0465**	
	100%	**0.0466**	

for all overlap percentages considered, the performance of the CIMF approach was better than the performance of the baseline. This is consistent with results from these approaches for Overlap Scenario I, reported in Tabls 8.6, and provides additional evidence regarding the effectiveness of the CIMF approach to transfer knowledge from source domains with varying amounts of overlapping users with the target. Furthermore, for the Conference and Reference domains, the performance improvement of CIMF relative to IMF was smaller, similar to results in Table 8.6, suggesting that, for these domains, the CIMF approach was unable to leverage from the auxiliary domains.

Finally, as shown in Table 8.7, in three of the six recommendation tasks (Friend, Co-Author, Reference), performance of the CIMF approach increased as the number of over-

lapping users between the source and target domains increased from 25% to 50%, and then to 100%. However, for the Artist, Tag, and Conference domains, performance of CIMF approach decreased when the user overlap increased from 25% to 50% and increased when the user overlap increased from 50% to 100%. This suggests that an increase in user overlap percentage may not always result in an increase in the performance of CIMF approach, similar to results from Adsorption-based cross domain approaches for User Overlap Scenario II.

8.5 Usefulness of Domain Knowledge

Another goal of this study was to analyze if knowledge about the data domain (e.g., close connections versus loose connections among users) or characteristics of the dataset (e.g., density of the feedback matrix) can be useful in selecting the most suitable cross-domain CF approach to use for a particular recommendation problem, similar to the study in the single-domain setting. The rest of this section discusses in detail this research question and presents an analysis of experimental results from Tables 8.1 and 8.3 in the cross-domain setting to answer the research questions.

8.5.1 Research Question and Analysis

1. **Does knowledge about data or domain characteristics help in choosing the cross-domain approach to use among Adsorption-based and MF-based cross-domain approaches?**

   As hypothesized, for the dense Artist domain (median number of items preferred is 33), Adsorption-based cross-domain approaches seems to be a more suitable choice to recommend artists. Similarly, for the sparse Tag (median items preferred is 9) and Friend domains (median items preferred is 6) that exhibit loose neighborhood relations, the proposed CIMF approach had significantly better performance as compared to the

Adsorption-based cross-domain approaches.

Similarly, for the DBLP dataset, as hypothesized, analysis of the results suggests that, for the sparse Conference domain (median items preferred is 4), the CIMF had significantly better performance as compared to the Adsorption-based cross-domain approaches. However, contrary to the hypothesis that Adsorption-based cross-domain approaches may be a better approach for the Co-author domain with strong neighborhood relation and for the dense Reference domain, CIMF had better performance for both domains. One reason for the improved performance of CIMF approach as compared to Adsorption-based approaches for the Co-author and Reference domains may be attributed to a source domain that exhibits loose neighborhood relations among users, and the CIMF approach may be more successful in capturing global user information as latent factors and transferring the latent factors to the target domain.

It has to be noted that stronger conclusions regarding the usefulness of domain knowledge and data characteristics to determine the CF approach to use cannot be made in this study because of the following reason: knowledge is transferred from more than one source domain with various domain characteristics to the target domain. In order to gain better insights into the usefulness of domain and data knowledge to select the cross-domain CF approach to use for a recommendation task, a study with more controlled experimental setting for knowledge transfer must be conducted.

Chapter 9

Conclusions and Future Work

This dissertation is concluded by summarizing the important contributions and significance of the achieved results and conclusions drawn thereof. An outlook on further research directions for future work is also provided.

9.1 Summary and Conclusions

The motivation for this dissertation was driven by current developments on the Web, where people often have to cope with large amounts of information that is increasing daily, commonly referred to as the information overload problem. Unfortunately humans are unable to sift through all the information manually in order to find relevant resources. Automated methods, such as recommender systems, are therefore useful to facilitate the information-seeking process by considering user preferential behavior expressed in terms of implicit feedback. Based on this motivation, the work described in this dissertation primarily focused on identifying the influence of data and domain characteristics on the performance of two state-of-the-art CF approaches, Adsorption and MF, in the single-domain setting. Furthermore, given the heterogeneity of online information and the fact that user interests span many types of items from various domains, the important contributions of this dis-

sertation in the cross-domain setting include the proposed Adsorption-based and Matrix Factorization-based cross-domain approaches. The rest of this section summarizes contributions and important conclusions of this work in the single-domain (Section 9.1.1) and cross-domain settings (Section 9.1.2).

9.1.1 Summary and Conclusions of Single-Domain Study

The work in the single-domain setting summarized the experiences and lessons learned from experiments with two state-of-the-art collaborative filtering approaches, Adsorption and Matrix Factorization, and three implicit feedback datasets from three different domains. Overall, the study showed that knowledge about the characteristics of the domain and of specific datasets can be used to guide an analyst towards the most appropriate algorithm to use, thus saving valuable time [Parimi and Caragea, 2015d]. Furthermore, this study investigated the application of three simple custom similarity functions and modularity-based community detection techniques [Parimi and Caragea, 2014, 2015d] in order to generate user neighborhoods for Adsorption. Experimental results from three real-world datasets from various domains suggested that domain knowledge and knowledge about the user characteristics in the dataset (frequent and infrequent user information) can also be useful for selecting the neighborhood approach for Adsorption. Finally, for data domains with timestamps, the results showed no decrease in performance when using short-histories compared to long-histories. Moreover, for large-scale recommender systems, short-histories are computationally more feasible, suggesting that they should be preferred over long-histories, irrespective of the CF approach [Parimi and Caragea, 2015d].

9.1.2 Summary and Conclusions of Cross-Domain Study

The work on cross-domain recommender systems described in this dissertation was primarily focused on the following hypothesis:

User preferences from several auxiliary domains can be leveraged to improve user personalization accuracy in one domain.

To verify this hypothesis, several assumptions of existing approaches for cross-domain recommender systems were identified [Parimi and Caragea, 2015e,a,b], and two knowledge aggregation approaches, specifically aggregation of neighborhoods (WAN) and aggregation of recommendations (WAR) that require a partial user overlap between domains, were proposed based on the Adsorption algorithm used in single-domain setting [Parimi and Caragea, 2015e,b]. Analysis of results on six target domains from two datasets suggested the usefulness of the two proposed approaches for improving target accuracy as compared to the single-domain approach. The analysis also suggested that the amount of information transferred from different domains should be carefully controlled through the use of weights in order to avoid performance decrease.

In order to address the problem of manually determining weights for source and target domains when aggregating knowledge, a solution based on the alignment score of user neighborhoods was proposed [Parimi and Caragea, 2015b]. Analysis of results on six target recommendation tasks suggested that the proposed approach to compute weights was effective in improving target recommendation accuracy as compared to single-domain approach, and was comparable to the WAN and WAR approaches with manual weights.

Furthermore, the performance of the two Adsorption-based approaches was studied in the unacquainted world scenario and in various user overlap scenarios [Parimi and Caragea, 2015b]. Experimental results indicated that the approaches were highly effective in overcoming the cold-start user problem in the target domain. Results also showed that the two proposed approaches can be effective when reasonable user overlap (50% according to the results) is present between source and target domains [Parimi and Caragea, 2015b,c].

Finally, a cross-domain matrix factorization approach was also proposed in order to ad-

dress the cross-domain recommendation problem and to verify the hypothesis about leveraging knowledge from multiple domains in order to improve recommendation accuracy in one domain [Parimi and Caragea, 2015a]. The proposed CIMF approach first identifies user and item latent factors in the source domains and then integrates the source user latent factors into the target through a regularization technique. Also, the algorithm requires only a partial user overlap as opposed to a complete user overlap between source and target domains. The experimental study on six target recommendation tasks showed that the proposed approach was effective in improving the MAP scores in all target domains considered, as compared to single-domain approaches. Furthermore, the results from two user overlap scenarios suggested that the CIMF approach can utilize varying amounts of user overlap between different domains in order to improve recommendation accuracy [Parimi and Caragea, 2015a,c].

9.2 Future Work

The directions to continue the research presented in this dissertation are manifold. This section discusses some of these directions based on the following perspectives: approach and experimental design.

9.2.1 Approach Perspective

First, potential extensions to the cross-domain approaches proposed in this work are discussed. The CIMF approach proposed in this work, described in Section 5.4, to address the cross-domain recommendation problem, described in Section 5.2, is a bi-factorization technique, i.e., the user-item preference matrix given as input to the algorithm is partitioned into two matrices: a user-factor matrix and an item-factor matrix. However, the literature on matrix factorization techniques for recommender systems suggests that the bi-factorization technique used in this work integrates both domain dependent and inde-

pendent semantic concepts into user and item latent factors [Hu et al., 2008; Koren et al., 2009; Pan et al., 2010]. In order to capture the user and item latent factors and to absorb the domain dependent effects of a preference matrix into a separate matrix, approaches such as [Pan et al., 2010; Li et al., 2009a; Gao et al., 2008] have used a variation of orthogonal non-negative tri-factorization techniques proposed by Ding et al. [2006] to transfer knowledge from sources to target. However, as described in Section 5.1.1, these approaches factor the user-item preference matrix, assuming explicit feedback, whereas the work in this dissertation on cross-domain recommender systems assumed implicit user feedback in both the source and target domains. Therefore, the goal is to extend the tri-factorization techniques used in [Ding et al., 2006; Pan et al., 2010] and adapt it to make use of implicit user feedback by representing the feedback value using two variables, the confidence and preference variables, as described in Section 5.4. Another future research direction is to design hybrid cross-domain approaches based on Adsorption-based and MF-based cross-domain approaches, and evaluate their effectiveness for various target recommendation problems.

9.2.2 Experimental Design Perspective

In order to gain additional insights about the effectiveness of the two Adsorption-based cross-domain approaches and the Matrix Factorization-based cross-domain approach, the goal is to extract and experiment with more large-scale (more users, items, and user-item preferences) cross-domain datasets. Furthermore, motivated by the work in [Cremonesi et al., 2011] that suggested that the density of the preference matrix influences the performance of a cross-domain recommendation approach, another future goal would be to simulate various sparsity levels in the source and target preference matrices and experiment with the proposed cross-domain approaches in order to understand their effectiveness at different sparsity levels.

Finally, the approach proposed in Section 5.3.3 to automatically compute weights depends on alignment of user-user graphs between a pair of domains. However, a threshold on user-user similarity (zero in this work) can be used to control the number of edges in a

graph, consequently affecting the conditional probability weights. Therefore, another future goal is to experiment with various thresholds to investigate if changing the threshold leads to an improvement in MAP scores for the WAN and WAR approaches.

Bibliography

Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749.

Anastasakos, T., Hillard, D., Kshetramade, S., and Raghavan, H. (2009). A collaborative filtering approach to ad recommendation using the query-ad click graph. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 1927–1930. ACM.

Balabanović, M. and Shoham, Y. (1997). Fab: Content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72.

Baluja, S., Seth, R., Sivakumar, D., Jing, Y., Yagnik, J., Kumar, S., Ravichandran, D., and Aly, M. (2008). Video suggestion and discovery for youtube: Taking random walks through the view graph. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 895–904.

Bell, R., Koren, Y., and Volinsky, C. (2007). Modeling relationships at multiple scales to improve accuracy of large recommender systems. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, pages 95–104. ACM.

Bell, R. M. and Koren, Y. In *Proceedings of the 1st KDDCup and Workshop*, 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining., San Jose, California.

Bennett, J., Lanning, S., and Netflix, N. (2007). The netflix prize. In *In KDD Cup and Workshop in conjunction with ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Berkovsky, S., Kuflik, T., and Ricci, F. (2007a). Cross-domain mediation in collaborative filtering. In *User Modeling 2007*, volume 4511 of *Lecture Notes in Computer Science*, pages 355–359. Springer Berlin Heidelberg.

Berkovsky, S., Kuflik, T., and Ricci, F. (2007b). Distributed collaborative filtering with domain specialization. In *Proceedings of the 2007 ACM Conference on Recommender Systems*, RecSys '07, pages 33–40. ACM.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, P10008:1–12.

Brandes, U., Delling, D., Gaertler, M., Gorke, R., Hoefer, M., Nikoloski, Z., and Wagner, D. (2008). On modularity clustering. *IEEE Trans. on Knowl. and Data Eng.*, 20(2):172–188.

Breese, J. S., Heckerman, D., and Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, UAI'98.

Burke, R., Vahedian, F., and Mobasher, B. (2014). Hybrid recommendation in heterogeneous networks. In *User Modeling, Adaptation, and Personalization*, volume 8538 of *Lecture Notes in Computer Science*, pages 49–60. Springer International Publishing.

Cantador, I., Brusilovsky, P., and Kuflik, T. (2011). 2nd workshop on information hetero-

geneity and fusion in recommender systems. In *Proceedings of the 5th ACM conference on Recommender systems (HetRec 2011)*, RecSys 2011.

Cantador, I. and Cremonesi, P. (2014). Tutorial on cross-domain recommender systems. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, pages 401–402. ACM.

Cremonesi, P., Koren, Y., and Turrin, R. (2010). Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 39–46.

Cremonesi, P. and Quadrana, M. (2014). Cross-domain recommendations without overlapping data: Myth or reality? In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, pages 297–300. ACM.

Cremonesi, P., Tripodi, A., and Turrin, R. (2011). Cross-domain recommender systems. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops*, ICDMW '11, pages 496–503. IEEE Computer Society.

Dean, J. and Ghemawat, S. (2008). Mapreduce: Simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113.

Desrosiers, C. and Karypis, G. (2011). A comprehensive survey of neighborhood-based recommendation methods. In *Recommender Systems Handbook*, pages 107–144. Springer.

Ding, C., Li, T., Peng, W., and Park, H. (2006). Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 126–135.

Fernández-Tobías, I., Cantador, I., Kaminskas, M., and Ricci, F. (2012). Cross-domain recommender systems: A survey of the state of the art. *Proceedings of the 2nd Spanish Conference on Information Retrieval. CERI.*

Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, 486(35):75 – 174.

Gao, S., Luo, H., Chen, D., Li, S., Gallinari, P., and Guo, J. (2008). Cross-domain recommendation via cluster-level latent factor model. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*, ECML PKDD '08, pages 161–176.

Gemmell, J., Schimoler, T., Mobasher, B., and Burke, R. (2012). Resource recommendation in social annotation systems: A linear-weighted hybrid approach. *Journal of Computer and System Sciences*, 78(4):1160 – 1174.

Herlocker, J. L., Konstan, J. A., Borchers, A., and Riedl, J. (1999). An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99.

Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53.

Hofmann, T. (2003). Collaborative filtering via gaussian probabilistic latent semantic analysis. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, pages 259–266. ACM.

Hu, L., Cao, J., Xu, G., Cao, L., Gu, Z., and Zhu, C. (2013). Personalized recommendation via cross-domain triadic factorization. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 595–606. International World Wide Web Conferences Steering Committee.

Hu, Y., Koren, Y., and Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ICDM '08, pages 263–272. IEEE Computer Society.

Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., and Wu, A. Y. (2002). An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):881–892.

Karatzoglou, A., Amatriain, X., Baltrunas, L., and Oliver, N. (2010). Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 79–86. ACM.

Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., and Riedl, J. (1997). Grouplens: Applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3):77–87.

Konstas, I., Stathopoulos, V., and Jose, J. M. (2009). On social networks and collaborative recommendation. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 195–202. ACM.

Koren, Y. (2008). Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 426–434. ACM.

Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37.

Kunegis, J. (2013). KONECT – The Koblenz Network Collection. In *Proceedings of International Web Observatory Workshop*.

Ley, M. (2002). The DBLP computer science bibliography: Evolution, research issues, perspectives. In *Proceedings of Int. Symposium on String Processing and Information Retrieval*.

Leydesdorff, L. (2007). Betweenness centrality as an indicator of the interdisciplinarity of scientific journals. *Journal of the American Society for Information Science and Technology*, 58(9):1303–1319.

Li, B. (2011). Cross-domain collaborative filtering: A brief survey. In *IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 1085–1086.

Li, B., Yang, Q., and Xue, X. (2009a). Can movies and books collaborate?: Cross-domain collaborative filtering for sparsity reduction. In *Proceedings of the 21st International Jont Conference on Artifical Intelligence*, IJCAI'09.

Li, B., Yang, Q., and Xue, X. (2009b). Transfer learning for collaborative filtering via a rating-matrix generative model. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09.

Linden, G., Smith, B., and York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80.

Loni, B., Shi, Y., Larson, M., and Hanjalic, A. (2014). Cross-domain collaborative filtering with factorization machines. In *Advances in Information Retrieval*, Lecture Notes in Computer Science, pages 656–661. Springer International Publishing.

Ma, H., King, I., and Lyu, M. R. (2012). Mining web graphs for recommendations. *IEEE Transactions on Knowledge and Data Engineering*, 24(6):1051–1064.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Mooney, R. J. and Roy, L. (2000). Content-based book recommending using learning for text categorization. In *Proceedings of the Fifth ACM Conference on Digital Libraries*, DL '00, pages 195–204.

Newman, M. E. J. (2004). Fast algorithm for detecting community structure in networks. *Physics Review E*, 69:066133.

Pan, W., Liu, N. N., Xiang, E. W., and Yang, Q. (2011). Transfer learning to predict missing ratings via heterogeneous user feedbacks. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Three*, IJCAI'11, pages 2318–2323.

Pan, W., Xiang, E. W., Liu, N. N., and Yang, Q. (2010). Transfer learning in collaborative filtering for sparsity reduction. In *Proceedings of the Twenty-Fourth AAAI Conference*.

Parimi, R. and Caragea, D. (2014). Community detection on large graph datasets for recommender systems. In *Proceedings of the 4th IEEE ICDM Workshop on Data Mining in Networks*, ICDM 2014.

Parimi, R. and Caragea, D. (2015a). Cross-domain matrix factorization for multiple implicit-feedback domains. In *International Workshop on Machine learning, Optimization and big Data (MOD) (Accepted)*.

Parimi, R. and Caragea, D. (2015b). Enhancements to adsorption-based approaches for cross-domain recommender systems. In *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM) (submitted)*.

Parimi, R. and Caragea, D. (2015c). Framework for evaluating cross-domain recommender systems with various degrees of user overlap between target and source domains. In *IEEE/WIC/ACM Web Intelligence Conference (submitted)*.

Parimi, R. and Caragea, D. (2015d). How to choose a recommender system: Insights and experiences for large-scale user personalization. In *4th International Congress on BigData (accepted)*.

Parimi, R. and Caragea, D. (2015e). Leveraging multiple networks for author personalization. Scholarly Big Data, AAAI Workshop.

Pazzani, M. and Billsus, D. (1997). Learning and revising user profiles: The identification of interesting web sites. *Machine Learning*, 27(3):313–331.

Sahebi, S. and Brusilovsky, P. (2013). Cross-domain collaborative recommendation in a cold-start context: The impact of user profile size on the quality of recommendation. In *User Modeling, Adaptation, and Personalization*, Lecture Notes in Computer Science, pages 289–295.

Sahebi, S. and Cohen, W. (2011). Community-based recommendations: a solution to the cold start problem. In *Workshop on Recommender Systems and the Social Web (RSWEB), held in conjunction with ACM RecSys?11*.

Said, A. and Bellogín, A. (2014). Comparative recommender system evaluation: benchmarking recommendation frameworks. In *Eighth ACM Conference on Recommender Systems, RecSys '14, Foster City, Silicon Valley, CA, USA - October 06 - 10, 2014*, pages 129–136.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 285–295. ACM.

Sarwar, B. M., Karypis, G., Konstan, J., and Riedl, J. (2002). Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering.

Shapira, B., Rokach, L., and Freilikhman, S. (2013). Facebook single and cross domain data for recommendation systems. *User Modeling and User-Adapted Interaction*, 23(2-3):211–247.

Singh, A. P. and Gordon, G. J. (2008). Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD*.

Singh, R., Xu, J., and Berger, B. (2008). Global alignment of multiple protein interaction networks with application to functional orthology detection. *Proceedings of the National Academy of Sciences*, 105(35):12763–12768.

Su, X. and Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009:4:2–4:2.

Sun, Y. and Han, J. (2012). *Mining Heterogeneous Information Networks: Principles and Methodologies*. Morgan & Claypool Publishers.

Takács, G., Pilászy, I., Németh, B., and Tikk, D. (2008). Matrix factorization and neighbor based algorithms for the netflix prize problem. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys '08, pages 267–274.

Takács, G., Pilászy, I., Németh, B., and Tikk, D. (2009). Scalable collaborative filtering approaches for large recommender systems. *Journal of Machine Learning Research*, 10:623–656.

Talukdar, P. P. and Crammer, K. (2009). New regularized algorithms for transductive learning. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part II*, ECML PKDD '09, pages 442–457. Springer-Verlag.

Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., and Su, Z. (2008). Arnetminer: Extraction and mining of academic social networks. In *International Conference on Knowledge Discovery and Data Mining (KDD).*, pages 990–998.

Vahedian, F. (2014). Weighted hybrid recommendation for heterogeneous networks. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, pages 429–432. ACM.

Wang, C. and Blei, D. M. (2011). Collaborative topic modeling for recommending scientific

articles. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 448–456. ACM.

Winoto, P. and Tang, T. (2008). If you like the devil wears prada the book, will you also enjoy the devil wears prada the movie? a study of cross-domain recommendations. *New Generation Computing*, 26(3):209–225.

Xin, L., E, H., Song, J., Song, M., and Tong, J. (2014). Book recommendation based on community detection. In *Pervasive Computing and the Networked World*, volume 8351 of *Lecture Notes in Computer Science*, pages 364–373.

Yan, J., Liu, N., Wang, G., Zhang, W., Jiang, Y., and Chen, Z. (2009). How much can behavioral targeting help online advertising? In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 261–270. ACM.

Yang, Y., Chawla, N. V., Sun, Y., and Han, J. (2012). Predicting links in multi-relational and heterogeneous networks. In *Proceedings of ICDM*.

Ying, J.-C., Shi, B.-N., Tseng, V., Tsai, H.-W., Cheng, K. H., and Lin, S.-C. (2013). Preference-aware community detection for item recommendation. In *Technologies and Applications of Artificial Intelligence (TAAI), 2013 Conference on*, pages 49–54.

Yuan, Q., Chen, L., and Zhao, S. (2011). Factorization vs. regularization: Fusing heterogeneous social relationships in top-n recommendation. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, pages 245–252. ACM.

Zhou, Y., Wilkinson, D., Schreiber, R., and Pan, R. (2008). Large-scale parallel collaborative filtering for the netflix prize. In *Proceedings of the 4th International Conference on Algorithmic Aspects in Information and Management*, AAIM '08, pages 337–348. Springer-Verlag.

Ziegler, C.-N., McNee, S. M., Konstan, J. A., and Lausen, G. (2005). Improving recommendation lists through topic diversification. In *Proceedings of the 14th International Conference on World Wide Web*, WWW '05, pages 22–32. ACM.