EXPLOITING RATINGS AND TRUST TO RESOLVE THE DATA SPARSITY AND COLD START OF RECOMMENDER SYSTEMS

by

GUIBING GUO

A thesis submitted to
the School of Computer Engineering
of the Nanyang Technological University
in fulfillment of the thesis requirements
for the degree of Doctor of Philosophy

2014

Abstract

Collaborative filtering (CF) is a widely used technique for recommender systems. The essential principle is that users with similar preference in the past are likely to give similar ratings on the items of interest in the future. However, collaborative filtering inherently suffers from two severe issues: *data sparsity* and *cold start*. The former issue refers to the difficulty in finding sufficient and reliable similar users, given that users generally rate only a small portion of items, while the latter issue refers to the difficulty presented by the *cold-start users* who rate zero or only a few items. Both issues are due to a lack of user ratings, and severely prevent recommender systems from generating accurate and personalized recommendations.

To help resolve these issues, we have worked on two lines of research in this thesis by exploiting the value of both ratings and trust. Firstly, we propose two approaches to leverage user ratings for recommender systems. The first approach is to design a Bayesian similarity measure based on Bayesian inference, taking into consideration both the direction and length of rating vectors. We posit that not all the rating pairs should be equally counted in order to accurately model user correlation. Three different evidence factors are designed to compute the importance weights of rating pairs. Further, our principled method reduces the correlation due to chance and potential system bias. Experimental results on six real-world data sets show that our approach achieves superior accuracy in comparison with other counterparts. This method aims to make better use of existing user ratings.

Secondly, we propose a new information source for recommender systems, called *prior ratings*. Prior ratings are based on users' experiences of virtual products represented in a mediated environment, and they can be submitted prior to purchase. A conceptual model of prior ratings is proposed, integrating the environmental factor *presence* whose effects on product evaluation have not been studied previously. A user study conducted in website and virtual store modalities demonstrates the validity of the conceptual model, in that users are more willing and confident to provide prior ratings in virtual environments. A method is proposed to

show how to leverage prior ratings in collaborative filtering. Experimental results indicate the effectiveness of prior ratings for recommender systems. By eliciting more kinds of user ratings, user preference can be better modelled and thus recommendations are improved.

The second research line is to adopt additional trust information to help model user preferences. In this thesis, we propose two approaches including one memory-based and one model-based. Firstly, the ratings of trusted neighbors are merged together to generate a new and more complete rating profile for the *active users* (who seek recommendations). Based on the new rating profile, a CF technique can be applied to find more reliable similar users, and thus recommendations can be better generated with higher accuracy and coverage. This strategy is especially useful for the cold-start users as their preference is approximated by the trusted neighbors. The underlying assumption is that trust and similarity are strongly and positively correlated which has been justified to be generally true in the literature. This strategy is applied and evaluated in three real-world data sets. Experimental results show that our approach can effectively cope with the concerned issues both in accuracy and coverage relative to other counterparts. The main strength of this memory-based approach is that user preferences can be complemented and derived by the ratings of trusted neighbors.

Secondly, we focus on how to take better advantage of social trust in a matrix factorization model. Although a number of trust-based recommendation models have been proposed in the literature, even the state-of-the-art trust-based models can be inferior to other well-performing ratings-only recommendation methods. By analyzing the social trust data from four real-world data sets, we conclude that not only the explicit but also the implicit influence of both ratings and trust should be taken into consideration in a recommendation model. Hence, we build on top of a state-of-the-art recommendation algorithm SVD++ which inherently involves the explicit and implicit influence of rated items, by further incorporating both the explicit and implicit influence of trusted users on the prediction of items for an active user. To our knowledge, the work reported is the first to extend SVD++ with social trust information. Experimental results on the four real-world data sets demonstrate that our approach TrustSVD achieves better accuracy than other both trust-based and ratings-only counterparts (ten in total), and can better handle the concerned issues.

To summarize, we have proposed four different approaches to exploit the value of ratings and trust in order to cope with the problems of data sparsity and cold start, improving the recommendation performance in terms of predictive accuracy and coverage.

Acknowledgements

I would like to express my deepest appreciation and gratitude to my advisors, Dr. Jie Zhang and Dr. Daniel Thalmann, for the continuous and patient guidance, encouragement, inspiration, support and mentorship they have provided to me over the past four years. They have made this a thoughtful and rewarding journey. I would also like to thank Dr. Neil Yorke-Smith, who as a good mentor, friend and collaborator, spent countless hours proofreading and revising papers, discussing my research, and was always willing to help and give his best suggestions.

I appreciate the research support of the Institute for Media Innovation for providing me a Ph.D grant. I also want to thank my colleagues, Dr. Hui Fang, Dr. Yuan Liu, Dr. Siwei Jiang, Ms. Athirai A. Irissappane, Mr. Chang Xu, Ms. Huanghuang Zhang, Ms. Xuan Liu, Dr. Mohamed Elgendi, Dr. Yang Xiao for the fruitful and pleasant discussion. Without them, the research cannot be such an enriching and enjoyable experience. Many thanks to all the staff and Ph.D students from the Institute for Media Innovation for their friendships and supports.

Finally, my immense gratitude goes to my family who were always there cheering me up and stood by me through all the good and bad time.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The way people operate online has been greatly changed by the emergence of Web 2.0. As described by Rosa et al. [1], "Online activities can no longer be characterized by just searching or browsing. Usage is evolving to interacting, and quickly to creating and sharing content". Zhou et al. [2] also point out that "The web users are no longer mere consumers of information", but the "producers of information". For example, they upload movies to YouTube, share their photos in Flickr, post their activities in Facebook, invite friends to hang out from Google+, write blogs in Blogger, etc. The available choices grow up exponentially, and it becomes more and more difficult and challenging for users to locate and operate with information of interest. This problem is recognized and well-known as *information overload*. Although traditional search engines (e.g., Google) are heavily used for information retrieval, it is reported in Pearce et al. [3] that these engines are well-designed for the tasks with specifically defined requirements (e.g., "I want to watch *Ice Age 4*") but less competent for the tasks with unclear or exploratory specifications (e.g., "I want to watch some movies more interesting and relaxing").

Recommender systems [4, 5, 6] which are regarded as 'exploring engines' are applied in real-world electronic commerce (e-commerce) applications, assisting users in discovering information of interest from a plethora of choices to overcome the information overload problem. These systems can automatically learn users' preferences from their past explicit (e.g., ratings) or implicit (e.g., user behaviors, purchase patterns, etc.) feedback, and recommend relevant items (e.g., products, services, etc.), i.e., personalized recommendations to users. Specifically, recommender systems select a small number of items for an active user (who desires recommendations) or predict the quality of unseen items [5]. Typical applications include Amazon for books, Google News for news and stories, CiteULike for academic papers, Youtube for

videos, and Last.fm for music recommendations. Recommender systems can not only improve user's satisfaction [7] and involvement [8], but also have a positive influence on sales [9] by building user's loyalty [10] and trust [11].

1.1 Research Challenges

Although many kinds of methods have been proposed for recommender systems in the literature, collaborative filtering (CF) [5] is one of the most well-known and widely adopted techniques. The underlying principle is that users with similar preference in the past are likely to favor the same items in the future [12, 6]. Specifically, CF identifies a set of like-minded (similar) users (called *nearest neighbours*[1.1]) for an active user by comparing her historic rating records with others. Then, the ratings of nearest neighbours will be aggregated to make a prediction for an item that she has not rated. This procedure is quite intuitive and has been adopted by many real applications in practice. For example, Amazon implements an item-to-item CF [13], and CiteULike allows users to choose their preferred algorithm between item-based and user-based CF[1.2]. However, CF is far from being perfect in terms of predictive performance, considering that it inherently suffers from two severe problems:

- **Data sparsity** refers to the difficulty for accurate recommendations as a user usually has only rated a very limited portion of items due to a lack of proper incentives or knowledge to rate them or sheer number of items. As a matter of fact, the sparsity of the reported matrix of user-item ratings is usually greater than 99% [14, 15].

- **Cold start** refers to the difficulty for accurate recommendations to the new or cold-start users who have not rated at all or only rated a few number (oftentimes less than 5) of items [16]. Cold start is an extreme case of the data sparsity problem.

The key issue of the two problems is that only limited rating information is available for preference modelling, whereby inherently and severely hindering the recommendation performance. Although many algorithms have been proposed to date, these issues have not been well-addressed yet. Given the sparsity of user ratings, in this thesis we propose two different

[1.1] In reality, only the top-N most similar users will be chosen and regarded as the nearest neighbours.
[1.2] Users can set specific algorithms via *MyCiteULike* $\rightarrow$ *Recommendations* $\rightarrow$ *Settings*

lines of research by: (1) making better use of existent ratings and eliciting more user ratings; and (2) incorporating additional information (i.e., social trust) to help model user preference. For each line, we propose two different approaches to handle the concerned issues, and demonstrate their effectiveness on a number of real-world data sets.

1.2 Ratings-only Recommender Systems

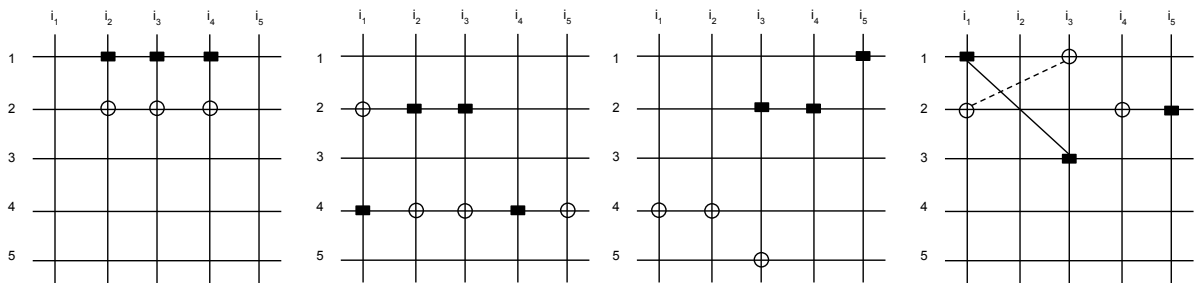
The first research line we take is to exploit the value of user ratings for preference modelling. Specifically, we first propose a Bayesian similarity measure [17] which produces more reliable and realistic similarity measurements than traditional similarity measures. Then, we propose a new information source *prior ratings*[1.3] [18, 19] in virtual reality environments, and validate its conceptual model as well as its usefulness for recommender systems. The first method aims to alleviate the concerned issues using existing ratings while the second intends to elicit more user ratings based on virtual product experience in order to inherently resolve these issues.

1.2.1 Bayesian Similarity Measure

The essence of CF is to discover similar users based on their rating profiles. Similarity plays an important role in CF techniques. First, it serves as a criterion to select a group of similar users whose ratings will be aggregated as a basis of recommendations. Second, it is also used to weight the ratings so that more similar users will have greater impact on the recommendations. Hence, similarity computation has direct and significant influence on the performance of CF. It is widely applied in two main categories of CF techniques, i.e., memory-based [20, 21] and model-based [22, 23] approaches. Memory-based approaches make recommendations by aggregating the preferences of similar users (items), where the similarity of a user (item) with each of the other users (items) is computed by some similarity measure. Differently, model-based approaches first train a recommendation model using historical user-item interaction data, and then apply the learned model to predict future potential interactions (i.e., recommendations) between users and items. This thesis focuses on user-based collaborative filtering.

Cosine similarity (COS) and Pearson correlation coefficient (PCC) [24] are the most usually-adopted methods to calculate user similarity in CF. COS defines user similarity as the cosine

[1.3]Ratings issued by users themselves before them having experienced the real products



(a) Flat-value (b) Opposite-value (c) Single-value (d) Cross-value

Figure 1.1: The problems of traditional similarity measures, i.e., Pearson correlation coefficient and cosine similarity. The filled rectangles and empty circles represent the ratings given by user u and user v, respectively.

value of the angle between two vectors of ratings (the *rating profiles*); PCC defines user similarity as the linear correlation between the two rating profiles. Formally, they are defined as follows:

$$s_{u,v} = \frac{\sum_{i \in I_{u,v}} r_{u,i} \cdot r_{v,i}}{\sqrt{\sum_{i \in I_{u,v}} r_{u,i}^2} \sqrt{\sum_{i \in I_{u,v}} r_{v,i}^2}} \tag{COS}$$

$$s_{u,v} = \frac{\sum_{i \in I_{u,v}} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{u,v}} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_{u,v}} (r_{v,i} - \bar{r}_v)^2}} \tag{PCC}$$

where $s_{u,v}$ is the similarity between user u and user v computed based on their ratings on the set $I_{u,v}$ of commonly rated items, $r_{u,i}$ denotes the rating given by user u on item i, and $\bar{r}_u$ and $\bar{r}_v$ represent the average rating given by user u and user v, respectively. Despite the popularity and simplicity of the two methods, it is well recognized that they only consider the direction of rating vectors but ignore the length [25]. Ahn [26] points out that the computed similarity could even be misleading if vector length is ignored. Both PCC and COS are also known to suffer from several inherent drawbacks [26]. These drawbacks can be summarized in the following four specific cases, and illustrated in Figure 1.1 where two users u and v have given their ratings (from 1 to 5) on five items (denoted by $i_1, \ldots, i_5$). The ratings are represented by filled rectangles and empty circles for user u and user v, respectively.

- **Flat-value problem**: if all the rating values are flat, e.g., $[1, 1, 1]$ given by user u on three items i_2, i_3, i_4 with values 1, or $[2, 2, 2]$ given by user v on the same items with values 2,

PCC is not computable as the correlation formula denominator becomes 0, and COS is always 1 regardless of the rating values. In our example of Figure 1.1(a), since both users give low ratings to all the three items, their similarity should be computable (because of existing *rating pairs*[1.4]) and less than 1 (not exactly the same).

- **Opposite-value problem**: if two users specify opposite ratings on the commonly-rated items, PCC is always -1. As shown in Figure 1.1(b), user u disagrees with the ratings given by user v on the commonly rated items and the computed PCC is -1. However, two comments can be made: (1) the number of co-rated items has no effect on the computed PCC; and (2) their opinions are not extremely opposite in terms of rating semantics. Hence, the computed PCC value -1 could be misleading.

- **Single-value problem**: if two users have only rated one item in common, PCC is not computable, and COS results in 1 regardless of the rating values. In Figure 1.1(c), although both users rated three items, only one item is commonly rated. In this case, PCC is not computable and COS yields 1. Both similarity measures cannot effectively reflect the situation where two users disagree with each other on the co-rated item.

- **Cross-value problem**: if two users have only rated two items in common, PCC is always -1 when the rating vectors cross each other, e.g., $[1, 3]$ and $[2, 1]$; otherwise PCC is 1 if computable. Both users in Figure 1.1(d) have rated two items in common and their ratings are crossing with each other. Then, PCC is computed as -1 indicating that they have distinct opinions about items i_1 and i_3. However, although the rating values are different and crossing, they both tend to give low ratings to these items and hence their opinions are similar to some extent.

To address the above issues and propose a better similarity measure, we design a novel Bayesian approach [17] by taking into account both the direction and length of rating vectors. An attractive advantage of Bayesian approaches is that one can infer in the same manner from a small sample as from a large sample [27]. This is especially useful when the length of rating vectors is short, which is often the case caused by the cold start and data sparsity issues. We apply the Dirichlet distribution to accommodate the multi-level distances between two ratings

[1.4] A rating pair is defined as two ratings that are given by two users on a certain commonly rated item.

towards the same item (the *rating pair*). Similarity is defined as the inverse normalization of user distance, which is computed by the weighted average of rating distances and of importance weights corresponding to the amount of rating pairs falling in that distance. Three different evidence factors, namely rating consistency, Gaussian singularity and rating semantics are developed to compute the weights of rating pairs. We further exclude the probability of the scenario where users happen to be 'similar' due to a small number of co-rated items, termed as *chance correlation*, and remove the potential system bias caused by the formulation of Bayesian similarity. Experimental results based on six real-world data sets[1.5] show that our approach achieves superior accuracy.

1.2.2 Prior Ratings

User ratings are crucial for recommender systems in e-commerce in order to provide quality personalized product recommendations. However, users can lack motivation to provide ratings (why should I bother to report my experience of an item?), and ratings can generally be given only after purchase (how can I share my experience of an item I have not tried?). As we have seen, without sufficient rating information for preference modelling, the effectiveness of recommender systems is hindered by the data sparsity and cold start problems [5].

Although many approaches have been proposed to address these problems either by furthering the use of existent ratings [26, 17], or by including additional information [16, 28, 29, 30, 20] as we discuss in next section, few researchers have attempted to elicit more user ratings from the perspective of user interfaces, so as to inherently mitigate the severity of these problems. On the other hand, Virtual Reality (VR) environments (e.g., Second Life [31]), have received considerable attention because of their ability to provide users with immersive virtual user experiences. Users can experience media more richly and can interact in real time with *virtual products*—the 'second existence' of real products in a mediated environment [32]. Although these environments offer potentially useful information for preference modelling, research on e-commerce in VR is still in its infancy.

In this thesis, we propose a new information source, called *prior ratings*, built upon *virtual product experiences*[1.6] [33], for recommender systems. Specifically, prior ratings can be issued

[1.5] All the used data sets are publicly available and often used as benchmarks in other literature research.
[1.6] Experience gained by interacting with the products represented in virtual reality environments

prior to purchase by interacting with virtual products in a mediated environment. The aim of this work is to study (1) the concept and nature of prior ratings with respect to product attributes and environmental factors; and (2) the usefulness of prior ratings in coping with the data sparsity and cold start problems of recommender systems.

In particular, first, we propose a conceptual model of prior ratings to provide a principled foundation, integrating the environmental factor *presence* whose effects on product evaluation have not been studied previously. Five hypotheses and two research questions are proposed to verify the validity of the conceptual model. We recruited volunteers and performed user studies in both 2D (website) and 3D (virtual store) user interface modalities. The results demonstrate the validity of the conceptual model under our experimental settings, and indicate that users are more willing and confident to give prior ratings in a VR store (due to a stronger sense of presence) than in a website.

Then, second, by integrating the prior rating and confidence data collected from the user studies into a novel adapted collaborative filtering technique that we develop, we empirically demonstrate the usefulness of prior ratings in improving recommendation performance in terms of predictive accuracy and coverage.

Summarized, the major contributions of our work are in three-fold: (1) we introduce a new information source (and its conceptual model) called *prior ratings*, which holds potential to benefit recommender systems in e-commerce; (2) we design a user study to validate the conceptual model of prior ratings; and (3) we propose and evaluate a collaborative filtering technique to demonstrate how to leverage prior ratings in predicting the ratings of products. Our work sheds light on inherently alleviating the data sparsity and cold start problems by the design of user interfaces with rich media and interactions by which confident prior ratings can be elicited from users.

1.3 Trust-aware Recommender Systems

To help resolve the concerned issues and model user preferences more accurately, additional information from other sources is widely adopted and incorporated into CF, indicating the second research line. Typical user information includes friendship [28], membership [34, 29] and social trust [35, 36], where trust is believed less ambiguous and more reliable than friendship

and membership. In this thesis, trust is defined as *one's belief toward others in providing accurate ratings relative to the preferences of the active user*. Both implicit trust (e.g., [37, 38]) and explicit trust (e.g., [39, 16, 40, 41]) have been investigated in the literature. The former trust is inferred from user behaviors such as ratings whereas the latter is directly specified by users. For example, a user in Epinions.com can add other users into her web-of-trust if she finds out that their product reviews are consistently valuable. Our present works mainly focus on the use and value of explicit trust, and leave the discussion of implicit trust to Chapter 7.

Trust-aware recommender systems have attracted more and more attention in the literature, given that trust is strongly and positively correlated with user similarity [42]. They are developed based on the phenomenon that friends often influence each other by recommending items. These approaches can be further classified into two categories: *memory-based* and *model-based* methods (explained in the subsequent sections). Our works continue the two research directions to design better trust-aware recommender systems.

1.3.1 The Merge Method

Memory-based approaches [39, 16, 43, 40] aim to retrieve candidate users (i.e., *nearest neighbours*) from the user space by integrating social trust in addition to user similarity. Although some improvements have been achieved, it is not much that they have achieved till now [44]. In fact, in some cases trivial methods may beat the state-of-the-art methods [41].

We propose a novel trust-based approach called "Merge" [20, 45] by incorporating the trusted neighbours explicitly specified by the active users in the systems, aiming to improve the overall performance of recommendations and to ameliorate the data sparsity and cold-start problems of CF. Specifically, we merge the ratings of trusted neighbours of an active user by averaging the ratings on the commonly rated items according to the importance weights of the trusted neighbours. The importance weight is computed as a linear combination of three parts: trust value, rating similarity and social similarity (i.e., ratio of commonly trusted neighbours). Note that the active user herself is seen as one of her trusted neighbours such that the ratings of the active user will be retained and kept unchanged. Only the ratings of the trusted neighbours on the other items that the active user has not rated will be merged and used to complement her own ratings, in order to form a more complete rating profile. This strategy is especially effective for the cold-start users since they have few ratings that cannot be effectively used

by CF. For the other users like *heavy users* who rated many items and whose preferences are already well exposed by their ratings, the need or necessity for merging ratings of trusted neighbours is not that strong. Nevertheless, we do claim that by adopting merged ratings from trusted neighbours, even the heavy users can benefit in terms of accuracy and coverage. The quality of the merged rating is measured by the *confidence* considering the number of ratings involved and the ratio of conflicts between positive and negative opinions (ratings). Generally, the more ratings it has and the less conflict exists, the higher confidence will be. Note that the confidence of items rated by the active user is always the highest.

The set of merged ratings (along with the confidence) is then used to represent the active user's preferences and to find similar users based on user similarity. Further, the rating confidence is also taken into account in the computation of user similarity. Finally, the Merge method is incorporated into a conventional CF to generate recommendations. Experiments on three real-world data sets[1.7] are conducted to demonstrate the effectiveness of our method in terms of accuracy and coverage. The results confirm that our method achieves promising recommendation performance, especially effective for the cold-start users comparing with the other counterparts. Although the idea of incorporating trust information into recommender systems is not new, our work is likely the first to effectively complement user rating profiles based on the ratings of trusted neighbours. Hence, our method shades light on a new way to build an effective trust-aware recommender system.

1.3.2 The TrustSVD Model

The Merge method provides a memory-based solution by forming a new and more complete profile of user preference based on the ratings of trusted neighbours. However, model-based approaches, especially the matrix factorization models have gained increasing attention due to the Netflix competition[1.8], since they usually achieve superior recommendation performance to memory-based approaches [46]. Matrix factorization techniques aim to decompose the user-item rating matrix into two small ranks of user-feature and item-feature matrices. Then, the prediction is generated by the inner product of a user's feature vector and an item's feature vector. Trust-based models further regularize the decomposition process by incorporating the

[1.7] For trust-aware recommender systems, only a few data sets with trust information are publicly available. To our best knowledge, most of them have been tested in our experiments for both the Merge and TrustSVD methods.

[1.8] http://www.netflixprize.com/

influence of social trust. However, even the best performance reported by the latest work [47] may be inferior to that of other state-of-the-art models which are solely based on user-item ratings. For instance, a well-performing trust-based model [48] obtains 1.0585 on data set Epinions.com in terms of Root Mean Square Error (RMSE), whereas the performance of a user-item baseline (see Koren [49], Section 2.1) can achieve 1.0472 in terms of RMSE on the same data set.[1.9]

To investigate this phenomenon, we conduct an empirical trust analysis based on four real-word data sets (FilmTrust, Epinions, Flixster and Ciao) through which two important observations are concluded. First, trust information is also very sparse, yet complementary to rating information. Hence, focusing too much on either one kind of information may achieve only marginal gains in predictive accuracy. Second, users are strongly correlated with their trust neighbours whereas they have a weakly positive correlation with their *trust-alike* neighbours (e.g., friends). Given that very few trust networks exist, it is better to have a more general trust-based model that can operate well on both trust and trust-alike relationships. These observations motivate us to consider both the influence of ratings and trust in a trust-based model. The influence can be explicit (real values of ratings and trust) or implicit (who rates what (for ratings) and who trusts whom (for trust)). The former influence is the basis of many matrix factorization models while the latter influence of ratings (e.g., who rated what) has been demonstrated useful in providing accurate recommendations [49, 50]. We will later show that implicit trust can also provide added value over explicit trust.

Thus we propose a novel trust-based recommendation model *TrustSVD* [51]. Our approach builds on top of a state-of-the-art approach SVD++ [49] where both the explicit and implicit influence of user-item ratings are involved to generate predictions. We extend on prior work by considering the influence of trusted users on the rating prediction for an active user. To our knowledge, this work is the first to extend SVD++ with social trust information. Specifically, on one hand the implicit influence of trust (who trusts whom) can be naturally added to the SVD++ model by extending the user modeling. On the other hand, the explicit influence of trust (trust values) is used to constrain that user-specific vectors should conform to their social trust relationships. This ensures that user-specific vectors can be learned from their trust

[1.9] Smaller RMSE values indicate better predictive accuracy. The result is reported by the well-known recommendation toolkit MyMediaLite (www.mymedialite.net/examples/datasets.html).

information even if a few or no ratings are given. In this way, the data sparsity and cold start issues can be better alleviated. Our novel model thus incorporates both explicit and implicit influence of ratings as well as trust. Further, a weighted-λ-regularization technique is used to help avoid over-fitting for model learning. Experimental results on four real-world data sets demonstrate that our approach achieves significantly better accuracy than other trust-based counterparts as well as other ratings-only well-performing models (ten models in total), and is more capable of coping with cold start situations.

1.4 Thesis Organization

The rest of this thesis is organized as follows.

Chapter 2 provides an overview of related research on general recommendation approaches to justify the directions and importance of our research, followed by the related work of ratings-only and trust-aware recommender systems. More efforts are needed to further resolve the data sparsity and cold start problems.

Chapter 3 elaborates a new similarity measure, called *Bayesian similarity* to measure user similarity. In particular, it takes into account both the direction and length of rating vectors, thereby to address the issues of traditional similarity measures. A number of factors are proposed to efficiently compute the weights of observed rating pairs. Empirical results demonstrates that Bayesian similarity achieves better performance than other similarity measures.

Chapter 4 studies a new viewpoint of user preference by introducing a new kind of user ratings, named *prior ratings*. The essential purpose is to elicit more user ratings in order to inherently resolve the data sparsity and cold start problems. Prior ratings are those given by users based on their virtual product experience before having experienced the real products. We have proposed and validated a conceptual model of prior ratings, and further shown that prior ratings are useful for recommendations.

Chapter 5 propose a memory-based approach which integrates the information of social trust. Specifically, it merges the ratings of trusted neighbours to form a more complete rating

profile through which user preference can be better modelled. We compare our approach with a number of other trust-based approaches, and find that our approach outperforms the others in precision and coverage.

Chapter 6 investigate a model-based approach with the incorporation of social trust, called *TrustSVD*. Specifically, it incorporates both the explicit and implicit influence of user ratings and social trust into a matrix factorization model. This is motivated by the indications drawn from the trust analysis on four real-world data sets. Empirical results show that our approaches significantly performs better than the other counterparts.

Chapter 7 concludes the present research and outlines the future work for better recommender systems in terms of ratings and trust. Briefly, for ratings, multiple ratings can be given by a user to one and the same item, and multi-criteria ratings should be studied more for recommender systems. For trust, implicit trust should be inferred to enrich trust information, and distrust can provide added value over trust to recommender systems.

Chapter 2

Literature Review

In this chapter, we provide an overview of related research on general recommendation approaches, and justify the directions and importance of our research work in Section 2.1. Section 2.2 describes a number of related works regarding ratings-only recommender systems. First, we summarize many traditional and newly-proposed similarity measures and discuss their inability to cope with cold conditions. Second, we discuss the ways to elicit user ratings in the literature and indicate how our new information source differs from the traditional ones. Section 2.3 depicts the literature review of trust-aware recommender systems including both memory-based and model-based approaches, and points out their strength and limitations in the light of recommendation performance.

2.1 General Recommendation Approaches

Recommender systems have been extensively studied for decades and there have been many algorithms proposed to implement them. Generally, they can be classified into two broad categories: memory-based approaches and model-based approaches. The former approaches make recommendations by aggregating the preferences of similar users (items), i.e., the historical user-item ratings. Some similarity measures are used to compute user (item) similarity, and then the ratings of similar users (items) are aggregated to generate a prediction for an unknown item. Typical examples include content-based filtering [52, 53], collaborative filtering (CF) [54, 55] and hybrid approaches [56, 25]. Content-based approaches depend on descriptive information of items to recommend users similar items. The issues are that content information may not be available or too costly to collect, and that only similar (and no diverse) items can

be recommended. In contrast, CF makes recommendations only based on user-item ratings, independent on the content of items. The intuition behind CF[2.1] is that items appreciated by the similar users will also be appreciated by the active user. Hybrid recommender systems combine two or more kinds of recommendation techniques to gain better performance with fewer of the drawbacks of any individual one. The major drawback lies in the complexity of algorithms and implementations. Jannach et al. [57] report that CF receives much more attention than the others according to the statistics analyzed from recent publications.

Although memory-based approaches are adopted in some real applications such as CiteULike[2.2], Youtube[2.3] and Last.fm[2.4], they have been recognized to be ineffective to large-scale data sets. By contrast, model-based approaches learn and model users' rating patterns by employing statistical and machine learning techniques to user-item interaction data sets (offline). The learned models are then used to predict ratings of unknown items (online). Hence, they can better adapt and scale up to large-scale data sets. Typical examples include aspect models [58], regression models [59], probabilistic models [60], ranking models [61], latent factor models [62, 63], clustering-based approaches [64, 65], etc. Among them, the most well-known models are matrix factorization-based approaches [46], such as SVD [46], NNMF [66], tensor factorization [67]. The basic idea is that both users and items can be characterized by a number of latent features, and thus the prediction can be computed as a inner product of user-feature and item-feature vectors.

Furthermore, model-based approaches usually achieve better recommendation performance than memory-based ones. The reason is that not only ratings of two users but also ratings of other users are adopted to learn the features of users and items, and thus better handle the data sparsity and cold start issues. For example, Gunawardana and Meek [68] report to capture pairwise item interactions by using a Boltzmann machine, whose parameters are associated with item contents. They show that better performance is achieved in the case of cold-start situations. Gantner et al. [69] attempt to learn a function mapping user/item attributes to latent features of a matrix factorization model. With such mappings, the latent factors learned by a matrix factorization can be applied to new users or new items. Liu et al. [70] propose a

[2.1] This thesis is mainly concerned with user-based collaborative filtering.
[2.2] http://www.citeulike.org/
[2.3] http://www.youtube.com/
[2.4] http://www.last.fm/

representative-based matrix factorization method that aims to find out the most representative users and items in the system. Then, for the cold-start users, their preferences can be elicited by asking them to rate the most representative items; the same holds for the cold-start items. To combat the data sparsity problem, Ahmed et al. [71] propose a method to learn user preferences over item attributes by applying a personalized Bayesian hierarchical model, which combines both globally and locally learned user preferences.

However, it is hard for recommender systems to explain how recommendations are generated since the features used are usually latent and unobservable [72]. Another disadvantage is that the computation complexity for model building is usually expensive and time-consuming, especially when re-computation is required to fold in newly arriving ratings [5]. In contrast, it is easier for memory-based approaches to explain and justify the recommendations, and to incorporate newly-issued ratings into up-to-date recommendations than model-based ones. Hence, both memory-based and model-based approaches have their own merits and demerits. A lesson learned from the Netflix competition is that no single approach can always achieve the best performance, and different methods generally reveal different patterns of rating data [73]. Hence, it is necessary to further improve the performance of both kinds of recommendation approaches. Our works follow these directions to further improve recommendation performance.

2.2 Ratings-only Recommender Systems

User-item ratings are the core evidences for recommender systems to make use of and to infer user preference for an unknown item, no matter whether memory-based or model-based approaches are adopted or designed. The concerned issues, data sparsity and cold start are mainly due to a lack of sufficient ratings to model user preference, and thus to clearly identify users with similar preferences. To resolve these issues, the first research line we work on is to further exploit the value of ratings for recommendations. Specifically, this section gives the literature review in terms of how ratings are used to find similar users, and of how to elicit more user ratings according to their interactions and experience with items of interest.

2.2.1 Similarity Measures

The 'traditional approaches' of Pearson correlation coefficient (PCC) and cosine similarity (COS) are the most widely adopted similarity measures in the literature. Although it is re-

ported that PCC works better than COS in CF [24]—as the former performs data standardization whereas the latter does not—others show that COS rivals or outperforms PCC in some scenarios [74]. In other words, there is no consistent conclusion about the performance of PCC and COS in different cases. However, the literature rarely has sought to investigate the reasons for such phenomena, rather simply attributing them to the difference of data sets. We provide a reasonable and insightful explanation by conducting an empirical study on the nature of PCC, COS, and our method in Section 3.2.

Various similarity measures have been proposed in the literature, given the ineffectiveness of the traditional approaches [74]. Broadly, they can be classified into two categories. First, some researchers attempt to modify the traditional measures in some way. One direction is to weight computed similarity by taking into consideration the properties of commonly rated items. Breese et al. [24] adopt the inverse user frequency as weights to restrict the contribution of popular items in the PCC computation. The intuition is that two users who agree on popular items are less likely to be similar than those who agree on less popular items, because users generally tend to like popular items. Said et al. [75] design several weighting schemes based on the intuition that popular items have less impact on the similarity computation of PCC and COS than those receiving few number of ratings. The results show that the weighting schemes have only little effect on COS in all cases while more discernible impact for PCC is only observable on some data sets for the users who rate a lot of items. Breese et al. [24] also propose to use *case amplification* ρ to transform the PCC value from w to w^{ρ}. A typical value of ρ in their experiments is 2.5. The transformation helps emphasize the high weights closer to one, and suppress the low weights closer to zero. As a result, it can reduce noise in the data. With the recognition of inability of PCC in cold conditions, Ma et al. [25] propose a significance weight factor $\min(n, \gamma)/\gamma$ to devalue the PCC value when the number n of co-rated items is small, where γ is a constant and generally determined empirically. Similarly, Koren [76] suggests $(n-1)/(n-1+\lambda)$ as a shrunk factor, where n is the number of co-rated items, and λ is a parameter determined by cross validation. Candillier et al. [77] use Jaccard similarity as a weighting factor and combine it with other similarity measures (e.g., PCC) to appreciate the influence of co-rated items. Those weighting schemes can be regarded as the confidence of computed similarity. If similarity is based on sufficient ratings, we have a high confidence that the value is reliable and reflecting realistic user correlation; otherwise it has high chance

that the value will be error-prone and even misleading. Another direction is to enhance the similarity from the viewpoint of the properties of raters (i.e., users). Shi et al. [78] categorize users into three different pools: 'positive', 'neutral' and 'negative' according to their rank preferences of items. Then, along with the similarity based on all ratings, three pool-based similarities are computed which will be integrated to produce recommendations. Ortega et al. [79] adopt the concept of Pareto dominance to preprocess and narrow the whole user set to a set of users dominating others in terms of rating distances. Traditional similarity methods are then applied to compute user similarity. Although many approaches have been proposed, none of them makes any changes to the calculation of PCC or COS itself. As a result, the inherent issues mentioned in Section 1.2.1 are not addressed.

Second, other researchers propose *new* similarity measures in place of the traditional measures. Shardanand and Maes [80] propose a measure based on the mean square difference (MSD) normalized by the number of commonly rated items. However, as we will show in Section 3.3, its performance is generally worse than PCC or COS. Lathia et al. [81] develop a concordance-based measure which estimates the user correlation based on the number of concordant, discordant and tied pairs of common ratings. That is, it finds the proportion of agreement between two users. Since it depends on the mean of ratings to determine the concordance, this approach also suffers from the flat-value and single-value problems where user similarity is not computable. Ahn [26] proposes the PIP measure based on three semantic heuristics: *Proximity*, *Impact* and *Popularity*. The motivation is to explicitly consider the semantic meanings behind numerical rating values. For example, value 5 indicates a stronger preference than a value of 4, while both values mean that the user likes a specific item. Hence, the nuanced difference in semantics may matter and result in different similarity measurements. PIP attempts to enlarge the discrepancies of similarity between users with semantic agreements and those with semantic disagreements in ratings. However, the computed similarity is not bounded and often greater than 1, resulting in less meaningful user correlation. Bobadilla et al. [82] propose the *singularities measure* (SM) based on the intuition that users with mutually close ratings different from the majority (high singularity) are more similar than those with close ratings consistent with the others (low singularity). It bears similarity to the idea of popularity of items but differs in that singularity further investigates the deviation between one's rating and the majority's, and that even a popular item may receive distinct ratings from different users.

Although SM considers the mean of agreements, the length of rating vector is not taken into consideration. It tends to treat users with similar opinions as un-correlated if all of their ratings are consistent with others'. SM was evaluated only on a single data set in comparison with traditional approaches. We evaluate it more thoroughly as part of our work.

Although these various approaches proposed to date have achieved improvements to some extent, two main criticisms can be suggested. First, a better similarity measure is always expected to consistently perform better across different data sets. Second, most of these approaches are merely based on heuristics (or intuitions) and lack a fundamental theoretical underpinning. We aim to develop a principled method based on existent ratings in order to achieve a significant improvement in terms of predictive accuracy.

2.2.2 Rating Elicitation

Other than the existent ratings, researchers also adopt additional information to help model user preference such as social annotation (tag) and friendships [28, 22], membership [29], social trust [16, 83, 63, 84, 48]. In this way, although improvements have been demonstrated to some extent, the cold start problem remains a difficult issue to address. The reasons can be explained in three aspects. First, these kinds of information suffer from a number of inherent issues. Specifically, the semantics of friendship are ambiguous and error-prone. For example, friends may have different preferences because friendships can be built based on other relations (e.g., working affiliation) rather than common interest in items. It is usually low cost for a user to get connections with other users or even strangers (e.g., Facebook). Trust is only supported by few real systems (e.g., `epinions.com` and `ciao.co.uk`). Trust information is also very sparse [85], i.e., the density of trust information is even much smaller than that of rating information in some data sets, since not all users who give ratings will be socially connected with other users. In many other systems, trust information is not available and thus we have to infer trust from users' behavioural patterns [85, 86]. Another problem of social relationships is that they usually exist in the forms of social connections with no connection strength specified or available. For example, in trust networks, we only know the relationships about who trusts whom, but it is unknown to what extent one will trust another. One explanation is due to the concern of, e.g., privacy. It is a commonplace that not all socially connected users should be equally weighted for recommendations. Fang et al. [47] suggest to refine the trust values by

training a support vector regression model based on four general decomposed trust factors, before taking them as input to a matrix factorization model. They show that better performance can be achieved based on the refined trust values. The unweighted social relationships may further limit the utility of social recommender systems. Second, for 'extremely' cold-start users who have rated no items and linked to no one, it is difficult for a social recommender to provide accurate personalized recommendations. This is because user preferences cannot be inferred and modelled from their past rating behaviours. Third, even with the social information, the performance of cold-start users is still much worse than that of normal users, as demonstrated by the work of Yang et al. [48]. There is much room left for further improvements. Hence, the cold start problem has not been well handled by the existing approaches and information sources. More efforts are required to further alleviate the cold start problem, including developing more advanced recommendation algorithms and designing new information sources.

Our work follows the line of incorporating additional information in recommender systems, but differs in that we focus on introducing a new information source, called *prior ratings* (ratings given by users themselves prior to them having experienced real products), rather than the specific techniques to utilize such information. However, we do design a collaborative filtering technique to demonstrate the use of prior ratings.

Only a few works have attempted to study the concerned problems from the perspective of user interfaces. For example, Carenini et al. [87] recognize that traditional recommender systems support only a limited model of interaction to elicit new users' ratings. They explore a set of elicitation techniques leading to a more conversational and collaborative interaction model. Offline experiments show that the effectiveness of recommender systems can be improved by applying these techniques. However, whether the new model of interaction is accepted by users and useful for online recommendations in practice is unknown. McNee et al. [88] find that allowing systems to choose items for new users to rate works better than letting users choose the items, in order to bootstrap and build a recommendation model. Dong et al. [89] develop a browser plugin to provide users with suggestions on writing better product reviews. Other users can hence better understand the performance of products before making a purchase decision. Most of these studies focus on interface design or assistance, so that users are more comfortable, enabled, or loyal in providing ratings. They are not particularly dedicated to resolving the two concerned recommender systems problems. By contrast, our motivation is to tackle the concerned problems through a new information source in a richer virtual environment.

Contemporary websites are implementing novel interfaces and interactions to better elicit user preferences. For example, brides.com allows users to virtually try on wedding dresses by uploading their own photos and adjusting the specific positions of dresses to fit. As another example, ray-ban.com offers users a virtual mirror through which users can calibrate their faces using a computer camera, and virtually try on different kinds of glasses. However, the available media and interactions are limited in comparison with the capabilities of virtual reality (VR) (e.g., Second Life [31]). The emergence of 3D VR environments offers more adequate information which can be used to model user preference. Although the need to design new recommender agents for e-commerce in VR has been recognized [90], research on recommender systems in VR is still in its infancy. Eno et al. [91] summarize several ways to model user preferences in VR. Shah et al. [92] recommend to users locations of interest by analyzing users' login data to help them navigate in VR. Hu and Wang [93] propose a system for virtual furniture recommendation according to users' interest and requirements. Although a controlled prototype has been implemented, the features of VR are not exploited to elicit more user ratings.

In this thesis, we propose prior ratings as a means to make use of the information conveyed by the rich media and the real-time interactions in VR. Prior ratings represent a new information source distinct from the existing information sources noted earlier. First, prior ratings are issued by real users: hence they directly reflect users' preferences of products as well as standard type of user ratings. In this regard, they could be more reliable than other kinds of information, such as friendship and trust. Second, prior ratings differ from other extra information sources in that they do not depend on additional structures (e.g., social network) as required by the latter. Prior ratings only rely on the representations of virtual products in mediated environments, but these environments are the commonplace basis of e-commerce applications. Prior ratings are useful to deal with data sparsity and cold start because (1) more user ratings are incorporated to alleviate the sparsity of rating data; and (2) prior ratings can help model user preferences even if posterior ratings are few or none, and thus ensure the functionality of the recommenders. To our knowledge, there is no work that has defined the concept of and investigated the effectiveness of prior ratings for recommender systems.

2.3 Trust-aware Recommender Systems

To better model user preferences for the cold-start users who only rated a few items, additional user information is often adopted. For example, Konstas et al. [28] take into consideration both the social annotation (tag) and friendships inherently established among users in a music track recommender system. Due to the ambiguity of friendship, friends may have different preferences in items. In contrast, users joining the same online community[2.5] are more likely to have similar preferences [34]. Hence by leveraging data from multiple channels including memberships in a project, Guy et al. [29] build a system named SONAR to recommend people of interest to active users. Comparing with friendship and membership, trust information is of less ambiguity and more relevant to user similarity [35, 42, 36]. Research has shown that people prefer recommendations from friends to those made by recommender systems such as Amazon.com [94], and that users prefer to receive recommendations from the systems where they had positive experience before [95]. Formally, trust is strongly and positively correlated with user similarity [42]. Massa et al. [96] contend that asking users to provide a few trusted neighbours is more meaningful to bootstrap recommender systems than asking users to provide a few item ratings. Another one of the initial motivations for trust-aware recommender systems is to provide better personalized recommendations for the people who disagree with the average [97]. In this thesis, we focus on the value of *explicit trust* which is directly specified by users, and leave the discussion of *implicit trust* which can be inferred from other kinds of information as a part of future work.

Similar to general recommender systems, trust-aware recommenders can be also broadly split into two categories: memory- and model-based approaches. Next we present an overview of recommender systems enhanced by social trust, and how they are associated with our works.

2.3.1 Memory-based Approaches

Many memory-based approaches have been proposed to make use of social trust. For example, Jamali and Ester [98] design the *TrustWalker* approach to randomly select trusted neighbours in the trust networks, where users are represented as nodes and trusted neighbours are connected with each other by trust links (i.e., edges), the strength of which indicates the trustworthiness between two users. Trust information of the selected neighbours is combined with an

[2.5] We refer community to as an interest group of users.

item-based technique to predict item ratings. In contrast, our work focuses on generating predictions by combining trust information with a user-based technique. Liu and Lee [99] report that more accurate prediction algorithms are possible by incorporating trust information into traditional collaborative filtering. They do not directly use trust to substitute similarity but rather amplify similarity measurements by taking into account the number of messages exchanged among users. Hence, this approach is message specific. Further, a number of hybrid approaches incorporating trust are also proposed, such as [100, 101, 102], demonstrating that good performance can be achieved by combining both user- and item-based CF approaches. However, in this thesis we focus on how to further improve the user-based CF using explicit trust.

The closest approaches to ours are as follows. Massa and Avesani [16] analyze the drawbacks of conventional CF-based recommender systems, and elaborate the rationale why incorporating trust can mitigate those problems. They propose the *MoleTrust* algorithm, which performs depth-first search, to propagate and infer trust in the trust networks. Empirical results show that the coverage is significantly enlarged but the accuracy remains comparable when propagating trust. Similarly, Golbeck [39] propose a breadth-first search method called *TidalTrust* to infer and compute trust value. Both approaches substitute similarity with trust to predict item ratings, and the performance of the two algorithms is close [43]. Hence, we will only compare our method with one of them, namely MoleTrust in this thesis. In addition, Chowdhury et al. [40] propose to enhance CF by predicting the ratings of similar users who did not rate the target items according to the ratings of their trusted neighbours, so as to incorporate more similar users for recommendation. However, it performs badly in cold conditions where only few ratings are available, which is the main concern of the present research. Another recent work using the explicit trust network is proposed by Ray and Mahanti [41]. They improve the prediction accuracy by reconstructing the trust networks. More specifically, the trust links between two users will be removed if their similarity is lower than a certain threshold. Empirical results show that good accuracy can be achieved at the cost of poor coverage, and it fails to function in the case of cold conditions where user similarity may not be computable.

In addition, most previous works are only evaluated on a single data set [39, 16, 40, 41], lacking a necessary understanding of their methods' generality. Besides, the reported results often show that they are only able to achieve improvements in either accuracy or coverage, but

not in both. Further, the cold start problem has not been well addressed yet, and proposing better trust-aware recommender systems remains a big challenge [44]. The purpose of our work is to take a step further in addressing this challenge by proposing a novel approach to incorporate trusted neighbours in CF.

2.3.2 Model-based Approaches

More recently, trust-aware recommendation models have attracted much attention and been widely studied in the literature. The general objective is to learn better recommendation models (than ratings-only ones) by further considering user-side constraint, i.e., the social trust including both connection links and strength. For example, Guo et al. [103] cluster users by multiviews of similarity and trust, in order to resolve the relative low accuracy and coverage issues of clustering-based recommendations. They also make use of both ratings and trust to properly cluster cold-start users, i.e., mitigating the cold-start problem. The most popular and widely studied recommendation models are matrix factorization-based models [46] which aim to factorize the user-item rating matrix into two low-rank user-feature and item-feature matrices. Then the predictions can be generated by the inner products of user- and item-specific latent feature vectors [62]. Here are some representative model-based approaches with social trust. Ma et al. [104, 83, 22] developed several approaches by adding different trust regularization terms to a matrix factorization model. Specifically, they propose a social regularization method (*SoRec*) by considering the constraint of social relationships [104]. The idea is to share a common user-feature matrix factorized by ratings and by trust. Ma et al. [83] then propose a social trust ensemble method (*RSTE*) to linearly combine a basic matrix factorization model and a trust-based neighbourhood model together. Ma et al. [22] further propose that the active user's user-specific vector should be close to the average of her trusted neighbours, and use it as a regularization to form a new matrix factorization model (*SoReg*). Jamali and Ester [63] build a new model (*SocialMF*) on top of SoRec by reformulating the contributions of trusted users to the formation of the active user's user-specific vector rather than to the predictions of items. Yang et al. [105] highlight the domain-specific property of trust and propose three different approaches to infer the trust in each friend circle. By substituting the general trust with inferred circle-based trust, they show that the performance of the SocialMF model can be further improved. Zhu et al. [106] propose a graph Laplacian regularizer to capture

the potentially social relationships among users, and form the social recommendation problem as a low-rank semidefinite problem. However, empirical evaluation indicates that very marginal improvements are obtained in comparison with the RSTE model. More recently, Yang et al. [48] propose a hybrid method (*TrustMF*) that combines both a truster model and a trustee model from the perspectives of trusters and trustees, that is, both the users who trust the active user and those who are trusted by the user will influence the user's ratings on unknown items. Tang et al. [84] consider both global and local trust as the contextual information in their model, where the global trust is computed by a separate algorithm. Yao et al. [107] take into consideration both the explicit and implicit interactions among trusters and trustees in a recommendation model. Fang et al. [47] stress the importance of multiple aspects of social trust. They decompose trust into four general factors and then integrate them into a matrix factorization model.

All these works have shown that a matrix factorization model regularized by trust outperforms the one without trust. That is, trust is helpful in improving predictive accuracy. However, it is also noted that even the latest work [47] may be inferior to other well-performing ratings-only models. We argue that the main reasons are in two-fold. First, the existing trust-based models consider only the explicit influence of ratings. The utility of ratings is not well exploited. We will later show that trust information may be sparser than rating information. This motivates us to build a new trust-based model based on SVD++ [49] that inherently and well considers both the explicit and implicit influence of ratings. Second, these trust-based models do not consider the explicit and implicit influence of trust simultaneously. As will be explained later in Section 6.1, this may lead to deteriorated performance when being applied to social relationships with smaller correlations (with user similarity). Therefore, we incorporate into SVD++ both explicit and implicit influence of social trust, to enhance the generality of our approach. By doing so, a better way to utilize user-item ratings and user-user trust is proposed in this thesis.

2.4 Concluding Remarks

Recommender systems have been extensively developed and studied for decades and many different kinds of approaches have been proposed so far. It would be impossible to cover all

existent approaches in this thesis. Instead, we first provided a general overview about the main categories of methodologies present in the literature. Specifically, those methods can be broadly classified into two types: memory-based and model-based approaches. For each category, we discussed a number of representative approaches and summarized their advantages and disadvantages in general. In this way, we explained and justified the directions and importance of our research works to resolve the data sparsity and cold start problems.

Then, corresponding to the two lines of research we currently conducted, more specific and detailed related work was reviewed. Specifically, on one hand for ratings-only recommender systems, we conducted the literature review from two perspectives of: (1) ways to utilize existent ratings to measure user similarity which is a core part of collaborative filtering techniques; (2) ways to elicit more (kinds of) user ratings in order to inherently resolve the concerned issues. Existing similarity measures are ineffective to cope with the cold start situations as only few ratings are available. Incorrect measurements can cause performance degradation in a great part. The critical lack of ratings directs us to further leverage the value of ratings (1) by proposing a principled similarity measure to better utilize existent ratings in Chapter 3; and (2) by introducing a new source of ratings to increase the amount of usable ratings in Chapter 4.

On the other hand, although many trust-based approaches (including both memory- and model-based) have been proposed in the literature as summarized in this chapter, it is not much the research community achieved and better ways to incorporate trust and ratings are still expected, as such combinations are believed leading to better algorithms. Most importantly, the existing works cannot provide satisfying performance when dealing with the data sparsity and cold start problems, and more efforts should be made to resolve these issues. We continue the two kinds of trust-based approaches: (1) by merging the ratings of trusted neighbours to complement user profiles (memory-based) in Chapter 5; and (2) by incorporating both the explicit and implicit influence of ratings and trust in a concrete recommendation model (model-based) in Chapter 6.

Chapter 3

BS: A Bayesian Similarity Measure

In this chapter, we focus on the first research line to resolve the data sparsity and cold start problems, i.e., to exploit the value of ratings for better recommendations. Section 1.2.1 has summarized a number of issues from which traditional similarity measures suffer. The essential reason is that most of them focus on the direction of rating vectors while ignoring the length. We argue that the length of rating vectors is important to compute user similarity, and should be taken into consideration in order to design a better similarity measure. In this chapter, we will present a novel similarity measure based on Bayesian inference, termed as *Bayesian similarity* by considering both the direction and length of rating vectors. We aim to generate more realistic similarity measurements and thus improve the recommendation performance by finding more reliable similar users—a better way to make use of existing ratings.

The rest of this chapter is organized as follows. Our approach is elaborated in Section 3.1, including evidence weights, chance correlation, and system bias as the three main components. Then, we exemplify the differences between the traditional approaches and Bayesian similarity, and conduct a more general study on the nature of those similarity measures in Section 3.2. Finally, our approach is evaluated on six real-world data sets in Section 3.3, followed by the conclusion in Section 3.4.

3.1 Bayesian Similarity

The proposed Bayesian similarity measure is distinct from Pearson correlation coefficient (PCC) and cosine similarity (COS), and aims to solve the issues of these traditional similarity measures. It takes into consideration both the direction (rating distances) and the length (rating

amount) of rating vectors. Specifically, the rating distances are modelled by the Dirichlet distribution based on the amount of observed evidences, each of which is a pair of ratings (from the two vectors) towards a commonly rated item. Then the overall user similarity is modelled as the weighted average of rating distances according to their importance weights, corresponding to the amount of new evidences falling in the distance. Further, we consider the scenario where users happen to be 'similar' due to the small length of the rating vectors, termed as *chance correlation*. Therefore, the length of the rating vectors is taken into account in our approach via (1) the modelling of Dirichlet distribution, and (2) the chance correlation.

3.1.1 Dirichlet-based Measure

The *Dirichlet distribution* represents an unknown event by a prior distribution on the basis of initial beliefs [108]. As new evidences come in, the beliefs of the event can be represented and updated by a posterior distribution. The posterior distribution well suits a similarity measure since the similarity is updated based on the records of new ratings of commonly-rated items issued by two users. In addition, most existing recommender systems underpin users' ratings on a number of discrete values (e.g., 1 to 5) which can be well handled by the Dirichlet distribution.

We first introduce a number of notations, and mathematically model the similarity computation using the Dirichlet distribution. Let $(r_{u,k}, r_{v,k})$ be a pair of ratings (i.e., a rating pair) reported by users u and v on item k. The rating values are drawn from a discrete rating scale $\mathcal{L} = \{l_1, \ldots, l_n\}$ ($l_{j+1} > l_j, j \in [1, n-1]$) predefined by a recommender system, where n is the number of different rating values. We define the *rating distance* as the absolute difference between two user ratings, i.e., $d = |r_{u,k} - r_{v,k}|$.[3.1] We use the rating distance rather than rating difference in order to ensure the symmetry of similarity measure, i.e., $s_{u,v} = s_{v,u}$, where $s_{u,v}$ denotes the similarity between users u and v. According to the rating scale $\mathcal{L}$, we can define $\mathcal{D} = \{d_1, \ldots, d_n\}$ as a set of possible rating distances, where $d_i = |l_{j+i-1} - l_j|, d_{i+1} > d_i$, for any $i, j \in [1, n]$ and $d_1 \leq d_{i+j-1} \leq d_n$. For example, d_1 is the distance between two identical rating values l_j, i.e., $d_1 = 0$, and $d_n = l_n - l_1$ is the maximum rating distance between any two rating values.

[3.1] For simplicity, we omit the subscripts u, v, k from the rating distance d.

Let $\boldsymbol{x} = (x_1, \ldots, x_n)$ be the probability distribution vector of D, i.e., $P(D = d_i) = x_i$, which satisfies the additivity requirement $\sum_{i=1}^{n} x_i = 1$. The probability density of the Dirichlet distribution for variables $\boldsymbol{x} = (x_1, \ldots, x_n)$ with parameters $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$ is given by:

$$p(\boldsymbol{x}|\boldsymbol{\alpha}) = \frac{\Gamma(\alpha_0)}{\prod_{i=1}^{n} \Gamma(\alpha_i)} \prod_{i=1}^{n} x_i^{\alpha_i - 1}, \tag{3.1}$$

where $x_1, \ldots, x_n \geq 0$, $\alpha_1, \ldots, \alpha_n > 0$ and $\alpha_0 = \sum_{i=1}^{n} \alpha_i$, $\Gamma(x+1) = x\Gamma(x)$ is the gamma function. The parameter α_i can be interpreted as the amount of *pseudo-observations* of the event in question, i.e., rating pairs that are observed before real events happen. Hence, α_0 is the total amount of prior observations. It is important to set appropriate values for the parameters α_i as they will significantly influence the posterior probability.

Table 3.1: The distribution of prior rating evidences

	$\boldsymbol{l_1}$	$\boldsymbol{l_2}$	...	$\boldsymbol{l_{n-1}}$	$\boldsymbol{l_n}$
$\boldsymbol{l_1}$	d_1	d_2	...	d_{n-1}	d_n
$\boldsymbol{l_2}$	d_2	d_1	...	d_{n-2}	d_{n-1}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\boldsymbol{l_{n-1}}$	d_{n-1}	d_{n-2}	...	d_1	d_2
$\boldsymbol{l_n}$	d_n	d_{n-1}	...	d_2	d_1

Before observing any rating pairs, and without any prior knowledge to the contrary, we assume that ratings from two users are random and uncorrelated as illustrated in Table 3.1. The rows and columns represent the first and second elements in the rating pairs whose values are taking from the predefined rating scale $\mathcal{L}$, and each entry is the rating distance of a corresponding rating pair. Therefore, there are n^2 pseudo-observations corresponding to all the possible combinations of rating values. Thus, parameter α_i will be the number of pseudo-observations located in rating distance d_i.

To be generic, let p_j be the prior probability of a rating being value l_j taken out of the rating scale $\mathcal{L}$. Then we set the values of parameters α_i as follows:

$$\alpha_i = \begin{cases} \sum_{j=1}^{n} n^2 p_j^2 & \text{if } i = 1; \\ 2\sum_{j=1}^{n-i+1} n^2 p_j p_{j+i-1} & \text{if } 1 < i \leq n. \end{cases} \tag{3.2}$$

To explain and observe that the case of rating distance d_1 only occurs when both ratings in a rating pair are identical, i.e., (l_j, l_j), the probability of an identical rating pair with both rating

values l_j is the multiplication of their respective probabilities, i.e., p_j^2. Hence, the estimated number of pseudo-observations (at the rating distance of d_1) is $n^2 p_j^2$, and thus the total number of such kind of pseudo-observations is the summation over all the possible rating values, i.e., $j \in [1, n]$. For the other distance levels $d_i, 1 < i \leq n$, two possible combinations (l_j, l_{j+i-1}) and (l_{j+i-1}, l_j) could produce the same rating distance, leading to the estimated number of pseudo-observations being $n^2(p_j p_{j+i-1} + p_{j+i-1} p_j) = 2n^2 p_j p_{j+i-1}$. We then iterate all the possible rating values to yield the total number of pseudo-observations of this kind.

Since the values of parameter α_i have important influence on the computation of posterior probability for the Dirichlet distribution, we then proceed to determine the values of probabilities p_j (see Equation 3.2) for this purpose. In this work, we consider two possible ways, i.e., to learn from training data or to presume a simple uniform distribution. The experimental results show that the uninformed parameters (i.e., $p_j = 1/n$) work as fine as learning from the training data. One possible explanation is that learning the exact distribution of ratings from the training set may give rise to certain over-fitting.

3.1.2 Evidence Weights

New evidence for the Dirichlet distribution is often represented by a vector. Specifically, we can represent the rating pair $(r_{u,k}, r_{v,k})$ by a vector $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_n)$ where $\gamma_i = 1$ (where i is such that $d_i = |r_{u,k} - r_{v,k}|$) and the remaining entries equal zero. For example, a rating pair (5, 3) on a certain item can be represented as $\boldsymbol{\gamma} = (0, 0, 1, 0, 0)$ if the rating scale is the integers from 1 to 5. Such an evidence vector treats all evidences equally. However, in this work we claim that not all the evidences should and will be considered as equally useful for similarity computation. Three evidence factors are taken into account for this purpose.

Rating consistency. The first factor that we propose posits that realistic user similarity should be calculated based on the (reliable) items with consistent ratings, and using the (unreliable) items with inconsistent ratings is risky and may cause unexpected influence on similarity computation. The motivation is because of the observation that most users tend to give positive ratings overall, for example, in Epinions.com most users give rating values 4 and 5 (out of 5). Hence, it would be valuable to focus more to distinguish the ratings on the consistent items.

Rating consistency is determined by two factors: (1) the standard deviation σ_k of ratings on item k; and (2) the rating tendency of all users. First, generally, the value of σ_k reflects the extent of inconsistency of user ratings on item k. We define the acceptable range of rating deviations by $c\sigma_k$, where c is a scale constant that can be adapted for different data sets. Second, however, the value of σ_k may be less meaningful if the ratings on all items are highly deviated, i.e., users tend to disagree with each other in general. In this case, we consider the distance between the mode r_m and mean r_μ of ratings, i.e., $d_{m,\mu} = |r_m - r_\mu|$. Since the mode represents the most frequently occurred value, the distance $d_{m,\mu}$ reflects the tendency of all user ratings. The greater the value of $d_{m,\mu}$ is, the more deviated user ratings are indicated and the less meaningful σ_k will be. When $d_{m,\mu} > 1$,[3.2] σ_k is not meaningful at all.

Hence, the *important evidences* will be those whose rating distance for reliable item k is within a small range $c\sigma_k$, given that users achieve agreements in most cases. We define the evidence weight of γ_i as:

$$\varphi_k^i = \begin{cases} 1 & \text{if } c\sigma_k = 0; \\ 1 - \frac{d_i}{c\sigma_k} & \text{if } 0 \leq d_i < 2c\sigma_k; \\ -1 & \text{otherwise,} \end{cases} \tag{3.3}$$

where the upper bound $2c\sigma_k$ is chosen to restrict the value range to be $(-1, 1]$.

Let σ be the standard deviation of all ratings in a recommender system. We restrict the important evidences within a range $c\sigma$ no more than the minimal rating value l_1, i.e., $c = l_1/\sigma$. In case that the distributions of user ratings are unknown or that users generally do not have consensus ratings, we may set $c = 0$ so as not to consider evidence weights, or simply fall back to treating all evidences as equally important. The settings of c in different data sets used in our experiments are given in Table 3.3. For BookCrossing[3.3], the mean and mode values are 7.6 and 8.0, respectively. Since $d_{m,\mu} = 0.4 \leq 1$, the value of c is given by $c = l_1/\sigma = 1.0/1.84 \approx 0.5$, where the standard deviation of all ratings is $\sigma = 1.84$. Therefore, for an item with standard deviation σ_k, the smaller rating distance d_i is, the more important the evidence is. In contrast for Epinions, the mean and mode values are 3.99 and 5.0, and hence $d_{m,\mu} = 1.01 > 1$. In this case, we will set $c = 0$ to treat all evidences equally.

[3.2] The value 1 is empirically determined based on the analysis of specifications of six real-world data sets that we will use in Section 3.3.

[3.3] Refer to Section 3.3.1 for the description of data sets used in our experiments.

Gaussian singularity. Another commonly-used factor for similarity computation is called *singularity* [82]. The intuition is that two users agree more if their ratings are close to one another but distinct from the majority, than if their ratings are close to the value of most users. Bobadilla et al. [82] formulate singularity based on opinions, i.e., positive or negative ratings. Specifically, a rating that is greater than a certain rating value (e.g., 3 in the range [1, 5]) is regarded as a positive opinion; otherwise it is negative. They define the singularity of a positive (respectively, negative) rating as the proportion of negative (respectively, positive) opinions relative to the set of all opinions. In other words, a positive rating has high singularity if most ratings are negative.

However, this formulation ignores the differences between positive (or negative) opinions, that is, two ratings 4 and 5 are indifferently treated as positive opinions with the same singularity. Hence, we propose a more refined and general definition of singularity—the likelihood that a rating is not attributable to the rating distribution. We use the assumption that a user's ratings given on all the items follow a Gaussian distribution, which can be adapted to both discrete and continuous rating scales. We term it *Gaussian singularity*.

Specifically, according to all the ratings of user u on item k, we can fit a Gaussian distribution $R \sim \mathcal{N}(\mu, \sigma)$, where μ and σ represent the average and standard deviation of user ratings. Then the singularity $\psi_{u,k}$ of a rating $r_{u,k}$ is computed using the probability density function as follows:

$$\psi_{u,k} = 1 - \frac{1}{\sqrt{2\pi}\sigma} exp\Big(- \frac{(r_{u,k} - \mu)^2}{2\sigma^2} \Big). \tag{3.4}$$

Thus, the singularity of a pair of ratings $(r_{u,k}, r_{v,k})$ is computed by:

$$\psi^i_{u,v,k} = \psi_{u,k} * \psi_{v,k}, \tag{3.5}$$

where i refers to the subscript of rating evidence γ_i.

To illustrate, if two users u and v have each rated a number of items, we can use the ratings to fit two Gaussian probability distributions of ratings, respectively for each user. Suppose that for user u, the fitted distribution is $R_u \sim \mathcal{N}(4.2, 1.0)$, i.e., with the mean 4.2 and standard deviation 1.0. Then, the singularity of giving a rating $r_{u,k} = 5$ to item k is $\psi_{u,k} = 0.71$. Similarly for user v, given a fitted distribution $R_v \sim \mathcal{N}(3.0, 1.0)$, the singularity of rating item k as $r_{v,k} = 5$ is: $\psi_{v,k} = 0.95$. As expected, $\psi_{v,k}$ is greater than $\psi_{u,k}$ since it is more singular

for user v to give a rating 5. Together, the overall singularity for two users giving both ratings as 5 (i.e., evidence γ_i) is: $\psi^i_{u,v,k} = 0.71 * 0.95 \approx 0.67$.

Although the Gaussian distribution is adopted in our work, it must be recognized that not all users' ratings follow exactly a Gaussian distribution. Other kinds of probability distributions may be used as an alternative. However, since we find that the Gaussian distribution produces good results (for most users), we will not work with alternative distributions here.

Rating semantics. Ahn [26] stresses the importance of considering the underlying semantics of a rating scale in the computation of user similarity. Specifically, Ahn [26] defines three semantic factors, namely *Proximity*, *Impact*, and *Popularity*. However, the formulation of these factors is not bounded (often greater than 1) and thus cannot be used as evidence weights in our method. Hence, we adapt and generalize the original definitions and give new formulations for each factor.

- **Proximity** reflects the difference of two ratings in terms of positive and/or negative opinions. For example, a pair of ratings (5, 3) is closer with each other than a pair of ratings (4, 2). Although the rating distance is the same, the former pair of ratings are both positive whereas the latter contains different opinions. Note that a rating that is less than the median of a rating scale is regarded as negative opinions; otherwise it is positive. Hence, we can define the agreement of two ratings as:

$$\text{agreement} = \begin{cases} \text{True} & \text{if } (r_{u,k} - r_{med})(r_{v,k} - r_{med}) \geq 0; \\ \text{False} & \text{otherwise.} \end{cases} \tag{3.6}$$

  where r_{med} is the median rating of a rating scale predefined by a recommender system, given by $r_{med} = (l_1 + l_n)/2$. Then the proximity is defined by:

$$pr^i_{u,v,k} = \begin{cases} 1 - \dfrac{d_i}{d_n} & \text{if agreement is True;} \\ -\dfrac{d_i}{d_n} & \text{otherwise.} \end{cases} \tag{3.7}$$

  where d_n is the maximal distance implied by a rating scale. Unlike the Gaussian singularity focusing on the differences of specific rating values, the *Proximity* views user ratings from a more abstract level—opinion. That is, both ratings 4 and 5 (out of 5) are regarding as the same positive opinions, but they differ in the level of singularity.

- **Impact** considers the extent to which an item is preferred or disliked by users. For example, a rating 1 (out of 5) means a user does not like an item at all while a rating 4 indicates a strong preference. To facilitate discussion, we denote $\mu = (r_{u,k} + r_{v,k})/2$ as the average rating of the pair. For a pair of ratings, we consider three cases:

  (1) If both ratings are positive, the greater μ is, the more preferred the item is.

  (2) If both ratings are negative, the smaller μ is, the more disliked the item is.

  (3) If the opinions are different, the smaller μ is, the less distinct two opinions are in terms of like and dislike.

  Based on considerations, we obtain the following formulations of the impact factor:

$$im^i_{u,v,k} = \begin{cases} \dfrac{\mu}{l_n} & \text{if case 1;} \\ 1 - \dfrac{\mu}{l_n} & \text{if case 2;} \\ -\dfrac{\mu}{l_n} & \text{if case 3.} \end{cases} \tag{3.8}$$

  where l_n is the maximal rating value predefined by a recommender system. Both cases 1 and 2 show positive impact on user similarity since users agree with each other, whereas case 3 has negative impact due to the disagreement in user opinions.

- **Popularity** is similar to singularity in that it gives bigger value to the ratings whose values are further away from the average rating of a specific item. For example, consider identical rating pairs (4, 5) for two items k and p: the proximity and impact measures for the two items will be the same. However, if the average rating of item k is 3 and that of item p is 4, then the first pair on item k should be more important since it reflects the similarity of two users better. We denote $\bar{r}_k$ as the average rating of a specific item, and $\bar{d}_k = |(r_{u,k} + r_{v,k})/2 - \bar{r}_k|$ as the distance between rating pair and the average. Hence, we compute the popularity as follows:

$$po^i_{u,v,k} = \begin{cases} \dfrac{\bar{d}_k}{d_n} & \text{if } (r_{u,k} - \bar{r}_k)(r_{v,k} - \bar{r}_k) \geq 0; \\ -\dfrac{\bar{d}_k}{d_n} & \text{otherwise.} \end{cases} \tag{3.9}$$

Having defined our three PIP factors, following Ahn [26], the rating semantics is defined by multiplying them together:

$$\eta^i_{u,v,k} = pr^i_{u,v,k} * im^i_{u,v,k} * po^i_{u,v,k}. \tag{3.10}$$

Factor integration. The proposed three evidence factors, namely rating consistency, Gaussian singularity, and rating semantics, reflect the different aspects of user ratings and rating pairs. Some of the factors can partially overlap, such as singularity and semantics, and some can be partially opposite, such as singularity and consistency. Hence, an effective combination of these three factors may bring the benefits and combat the drawbacks of each factor simultaneously. Specifically, without lack of generality and simplicity, in this work the commonly-used linear combination is adopted as follows. For the sake of simplicity, we drop the dependency subscripts u, v, k.

$$e_i = \beta_1 * \varphi^i + \beta_2 * \psi^i + \beta_3 * \eta^i, \tag{3.11}$$

where e_i is the overall evidence weight of a rating pair γ_i; β_1, β_2 and β_3 indicate the relative importance of the three factors, namely rating consistency, Gaussian singularity and rating semantics, respectively; they are constrained by $\beta_1 + \beta_2 + \beta_3 = 1$ and $\beta_1, \beta_2, \beta_3 \in [0, 1]$. Tuning the best settings of parameters β_1 and β_2 is typically done by cross validation. Hence, a linear combination (with two freedom degrees) can greatly reduce the searching space (in the range of $[0, 1]$) than using an affine combination with three independent parameters each of which varies in the whole space of real values. Nevertheless, as we mentioned, the overlapping between singularity and semantics may result in the dominance of one factor over the other. Hence, together with rating consistency, the best settings can be achieved. We will elaborate more in detail in Section 3.3.3.

3.1.3 Distance Similarity

Now the Dirichlet distribution can be updated based on the observations of new evidences. Specifically, for an observation of a vector γ, the posterior probability density distribution will be $p(\boldsymbol{x}|\boldsymbol{\alpha} + \boldsymbol{\gamma})$. This procedure can be conducted sequentially to update the posterior probability density distribution when any new rating pairs come in. Upon observation of N rating pairs $\boldsymbol{\gamma}^1, \ldots, \boldsymbol{\gamma}^N$, the latest posterior probability density function becomes $p(\boldsymbol{x}|\boldsymbol{\alpha} + \sum_{j=1}^{N} \boldsymbol{\gamma}^j)$. Hence, the probability of rating distance of a new rating pair being d_i given the observed data will be equivalent to the expected value of the probability variable x_i:

$$p(\mathcal{D} = d_i|\gamma^0) = E(x_i|\gamma^0) = \frac{\alpha_i + \gamma_i^0}{\alpha_0 + \gamma^0}, \tag{3.12}$$

where $\gamma_i^0 = \sum_{j=1}^{N} \gamma_i^j e_i^j$ and $\pi = \sum_{i=1}^{N} \gamma_i^0$. Note that γ_i^j represents the i-th component of the j-th observation γ^j and e_i^j denotes the evidence weight of the j-th observation given by Equation 3.11, hence γ_i^0 is the amount of accumulated evidences whose rating distance is d_i.

Based on the posterior probability of each rating distance, we define *user distance* as the weighted average of rating distances d_i according to their importance weights w_i:

$$d_{u,v} = \frac{\sum_{i=1}^{n} w_i \cdot d_i}{\sum_{i=1}^{n} |w_i|}, \tag{3.13}$$

where $d_{u,v}$ denotes the distance between two users u and v, and w_i represents the importance of the rating distance d_i according to the amount of cumulated evidence γ_0^i between users u and v. For simplicity, we neglect the symbols u, v for importance weights. Intuitively, the more new evidences that are accumulated at a rating distance d_i, the more important the distance d_i will be. Hence, the importance weight of d_i is computed by:

$$w_i = \max\big(0, p(d_i|\pi) - p(d_i)\big) = \max\Big(0, \frac{\alpha_i + \gamma_0^i}{\alpha_0 + \pi} - \frac{\alpha_i}{\alpha_0}\Big) = \max\Big(0, \frac{\alpha_0\gamma_0^i - \alpha_i\pi}{\alpha_0(\alpha_0 + \pi)}\Big), \tag{3.14}$$

where we constrain $w_i > 0$ in order to remove the situation where posterior probability is less than the priori probability, which can arise when a rating level receives very few evidences (relative to all the evidences). We then normalize the distance to derive user similarity:

$$s'_{u,v} = 1 - \frac{d_{u,v}}{d_n}, \tag{3.15}$$

where $s'_{u,v}$ denotes the 'raw' similarity between two users u and v, and d_n is the maximum rating distance.

3.1.4 Chance Correlation

Until now, we have defined user similarity according to the distributions of rating distances. However, it is possible that two users are regarded as similar just because their rating distances happen to be relatively small, especially when the number of ratings is small. Hence it would be useful to reduce such correlation due to chance, or *chance correlation* for short. As described in Section 3.1.3, γ_i^0 out of γ^0 evidences located at the level of distance d_i. Recall that the prior probability of rating pairs with rating distance d_i is α_i/α_0, and so the chance that γ_i^0 evidences

fall in that level independently will be $(\alpha_i/\alpha_0)^{\gamma_i^0}$. Hence, the chance correlation is computed as the probability that amount of evidences fall in different rating distances independently:

$$s''_{u,v} = \prod_{i=1}^{n} (\frac{\alpha_i}{\alpha_0})^{\gamma_i^0}, \tag{3.16}$$

where $s''_{u,v}$ is the chance correlation between users u and v. It is observed that small values of γ_i (i.e., few evidences) lead to large chance correlation while big values of γ_i (i.e., many evidences) result in indiscernible chance correlation.

3.1.5 System Bias and Bayesian Similarity

The final consideration we treat is that similarity measures usually possess a certain level of *system bias*, i.e., the estimated similarity tends to be higher or lower to some extent than the realistic similarity. More intuitively, the system bias is partially due to the bias caused by the formulation of similarity measures. For example, PCC removes user averages when computing user similarity whereas COS does not. We will elaborate this issue later in Section 3.2.2. Therefore, user similarity is derived by excluding the chance correlation and system bias from the 'raw' similarity, and further bounded no smaller than 0 by a maximum function, i.e.,

$$s_{u,v} = \max(s'_{u,v} - s''_{u,v} - \delta, 0), \tag{3.17}$$

where $s_{u,v}$ denotes the user similarity between users u and v, and δ is a constant representing the general system bias. As analyzed in Section 3.2.2, our method will generally hold a limited system bias around 0.04, i.e., $\delta = 0.04$, given that only rating consistency is used to compute evidence weights. However, if three evidence factors are effectively combined together, since they may complement with each other, the system bias could be ignorable, i.e., $\delta = 0$ as discussed in Section 3.3.

3.1.6 Algorithm and Example

The pseudo-code of the computation of Bayesian similarity for two users u and v is presented in Algorithm 1. Specifically, the ratings of users u and v are taken as input and the computed Bayesian similarity is returned as the output. Initially, we declare two variables sum_d and sum_w (line 1) to accumulate the summation of weighted distances and importance weights

Input : user u's ratings R_u, user v's ratings R_v, Dirichlet parameters α_i
Output: Bayesian similarity between users u and v, i.e, $s_{u,v}$

1 set $sum_d \leftarrow 0$, $sum_w \leftarrow 0$;
2 **foreach** $(r_{u,k}, r_{v,k}), k \in I_{u,v}$ **do**
3 rating distance $d_i \leftarrow |r_{u,k} - r_{v,k}|$;
4 compute rating consistency φ_k^i by Equation 3.3;
5 compute Gaussian singularity $\psi_{u,v,k}^i$ by Equation 3.5;
6 compute rating semantics $\eta_{u,v,k}^i$ by Equation 3.10;
7 combine three factors to obtain evidence weight e_i by Equation 3.11;
8 compute importance weight w_i of rating distance d_i by Equations 3.14 and 3.12;
9 **if** $w_i > 0$ **then**
10 $sum_d \leftarrow sum_d + w_i * d_i$;
11 $sum_w \leftarrow sum_w + |w_i|$;
12 compute user distance $d_{u,v}$ by Equation 3.13: $d_{u,v} = sum_d / sum_w$;
13 compute the 'raw' similarity $s'_{u,v}$ by Equation 3.15;
14 compute chance correlation $s''_{u,v}$ by Equation 3.16;
15 **return** Bayesian similarity $s_{u,v}$ by Equation 3.17;

Algorithm 1: The Computation of Bayesian Similarity

(lines 9-11), respectively. For each rating pair with respect to the co-rated items (line 2), we first obtain the rating distance d_i (line 3), and then compute the three evidence factors subsequently (lines 4-6) which will be combined to yield the evidence weight w_i by Equation 3.11 (line 7). After accumulating all the values in sum_d and sum_w (lines 9-11), we can compute user distance $d_{u,v}$ by Equation 3.13 (line 12) and thus the 'raw' user similarity by Equation 3.15 (line 13). Once the chance correlation is computed (line 14), the Bayesian similarity for the two users can be derived and returned by Equation 3.17 (line 15), i.e., by removing the chance correlation and system bias from the 'raw' user similarity.

Regarding the time complexity of Algorithm 1, the most time-consuming part is the **foreach** loop in lines 2-11. Specifically, for each iteration, the computational complexity for each step (e.g., Equation 3.3 in line 4) can be completed in $O(1)$. Hence, the overall time complexity is $O(m)$, where m is the average number of co-rated items between two users. In other words, our similarity measure is linear to the number of items commonly rated by the two users. Therefore, the time complexity of Bayesian similarity is at the same order of magnitude as PCC and COS. This is also confirmed in our experiments where no significant difference is observed in terms of computational cost.

Here we give an intuitive example to show the procedure of Algorithm 1 step by step. Suppose that two users u and v have rated four items in common, and their rating profiles are $[2, 4, 4, 1]$ and $[4, 2, 2, 5]$, respectively. First of all, we need to determine the values of parameters α_i by Equation 3.2. We use the uniform distribution and thus $p_j = 1/5 = 0.2$ if rating values vary from 1 to 5 (i.e., $n = 5$). Accordingly, we can obtain: $(\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5) = (5, 8, 6, 4, 2)$ and $\alpha_0 = \sum_{i=1}^{5} \alpha_i = 25$. Hence, the prior probability distribution will be $(p(d_1), p(d_2), p(d_3), p(d_4), p(d_5)) = (0.2, 0.32, 0.24, 0.16, 0.08)$. For simplicity, we set $\beta_1 = 1, \beta_2 = \beta_3 = 0$, i.e., only the factor rating consistency is considered. Since the characteristics of the whole data set is unknown, according to Equation 3.3, the parameter c is set to 0 and the evidence weight $e_i = \varphi^i = 1$ for each rating pair. After collecting $\gamma^0 = 4$ new rating pairs, the posterior probability distribution turns to be $\big(p(d_1|\gamma_0), p(d_2|\gamma_0), p(d_3|\gamma_0), p(d_4|\gamma_0), p(d_5|\gamma_0)\big) = (0.172, 0.276, 0.310, 0.138, 0.103)$. Hence, the importance weights can be computed by Equation 3.14: $(w_1, w_2, w_3, w_4, w_5) = (-0.028, -0.044, 0.07, -0.022, 0.023)$. Using Equation 3.13, the user distance $d_{u,v}$ is obtained by: $d_{u,v} = (0.07*2+0.023*4)/(0.07+0.023) = 2.49462366$. Hence, the 'raw' similarity is derived by $s'_{u,v} = 1 - d_{u,v}/d_n = 1 - 2.49462366/4 = 0.376344$. Then, the chance correlation by Equation 3.16 is given by: $s''_{u,v} = 0.2^0 * 0.32^0 * 0.24^3 * 0.16^0 * 0.08^1 = 0.0011$, and the system bias is taken as 0.04. Overall, the Bayesian similarity is determined by: $s_{u,v} = \max(0.376344 - 0.0011 - 0.04, 0) = 0.335244 \approx 0.335$. This example is also presented as an instance (a_6) in Table 3.2, where the computed PCC is -1 and COS value is 0.681.

3.2 Similarity Measures Analysis

This section aims to provide intuitive examples of different similarity measures in dealing with the four issues summarized in Section 1.2.1, and to give insights into the nature of different similarity measures.

3.2.1 Examples

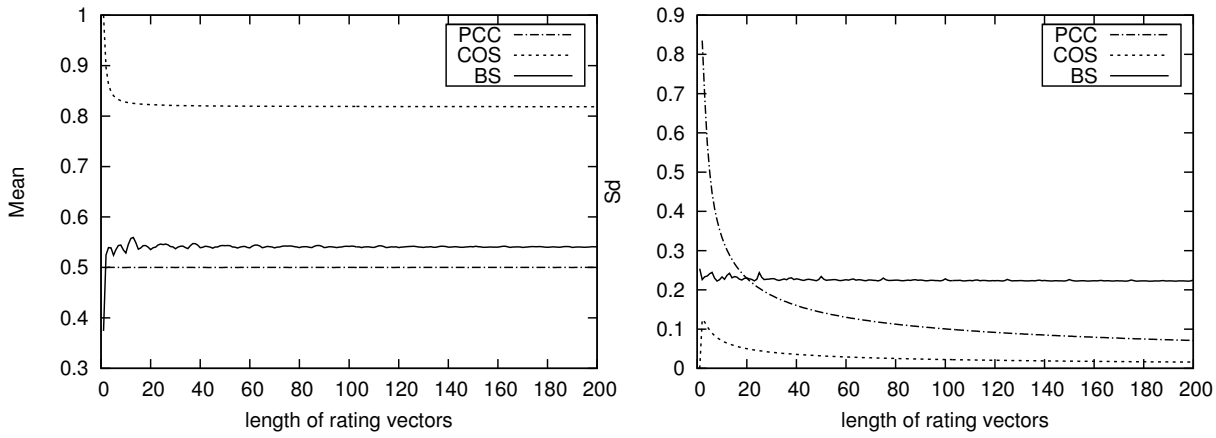
Earlier we summarized four specific problems from which PCC and COS suffer. Here we illustrate by examples the differences among the similarity values computed by our *Bayesian similarity* (BS) measure and the two traditional measures. We denote BS-1 as the variant of

our method that does not remove chance correlation. The results are shown in Table 3.2, where 'NaN' indicates not computable. All ratings in the table are integers in the range $[1, 5]$. We assume that the ratings are uniformly distributed, i.e., $p_j = 0.2$ for Equation 3.2. For simplicity, we only adopt the rating consistency to compute evidence weights, i.e., $\beta_1 = 1, \beta_2 = \beta_3 = 0$ for Equation 3.11, partially due to the observation that rating consistency works better than other factors which will be analyzed in Section 3.3.3.

Table 3.2: Examples of PCC, COS and BS similarity measures

Problems	**Examples**			**PCC**	**COS**	**BS**	**BS-1**
	ID	**Vector u**	**Vector v**				
Flat-value	a_1	[1, 1, 1]	[1, 1, 1]	NaN	1.0	0.952	0.96
	a_2	[1, 1, 1]	[2, 2, 2]	NaN	1.0	0.677	0.71
	a_3	[1, 1, 1]	[5, 5, 5]	NaN	1.0	0.0	0.0
Opposite-value	a_4	[1, 5, 1]	[5, 1, 5]	-1.0	0.404	0.0	0.0
	a_5	[2, 4, 4]	[4, 2, 2]	-1.0	0.816	0.446	0.46
	a_6	[2, 4, 4, 1]	[4, 2, 2, 5]	-1.0	0.681	0.335	0.336
Single-value	a_7	[1]	[1]	NaN	1.0	0.76	0.96
	a_8	[1]	[2]	NaN	1.0	0.39	0.71
	a_9	[1]	[5]	NaN	1.0	0.0	0.0
Cross-value	a_{10}	[1, 5]	[5, 1]	-1.0	0.385	0.0	0.0
	a_{11}	[1, 3]	[4, 2]	-1.0	0.707	0.332	0.383
	a_{12}	[5, 1]	[5, 4]	1.0	0.888	0.530	0.5616
	a_{13}	[4, 3]	[3, 1]	1.0	0.949	0.485	0.5623

It is observed that our method can solve the four problems of PCC and COS, and generate more realistic similarity measurements overall. Specifically, for the flat-value and single-value problems, PCC is non-computable and COS is always 1, whereas BS produces more reasonable similarities. In addition, BS generates higher similarity in a_1, a_2 than in a_7, a_8 respectively. Although the rating directions are the same, the former situations have a greater amount of rating evidences than the latter. Instead, BS-1 computes the same values in these cases where chance correlation is not considered. Overall, BS-1 tends to generate larger values than BS. The differences between BS and BS-1 could be non-trivial, especially when the length of rating vectors is short (e.g., $a_2, a_7, a_8, a_{12}, a_{13}$). Further, when the ratings are diametrically opposite (a_3, a_4, a_9, a_{10}), BS always gives 0 no matter how much information we have. However, COS continues to generate relatively high similarity; PCC may not be computable and hence these values are unreasonable. When the ratings are opposite but not extreme (a_5, a_6, a_{11}), PCC



(a) Mean

(b) Standard Deviation

Figure 3.1: The trends of similarity measures according to the variation of vector length

gives the extreme value -1 all the time and COS tends to produce high similarity, whereas the similarity calculated by BS is kept low. Finally, if the ratings are not crossing (a_{12}, a_{13}), PCC will yield 1 if computable and COS produces large values relative to BS even if some of the ratings are conflicting. Hence, these values are counter-intuitive and misleading, as pointed out by Ahn [26]. In contrast, our method can produce more realistic measurements.

3.2.2 Similarity Trend Analysis

We further investigate the nature of the three similarity measures in a more general manner. The trends of computed similarity values are analyzed when the length of rating vectors varies in a large range, using the same settings as in the previous subsection. In particular, a normal distribution is used to describe the distribution of user similarity. Since the similarity value is located in $[0, 1]$, the mean value of user similarity will be equal to the median of the normal distribution, i.e., 0.5. Note that for comparison purpose, PCC similarity is normalized from $[-1, 1]$ to $[0, 1]$ via $(1 + \text{PCC})/2$. We vary the length of rating vectors from 1 to 200. For each length, we randomly generate one million samples of two rating vectors and calculate the similarity for each pair by applying PCC, COS, and BS similarity measures. The mean and standard deviation for each length are summarized and shown in Figure 3.1.

For the mean value, PCC stays at the value of 0.5, while COS starts with high values and decreases quickly (length ≤ 10), reaching a stable state with the value of 0.82. Lin [109] con-

tends that one intuition a consistent similarity measure should obey is: the more commonality two users share, the more similar they are. In this regard, the COS similarity is counterintuitive in that it produces higher values when the length of rating vectors is short (i.e., when two users share less ratings), and lower values when the length of rating vectors is long (i.e., when two users share more ratings). In other words, the COS similarity is likely to be inconsistent when vector length is small. In contrast, BS begins with a low value at length 1 and then stays around 0.54 with a limited fluctuation when the length is short. Therefore, the BS similarity is more consistent than the COS similarity. These results indicate that in general for any two users: (1) PCC is able to remove system bias due to the data standardization involved; (2) COS always tends to generate high similarity around 0.82, i.e., with a large bias around 0.32; and (3) BS exhibits only a limited bias ($\delta = 0.04$) under the experimental settings. This phenomena is also observed by Lathia et al. [74] who find that in the MovieLens data set[3.4], nearly 80% of the whole community has COS similarity between 0.9 and 1.0, and that the most frequent PCC values are distributed around 0 (without normalization), which corresponds to 0.5 in our settings.

For the standard deviation, PCC makes large deviations when the length of vectors is less than 20, COS generates very limited deviation, whereas BS keeps a stable deviation around 0.22. A large deviation may cause the unstable values, i.e., inconsistent values are likely to be produced, while a small deviation may result in values that cannot be well distinguished from each other. In conclusion: (1) PCC is not stable and varies considerably when the vector length is short; (2) COS similarity is distributed densely around its mean value which makes it less distinguishable; and (3) BS tends to be distributed within a range of 0.22 which makes its value more easily distinguishable from others.

Note that our experiments assume that evidence weights are purely based on rating consistency. Under this condition, we find that our approach brings with it a limited system bias (0.04). As indicated by Equation 3.11, rating consistency can be combined with other evidence factors to form a more reliable and powerful factor to compute evidence weights. Thus, it is possible that the system bias can be further limited or eliminated by effectively combining the three evidence factors, and that the user similarity can be further distinguished by including more aspects of ratings. We will demonstrate the proposition in Section 3.3.4.

[3.4] movielens.umn.edu

3.3 Evaluation

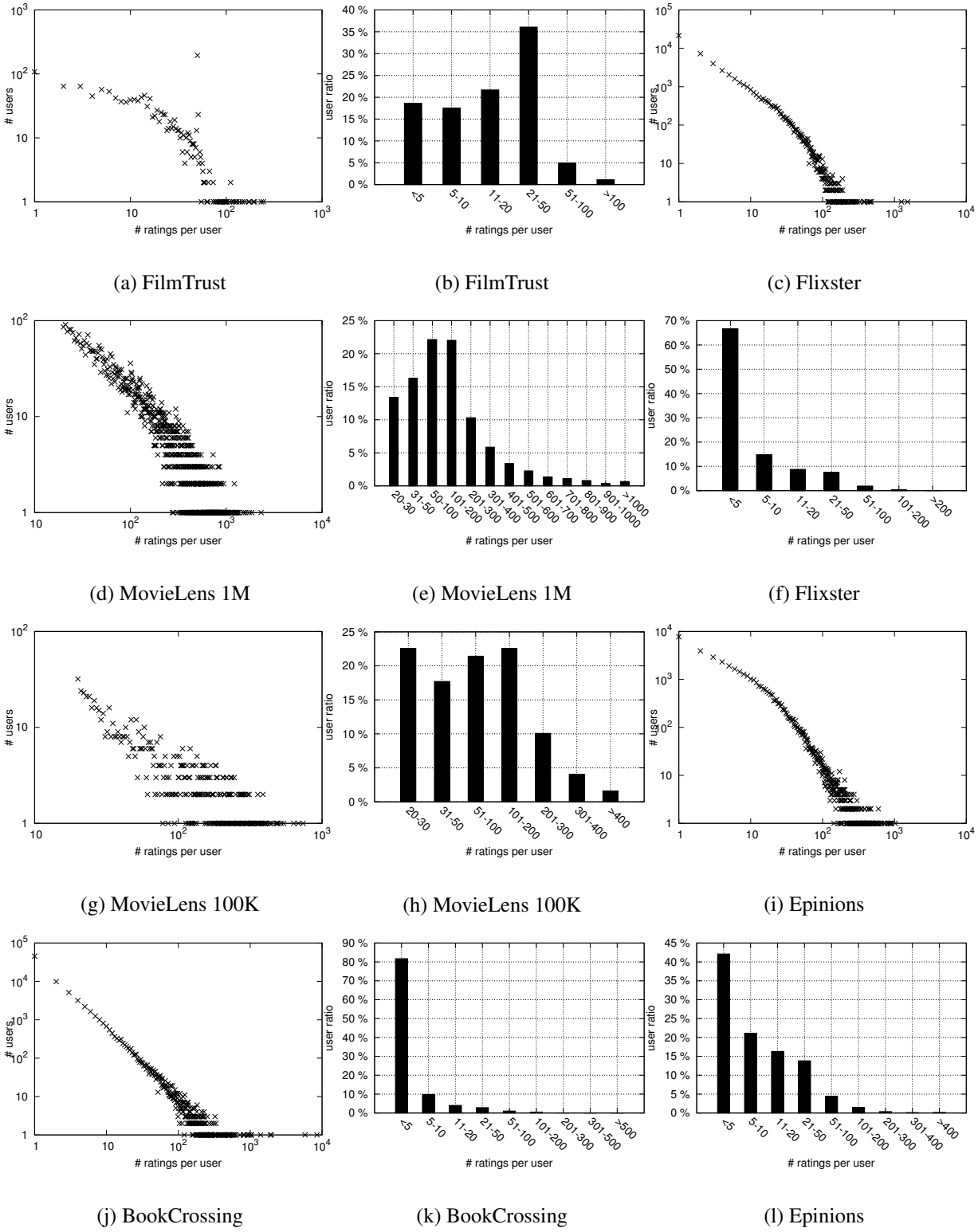
A series of experiments are conducted in this section to investigate: (1) the effects of different evidence factors as well as the best combinations of them on the performance of rating prediction; (2) the effects of chance correlation and system bias in our method; and (3) the comparison with other similarity measures in terms of predictive accuracy.

3.3.1 Data Sets

Six real-world data sets are used in our experiments; their statistics are illustrated in Table 3.3. They differ from each other in terms of predefined rating scales and density. BookCrossing.com is a free online book club to facilitate book sharing around the world. The data set[3.5] contains 433K ratings issued by 77.8K users on 186K books from the BookCrossing community. Epinions.com allows users to rate many different items (books, movies, etc.) by issuing an integer value from 1 to 5 and by adding textual review comments. The data set[3.6] includes 40.2K users, 139.7K items and 664.8K ratings. The remaining three data sets are all the online communities giving and sharing movie ratings with each other. Flixster[3.7] has the smallest rating density relative to the others and permits users to give more fine-grained and real-valued ratings from 0.5 to 5.0 with step 0.5. FilmTrust[3.8] is the smallest data set with only 35.5K user ratings. Notably, the two MovieLens data sets (100K and 1M)[3.9] have been pre-processed (by the GroupLens team) such that each user has rated at least 20 movies, resulting in the highest rating densities comparing with the others. The detailed specifications of all the data sets are presented in Table 3.3, together with the computed values of c (see Equation 3.3) in the last column for each data set.

In addition, the distributions of the number of users with respect to the number of ratings given by per user are illustrated in Figure 3.2. The figure shows that generally most users have only rated a small number of items (often no more than 20), and only a small portion of users have rated a large number of items. Since the MovieLens data sets have been pre-processed, each user has rated at least 20 items. On the other hand, different data sets show some distinct

[3.5] http://www.informatik.uni-freiburg.de/~cziegler/BX/
[3.6] http://www.trustlet.org/wiki/Epinions_datasets
[3.7] http://www.cs.sfu.ca/~sja25/personal/datasets/
[3.8] http://www.librec.net/datasets.html
[3.9] http://www.grouplens.org/node/12



(a) FilmTrust (b) FilmTrust (c) Flixster

(d) MovieLens 1M (e) MovieLens 1M (f) Flixster

(g) MovieLens 100K (h) MovieLens 100K (i) Epinions

(j) BookCrossing (k) BookCrossing (l) Epinions

Figure 3.2: The distributions of the number of users with respect to the number of ratings given per user across all the data sets

Table 3.3: The statistics of data sets used in the experiments

Data Sets	# Users	# Items	# Ratings	Scale	Density	c
BookCrossing	77.8K	186K	433K	[1, 10]	0.03%	0.5
Epinions	40.2K	139.7K	664.8K	[1, 5]	0.05%	0.0
Flixster	53.2K	18.2K	409.8K	[0.5, 5.0]	0.04%	0.0
FilmTrust	1508	2071	35.5K	[1, 5]	1.14%	0.6
MovieLens 100K	943	1682	100K	[1, 5]	6.30%	0.9
MovieLens 1M	6040	3952	1M	[1, 5]	4.47%	0.9

characteristics, for example, over 60% users have rated less than 5 items in Flixster and the ratio is even up to 80% in BookCrossing while the percentage is around 40% in Epinions and even less than 20% in FilmTrust. The distributions on the other ranges of rating amounts also present the differences to some extent. In conclusion, the data sets vary from each other and thus represent a number of different kinds of communities with different types of rating patterns among users.

3.3.2 Experimental Settings

We evaluate recommendation performance using the 5-fold cross validation method. The data set is split into five disjoint subsets; for each iteration, four folds are used as training data and one as a testing set. We apply the K-NN approach to select a group of similar users whose ranking is in the top K according to similarity; we vary K from 5 to 50 with step 5 in all the experiments. The ratings of selected similar users are aggregated to predict items' ratings by a mean-centring approach [110]:

$$p_{u,i} = \bar{r}_u + \frac{\sum_{v \in N_u} s_{u,v}(r_{v,i} - \bar{r}_v)}{\sum_{v \in N_u} |s_{u,v}|}, \tag{3.18}$$

where $p_{u,i}$ is the predicted rating for user u on item i, N_u is the set of top K nearest neighbours, $s_{u,v}$ is the user similarity between users u and v, $\bar{r}_u$ and $\bar{r}_v$ are the average of ratings given by users u and v, respectively. This approach is a classic user-based collaborative filtering.

To study more aspects of the utilities of different similarity measures on recommendation performance, we consider three different testing views in our experiments.

- **All Users** is the view where all the users' ratings in the testing set will be used.

- **cold-start users** refers to the view where only the testing ratings of cold-start users who rated less than 5 items in the training set will be predicted.

- **Niche Items** refers to the view where only the testing ratings of niche items which received less than 5 ratings in the training set will be evaluated.

The predictive accuracy is measured by two popular metrics, namely mean absolute error (MAE) and root mean square error (RMSE) between the prediction $p_{u,i}$ and the ground truth $r_{u,i}$ using the testing set:

$$\text{MAE} = \frac{\sum_{u,i\in\Omega} |p_{u,i} - r_{u,i}|}{|\Omega|}, \quad \text{RMSE} = \sqrt{\frac{\sum_{u,i\in\Omega}(p_{u,i} - r_{u,i})^2}{|\Omega|}} \tag{3.19}$$

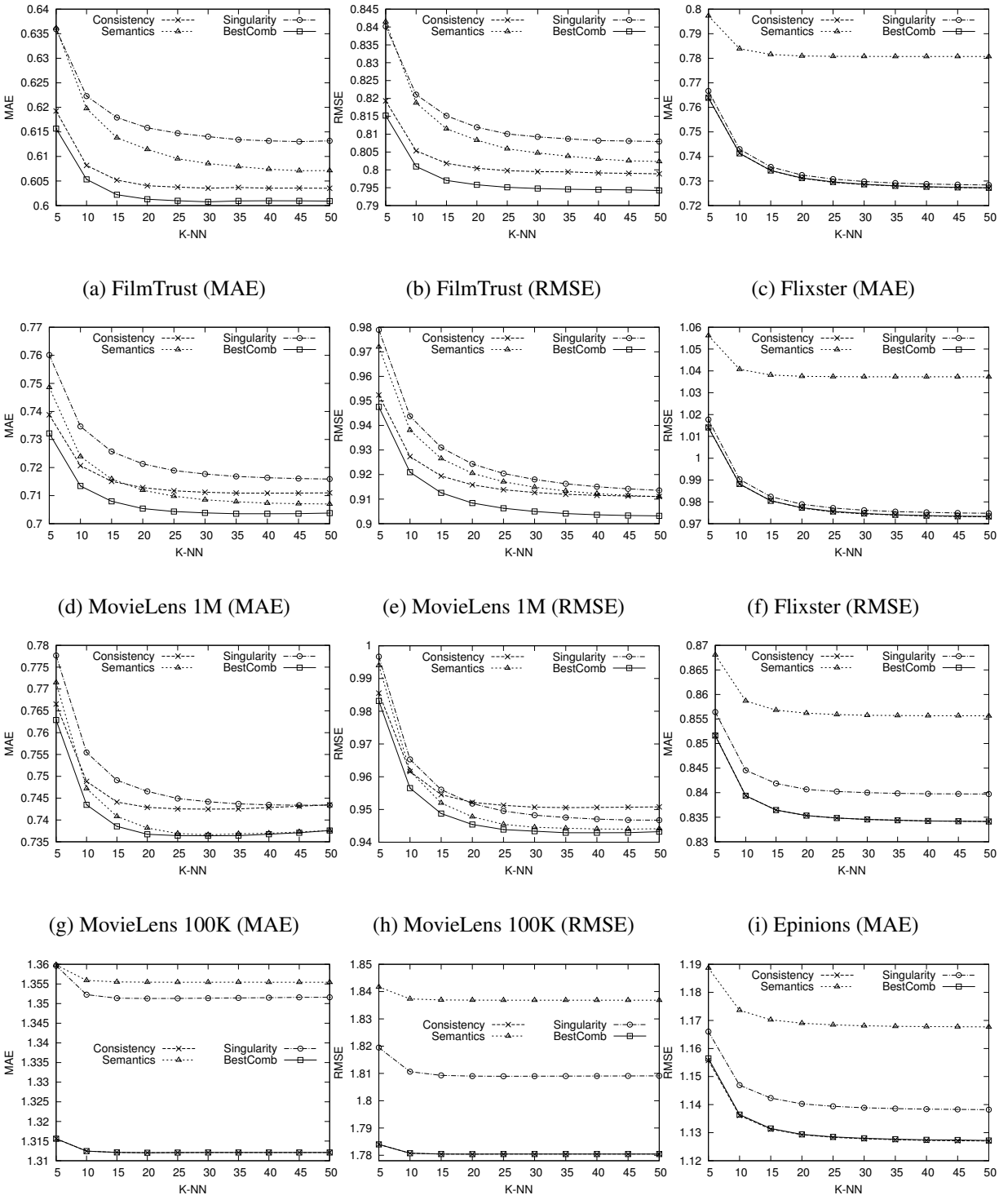
where Ω represents the testing set, and $|\Omega|$ is the cardinality of set Ω. Thus, lower MAE and RMSE values indicate better predictive accuracy. While our experiments use memory-based CF, we emphasize that similarity computation is equally relevant to model-based methods, including those based on matrix factorization such as Ma et al. [22] and Shi et al. [23].

3.3.3 Effects of Different Evidence Factors

Until now, we have introduced three different evidence factors to compute evidence weights, namely rating consistency, Gaussian singularity, and rating semantics. Hence it is necessary to investigate the impact of each evidence factor on the predictive performance as well as the best combination of three factors denoted by *BestComb*, obtained by tuning the values of parameters β_1 and β_2 in Equation 3.11. Specifically, we conduct an exhaustive grid search[3.10] of the possible combinations of (β_1, β_2) and obtain their performance based on 5-fold cross validation while setting the number of most similar users as 10 for predictions[3.11], i.e., $K = 10$. The experiments show that the best combinations of parameters (β_1, β_2) are (0.2, 0.1) on FilmTrust, (0.1, 0) on MovieLens 1M, (0.8, 0.1) on Flixster, (1, 0) on BookCrossing, (0.2, 0.2) on MovieLens 100K and (0.9, 0.1) on Epinions, respectively. It is noted that rating consistency (β_1) is more important on sparser datasets, i.e., Epinions, Flixster and BookCrossing (see Table 3.3), where users share less ratings in common. Then, we run 5-fold cross validation to show the

[3.10] The search space for each parameter is from 0 to 1 with step 0.1, with the constraint that $\beta_1 + \beta_2 \leq 1$.

[3.11] The setting $K = 10$ is just arbitrarily chosen . In fact, other values of K also hold the similar trends, indicating the suitability of our setting to investigate the effects of different evidence factors.



(a) FilmTrust (MAE) (b) FilmTrust (RMSE) (c) Flixster (MAE)

(d) MovieLens 1M (MAE) (e) MovieLens 1M (RMSE) (f) Flixster (RMSE)

(g) MovieLens 100K (MAE) (h) MovieLens 100K (RMSE) (i) Epinions (MAE)

(j) BookCrossing (MAE) (k) BookCrossing (RMSE) (l) Epinions (RMSE)

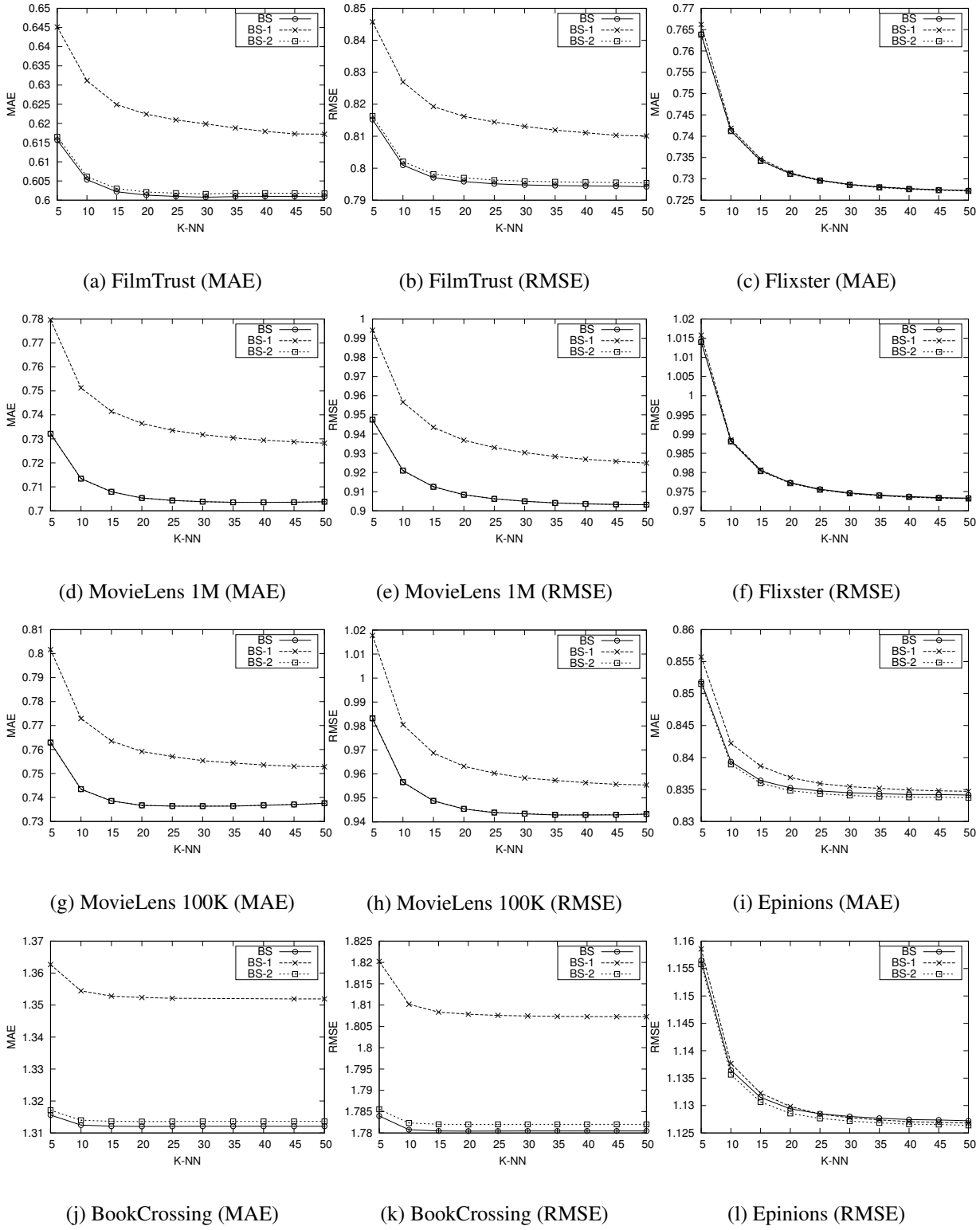
Figure 3.3: The predictive performance using different evidence factors for all users

predictive performance of different evidence factors. The results are illustrated in Figure 3.3. Significance tests (paired t-tests, confidence 0.95) are conducted between the best combination BestComb and the best of other single evidence factors. The significance test results are presented in Table 3.4.

In general, the method with the best combination of three evidence factors, i.e., BestComb, achieves the best performance across all the data sets, and different individual evidence factor may have various effect on different data sets. More specifically, rating consistency reaches comparable results with BestComb on the Flixster, BookCrossing and Epinions data sets, and outperforms Gaussian singularity and rating semantics on FilmTrust, but only performs worse than other single factors on two MovieLens data sets. One possible explanation is that users in MovieLens are likely to share more ratings in common since each of them has rated at least 20 movies. Gaussian singularity performs the worst on FilmTrust, MovieLens data sets, while rating semantics is demonstrated the worst on the rest of data sets. In conclusion, as a single evidence factor, rating consistency is likely to be more reliable and effective than rating semantics and Gaussian singularity, while BestComb can always outperform the others over all the data sets. This may be explained by the fact that rating consistency focuses more to distinguish similar ratings and that most users tend to give positive ratings, i.e., most ratings are likely to be similar to some extent. Recall that in Section 3.2.2 our method can produce more realistic and distinguishable user similarities. In contrast, Gaussian singularity tends to consider more dissimilar ratings while rating semantics attempts to assume that ratings are distributed randomly. A proper integration of these evidence factors may benefit from each single factor and give the best predictive accuracy. Further, we note that rating consistency and semantics consistently have more important impact (i.e., greater coefficients) than Gaussian singularity across all the data sets. In other words, concentrating more on the similar ratings and taking into account their rating semantics can give the best value for similarity computation. One possible reason is that rating semantics has some overlapping with Gaussian singularity as explained in Section 3.1.2.

3.3.4 Effects of Chance Correlation and System Bias

After determining the best settings for parameters β_1 and β_2 in Equation 3.11, we proceed to explore the effects of the other two components of our approach BS, namely chance correlation



(a) FilmTrust (MAE) (b) FilmTrust (RMSE) (c) Flixster (MAE)

(d) MovieLens 1M (MAE) (e) MovieLens 1M (RMSE) (f) Flixster (RMSE)

(g) MovieLens 100K (MAE) (h) MovieLens 100K (RMSE) (i) Epinions (MAE)

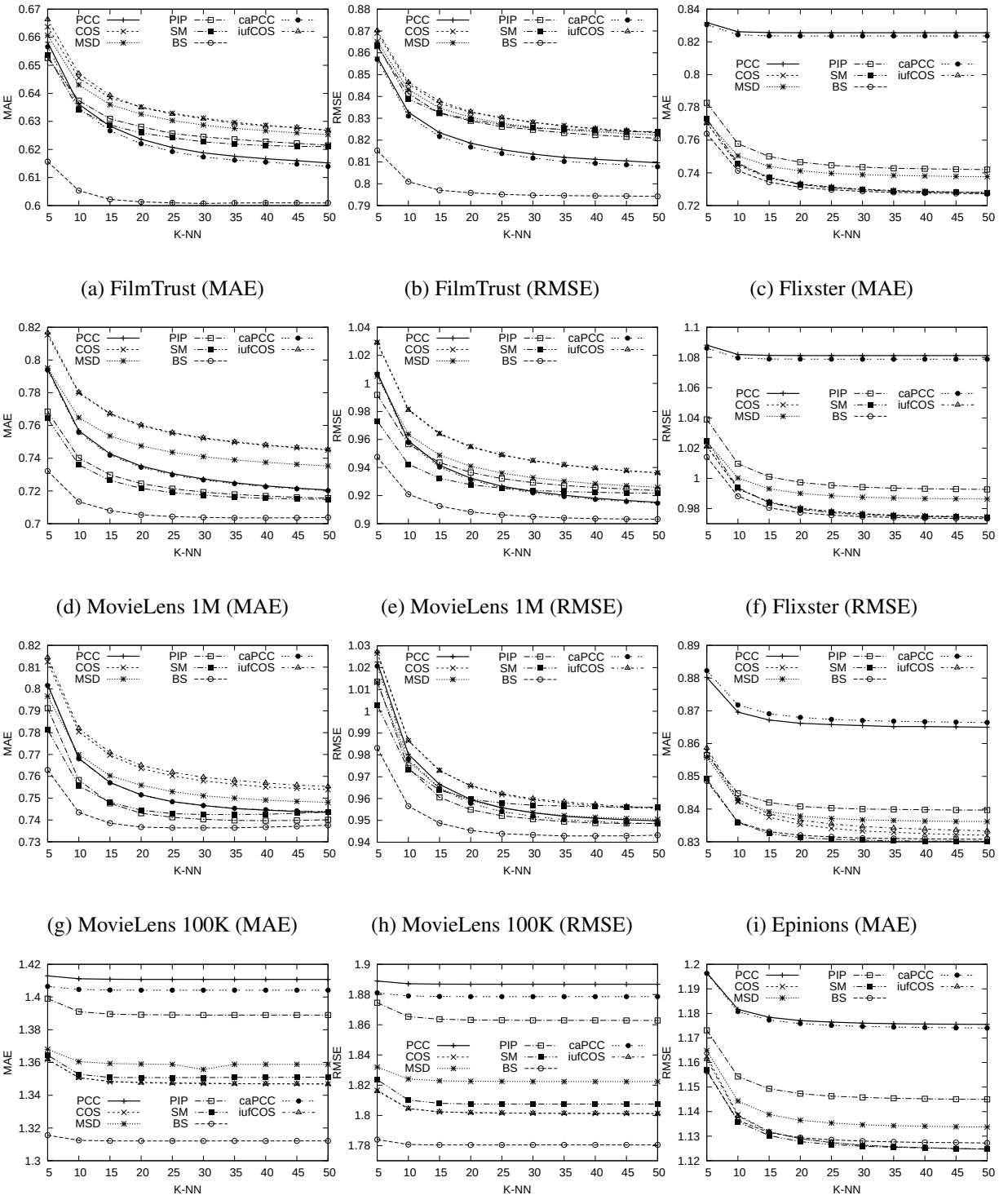
(j) BookCrossing (MAE) (k) BookCrossing (RMSE) (l) Epinions (RMSE)

Figure 3.4: The effects of disabling chance correlation (denoted by BS-1) or system bias (denoted by BS-2) from the Bayesian similarity on all users.

Table 3.4: Significance tests of the best combination *BestComb* w.r.t. the best of other single evidence factors in terms of MAE and RMSE across all the data sets, where p-values are denoted by the significance symbols: < 0.05 with *, < 0.01 with **, < 0.001 with ***; and 'NA' means not computable (or available).

Data Set (MAE)	**df**	t **value**	p **value**	**Best of Single Factors**
FilmTrust	9	-30.0271	1.232e-10***	Consistency
Flixster	9	4.1408	0.9987	Consistency
MovieLens 100K	9	-2.0664	0.03438*	Semantics
MovieLens 1M	9	-5.1098	3.183e-4***	Semantics
Epinions	9	NA	NA	Consistency
BookCrossing	9	NA	NA	Consistency
Data Set (RMSE)	**df**	t **value**	p **value**	**Best of Single Factors**
FilmTrust	9	-65.3828	1.156e-13***	Consistency
Flixster	9	3.5094	0.9967	Consistency
MovieLens 100K	9	-2.9583	8.002e-3**	Semantics
MovieLens 1M	9	-23.1675	1.237e-9***	Consistency
Epinions	9	4.6288	0.9994	Consistency
BookCrossing	9	NA	NA	Consistency

and system bias. We denote BS-1 and BS-2 as the variants that disable chance correlation (setting $s''_{u,v} = 0$) and system bias (setting $\delta = 0$) in Equation 3.17, respectively. The experimental results are presented in Figure 3.4. It is observed that BS consistently outperforms BS-1 across all the data sets, though it is only slightly better than BS-1 on Flixster. Hence, we conclude that chance correlation is critical in our approach as disabling it will greatly decrease the predictive accuracy. However, the effect of the system bias is not as significant as chance correlation. Specifically, BS-2 achieves comparable results with BS on most data sets and even exceeds BS on Epinions. That is, disabling system bias (i.e., $\delta = 0$) may cause only slight decrement or even sometimes reach slight increment relative to BS (with $\delta = 0.04$) in terms of predictive accuracy. As a conclusion, it is indiscernible in accuracy to disable system bias, though setting a small value (0.04) may result in slightly better performance. As explained in Section 3.2.2, the value 0.04 is obtained by using only the rating consistency to compute evidence weights. However, since a good combination usually requires the consideration of other evidence factors as demonstrated in previous subsection, it may lead to a more limited or ignorable system bias.



(a) FilmTrust (MAE) (b) FilmTrust (RMSE) (c) Flixster (MAE)

(d) MovieLens 1M (MAE) (e) MovieLens 1M (RMSE) (f) Flixster (RMSE)

(g) MovieLens 100K (MAE) (h) MovieLens 100K (RMSE) (i) Epinions (MAE)

(j) BookCrossing (MAE) (k) BookCrossing (RMSE) (l) Epinions (RMSE)

Figure 3.5: The predictive accuracy of comparative approaches on all users

3.3.5 Performance Comparison on All Users

The baseline approaches for comparison are PCC, COS, MSD [80], inverse user frequency-based COS (denoted by iufCOS) and case amplification-based PCC (denoted by caPCC) [24].[3.12] Breese et al. [24] empirically suggest that the best value of the case amplification parameter for caPCC is $\rho = 2.5$. We go further and adopt a grid search for each of our data sets in the value set $\{0.5, 1, 2, 2.5, 3, 5, 10\}$ to find out the optimal ρ values across all testing views. Experimental results show that the setting of $\rho = 0.5$ consistently achieves the best performance across all the test cases. Besides these five methods, we also compare with recent approaches, namely PIP [26] and SM [82] which show better performance than a number of baselines, as described in Section 2.2.1. The performance of these approaches is shown in Figure 3.5 in terms of MAE and RMSE.

The results show that BS outperforms traditional measures (i.e., PCC and COS, also MSD) consistently in all the data sets. Of the traditional measures, the performance of MSD is always between that of PCC and COS. PCC works better than COS on some data sets including FilmTrust, MovieLens 100K and 1M data sets and worse in the others. One explanation is that PCC only removes local bias (the average of ratings on co-rated items) rather than global bias (the average of all the ratings); hence it is not a standard data standardization. With an optimized case amplification value (i.e., $\rho = 0.5$), caPCC slightly beats PCC consistently across all the data sets. On the other hand, with a discount of inverse user frequency, iufCOS works very closely to (or slightly worse than) COS, indicating the uselessness of weighting schemes for COS as reported by Said et al. [75]. Of the newer methods, SM generally works better than PIP except on MovieLens 100K. Interestingly, PIP and SM outperform the traditional methods only on the two MovieLens data sets. This underscores the necessity of comparing performance on several different data sets. Adomavicius et al. [111] also show that the accuracy of CF recommendations is highly influenced by the structural characteristics of data sets. In line with this conclusion, we observe that the performance of PIP varies on different data sets relative to other baselines. This may be explained by the grid formulation of the PIP method. For example, the factor of proximity [26] is set in such a way that the distance between two ratings will be doubled if they disagree with each other. Such a kind of setting may or may not work for some data sets, since it depends in part on the rating scale used in the data set. By contrast,

[3.12] Other variants of PCC exist in the literature, but we will not compare all of them in this work.

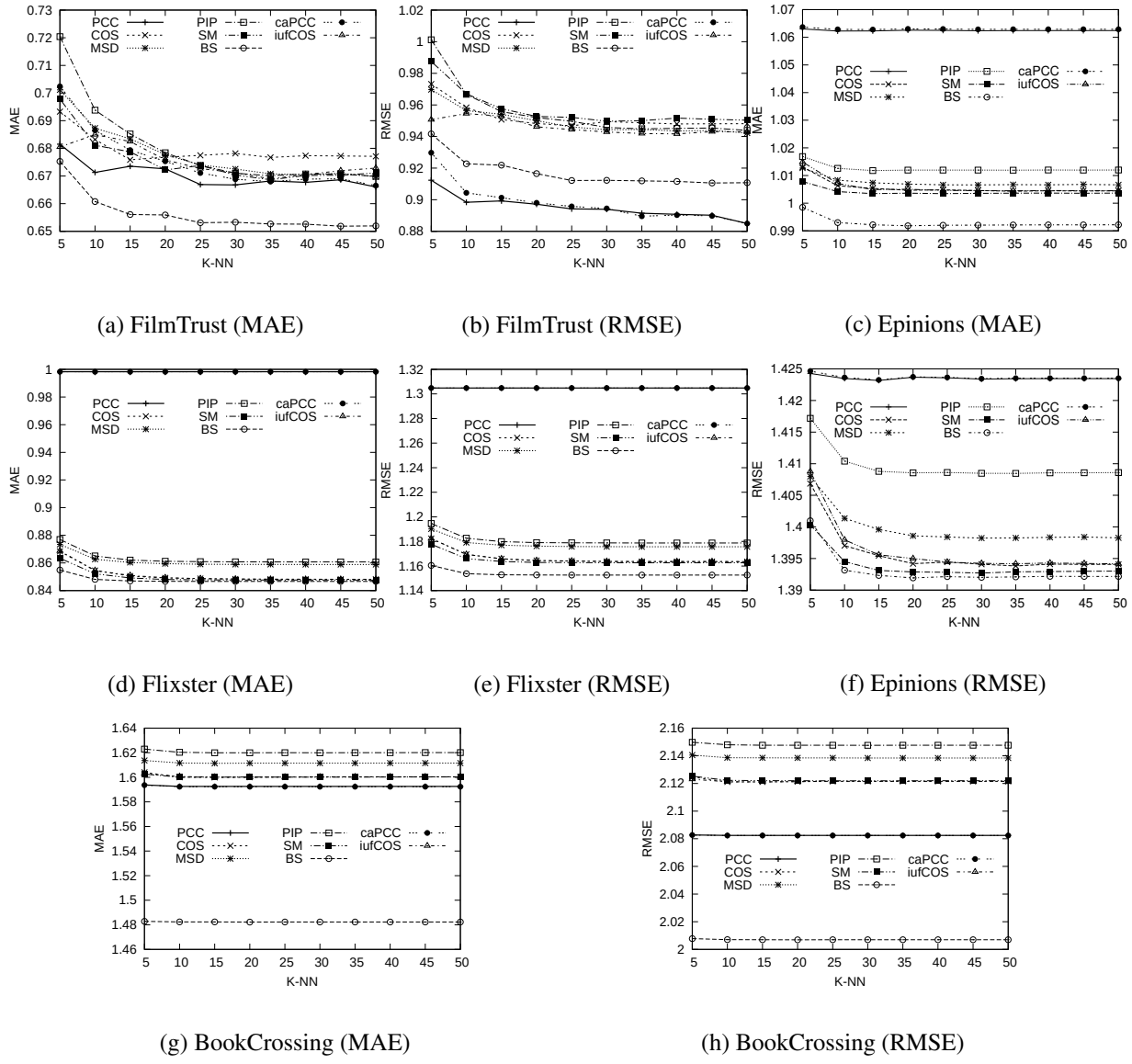
Table 3.5: Significance tests of the best combination *BestComb* w.r.t. the best of other methods on the view of all users in terms of MAE and RMSE. Note that for the Epinions data set, two tests are available where the second one is with the *second* in MAE (*third* in RMSE) best method. The p-values are denoted by the significance symbols: < 0.05 with *, < 0.01 with **, < 0.001 with ***.

Data Set (MAE)	**df**	t **value**	p **value**	**Best of Other Methods**
FilmTrust	9	-7.4407	1.965e-5***	caPCC
MovieLens 1M	9	-7.9515	1.162e-5***	SM
BookCrossing	9	-32.6342	5.859e-11***	COS
Flixster	9	-2.7009	0.01218*	SM
MovieLens 100K	9	-3.0699	6.678e-3**	PIP
Epinions	9	3.4852	0.9966	SM
Epinions	9	-4.1937	1.164e-3**	COS (the second best)
Data Set (RMSE)	**df**	t **value**	p **value**	**Best of Other Methods**
FilmTrust	9	-7.5298	1.790e-5***	caPCC
MovieLens 1M	9	-28.2894	2.096e-10***	SM
BookCrossing	9	-19.4044	5.926e-9***	COS
Flixster	9	-2.9437	8.194e-3**	SM
MovieLens 100K	9	-4.3736	8.939e-4***	PIP
Epinions	9	5.4422	0.9998	SM
Epinions	9	-34.0306	4.03e-11***	MSD (the third best)

our method performs better than both PIP and SM on all the data sets except Epinions, and exhibits greater improvements (with respect to the traditional approaches). On Epinions, BS and SM have very close performance and outperform the other methods.

In addition to the above experiments, we conduct a series of paired two sample t-tests on all the data sets to study the significance of accuracy improvement that our method achieves in comparison with the best of other methods (confidence level 0.95). The results are shown in Table 3.5, where the *alternative* hypotheses are: *the MAE (RMSE) of BS is significantly less than that of the best of other methods*. The resultant p values indicate that our method significantly outperforms all others on five out of the six data sets. Only on Epinions is BS slightly outperformed by another method (SM). However, this performance difference on Epinions between BS and SM is quite small: 0.00047 in MAE and 0.00162 in RMSE on average. In these cases, a further significance test is adopted to compare our method with the *second (or third) best* of other methods[3.13], i.e., COS in MAE (or MSD in RMSE), on Epinions. The results, also

[3.13]The purpose is to justify that our approach works relatively fine and comparable even in some special cases where it cannot beat all the other methods.



(a) FilmTrust (MAE) (b) FilmTrust (RMSE) (c) Epinions (MAE)

(d) Flixster (MAE) (e) Flixster (RMSE) (f) Epinions (RMSE)

(g) BookCrossing (MAE) (h) BookCrossing (RMSE)

Figure 3.6: The predictive accuracy of comparative approaches on cold-start users

in Table 3.5, show that BS achieves significantly better performance than the second/third best other method. Hence, looking across the range of data sets, we conclude that our method has the most robust performance of all the methods considered.

3.3.6 Performance Comparison on cold-start users

The performance on cold-start users is illustrated in Figure 3.6, and the corresponding significance tests are presented in Table 3.6 with respect to the best of other methods. Since users

Table 3.6: Significance test results on the view of cold-start users in terms of MAE and RMSE. The last test is between our method with the *second* best method in FilmTrust. The p-values are denoted by the significance symbols: < 0.05 with *, < 0.01 with **, < 0.001 with ***.

Data Set (MAE)	**df**	t **value**	p **value**	**Best of Other Methods**
FilmTrust	9	-13.0566	1.870e-7***	PCC
Flixster	9	-2.9292	8.389e-3**	SM
Epinions	9	-51.2374	1.032e-12***	SM
BookCrossing	9	-1680.479	<2.2e-16***	PCC
Data Set (RMSE)	**df**	t **value**	p **value**	**Best of Other Methods**
FilmTrust	9	19.2953	1.0	PCC
Flixster	9	-15.3618	4.585e-8***	SM
Epinions	9	-4.3494	9.259e-4***	SM
BookCrossing	9	-1495.982	<2.2e-16***	PCC
FilmTrust	9	-12.9557	1.999e-7***	iufCOS (the second best)

in the two MovieLens data sets rated at least 20 items, these data sets are not suitable for the experiments on cold-start users. A number of observations can be drawn from the experimental results. Firstly, different from the performance on all users, the differences between PCC and caPCC is negligible on cold-start users. In other words, the weighting scheme for PCC on cold-start users is not helpful. In contrast, although the curves of COS and iufCOS are still highly overlapped, the difference is that iufCOS works better than COS when K is small. Generally, PCC works better than COS in FilmTrust and BookCrossing but worse in Epinions and Flixster.

Secondly and surprisingly, PIP achieves poor performance (and even worse than MSD) in general, which seems conflicting with the conclusion reported by Ahn [26]. One possible explanation is that the experimental settings in Ahn [26] count the number of ratings used to calculate user similarity whereas we focus on the number of ratings issued by the users. In this regard, our setting is more realistic (for selecting cold-start users) as one interaction with other users (and thus used to compute user similarity) does not mean that the user only rated one item, and rather many items could be rated. In contrast, SM works relative better than the other baselines in all the data sets.

Lastly and most importantly, our approach BS in general works significantly better ($p < 0.05$, see Table 3.6) than the others across all the data sets except the FilmTrust in terms of RMSE. Specifically, as shown in Table 3.6 regrading the performance in FilmTrust, our approach BS works better than the best of other methods in MAE, but worse than PCC in RMSE.

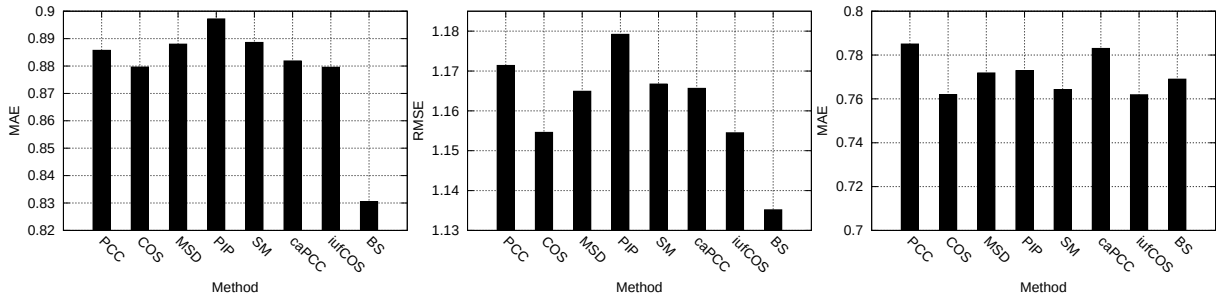
The lower MAE value indicates that the rating predictions by BS are closer to the ground truth than PCC (see Figure 3.6 (a)), while the higher RMSE value means that BS produces greater errors than PCC (see Figure 3.6 (b)). In other words, the rating predictions by BS tend to be either greatly approximated (mostly due to small MAE) or deviated (few due to relatively high RMSE) in FilmTrust. This can be attributed to the accuracy of computed factors. For example, when a user has only a few ratings, the average of her ratings could be deviated more than the case where many ratings are available. This may lead to incorrect estimation of the factors such as impact (see Equation 3.8), popularity (see Equation 3.9), Gaussian singularity (see Equation 3.4), etc. Another explanation for the variance between MAE and RMSE is due to the small size of FilmTrust, which makes the performance more sensitive to a number of high predictive errors. Nevertheless, from the last test presented in Table 3.6, the performance of our approach BS still performs significantly better than that of the second best of other methods.

3.3.7 Performance Comparison on Niche Items

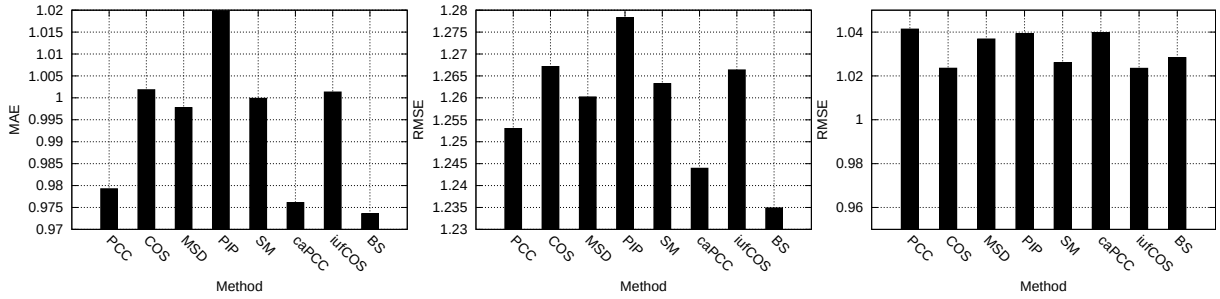
The performance on niche items is shown in Figure 3.7. By definition, niche items are those which received less than 5 ratings. Hence, there is no need to tune the number K of nearest neighbours for a specific user since the maximum number will be less than 5. Overall, the performance on niche items is similar to that on all users. Specifically, PCC is superior to caPCC while COS is of no difference from iufCOS. PIP shows no better results than the other baselines, but differently SM tends to act similarly as the others including MSD, COS and iufCOS. It is noted that our approach consistently outperforms all the others across different data sets except Epinions and Flixster where BS performs close to or only slightly worse than the best of other methods. It can be explained by that the characteristics of niche items are well captured by the factor of singularity and semantics, though rating consistency is of less help.

3.3.8 Summary and Discussion

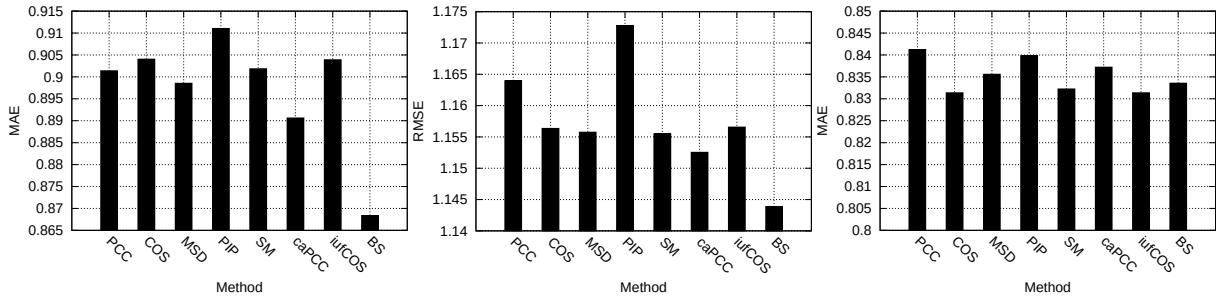
In summary, we have shown that in general our approach BS works better than the others in terms of both MAE and RMSE across a number of real-world data sets. Even in some special cases where our approach does not significantly outperform the others, the performance by BS is often equivalent to or only slightly worse than the best of other methods. Although not specialized for cold-start users or niche items, our approach demonstrates its generality



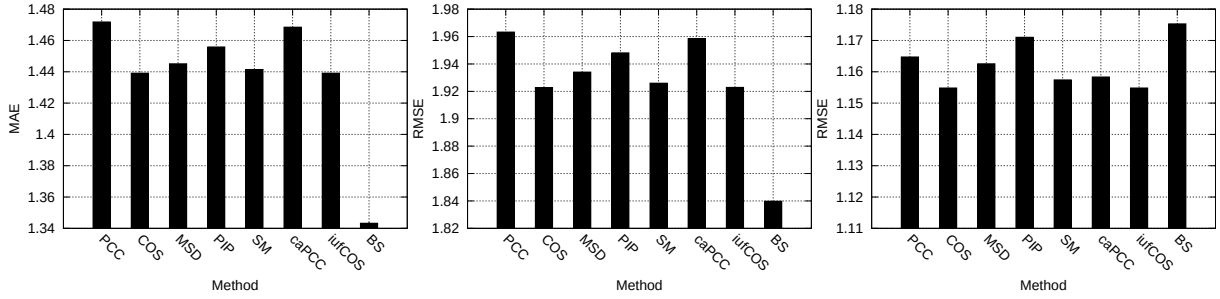
(a) FilmTrust (MAE) (b) FilmTrust (RMSE) (c) Flixster



(d) MovieLens 1M (MAE) (e) MovieLens 1M (RMSE) (f) Flixster



(g) MovieLens 100K (MAE) (h) MovieLens 100K (RMSE) (i) Epinions



(j) BookCrossing (MAE) (k) BookCrossing (RMSE) (l) Epinions

Figure 3.7: The predictive accuracy of comparative approaches on niche items

and good performance in different testing views. Further studying the performance on other samples of users or items would be an interesting part of the future work.

Nevertheless, it is worth noting that the used user-based KNN method for comparison is not competitive with advanced model-based approaches, such as matrix factorization models in terms of predictive accuracy[3.14] [76]. However, the user-based KNN method is suitable for the present work because our main objective is to verify the effectiveness of the Bayesian similarity measure (in comparison with others), rather than to justify that a user-based KNN with Bayesian similarity can beat all the other recommendation methods. In addition, we cannot over-emphasize that similarity measures can also be used in model-based approaches such as those of Ma et al. [22] and Shi et al. [23].

The main idea of our Bayesian similarity is to take into account the direction and length of rating vectors, and stress the importance of a number of evidence weighting factors in measuring user similarity. In principle, the same basic idea can be applied to measure item similarity by reformulating the weighting factors from the perspective of items rather than users. For example, the rating consistency can be modelled based on reliable users rather than reliable items, and the system bias is also applicable to item similarity. Designing a proper item similarity measure is an interesting topic and itself can be a separate line of research. Hence, in this study we will not discuss more on that topic.

3.4 Concluding Remarks

This chapter proposed a novel Bayesian similarity measure for recommender systems based on Bayesian inference, taking into account both the direction and length of rating vectors. We stressed the importance of evidence weights for the similarity computation and introduced three different evidence factors. We showed that an effective combination of these factors can achieve the best predictive accuracy. In addition, we found that removing chance correlation can significantly improve the computed user similarity, and that only a very limited or ignorable system bias may be caused by our method. Using typical examples, we exemplified that our Bayesian similarity was capable of addressing the four issues of traditional similarity measures (i.e., Pearson correlation coefficient and cosine similarity). More generally, we empirically

[3.14] A detailed comparison between UserKNN and matrix factorization methods can be found and reported at the page: http://www.librec.net/example.html

analyzed the trends of these measures, and found that our method can generate more realistic and distinguishable similarity measurements. Finally, the experimental results based on six real-world data sets further demonstrated the effectiveness and reliability of our approach in comparison with other counterparts in terms of predictive accuracy.

Chapter 4

Prior Ratings: A New Information Source

User ratings are the essence of recommender systems in e-commerce. Lack of motivation to provide ratings and eligibility to rate generally only after purchase restrain the effectiveness of such systems and contribute to the concerned issues, i.e., data sparsity and cold start. Although many approaches have been proposed to resolve these issues, only a few have attempted to elicit more ratings from the perspective of user interactions as we reviewed in Section 2.2.2. Different from Chapter 3 with existing ratings, this chapter introduces a new kind of user ratings, called *prior ratings* [18, 19], to enrich the user ratings with the aim to inherently resolve the data sparsity and cold start problems. We take advantage of user interactions with products represented in 3D virtual reality environments, i.e., virtual products. Although directly reflecting user preference, prior ratings differ from the traditional ratings in that prior ratings are based on virtual product experience in virtual reality environments rather than physical product experience in real world. Nevertheless, we will show that prior ratings are complementary to traditional ratings, and thus more ratings can be elicited and collected for better recommendations. For this purpose, we conduct a deep study about the nature of and the ways to incorporate prior ratings in collaborative filtering techniques.

The rest of this chapter is organized as follows. Section 4.1 details the proposed conceptual model of prior ratings, and proposes five related hypotheses and two research questions regarding its validity. Section 4.2 reports on a user study designed to validate the conceptual model. Then, Section 4.3 discusses the relations between prior ratings and other kinds of information sources for recommender systems as well as the limitations and implications of the user study and experimental results. Based on the prior ratings and confidence data collected from the user study, Section 4.4 designs a variant of the traditional collaborative filtering technique to

demonstrate the usefulness of prior ratings in improving recommendation performance. Finally, Section 4.5 concludes our present study.

4.1 Prior Ratings

We define the term *prior ratings* as users' assessment or judgement of preference of products in the light of their *virtual product experiences*, referring to the psychological and emotional states that users undergo while interacting with virtual products in a mediated environment [33], where a product can be only partially explored or experienced. Hence, prior ratings are reported by users based on their interactions with virtual products in a mediated environment, and they can be issued prior to purchase or after purchase (if any). Therefore, technically prior ratings could be given in any other mediated environments, such as augmented reality, as long as they can provide reliable virtual product experiences.

We refer to the 'standard' type of ratings derived from 'posterior' product experiences as *posterior ratings*. By 'posterior', we mean experiences of a tangible product obtained via direct trials or use of the product in a physical environment, where a product can be fully explored or experienced. Since tangible products can be fully experienced usually only after purchase, posterior ratings are primarily post-purchase ratings. Prior ratings and posterior ratings are distinct and complementary in that they reflect different forms of user experiences. The key difference is that prior ratings are based on limited yet reliable product experience whereas posterior ratings are based on full product knowledge and tangible product experience. In this regard, the environment where a product is represented is not the key difference. For example, if a movie's ratings are based on a 30-second trailer, users may not know much about the movie and such experience is unreliable. The ratings are neither prior nor posterior ratings. However, if the ratings are based on 10-minute trailer from which users can gain much more knowledge and understanding about the movie, then the experience can be regarded as reliable experience, and those ratings are prior ratings. Eventually, if users have watched the movie in some cinema, those ratings with full experience of the movie become posterior ratings. In this work, we focus on two general cases where mediated environments can provide reliable virtual product experience, and physical environments enable complete product experience.

Two kinds of mediated environments are investigated: traditional 2D websites (WS) and 3D virtual store (VR) environments. They differ in richness of both media and of interactions

through which product information can be delivered. WS only supports limited media and user interactions; VR real-time interactions enable users to possess a strong sense of being in a mediated environment and gain a lifelike shopping experience [112]. Specifically, products are represented in 3D virtual models through which users can view, rotate, zoom, customize and even try them on. Considering the rich virtual product experiences that users obtain in VR, we posit that VR will motivate users to express their opinions by providing prior ratings to the products of interest, and hence make more informative purchase decisions while shopping in VR.

Hypothesis 1 *Users are more willing to provide prior ratings to the items (e.g., products) that they have interacted with in VR than in WS.*

Although prior ratings can be submitted in both WS and VR as long as the user interfaces enable the rating functionality, the confidence level may differ. Specifically, due to limited media and interactions available in WS, users may have less adequate information than in VR as a basis for their ratings. Jiang and Benbasat [113] also contend that virtual products in VR help improve the perceived *diagnosticity* of products—the extent to which users believe a particular shopping experience is helpful to understand the quality and performance of a product. Therefore, users may feel more capable of forming direct, intuitive and concrete opinions about products in VR than in WS in terms of both rating confidence and rating values.

Hypothesis 2 *(a) Users have more confidence in providing prior ratings in VR than in WS; (b) the average value of prior ratings in VR is closer to that of posterior ratings than that of prior ratings in WS.*

4.1.1 Conceptual Model of Prior Ratings

We now present a conceptual model of prior ratings as shown in Figure 4.1. Such a model allows a principled basis for the elicitation and analysis of prior ratings. The objective of our conceptual model is to investigate a comprehensive understanding of the nature of prior ratings. Specifically, (1) how prior ratings are given by users, and (2) how other factors such as the presence of virtual reality and the attributes of virtual products impact on users' evaluation

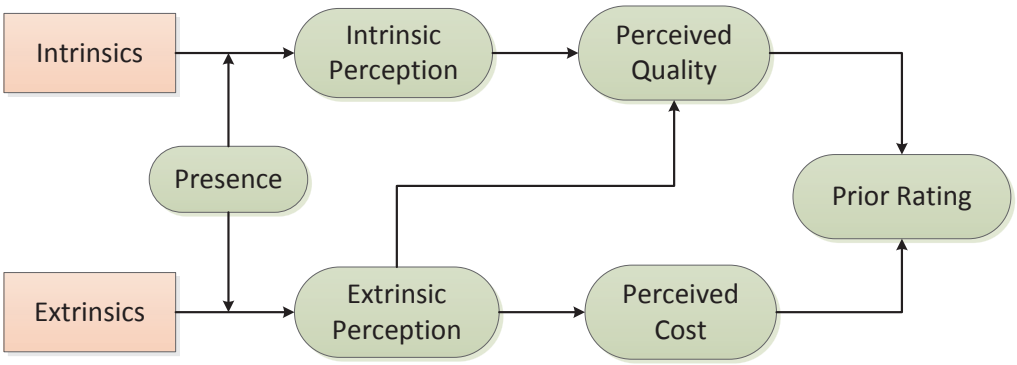


Figure 4.1: The conceptual model of prior ratings

of prior ratings. Only after a proper understanding of prior ratings, we will be able to show how to leverage them in a newly-designed collaborative filtering technique so as to resolve the data sparsity and cold start problems—which are our main concern in this thesis—in Section 4.4. Note that the conceptual model is not used to justify the effectiveness of prior ratings in resolving the data sparsity and cold-start problems, but to give a better comprehension and perception of prior ratings.

For a specific product, a number of intrinsic and extrinsic *attributes* are associated with it (see details in Sections 4.1.3 and 4.1.4). In different environments, the perceptions of these attributes can differ according to the types of media and interactions that deliver information about them. For example, VR environments may have better perceptions of products than traditional websites as the former generally enables richer media and real-time interactions. The intrinsic and extrinsic *perceptions* indicate the quality of products as perceived directly and indirectly, respectively. In contrast, the *perceived cost* (e.g., time, price) refers to the cost that users have to bear in order to obtain the products. A prior rating is an overall evaluation of preference of products in terms of both perceived quality and cost, i.e., a combination of what we 'get' and what we 'give'.

We next proceed to elaborate the details of the conceptual model.

4.1.2 Presence

Presence is defined as users' sense of "being there", the extent to which they experience the virtual environments as real or present and temporarily ignore where they are physically

present [114]. Two major determinants have been identified, namely *vividness* and *interactivity* [115]. First, vividness reflects the representational richness of a mediated environment as defined by its formal media through which information can be presented. Two important elements of vividness are *sensory breadth* which refers to the number of sensory dimensions simultaneously presented, and *sensory depth* which refers to the resolution within each perceptual channel. Second, interactivity is defined as "the extent to which users can participate in modifying the form and content of a mediated environment in real time" [115]. Three important elements, namely *speed*, *range*, and *mapping* describe the specification of a mediated environment in terms of response time, the amount of manipulable attributes, and the projections between human and environmental actions.

Hence, presence in this work is captured as the extent to which being in a mediated environment feels like being in a real environment[4.1], given the richness in media and interactions. Picciano [116] reports that the sense of social presence (i.e., the sense of belonging in a course and group) has a positive and statistically significant influence on the performance of students' written assignments in an online course. Phang and Kankanhalli [117] study how the perceptions of virtual world can enhance online learning. They show that in 3D environments, presence can enhance students' concentration and enjoyment during the learning process, and thus improve students' learning outcomes. These two works show that (1) it is important for learners to perceive a realistic classroom experience; and (2) such sense of being there can help them concentrate more on the learning contents. In the case of e-commerce recommendations, the presence of virtual reality enables users a lifelike shopping experience, and thus users may concentrate more on the product experience and evaluation. In addition, Heeter [118] stresses the importance of being able to change virtual environments, for instance, moving and painting a 3D object. A higher sense of presence can enable user interactions with 3D environments to be easier and more responsive. In our case, the 3D models of virtual products can respond to users' actions, e.g., rotating and zooming, and hence users may gain more direct comprehension about the properties (attributes) of products. Considering that the information concerning product attributes is conveyed by media channels and user interactions, presence can be an important environmental factor that will influence the perceptions of product attributes.

[4.1] Compare question 2 for the tested environments in Figure 4.3.

Hypothesis 3 *Presence has positive influence on the perceptions of both intrinsic and extrinsic attributes.*

Note that a higher sense of presence does not necessarily mean better perceived quality. Perceived quality is based on the perceptions of product attributes; presence is a moderator of the perceptions of product attributes.

4.1.3 Intrinsic Attributes

Intrinsic attributes (e.g., workmanship, size) have a direct impact on perceived quality during the goal-directed process of pre-purchase product evaluation [119]. Goering [120] also considers that intrinsic quality of a product has an important influence on the perceived quality of a product. Specifically, the higher the intrinsic quality of a product is, the better it will perform. In addition, intrinsic attributes can also work as cues to infer product quality [121]. For example, the attribute 'nutrition content' can be used as a cue to assess the quality of a breakfast cereal.

The specific intrinsic attributes embedded can vary between different products. In this work, we classify intrinsic attributes into three types, namely *appearance*, *material*, and *functionality*. Appearance refers to the attributes related to the superficial representation of products, such as patterns, form, size, etc. Material refers to the attributes associated with what products are made of, such as fabric properties, weight, etc. Functionality refers to the attributes indicating the utility of products or the actions that products can perform or that can be performed on products. For example, an electronic watch contains the functionality of stopwatch and it may 'fit' someone well.

More generally, Nelson [122] identified two different types of product attributes: *search attributes* and *experience attributes*. The former type refers to the attributes the information of which can be conveyed most effectively through secondhand sources, whereas the latter refers to the attributes the information of which can be evaluated most effectively by using products directly. Therefore, by definition, appearance and material attributes are more likely to be search attributes as their information can be easily obtained by searching. In contrast, functional attributes tend to be experience attributes since the effectiveness of functions requires the interactions with products, i.e., direct experiences.

Question 1 *What are the major intrinsic attributes that influence the perceived quality of products in WS and VR?*

4.1.4 Extrinsic Attributes

Unlike intrinsic attributes, extrinsic attributes (e.g., price, product type) have no direct indications of perceived quality. Rather, they are often used as cues to infer the quality of products when the information of intrinsic attributes is incomplete [123]. For example, considerable theoretical and empirical evidence [124, 125] shows that price is often used by users to infer the quality of products when it is the only available cue or when there is inadequate information about intrinsic attributes [123]. The rationale is that more cost is often required to produce high-quality products than low-quality products and the probability to charge high prices for low-quality products is low due to competitive pressures [126].

Other than price, brand and store are also well-studied in the literature. Brand name serves as a 'shorthand' for perceived quality by providing users with a bundle of information about the product [127]. It helps reduce the perception of risk prior to purchase in terms of financial, time, performance and psychological risk [128]. In comparison with price and brand name, store name also has a positive but small (not significant) impact on perceived quality [125]. Dodds et al. [123] also show that favourable brand and store information positively influence the perceptions of quality and value.

In addition, products can be categorized into different types in the light of different kinds of product attributes. For example, according to the definitions of search and experience attributes, products can be classified as search products and experience products [122]. Search products are those whose dominant attributes are search attributes and hence full information of them can be known prior to purchase without direct experience. In contrast, experience products are those whose dominant attributes are experience attributes and hence full information of them cannot be known until use of products (direct experience). Therefore, we have to highlight that not all products are suited in VR; specifically, experience products may perform better in VR whereas search products may perform better in WS in terms of user efforts in retrieving product information. In this work, we refer to product types as the *categories* of products. For example, there could be action and comedy movies. The categories are not deterministic and can be varied in different systems. Another reason for our definition of product type is that recommender systems usually focus on some specific product domains such as music and video rather than generic products.

Other extrinsic factors investigated in the literature may also have an effect on perceived quality, such as warranty [129], packaging [130], advertising [131], etc. We do not consider every kind of extrinsic attributes, but examine the extrinsic factors that will most significantly influence the quality of products. As noted, WS is most efficient to deliver information of search attributes and VR to convey information of experience attributes. Further, intrinsic attributes rely more on direct experience whereas information of extrinsic attributes can be found without use of products. Therefore, for a product that can be represented in both WS and VR, we come to the following question and assumption.

Question 2 *What are the major extrinsic attributes that influence the perceived quality of products in WS and VR?*

Hypothesis 4 *Users depend more on extrinsic attributes than intrinsic attributes to evaluate the product quality in WS, whereas users depend more on intrinsic attributes than extrinsic attributes to evaluate the product quality in VR.*

Besides quality, extrinsic attributes also contribute to *perceived cost*, a combination of monetary and non-monetary attributes [124]. The former usually refers to price, and the latter includes energy, efforts and other costs (e.g., time, shipping). Other extrinsic attributes (e.g., warranty) may have no or a less influence on perceived cost in the presence of price.

4.1.5 Prior Ratings

For a given product in a pre-purchase phase, users go through a process (perhaps subconscious) of evaluating the benefits that they can get and the cost that they have to incur. The outcome of this process helps determine whether users will like the product in question. Other than perceived quality, we posit that prior ratings could also be positively enhanced if the perceived cost is acceptable. Intuitively, for a specific product interested in by a user in terms of quality, if the price of the product turns out to be acceptable, it is likely that the user will like the product as a whole. Recall that due to competitive pressures the price of a product is usually correlated with its quality, i.e., within a normal range [126]. Therefore, we reach the following assumption.

Hypothesis 5 *Perceived quality has significantly positive influence on prior ratings, and perceived cost will also positively influence prior ratings, given that price is within a normal range.*

(a) website

(b) virtual store

Figure 4.2: Website and virtual store modalities

4.2 User Study of Prior Ratings

In the previous section we introduced the concept of prior ratings and provided a conceptual model for them by drawing on various sources in the literature. This section reports a user study to validate the concept of prior ratings.

We developed two user interfaces with different levels of presence, namely *website* and *vir-*

tual store (as seen in Figure 4.2)—corresponding to the mediated environments of WS and VR, respectively. Both user interfaces 'sell' our t-shirts whose source was the real-life commerce website 80stees.com. From this website we derived 50 t-shirts in total as the products which will be evaluated by users in both interfaces. These t-shirts have average posterior ratings (on 80stees.com) in the range $[3.2, 4.9]$ (out of 5) given by the real t-shirt buyers. The virtual store was built using OpenSimulator.org, an open source project for simulating 3D environments. T-shirts were displayed and arranged without a predefined order on the walls of a virtual store. Users can interact with them by viewing, rotating, zooming, and even virtually trying on and customizing the t-shirts (on their avatar). They can also adjust the avatar's shape as desired to meet their personal specifications. In contrast, no interactions were available in WS: users can only imagine what the t-shirt would be like from text descriptions and static images. Six attributes were identified and studied in the experiments: three intrinsic attributes (appearance, material, fit) and three extrinsic attributes (price, category, store). Appearance included colour, image pattern and size; material corresponded to fabric features; fit indicated how t-shirts can perform on avatars; category was the classification of t-shirts used by 80stees.com, such as '80s cartoon t-shirts'; store referred to the design of user interfaces.

4.2.1 Pilot Study

In order to guide the above choices, and to understand whether our experimental settings are reasonable and useful, we conducted a questionnaire-based pilot study. Participants were asked to imagine online shopping for t-shirts and rate what product attributes (see Table 4.1) they were most concerned with. For each attribute, users rated its importance from 1 ("of very little or no importance") to 5 ("of utmost importance"). In August 2012, we recruited 23 volunteers to participate in the online questionnaire, by sending emails of participation to the students and staff on the campus of a technical university. Based upon subjects' ratings, a one sample t-test for each attribute was conducted. To be specific, the *null* hypothesis was: *the mean importance of the attribute in question is moderate* ($mean = 3$), and the alternative hypothesis was set: *the mean importance of the attribute in question is greater than moderate* ($mean > 3$). A small $p < 0.05$ value will reject the null hypothesis and accept the alternative hypothesis.

The results from Table 4.1 show that four major attributes are mostly ($mean > 4$ and $p < 0.001$) concerned with by users when purchasing t-shirts online, namely appearance,

Table 4.1: Results of pilot study: importance of attributes

	Attributes	Mean	p-value
Intrinsic	*appearance*	4.348	*1.29e-07*
	material	4.174	*2.38e-06*
	fit	4.304	*2.041e-08*
Extrinsic	*price*	4.130	*7.62e-09*
	situation	3.044	0.433
	customization	2.522	0.0227
	rating	2.478	0.0152
	brand	2.826	0.769
	store	2.860	0.280
	recommendation	2.826	0.253
	category	2.522	0.0266
	warranty	2.652	0.123
	promotion	2.870	0.301
	shipping	2.957	0.426

material, fit and price. Other attributes are not significant. For example, whether the t-shirts can be customized is not important at all. One subject commented that "If it is a t-shirt I do not care 'experts' recommendation". In addition, subjects also suggested other expected attributes regarding the functionalities of t-shirts: "Matches my other clothes well", "fitness", "have enough text or image details" and have a "Recommender System" or to "have a dummy try-on".

In conclusion, considering that users usually have past experiences about t-shirts in real life, they are confident to evaluate the performance of t-shirts if sufficient information of the four major attributes is available online, especially if it is convenient to visualize or measure the wearing effects. We therefore selected the four significant attributes, together with the standard attributes store and category, for the experiments.

4.2.2 Method and Participants

The user study consisted of one session, structured as follows. All subjects started with a video introduction to the user study, including operations in two different environments. Specifically, for WS, subjects were guided to scan the overall and specific reviews of other customers along

Table 4.2: Demographics of subjects in the user study

Feature	Description
Gender	Male (24), Female (6)
Age	$\leq$20 (1), 20-29 (24), 30-39 (5)
Degree	Doctoral (16), Master (4), Bachelor (9), College (1)
Staff	graphics (2), control system (2), engineering (1), telepresence (1)
Students	Computer Engineering (13), Electrical and Electronic Engineering (11)
Shopping[4.17]	1-2 times/week (3), 1-2/month (8), 1-2 months (16), never (3)
VPEs[4.18]	$<$1 month (4), $<$3 months (5), $<$1 year (4), 1-2 years (2), never (15)

with the t-shirt specifications.[4.2] For VR, subjects were introduced to try the VR hands-on, to become familiar with the functionalities of VR such as navigating, zooming in and out, virtual try-on, etc. Once subjects were comfortable, they proceeded. Each subject experienced and evaluated eight different, randomly-chosen t-shirts in each environment by giving ratings to the questions about product attributes.

Rating values were integers from 1 ("strongly disagree") to 5 ("strongly agree"). Subjects could also add textual comments for each t-shirt. To eliminate the influence of ordering, subjects were randomly determined into two groups. Specifically, of 30 volunteers recruited on a university campus, 16 subjects executed the user study first in WS and then VR, and 14 proceeded inversely. After subjects finished evaluating eight t-shirts in each environment, they were asked to rate the environment regarding the confidence (and state their reasons) and comfort in giving ratings, and the feelings of sense of presence. Finally, subjects could opt to state whether and in which environment they are willing and prefer to provide prior ratings. Subject demographics are reported in Table 4.2, and the questions shown in Figure 4.3.

4.2.3 Results and Analysis

Data cleaning is adopted to rule out noise of user data. In particular, the data from three users were discarded: one subject only completed the user study in virtual store, and two others stated that they were unfamiliar with the functionalities of VR even after an interactive introduction. A further user informed us that his evaluations on the first t-shirt in virtual store were not

[4.2] Users in VR can also read through the product reviews.
[4.17] That is, the frequency of shopping online.
[4.18] That is, the frequency of prior virtual product experiences.

To what extent do you agree or disagree with the following statements?
For each t-shirt:

(1) *The t-shirt is good looking in terms of color, patterns, style, etc.*

(2) *The t-shirt is made of good material.*

(3) *The t-shirt fits you well.*

(4) *The category of this t-shirt is of your favor.*

(5) *The price of this t-shirt is acceptable, including price and shipping fees.*

(6) *The website (virtual store) is well-designed.*

(7) *In total, the quality of this t-shirt is good.*

(8) *You need to spend a lot to obtain this t-shirt in price, time, effort, etc.*

(9) *In total, this t-shirt is worthy purchasing.*

(10) *Overall, you like this t-shirt.*

For each environment:

(1) *You are confident about your ratings. When you gave ratings, you feel confident and no hesitations to make a judgement.*

(2) *It feels the same that inspecting the t-shirt in the environment is just as if you were in a real store and had a real t-shirt in hand.*

(3) *You are comfortable to give ratings in the tested environment.*

(4) *You are (not) confident in your ratings because (state your reasons)*

For willingness (optional):

(1) *Are you willing to rate the t-shirt of your interest or interacted with?*

(2) *If yes, state your reason and indicate how confident in your ratings?*

(3) *If no, state your reason. In what conditions, you will rate the t-shirts?*

Figure 4.3: Questions in the user study

reflecting his real feelings due to misunderstanding of some terms in the first place. Thus his ratings on that t-shirt were also removed. After data cleaning, we had data of 27 users: 15 who tried WS then VR, and 12 in the inverse order. In total we collected 215 rating records from WS and 218 from VR. The statistics (in percentage) of collected ratings is illustrated in Table 4.4, including both prior and posterior ratings.[4.19]

For Hypothesis 1, of 19 subjects who answered our questions regarding the willingness to rate t-shirts, 18 gave positive responses. More specifically, most subjects preferred to rate products in VR (14) rather than in WS (2). Two other subjects did not explicitly state their

[4.19]Posterior ratings were collected and issued by the real website users who had purchased t-shirts in the past.

preference. Most subjects expressed that the reasons were "it can provide more detail information" and "this environment (VR) has really high engagement. I'd like to share my feeling". Only one subject did not want to provide prior ratings ("time consuming") but did indicate the willingness if "benefits or lucky draw" were offered. Thus, Hypothesis 1 is supported.

For Hypothesis 2(a), we conducted a number of paired two sample t-tests to investigate the mean differences of environmental factors, namely confidence, comfort, and presence. Table 4.3 reports the results. Since all $p < 0.01$, we find that users in VR have greater confidence and feel more comfortable in their prior ratings than in WS. The mean confidence in VR (3.778) is larger than that in WS (3.296). This may be partially explained by the fact that users have stronger sense of presence in VR than in WS.

Subjects also expressed their reasons of giving a specific rating to the confidence of tested environment, for instance:

- *I could not dress the T-shirt on my own body to check the looking effect, size and material. The image of T-shirt on model might not be accurate and it's captured only from one side of view.* [in WS]
- *Everyone has a different figure and it is hard to image what it will look like when I wear this shirt.* [in WS]
- *Virtual environment gives a better understanding about how the t shirts looks on you.* [in VR]
- *It seems like that I was just staying in a real store, and then I can see the outfit directly. So I can make the judgement confidently.* [in VR]

These comments suggest that users in VR possess more confidence in their ratings because they can try t-shirts on their 'own' body rather than have to imagine the real wearing effect in WS. They also feel a stronger sense of presence in VR as if being in a real store. Thus, Hypothesis 2(a) is supported.

For Hypothesis 2(b), as stated in Table 4.4, the final collected data consists of 215 prior ratings in WS ($R.ws$) and 218 records in VR ($R.vr$). Since all users only rated a handful of t-shirts, they are all regarded as cold-start users. The correlation between posterior ratings (R_p) and $R.ws$, denoted as $corr(R_p, R.ws)$ is -0.42 whereas $corr(R_p, R.vr) = 0.23$, signifying

Table 4.3: Evaluations of the environmental factors

	Mean.ws	Mean.vr	Diff.	p-value
confidence	3.296	3.778	0.482	*3.300e-3*
comfort	3.444	3.963	0.519	*6.653e-3*
presence	2.185	3.222	1.037	*1.420e-4*

Table 4.4: The distributions of collected ratings

Scales	R_p	$R.ws$	$R.vr$
1	3.82%	11.63%	3.67%
2	4.08%	18.60%	10.55%
3	7.15%	35.81%	27.52%
4	27.77%	25.12%	42.66%
5	57.18%	8.84%	15.60%
1, 2, 3	15.05%	66.04%	41.74%
4, 5	84.95%	33.96%	58.26%
Total	1469	215	218

that the distribution of posterior ratings is distinct from prior ratings in WS, but marginally yet positively similar to prior ratings in VR. To have a better viewpoint, we classify the two rating values (i.e., 4, 5) which are larger than median scale value (i.e., 3) as positive, and the remainder (i.e., 1, 2, 3) as negative. Then we obtain clearer correlations: $corr(R_p, R.ws) = -1$ and $corr(R_p, R.vr) = 1$. In addition, the average posterior rating is 4.13 whereas the values for $R.ws$ and $R.vr$ are 2.94 and 3.56, respectively. In conclusion, prior ratings in VR are much closer to posterior ratings than those in WS. Thus, Hypothesis 2(b) is supported.

For Hypothesis 3, we conducted multiple linear regressions, each of which used 'presence' as independent variable and one of intrinsic or extrinsic attributes in WS and VR as dependent variable. The results are illustrated in Table 4.5. We see that presence in WS is most influential ($p < 0.01$) on material and store; in VR it is influential ($p < 0.001$) on all attributes except category. Hence, presence in WS has smaller effects on the perceptions of product attributes than that in VR. This can be attributed to the lower level of presence in WS as shown in Table 4.3. However, for attributes whose information can be adequately communicated by basic media (i.e., text descriptions, static images), such as category, presence may be of limited influence. One possible explanation for the different effects of price in two environments is

Table 4.5: The influences of presence on attributes

Environment	Attributes	Estimate	t Value	Pr($>\|t\|$)
WS	appearance	0.142	2.131	0.0342
	material	0.270	3.822	*1.740e-4*
	fit	0.187	2.452	0.0150
	category	0.130	1.880	0.0614
	price	0.0921	1.294	0.197
	store	0.269	3.216	*1.500e-3*
VR	*appearance*	0.0860	1.259	*<2e-16*
	material	0.244	3.388	*8.370e-4*
	fit	0.216	3.349	*9.580e-4*
	category	0.0698	1.092	0.276
	price	0.209	3.295	*1.150e-3*
	store	0.468	7.623	*7.740e-13*

Table 4.6: The evaluations of perceived quality

Environment	Attributes	Estimate	t Value	Pr($>\|t\|$)
WS	appearance	-0.0665	-1.152	0.250
	material	0.283	5.729	*3.52e-08*
	fit	0.125	2.130	0.0343
	category	0.311	5.115	*7.11e-07*
	price	0.0462	0.975	0.331
	store	0.212	3.748	*0.000231*
VR	*appearance*	0.1958	3.217	*0.00150*
	material	0.1413	2.941	*0.00363*
	fit	0.2467	4.748	*3.79e-06*
	category	0.1081	2.044	0.04222
	price	0.1999	4.795	*3.07e-06*
	store	-0.0059	-0.126	0.89976

that price in WS may be ignored as a cue to infer user preference as we will explain later for Hypothesis 5. Thus, Hypothesis 3 is partially supported.

For Questions 1 and 2 and Hypothesis 4, we conducted a multi-variable linear regression with intrinsic and extrinsic attributes as independent variables and 'perceived quality' as dependent variable. The results, presented in Table 4.6, show that three attributes in WS are the major concerns for product quality, namely material, category and store. In addition, attribute

Table 4.7: The evaluations of prior ratings

Environment	Attributes	Estimate	t Value	Pr($>\|t\|$)
WS	*quality*	0.619	8.129	*3.58e-14*
	cost	0.0613	0.790	0.430
VR	*quality*	0.670	10.521	*< 2e-16*
	cost	0.141	2.206	*0.028*

'fit' is also considered important but has smaller influence. A number of subjects commented that "It is difficult to judge the t-shirt", "I don't think t-shirt would fit me well. It doesn't even fit the model well." and "cannot see design". Note that the regression coefficients of category and store are greater than material and fit, which means that perceived quality relies more on extrinsic attributes than on intrinsic attributes. In contrast, the most important attributes in VR are appearance, material, fit and price. Most comments were focused on these four attributes, for example, "Nice and simple yet beautiful color", "I am a fan of ninja turtle, but it is not 100% cotton [sic].", "This shirt looks cute on the girl" and "The price is very very cheap. good quality price ratio". Hence, subjects relied more on intrinsic attributes than extrinsic attributes to evaluate the quality of t-shirts in VR.

Of the four major attributes identified from pilot study, we find that only one of them (material) is revealed in WS whereas all four attributes are correctly recognized in VR. One possible explanation is that when users have less or no direct experiences with products, they may tend to use extrinsic attributes (i.e., category, store, price) as cues to infer the product quality. On the other hand, if users have effective and direct interactions with products and thereby gain sufficient direct product experiences, they may rely more on intrinsic attributes to evaluate products. Thus, Hypothesis 4 is supported.

For Hypothesis 5, we investigated the correlations among prior ratings, perceived quality and cost in WS and VR by applying a linear regression analysis. The results are shown in Table 4.7. It is observed that the coefficient of perceived quality is positive and large (> 0.6), and that it has a significant influence ($p < 0.001$) on prior ratings. In addition, the cost is demonstrated as relatively small yet positively important in VR ($0.14, p < 0.05$) rather than in WS ($0.06, p > 0.1$). The price of collected t-shirts ranges from US$3.99 to $32.00, and so the price is within a normal range in general in the context of the study. Besides, the mean, median

and mode of product price ratings are 3.25, 3, and 3, respectively (where 3 (out of 5) here means "slightly disagree or slightly agree" that the price is within a normal range). The results indicate that most subjects do not think the price is unacceptable. In other words, price tends to be indifferent when evaluating the preferences, and works as a confirmation that the perceived quality is matchable with the price and thus its quality is trustworthy. As a consequence, users are more likely to like products as a whole given good quality estimated. However, as users may not correctly judge the quality of products in WS, the price may fail to be or less considered when assessing their preferences. In addition, if the price seems exceptional—too high or too low—users may suspect that the real quality that they will received cannot be competitive with the quality that the product is claimed. In this case, the perceived cost may have negative influence on the prior ratings. Thus, Hypothesis 5 is supported.

4.3 Discussion

In this section, we discuss the motivation for users to give prior ratings, the usage of prior ratings, the similarities and differences between prior ratings and other information sources, the limitations to our current experiments, and the implication of our work in real applications.

4.3.1 Motivation to Provide Prior Ratings

As a new information source intended for recommender systems, it is important for users to be properly motivated to provide their feelings, understandings and evaluation towards the experience of virtual products, i.e., prior ratings. In this work, we consider a number of possible ways to motivate users to provide prior ratings.

First, it is easy, comfortable and convenient to rate virtual products. For posterior ratings, users have to wait for product delivery, try out products (offline) and then go back to rate them (online), if they decide to rate. The time and effort in such a procedure can cause users to forget or not wish to rate products after use. In addition, users may not have a fresh memory regarding their offline experience and thus bring in noise in their posterior ratings [132]. In contrast, the presence of virtual reality and responsive virtual products provide and enable users life-like shopping experience. More importantly, users can immediately go through the virtual products of interest, and preserve a fresh memory leading to the prior ratings closer to their real feelings. Hence, users may feel it low effort, comfortable, and convenient to give prior ratings.

Second, new and novel incentive mechanisms [133] can be developed to pro-actively encourage users to rate products. Virtual product experiences provides an opportunity for retailers to design novel ways to motivate users to try out more products and speak out their experiences, i.e., providing prior ratings. How to design incentive mechanisms is beyond the scope of the present work. Nevertheless, the media-rich new environments do have the potential to boost new opportunities for incentive mechanisms.

Third, an additional possibility is not to request users to rate explicitly, but to implicitly transform users' experience into prior ratings automatically. In this way, users can focus more on their virtual products and do not worry about giving ratings. As an initial attempt, we consider how to capture users' emotional signals during their experience with virtual products, and then convert these signals (in the form of electroencephalography EEG) into ratings prior to purchase [134]. Arapakis et al. [135] use web cameras mounted on top of computer screens to capture users' real-time facial expressions during watching videos, and show that better recommendation performance can be achieved by integrating with other implicit feedback. Literature works such as Yeasin et al. [136] have shown that facial expressions can be automatically transformed into numerical level of interest, i.e., user preference. Such techniques could be adapted in virtual reality to derive prior ratings.

Lastly, our evaluation in Section 4.4.2 will show that even with a small amount of prior ratings, the performance of recommendations can still be improved to some extent. In other words, we do not expect users to provide prior ratings of every product that they experience; a small amount of ratings is sufficient.

4.3.2 Usage of Prior Ratings

Since prior ratings are issued by users who have experienced the products of interest in virtual reality, the ratings are not used to recommend these products back to those who have already formed their evaluation and opinions (virtually or physically). Rather, prior ratings are used to help model user preference for a recommender system to recommend users with items that they are not aware of. Specifically, prior ratings can be used in at least two cases: (1) the prior ratings of an active user can be used to help identify other users who share similar preferences, and hence the system can recommend to the active user products that she has not experienced or purchased; and (2) the prior ratings of similar users can be used to form a proper prediction

on the unknown items for an active user. Further, in Section 4.4 we will propose a specific collaborative filtering technique that leverages prior ratings to improve the performance of recommendations, as a feasible solution to demonstrate the use of prior ratings in a real case.

4.3.3 Prior Ratings vs. Posterior Ratings

The most commonly used information source in recommender systems is posterior ratings given by users after they purchased and fully experienced the products of interest. As stressed earlier, the most important distinction between prior ratings and posterior ratings is that prior ratings are based on partial, limited (yet reliable) product experience whereas posterior ratings are based on full, complete product experience. These different kinds of product experience are generally provided under different settings, that is, prior ratings are mostly given in mediated environments while posterior ratings are likely issued in real and physical environments.

The values of prior and posterior ratings will be expected to differ (but have some correlation, see Hypothesis 2), because they are made by users based on different information. In general, both types of ratings indicate user preferences towards a certain product based on some form of user experience, but they differ in the confidence of user ratings. Specifically, prior ratings reflect users' virtual product experience according to their interactions with virtual products that are represented in a mediated environment. We find that environments with higher sense of presence motivate users to rate products[4.20] (see Hypothesis 1), and lead to more confident prior ratings and rating values closer to posterior ratings (see Hypothesis 2). In contrast, posterior ratings reflect more tangible forms of user experience based on 'physical' interactions with real products in real world; but the issue is that users often lack incentive to provide their ratings—one cause of the data sparsity problem.

Users' confidence of prior ratings may be lower than that of posterior ratings due to the less tangible form of product experience. We ascribe the lower confidence to the partial knowledge that can be conveyed by virtual products in a mediated environment. Nevertheless, we do require that virtual product experience should be reliable enough such that these ratings can be regarded as prior ratings.[4.21] Therefore, the confidence will not be too low to be meaningless.

[4.20] That is, we mean that users are more willing to give prior ratings in VR than in WS, rather than that users are more likely to give prior ratings than posterior ratings.

[4.21] For some domains, it may not be easy to simulate even reliable virtual product experience.

We use users' stated rating confidence, if any, to distinguish prior ratings from posterior ratings. In Section 4.4 we evaluate the impact of confidence on recommendation performance.

As pointed out by Nguyen et al. [132], posterior ratings could also be noisy *per se* as users do not have fresh memory when they get back to rate the products after the experience. Prior ratings, instead, can be given in a shorter time after the experience with virtual products, and easier to 'get back' to rate them. In other words, prior ratings could alleviate the motivation issue of posterior ratings to some extent.

Lastly, it is commonly understood that users usually browse or experience more products than they actually purchase, especially in the virtual environments where product information is easy to reach. In this regard, we suggest that prior ratings can complement traditional posterior ratings and help inherently alleviate the data sparsity and cold start problems.

4.3.4 Prior Ratings vs. Social Information Sources

We discuss two kinds of social relationships of users, namely friendship and social trust which are widely adopted in social recommender systems. It has been demonstrated that social networks provide an additional and useful source of information to improve the quality of recommendations [137].

Specifically, friendship is readily available from social networks and has been demonstrated to be helpful for recommender systems [104]. However, it has also been reported that online friendship sometimes cannot work well for recommenders due to its inherent ambiguity as a relational descriptor [138]. It becomes easy, simple and low cost to connect with other users in social networks, and it is not surprising to find that a number of strangers appear within a user's circle of friends. An even weaker connections could be only one-sided, for example, users in Twitter can easily 'follow' other users whereas the others are free not to link back. In contrast, trust relationships are much stronger than friendships as the former are often built upon positive evaluation towards the others in conducting some expected actions [139], e.g., providing reliable ratings. It has been shown that trust-based recommender systems are able to provide better performance [48]. However, a critical problem of trust information is its sparsity and the difficulty of building it. Existing publicly-available datasets that include trust information show that only a small portion of users has specified others as trustworthy [85]. Only a few online systems (e.g., `epinions.com`, `ciao.co.uk`) support the concept of trust, whereas

most other systems do not build user connections based on trust evaluation, or even do not have an inherent social network. In conclusion, friendships are more common and easy to collect, but more ambiguous towards user preferences; trust is much stronger than friendships but itself suffers from the data sparsity problem. In addition, both information sources require a social network structure inherently supported by a recommender system. Besides, these kinds of information are usually represented in the form of 'who connects whom' without a numerical value indicating the strength of social ties, whereas not all the friends/trusted users should be equally weighted. This problem may further limit the usefulness in recommender systems. Moreover, although social relationships-based recommender systems can help mitigate the cold start problem, recent work shows that the performance of cold-start users is still much worse than that of normal users [48]. Therefore, it is necessary to identify other possible information sources that are more reliable and less constrained, and can be effectively used to model user preferences for recommender systems.

Prior ratings are just such a kind of information sources that has the potential to overcome the drawbacks of existing information sources, and help reveal and model user preference to improve the performance of recommendations. First, prior ratings are issued by users based on their evaluation of virtual products of interest prior to purchase, which are similar to the posterior ratings with respect to real products. Prior ratings directly indicate user's likeness towards the products that they experienced. As a comparison, it makes sense that even friends or trusted users may have different preferences, especially when these relationships are built upon offline relations (e.g., classmates, colleagues, families). Second, prior ratings have no requirement and constraint of a social network that should be supported by a social recommender system. Third, prior ratings may less suffer from the data sparsity problem than social trust, considering the following viewpoints: (1) since users usually experience more products than they purchase (and could even more in the case of e-commerce environments), it has a potential to attract more prior ratings than posterior ratings which are richer than trust; and (2) as elaborated in Section 4.3.1, a number of ways can be used to motivate users to provide prior ratings. As a result of more available information, the data sparsity and cold start problems could be better resolved than other kinds of information sources. Lastly, prior ratings can co-exist with social information sources. Prior ratings refer to only individuals' personal preferences, and thus they do not prevent users from building connections with other users in virtual reality. In

fact, it is possible to infer implicit trust from prior ratings in the same way as from posterior ratings, to resolve the data sparsity problem of explicit trust [85, 86]. Ultimately it would be more powerful to make use of all possibly available information sources (e.g., prior ratings, posterior ratings, social relationships) in order to better resolve the data sparsity and cold start problems. Such future work is beyond the scope of this work.

4.3.5 Prior Ratings: An Alternative Information Source

To summarize the discussions in the last two subsections, we propose prior ratings as a good alternative information source to social relationships, and as a complementary to posterior ratings for modelling user preferences. The speciality of prior ratings lies in the differences from other information sources: they provide a unique source of information that is similar to (yet less confident than) posterior ratings, and is more reliable than social relationships as real users' preferences on products. Prior ratings are easy to issue, require no purchase payment, and have the potential to be denser than posterior ratings due to the fact that users often experience more products than they purchase. They are also distinguished in formulating the product experience prior to purchase in the form of usable information source. This kind of product experience exists for a long time in e-commerce, but has not been studied and investigated for recommender systems till now.

4.3.6 Limitations of Current Experiments

There are several potential limitations in our current experiments. First, certain attribute information (e.g., warranty, shipping) was not available for our user study. Although these are less relevant for the product type studied, they may be more important for other kinds of products. Second, due to a lack of devices, our prototype implementation uses only visual information in VR: users cannot touch the t-shirts and feel the material. Tactile feedback may be important for user evaluation of preferences. Nevertheless, as analyzed in Section 4.2.1, this limitation may not greatly influence the general conclusion since we exploited abstract attributes rather than some specific attributes. Further, as shown in Section 4.4, prior ratings can improve recommendation performance even if based solely on visual information, reducing the requirements of expensive VR devices. In this regard, t-shirts represent a kind of products with simple representations in VR. Third, most subjects in our study were computer or electrical engineering

students on a university campus, and the sample size was modest. A larger and more heterogeneous sample may allow for more confident generalization of our research findings.

4.3.7 Implications for Real Systems

Our work has practical implications for real systems. First, as stated by Hypotheses 1 and 2, users usually are more willing and feel more comfortable and confident in sharing their opinions in VR than in WS. This indicates that for the product sellers, it would be value to market products through e-commerce systems in VR, which can provide better online product experiences than traditional websites. Besides, as we will demonstrate in Section 4.4, prior ratings can benefit recommender systems by solving the data sparsity and cold start problems. In other words, the VR e-commerce systems can help expose more products to users, and recommend them more accurate products of interest.

Second, as pointed out by Hypothesis 3, a greater sense of presence can enhance users' perceptions of products, and thus form better opinions regarding the product qualities. Therefore, for the designers of VR, it is necessary to enhance the environmental presence by enabling richer types of interactions and media to better convey product information.

Third, our foundational work in introducing the concept of prior ratings opens up future discussions around customers' pre- and post-purchase product evaluations and their purchase intent decisions.

4.4 Leveraging Prior Ratings

Having defined prior ratings and observed their value, we now leverage prior ratings in product recommendations. Since no existing commonly used data sets contain information of prior ratings, we rely on the prior ratings collected from the user studies for quantified evaluation.

To facilitate discussion, we first introduce a number of notations. Let the sets of all users, all items, all ratings and all rating confidences be $\mathbb{U}$, $\mathbb{I}$, $\mathbb{R}$ and $\mathbb{C}$, respectively. We keep the symbols $u, v \in \mathbb{U}$ for users and $i, j \in \mathbb{I}$ for items. Let $r_{u,i} \in \mathbb{R}, c_{u,i} \in \mathbb{C}$ represent the rating and confidence given by user u on item i, respectively. The confidence $c_{u,i}$ indicates the degree to which users are 'certain' about their evaluation $r_{u,i}$ on item i. Since the collected data does not include the confidence of posterior ratings, we set it as 1.0 to ensure that users are more

certain in their posterior ratings. Then the task of a recommender can be modelled as: given a set of user-item-rating-confidence $(u, i, r_{u,i}, c_{u,i})$ quaternions, provide a prediction $(u, j, ?, ?)$ for user u on an unknown item j. The prediction pair is denoted as $(\hat{r}_{u,j}, \hat{c}_{u,j})$.

4.4.1 Prior Ratings-based CF (PRCF)

As a new information source, prior ratings have not been proposed and studied in the literature, and thus no recommendation algorithms have been developed based on both prior and posterior ratings so far. We integrate prior ratings with the conventional collaborative filtering (CF), based on a *confidence-aware* distance similarity. Note that we aim to evaluate the usefulness of prior ratings by providing a feasible solution to make use of prior ratings rather than to provide a perfect algorithm resulting in the best performance. The algorithms based on other popular techniques such as matrix factorization may be proposed and work better than our approach, but that is beyond the discussion of this work. We leave the exploration for a better recommendation algorithm based on prior ratings as a line of future research.

The first step of CF is to identify the like-minded users who have similar preferences with the active user. Since all the users only rated a few items, traditional similarity measures such as Pearson correlation coefficient and cosine similarity often fail to work effectively in this condition as we described in Section 3. More importantly, existing similarity measures cannot accommodate the new information source in terms of prior ratings and their confidences. In this work, we therefore propose a confidence-aware distance similarity by taking into consideration three factors: the distance of ratings, the distance of rating confidences and the semantics of rating values. Intuitively, the greater the rating distance is, the lower the similarity will be. This intuition also holds for the distance of rating confidences. Further, if two ratings have the same positive or negative opinions towards the same item, they are regarded as semantically indifferent. We regard a rating as positive if its value is greater than the median rating scale; otherwise it is negative. In addition, ratings generally have more influence on similarity than rating confidences. Hence, we compute user similarity as follows:

$$s_{u,v} = 1 - \frac{1}{3N} \sum_{i \in I_{u,v}} \Big(\frac{|r_{u,i} - r_{v,i}|}{R_{max} - R_{min}} + \frac{|c_{u,i} - c_{v,i}|}{|c_{u,i} - c_{v,i}| + 1} + sw_i \Big), \tag{4.1}$$

where $s_{u,v} \in [0, 1]$ is the similarity between users u and v based on their ratings $(r_{u,i}, r_{v,i})$ on commonly rated items $I_{u,v}$ with cardinal N, and R_{max} and R_{min} are respectively the maximum

and minimum rating values defined by a recommender system. The semantic weight sw_i is defined by:

$$sw_i = \begin{cases} \dfrac{1}{d_i + 1} & \text{if } d_i \geq 0; \\ \dfrac{|d_i|}{|d_i| + 1} & \text{otherwise,} \end{cases} \tag{4.2}$$

where $d_i = (r_{u,i} - R_{med})(r_{v,i} - R_{med})$ denotes the extent to which two users have the same opinions towards item i relative to the median value R_{med}. The settings of sw_i capture the intuition that if two users have closer opinions, the computed similarity $s_{u,v}$ will be greater and vice versa. The semantic weight is used to distinguish the case where two pairs of ratings have the same rating distance but possess distinct semantic meaning essentially. For example, assume that there are two pairs of ratings (5, 4) and (4, 3) from two users and that the rating values are integers from 1 to 5 predefined by a certain system. Although the rating distance is the same (1), the semantic meaning is different. Specifically, both ratings 5 and 4 are greater than the median rating scale (3) and hence positive whereas the ratings 4 and 3 are different opinions in real life. In other words, the rating distance 1 in the first pair reflects the difference in liking whereas the same distance in the second pair reflects the differences between liking and disliking.

The second step of CF is to select a set of similar users in order to predict the rating and confidence of an unknown item for an active user. Specifically, we adopt the users who have rated item j and whose similarity is greater than a predefined threshold, i.e., $U_{u,j} = \{v \mid s_{u,v} > \theta, \exists r_{v,j}, v \in U\}$, where θ is a similarity threshold. In this work, all users with positive correlations will be adopted, i.e., $\theta = 0$.

The third step of CF is to generate the predictions by using either simple weighted average (WA) or Resnick's formula (RF) [5]:

$$\hat{p}_{u,j} = \frac{\sum_{v \in U_{u,j}} s_{u,v} p_{v,j}}{\sum_{v \in U_{u,j}} |s_{u,v}|} \tag{WA}$$

$$\hat{p}_{u,j} = \bar{p}_u + \frac{\sum_{v \in U_{u,j}} s_{u,v}(p_{v,j} - \bar{p}_v)}{\sum_{v \in U_{u,j}} |s_{u,v}|} \tag{RF}$$

where $p_{u,j}$ corresponds to $r_{u,j}$ (or $c_{u,j}$), and $\hat{p}_{u,j}$ to $\hat{r}_{u,j}$ (or $\hat{c}_{u,j}$), respectively, and $\bar{p}_u$ represents the average rating or confidence reported by user u. WA and RF may produce different rating predictions and hence both are used to predict item ratings. Since the confidence of posterior

ratings is unavailable, we set it to 1.0 by default; thus the confidence difference in RF will be always 0 if only posterior ratings are available. Hence, we adopt WA only to predict rating confidences.

PRCF variants. One potential drawback of the PRCF approach is the reliance on rating confidences. In practice, users may not be motivated to provide the rating confidences, which could become for them an additional cognitive burden. Hence, we adapt PRCF to two scenarios. First, when no confidence data is available we set 0.5 as the default confidence for all prior ratings (considering that we set 1.0 as the default confidence for all posterior ratings). This algorithm variant is denoted as *PRCF-1*. Second, when only limited confidence data is available, i.e., only few users reported confidence while others did not, we use the average of all available confidence data as the default confidence for all prior ratings. This algorithm variant is denoted as *PRCF-2*.

Discussion. This section presented a new user-based collaborative filtering technique (i.e., PRCF) that we developed by exploiting the prior ratings. The new method is not trivial considering the following aspects. First, we proposed a new similarity measure based on both the values and confidence of prior ratings (see Equation 4.1), due to the inability of traditional similarity measures to accommodate the prior ratings. As we stressed earlier in Section 3, similarity measures are important in that they play two critical roles in collaborative filtering: (1) identifying a number of similar users; and (2) weighting the ratings of similar users to generate predictions. Second, our approach uses a new information source that is not supported by traditional approaches. Third, a new predictive metric will be proposed in next subsection (see Equation 4.5) to account for the rating confidence. Therefore, we regard PRCF as a new user-based collaborative filtering method rather than applying others' existing work.

4.4.2 Results and Analysis

Using the collected real data, we conduct a number of experiments to investigate the effectiveness of prior ratings in predicting item ratings. We further verify the usability of our collected modest data in revealing performance patterns. The leave-one-out approach is applied to the

rating data collected from user study (see Table 4.4). In particular, each rating is hidden iteratively and then predicted by adopting the proposed PRCF approach or its variants.

Predictive performance is usually estimated in terms of mean absolute errors (MAE) between predictions ($\hat{r}_{u,j}$) and the ground truth, and the rating coverage (RC) of predictable ratings over all testing ratings. In particular, they are computed as follows:

$$\text{MAE} = \frac{\sum_u \sum_j |\hat{r}_{u,j} - r_{u,j}|}{M}, \qquad \text{RC} = \frac{P}{M} \times 100\%, \tag{4.4}$$

where M is the total number of testing ratings, and P is the number of predicable ratings. Since MAE does not consider the influence of rating confidences ($\hat{c}_{u,j}$), we propose a confidence-aware metric, termed mean absolute confidence errors (MACE):

$$\text{MACE} = \frac{\sum_u \sum_j \hat{c}_{u,j} |\hat{r}_{u,j} - r_{u,j}|}{\sum_u \sum_j \hat{c}_{u,j}}. \tag{4.5}$$

Note that if all rating confidences such as of posterior ratings are the same, MACE will be the same as MAE. Generally, smaller MAE and MACE mean better predictive accuracy and higher value of RC indicates better coverage.

Performance of PRCF. We aim to investigate how posterior ratings can be better predicted by involving prior ratings. We denote R_p as the set of posterior ratings, R_{pw} the union set of R_p and $R.ws$, and R_{pv} the union set of R_p and $R.vr$. Two subsets will be used as testing data: *All* is the subset including all the posterior ratings, and *Pred* is the subset only including the posterior ratings that can be predicted by PRCF when no prior ratings are used. The performance of PRCF is reported in Table 4.8.

The results show that if only posterior ratings (R_p) are used, a small ratio (4.56%) of testing ratings is predicted. The rating coverage can be greatly increased by involving prior ratings (R_{pw}, R_{pv}) since more ratings are thus available. The difference in coverage between R_{pw} and R_{pv} is due to the fact that subjects in the user study randomly chose and rated different t-shirts in different environments. Another observation is that the accuracy based on WA and RF may be different in terms of MAE and MACE. MACE generates relatively smaller values than MAE as it considers rating confidences. For *All*, R_{pw} and R_{pv} may produce worse (WA) or better (RF) results than R_p, and R_{pw} has larger variations between WA and RF than R_{pv}. This may indicate that prior ratings in WS produce less reliable predictions than those in VR. Considering

Table 4.8: The predictive performance of PRCF

		All			Pred	
		MAE	MACE	RC	MAE	MACE
WA	R_p	0.916	0.916	4.56%	0.916	0.916
	R_{pw}	1.394	1.346	9.26%	1.070	1.048
	R_{pv}	1.044	1.009	7.49%	0.844	0.829
RF	R_p	0.957	0.957	4.56%	0.957	0.957
	R_{pw}	0.798	0.815	9.26%	0.926	0.910
	R_{pv}	0.929	0.919	7.49%	0.880	0.858

that the newly predicted item ratings also contribute to predictive errors, the performance on *Pred* is more comparable among different approaches and better demonstrates the effectiveness of prior ratings. Although the performance of R_{pw} may increase (RF) or decrease (WA), R_{pv} consistently and significantly obtains better accuracy with lower MAE and MACE. Specifically in *Pred*, 7.86% improvements in MAE and 9.50% in MACE can be achieved using WA while 8.0% (MAE) and 10.34% (MACE) are obtained using RF. In conclusion, prior ratings in VR can significantly improve both the coverage and accuracy whereas those in WS only show consistent improvements in the coverage rather than accuracy.

Performance of PRCF variants. We further study the effectiveness of prior ratings when rating confidences are missing or incomplete. The performance of PRCF variants is shown in Tables 4.9 and 4.10. Comparing with PRCF, PRCF-1 obtains similar but superior results in Table 4.9: prior ratings in VR can consistently improve the predictive accuracy, and the improvement is more significant especially in terms of MACE. Specifically for R_{pv} in *Pred*, 7.97% improvements in MAE and 11.35% in MACE can be achieved using WA while 8.46% (MAE) and 13.48% (MACE) are obtained using RF relative to the performance of R_p. One possible explanation is that the rating confidences reported by users are unreliable and noisy due to the fact that only visual information is available to evaluate product quality and performance. This indicates that a richer environment with higher sense of presence can help improve the quality of user evaluation, and hence enhance the utility of prior ratings in predicting the ratings of unknown items.

For PRCF-2, the default confidences adopted are 0.659 and 0.756, corresponding to the average confidences in WS and in VR after normalization, respectively. Analogously, the

Table 4.9: The predictive performance of PRCF-1

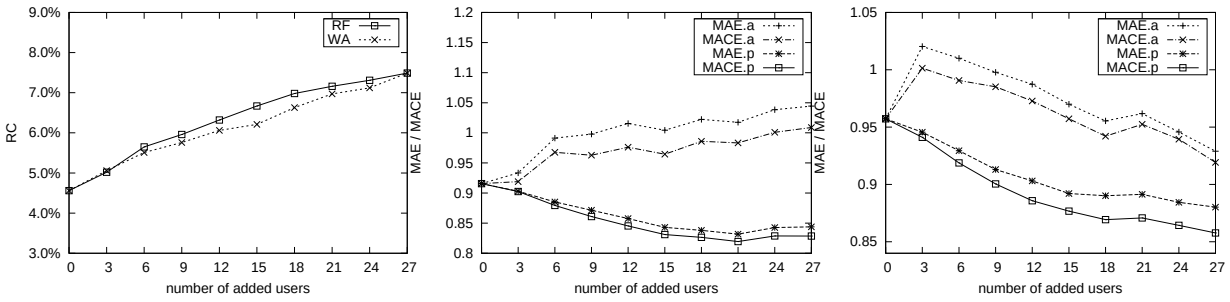
		All			Pred	
		MAE	MACE	RC	MAE	MACE
WA	R_p	0.916	0.916	4.56%	0.916	0.916
	R_{pw}	1.407	1.311	9.26%	1.095	1.057
	R_{pv}	1.046	0.964	7.49%	0.843	0.812
RF	R_p	0.957	0.957	4.56%	0.957	0.957
	R_{pw}	0.798	0.810	9.26%	0.925	0.896
	R_{pv}	0.926	0.877	7.49%	0.876	0.828

Table 4.10: The predictive performance of PRCF-2

		All			Pred	
		MAE	MACE	RC	MAE	MACE
WA	R_p	0.916	0.916	4.56%	0.916	0.916
	R_{pw}	1.407	1.351	9.26%	1.095	1.071
	R_{pv}	1.046	1.013	7.49%	0.843	0.830
RF	R_p	0.957	0.957	4.56%	0.957	0.957
	R_{pw}	0.798	0.805	9.26%	0.925	0.907
	R_{pv}	0.926	0.906	7.49%	0.876	0.855

results presented in Table 4.10 are similar to those in Table 4.8: R_{pw} works worse in WA but better in RF whereas R_{pv} achieves much better accuracy in both cases. We note that MAE in Tables 4.9 and 4.10 is the same. This may be due to two reasons. First, confidence in Equation 4.1 has smaller influence than ratings. Second, MAE does not take into account rating confidence. However, in terms of MACE, PRCF-1 performs better. Hence, MACE is important in evaluating overall predictive performance.

Since PRCF performs closely with PRCF-2, which in turn is inferior to PRCF-1, we conclude that prior ratings can effectively improve the recommendation performance when the rating confidences are robust and correctly based on sufficient product information. In case that the confidence data is missing or incomplete, prior ratings with a default smaller rating confidence offer a performance improvement. We can also recommend higher-fidelity mediated environments, such as VR to WS, because the former achieves consistent improvements in both coverage and accuracy. Although prior ratings in WS may also have a certain positive



(a) Rating coverage (b) Accuracy in WA (c) Accuracy in RF

Figure 4.4: The predictive accuracy and coverage with incremental prior ratings

influence on recommendations, its performance varies by the different prediction approaches. Lack of high confidences in prior ratings could be one of the critical reasons. The rationale is that prior ratings with low confidences *per se* are noisy and hence less useful.

Performance of incremental prior ratings. Finally, we examine whether the modest sample size in our experiments has validity. We randomly and incrementally (step $k = 3$) incorporate the prior ratings of new users. We apply PRCF to generate predictions, and repeat the whole process five times in order to generate different sequences of prior ratings. The mean results are shown in Figure 4.4. Note that when $k = 0$, only posterior ratings are used.

Figure 4.4(a) shows that, as more prior ratings are available, more ratings of unknown items can be predicted. The trends of rating coverage are similar in WA and RF. Figures 4.4(b, c) show the changes of predictive accuracy when the number of prior ratings is adjusted in WA and RF, respectively. For prediction set *All*, the predictive errors (MAE.a, MACE.a) in WA increase whereas in RF they decrease. As explained earlier, newly-predicted ratings may contribute to the overall errors (with view *All*), and hence the better view of performance is based on the results of prediction set *Pred*. The predictive errors (MAE.p, MACE.p) gradually decrease as the amount of prior ratings increase. This indicates that even a small number of prior ratings can improve recommendation performance.

4.5 Concluding Remarks

Towards the design of recommender systems for e-commerce in virtual reality (VR), this chapter proposed a new information source, called *prior ratings*. By leveraging the effective interac-

tions between users and virtual products represented in a mediated environment, prior ratings capture users' opinions about products that result from virtual product experiences, usually prior to purchase. We presented a conceptual model of prior ratings that provided their principled foundation. We conducted a user study in two different environments, namely website and virtual store, in which users virtually interacted with t-shirt products. Particularly, unlike in the traditional environments (WS), users felt more comfortable and were motivated to rate products in VR, and provided more confident prior ratings that were closer to posterior ratings due to the higher sense of presence. We found that the presence had positive influence on the perceptions of some experience-related product attributes, both intrinsic and extrinsic. To estimate product quality, users relied more on extrinsic attributes in WS while users relied more on intrinsic attributes in VR since direct experiences can be obtained. Besides, both perceived quality and cost positively influence prior ratings. We stress that prior ratings can complement posterior ratings and help ameliorate the data sparsity and cold start problems due to the fact that users usually browse or experience more product than they actually purchase. In conclusion, the results validated the conceptual model of prior ratings under our experimental settings. In addition, since higher presence may result in more confident prior ratings, it follows that the design of virtual stores should emphasize the sense of presence by increasing the media richness or the effectiveness of user interactions.

We furthered our contribution by demonstrating how to leverage prior ratings in predicting items' ratings using a collaborative filtering technique. Specifically, we introduced a new similarity measure by taking into account three important factors: the distances between ratings, the distances between confidences and the semantics of rating values, each of which captures the different considerations of user similarity. Using this measure, we proposed a confidence-aware performance measurement. Using the data collected from the user study, we conducted a number of experiments to evaluate the usefulness of prior ratings in improving the recommendation performance. The results indicated that prior ratings were effective and valuable, and held potential as a new information source to bootstrap recommender systems. We also showed that even a small number of prior ratings may benefit recommender systems.

Chapter 5

Merge: A Memory-based Approach

From this chapter onwards, we concentrate on the second research line to resolve the two severe problems, i.e., data sparsity and cold start of recommender systems. Specifically, we incorporate additional user information, in our case, of social trust to further improve the performance of recommendations. As we explained in Section 1.3, trust is more reliable than other kinds of social relationships, such as friendship and membership, due to the positive and strong correlation with user similarity. Two trust-based approaches will be presented in this direction, where one is a memory-based approach (this chapter) and the other a model-based approach (next chapter). The Netflix competition has shown that both kinds of recommendation methods are beneficial and necessary for the further development of recommender systems [73].

In this chapter, we propose a novel memory-based approach called "Merge" [20, 45] by merging the ratings of trusted neighbours to form a new and more complete rating profile for an active user. Specifically, the ratings of trusted neighbours on the commonly rated items are averaged according to the importance of trusted neighbours. The importance is determined by three factors, namely user similarity, trust value and social similarity. We further measure the quality of a merged rating by the confidence considering both the number of ratings involved and the ratio of conflicts between positive and negative opinions (ratings). Then, we adapt a conventional CF approach to generate recommendations based on the newly-formed rating profiles. Experimental results on three real-world data sets verify the effectiveness of our method in terms of accuracy and coverage, especially for the cold-start users. Although many trust-based approaches have been proposed previously, the novelty of our work lies in that it is the first work to effectively complement user rating profiles based on the ratings of trusted neighbours.

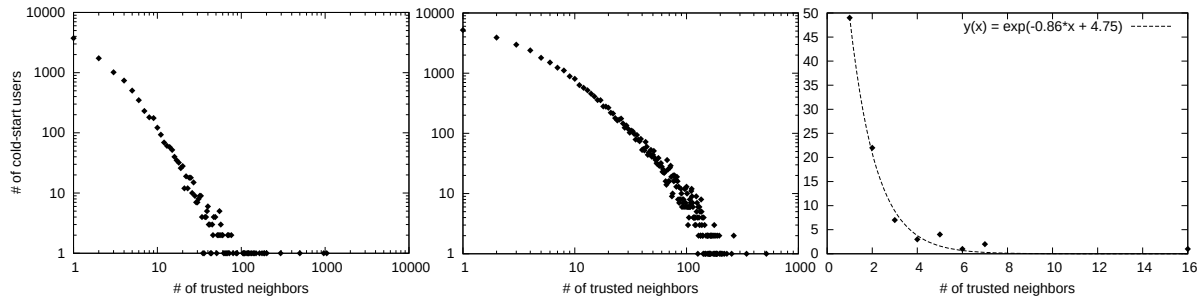
The rest of this chapter is organized as follows. The approach to merge the ratings of trusted neighbours is elaborated in Section 5.1. Then, we incorporate the merged ratings into a CF technique as described in Section 5.2. The use of our approach is exemplified in Section 5.3, followed by the highlights of our method given in Section 5.4. After that, experiments on three real-world data sets are conducted in Section 5.5 to verify the effectiveness of our method in predicting items' ratings, especially the effectiveness to address the cold start problem. Finally, Section 5.6 concludes our present work.

5.1 Merging Process

For the sake of convenience to discuss with, we introduce a number of notations to model the recommendation problem. Specifically, we denote the sets of all users, all items and all ratings as U, I and R, respectively. We keep the symbols u, v for the users and i, j for the items. Then $r_{u,i}$ represents a rating given by user u on item i, and takes a value in a certain rating scale, such as an integer from 1 to 5, predefined by a recommender system. Hence the task of a recommender can be modeled as: given a set of user-item-rating $(u, i, r_{u,i})$ triplets, provide a best prediction $(u, j, ?)$ for user u on an unknown item j. The predicted rating is denoted as $\hat{r}_{u,j}$. In a trust-aware recommender system, the active user u may have identified a set of trusted neighbours TN_u. For each trusted neighbour $v \in TN_u$, user u also specifies a trust value $t_{u,v} \in [0, 1]$ indicating the extent to which user u believes in user v's ability in giving accurate ratings. We presume that user u will always trust herself in giving accurate ratings. Thus, user u herself is also included in the trust neighbourhood TN_u, i.e. $u \in TN_u$ and $t_{u,u} = 1$. Besides, for simplicity, the set of items rated by user u is denoted by $I_u = \{i | r_{u,i} \in R, i \in I\}$, and the set of users who rated item i is denoted by $U_i = \{u | r_{u,i} \in R, u \in U\}$. Hence, the recommendation problem can be re-described as: given a set of user ratings $(u, i, r_{u,i})$ and a set of user trust $(u, v, t_{u,v})$, predict a best prediction $(u, j, \hat{r}_{u,j})$ for an active user u on a target item j. We are most concerned with the predictive accuracy of the predicted ratings (relative to the real preferences) and the percentage of target items that can be predicted.

5.1.1 Aggregating Trusted neighbours

The cold-start users are generally defined as the users who have rated very few items, oftentimes less than five items [16]. Hence, to better model user preference, additional information



(a) Epinions (b) Flixster (c) FilmTrust

Figure 5.1: The distributions of trusted neighbours for the cold-start users

is adopted. In our case, user's social trust information is utilized as users in the systems can specify other users as trusted neighbours. Since cold-start users usually are less active in the systems, they may not have a large number of trusted neighbours. We conduct experiments to show the statistics for cold-start users in real-world data sets, the specifications of which will be presented in Section 5.5.1. Figure 5.1 shows the distribution of trusted neighbours for the cold-start users in three different real-world data sets.

Both (a) and (b) in Figure 5.1 show the well-known *power law* property of social network. Specifically, most cold-start users have only few trusted neighbours and only few cold-start users have identified many trusted neighbours. From Figure 5.1 (c), it is observed that FilmTrust has much fewer amount of trusted neighbours than Epinions and Flixster. Besides, the described trending line of data points follows an exponential function. Nevertheless, the distribution of trusted neighbours is quite close: only few cold-start users have many trusted neighbours whereas most cold-start users have only few ones. Therefore, although social trust can be regarded as a (strongly and positively) additional information source to model user preference, the availability of trust information for cold-users is relatively limited.

Fortunately, trust can be propagated along with the web-of-trust. That is, if user A trusts user B and B trusts user C, it can be inferred that user A trusts user C to some extent. MoleTrust [16] and TidalTrust [39] are two typical algorithms to infer trust value. To better use trust information, it is necessary to propagate trust in order to find more (indirectly) trusted neighbours. In this work, we adopt the MoleTrust to infer the trust value of indirectly connected users. Note that the trust value in the data sets is binary, i.e., 0 or 1, where 0 means no

direct trust connections whereas 1 indicates that a user directly connects with and trusts another user. As a result, the inferred trust value by the MoleTrust will be also binary, and thus we cannot distinguish trusted neighbours in a shorter distance with those in a longer distance. This issue may deteriorate the performance of trust-based approaches. Hence, we adopt a weighting factor to devalue the inferred trust in a long distance:

$$t_{u,v} = \frac{1}{d} * t'_{u,v}, \tag{5.1}$$

where $t'(u, v)$ denotes the inferred trust value by the MoleTrust algorithm, d is the shortest distance between users u and v determined by a breath first search algorithm, and $t_{u,v} \in [0, 1]$ is the trust value that user u has towards another user v. In this way, directly specified trusted neighbours will be more trustworthy than the users in a long distance (but connected in the trust networks). Note that the greater d is, the more trusted neighbours will be inferred. However, the more cost will be taken and more noise is likely to be incorporated. According to the theory of six-degree separation [140], any two users in the social network can be connected (if possible) within small (less than six) steps. In this work, we restrict $d \leq 3$[5.1] to prevent meaningless searching and save computational cost for large-scale data sets. In fact, as we will show later, the Merge method works well enough when d is small.

Hence, a set of users can be identified as trusted neighbourhood for user u if the trust value of a user v is greater than a trust threshold:

$$TN_u = \{v | t_{u,v} > \theta_t, v \in U\}, \tag{5.2}$$

where θ_t is the trust threshold. Since the distance is restricted by $d \leq 3$, we presume that all connected trusted neighbours are useful and hence set $\theta_t = 0$ for simplicity. Although it is flexible to tune the trust threshold θ_t, it is not necessary to do so in practice. We defer the explanation till Section 5.1.3. In addition, the active user u herself is also regarded as a trusted neighbour in her trust neighbourhood, i.e., $u \in TN_u$ and $t_{u,u} = 1$. In other words, we presume that user u will always believe in her own ratings as they are accurately reflecting her real preferences.

[5.1] The same setting is used in [16, 40]. Better performance may be achieved by setting $d \leq 6$ and searching in a longer distance in the trust networks.

5.1.2 Merging the Ratings of Trusted neighbours

After determining the trust neighbourhood, a set of items can be identified as the candidate items for the merging process:

$$\tilde{I}_u = \{i | r_{v,i} \in R, \exists v \in TN_u, i \in I\}. \tag{5.3}$$

That is, $\tilde{I}_u$ consists of items that have been rated by at least one trusted neighbour from the trust neighbourhood. Then all the ratings of trusted neighbours on each item $j \in \tilde{I}_u$ will be merged into a single rating based on the weights of trusted neighbours:

$$\tilde{r}_{u,j} = \frac{\sum_{v \in TN_u} w_{u,v} \cdot r_{v,j}}{\sum_{v \in TN_u} |w_{u,v}|}, \tag{5.4}$$

where $\tilde{r}_{u,j}$ is the merged value for user u on item $j \in \tilde{I}_u$ based on the ratings of all the trusted neighbours, and $w_{u,v}$ denotes the importance weight of user v's ratings relative to the active user u. We claim that the importance weight $w_{u,v}$ is composed of three parts: trust value $t_{u,v}$, rating similarity $s_{u,v}$ and social similarity $j_{u,v}$. Hence, $w_{u,v}$ is computed as a linear combination of the three parts:

$$w_{u,v} = \alpha \cdot s_{u,v} + \beta \cdot t_{u,v} + \gamma \cdot j_{u,v}, \tag{5.5}$$

where $\alpha, \beta, \gamma \in [0, 1], \alpha + \beta + \gamma = 1$, indicate the extent to which the combination relies on rating similarity, trust value and social similarity, respectively. The rationale behind this computation, i.e., incorporating three parts rather than trust value only, is that people trusting each other may not share similar preferences [42]. Specifically, it is possible that trusted neighbours have low similarity. Ray and Mahanti [41] has shown that trusted neighbours with high similarity have a positive influence on predictive accuracy after eliminating those with low similarity. Therefore, it is necessary to consider both rating similarity and trust value.

Pearson correlation coefficient [4] is often used to compute user similarity based on ratings:

$$s_{u,v} = \frac{\sum_{i \in I_{u,v}} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{u,v}} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_{u,v}} (r_{v,i} - \bar{r}_v)^2}}, \tag{5.6}$$

where $s_{u,v} \in [-1, 1]$ is the similarity between two users u and v, and $I_{u,v} = I_u \cap I_v$ denotes the set of items rated by both users u and v. Since the active user $u \in TN_u$, we denote $s_{u,u} = 1$ for the purpose of consistency. In particular, $s_{u,v} > 0$ means positive correlation between users

u and v, $s_{u,v} < 0$ indicates opposite correlation and $s_{u,v} = 0$ implies no correlation. Alterative similarity measures could be cosine similarity [4], Bayesian similarity [17], etc.

The third component is the ratio of commonly trusted neighbours between two users u and v. The intuition is that two users are socially close if they share a number of trusted neighbours. In other words, a trusted neighbour who also shares some social friends will be regarded as more important than the user who has no friends in common with the active user. The social similarity is defined as the ratio of shared trusted neighbours over all the trusted neighbours, and computed by the Jaccard Index:

$$j_{u,v} = \frac{|TN_u \cap TN_v|}{|TN_u \cup TN_v|}, \tag{5.7}$$

where $j_{u,v} \in [0, 1]$ indicates the social similarity of two users u and v based on their trusted neighbours. Hence, the importance weight $w_{u,v}$ can be computed using Equation (5.5) since the three components are derived by Equations (5.1), (5.6) and (5.7), respectively. In this way, all the ratings of trusted neighbours on a certain item can be merged into a single value by Equation (5.4), i.e., the merged rating.

In addition, as indicated by Ray and Mahanti [41] and as a general belief, even trusted users may not share similar preference and so does the social similarity. In other words, the trust and social similarity may be noisy and inaccurate. Considering the cases with positive trust and social similarity but negative rating similarity may not make sense or be expected. Hence, we only consider the positively correlated users in this regard, i.e., $s_{u,v} > 0$. Another reason is to be consistent with the value range of trust and social similarity in Equations (5.1) and (5.7).

Furthermore, since user u always gives accurate ratings from her own viewpoint, all her ratings will be retained and kept unchanged during the merging process as it is not necessary for them to be approximated (by the ratings of other trusted neighbours) in any way. Thus we need to emphasize that only the ratings of trusted neighbours on the other items that user u has not rated will be merged. To put it simply, the active user will keep all her own ratings, and the ratings of trusted neighbours will be used to complement her own preferences so that a new more complete and accurate rating profile can be formed and used to represent the preferences of the active user.

5.1.3 Determining the Confidence of Merged Ratings

A merged rating for an active user on a certain item can be computed using Equation (5.4) based on the ratings of trusted neighbours. However, the quality or usefulness of the merged ratings is unknown. We term it as the *confidence* of the merged ratings, or *rating confidence* for short, which reflect the usefulness of the merged ratings and to what extent the merged ratings are reliable. Intuitively, two factors may have important influence: the number of ratings involved and the conflicts between positive and negative opinions among all these ratings.

More specifically, if an item receives many ratings from the trusted neighbours, the merged value is likely to be correct and reliable. In contrast, if an item only receives few ratings, the merged value tends to be noisy and unreliable. In fact, as shown in Figure 5.1, most cold-start users do not specify many other users as trusted neighbours, and by definition cold-start users rate only small number of items. In this work, we regard the rating whose value is greater than the median of a rating scale as a positive opinion and otherwise as a negative opinion:

$$\begin{cases} r_{v,i} \text{ is positive}: & \text{if } r_{v,i} > r_{med}; \\ r_{v,i} \text{ is negative}: & \text{otherwise}; \end{cases} \tag{5.8}$$

where r_{med} is the median rating value in the range from the minimum rating value r_{min} to the maximum rating value r_{max} predefined by a recommender system. The more consistent (i.e., less conflicts) between positive and negative opinions, the more reliable the merged rating will be. Therefore, only adopting the merged ratings may ignore the significant differences among different items and raise much noise in the merged rating profile, especially for those who have already rated many items, i.e., the *heavy users*. It is necessary to take into account the rating confidence for later rating predictions.

In conclusion, the measure of rating confidence should manage to reflect the differences in the number of ratings of trusted neighbours, and the differences in the conflicts between positive and negative opinions. Formally, the confidence $c_{u,j}$ of a merged rating $\tilde{r}_{u,j}$ is defined in the evidence space $< p_{u,j}, n_{u,j} >$ (refers to Wang and Singh [141]):

$$c_{u,j} = c(p_{u,j}, n_{u,j}) = \frac{1}{2} \int_0^1 \left| \frac{x^{p_{u,j}}(1-x)^{n_{u,j}}}{\int_0^1 x^{p_{u,j}}(1-x)^{n_{u,j}} d_x} - 1 \right| d_x, \tag{5.9}$$

where $c_{u,j} \in (0, 1]$ is the rating confidence of merged rating $\tilde{r}_{u,j}$ as a function of $p_{u,j}$ and $n_{u,j}$, referring to the number of positive, negative opinions (ratings) provided by the trusted

neighbours on item $j \in \tilde{I}_u$, respectively. The variable x denotes the probability of a given rating being positive. Hence, the greater the number $(p_{u,j} + n_{u,j})$ of ratings is, and the less conflicts between $p_{u,j}$ and $n_{u,j}$ will lead to greater confidence $c_{u,j}$. For consistency, the rating confidence of the ratings rated by the active users is always believed to be the highest, i.e., $c_{u,i} = 1$, for any item $i \in I_u$.

With the concept of rating confidence, we can now explain why it is not necessary to set or tune a proper trust threshold θ_t during the formation of trust neighbourhood in Equation (5.2). The reason is straightforward. Although less trustworthy users may be involved in the merging process, their influence to the merged rating is less than those with greater trust values, and the confidence measure can also mitigate their influence.

In summary, the merging process for each item $j \in \tilde{I}_u$ will produce two outputs: the merged rating $\tilde{r}_{u,j}$ and the corresponding rating confidence $c_{u,j}$. All the pairs of $(\tilde{r}_{u,j}, c_{u,j})$ will form a new rating profile to represent the preferences of the active users, based on which rating predictions can be generated more accurately.

5.2 Incorporating with Collaborative Filtering

Given the new rating profile on the item set $\tilde{I}_u$ after the merging process in Section 5.1, which represents the preferences of the active user u, we then apply a conventional CF technique to predict the rating of a target item j that has not been rated by user u. More specifically, we first probe a set NN_u of similar users (i.e., nearest neighbours) for user u based on the similarity between user u and other users who have rated item j. Then the ratings of these nearest neighbours will be aggregated to produce a prediction for user u on item j.

In general, Pearson correlation coefficient (PCC) is often adopted to measure the similarity between two users according to their ratings on the items that they commonly rated (see Equation (5.6)). In our case, other than the merged ratings, the confidence is also important to indicate the quality of the merged ratings. Since Equation (5.6) does not consider the rating confidence, we introduce a confidence-aware PCC to compute user similarity, denoted by *CPCC* for short:

$$s'_{u,v} = \frac{\sum_{i \in I_{u,v}} c_{u,i}(\tilde{r}_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{u,v}} c_{u,i}^2 (\tilde{r}_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_{u,v}} (r_{v,i} - \bar{r}_v)^2}}, \tag{5.10}$$

where $I_{u,v} = \tilde{I}_u \cap I_v$ is the set of items rated by both users u and v after the merging process, $\bar{r}_u$ and $\bar{r}_v$ are the average ratings for users u and v respectively, and $c_{u,i}$ is the confidence measurement regarding the merged rating $\tilde{r}_{u,i}$. The CPCC measure is inspired by the work of Xue et al. [142] in which the confidential weight of an item rated by the active user in their Equation (7) plays the same role as the rating confidence in our work, i.e., to discount the values of their ratings. For a real (non-merged) rating $r_{v,j}$ provided by a similar user v, we consider that its rating confidence is $c_{v,j} = 1$ and hence omitted in Equation (5.10).

After computing user similarity, a group of similar users are then selected into the nearest neighbourhood NN_u of the active user u. Herein we use the thresholding method, i.e., adopting the users whose similarity with the active user u is greater than a predefined threshold:

$$NN_u = \{v | s'_{u,v} > \theta_s, v \in U\}, \tag{5.11}$$

where θ_s is a predefined similarity threshold. An alternative method to determine the nearest neighbourhood is well known as top-K where the top K most similar users will be used. However, since in this work we focus on the performance of the cold-start users, the top-K method is less effective to determine the nearest neighbourhood than the thresholding method according to our experiments. Specifically, when we tune the values of K, no significant changes are observed in the performance of comparing methods. This may be due to the few similar users that can be identified based on the little rating information. Therefore, we use the thresholding rather than the top-K method to select nearest neighbours for the active users. We will investigate the effect of similarity threshold for our method in the experiments (see Section 5.5.4).

Finally, all the ratings of nearest neighbours are aggregated to produce a prediction on a target item j that the active user u has not rated. We use the simple weighted average method, i.e., to compute the average value of all ratings provided by the nearest neighbours v weighted by their similarity $s_{u,v}$ with the active user u. Formally, the prediction is computed by:

$$\hat{r}_{u,j} = \frac{\sum_{v \in NN_u} s'_{u,v} \cdot r_{v,j}}{\sum_{v \in NN_u} |s'_{u,v}|}, \tag{5.12}$$

where $\hat{r}_{u,j}$ represents the predicted value on item j. Hence it ensures that the users with greater similarity will have more influence on the predictions. An alternative prediction method is Resnick's formula [4] which in addition considers user bias in giving ratings. Nevertheless, we adopt the weighted average because the two most related works [16, 41] also take the same equation, leading to a fairer comparison in our experiments.

Table 5.1: The synthetic data set consisting of both (a) rating and (b) trust information

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9
u_1			5		**?**				
u_2	5		4		3			2	
u_3		4		3				1	
u_4	3		5		2				
u_5		4	4		3			3	
u_6		3	3	5	5				
u_7							5		4
u_8			4		2			1	
u_9			4		5			5	

(a) user-item rating matrix

	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9
u_1		1	1						
u_2			1	1					
u_3		1		1					
u_4					1				
u_5				1		1			
u_6			1	1					
u_7									
u_8									
u_9									

(b) user-user trust matrix

5.3 An Example

In this section, we intend to exemplify step by step the use of the Merge method to generate a prediction for a given item. Suppose there are nine users and nine items, denoted by u_k and i_j respectively, where $k, j \in [1, 9]$ in a certain system. Each user may rate a few items by giving an integer rating ranged in [1, 5] as shown in Table 5.1 (a). In addition, users may specify other users as trusted neighbours as shown in Table 5.1 (b), where an entry for example $(u_1, u_2, 1)$ indicates that user u_1 specifies user u_2 as a trusted neighbour. In this example, we are interested in generating a prediction on a target item i_5 (highlighted by the question mark) for an active user u_1. User u_1 has only reported a rating 5 on item i_3. She has indicated that users u_2 and u_3 as her trusted neighbours, and both trusted users also pointed out others as trusted neighbours. By linking all the trusted neighbours together, we form a trust network for user u_1 as illustrated in Figure 5.2. Specifically, users are represented as nodes and the trust links are denoted as edges among users. Note that trust information is asymmetric, that is, users u_1 trusting u_2 does not imply users u_2 trusting u_1.

Table 5.2: The computed trust values between user u_1 and others

	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9
d	0	1	1	2	3	4			
t_{u_1,u_k}	1.0	1.0	1.0	0.5	0.33	0.25			

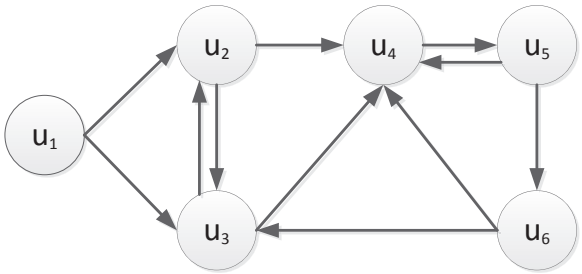


Figure 5.2: The trust network for a cold user u_1

The first step of the Merge method is to identify the trusted neighbours of the active user by allowing trust propagation in the trust network. According to Figure 5.2, trust values between the active user u_1 and other users can be inferred by Equation (5.1) and the results are presented in Table 5.2. In particular, as an active user, u_1 always trusts herself in giving accurate ratings and hence $t_{u_1,u_1} = 1.0$. Since users u_2 and u_3 are directly specified by user u_1, i.e., $d = 1$, their trust values will be 1.0. For user u_4, the minimum distance to user u_1 is 2, i.e., $d = 2$. The shortest path of trust propagation is $u_1 \rightarrow u_2$ (or u_3) $\rightarrow u_4$, and the other path could be $u_1 \rightarrow u_2 \rightarrow u_3 \rightarrow u_4$. Hence, the trust value is computed by $t_{u_1,u_4} = 1/2 = 0.5$. The minimum distance from users u_1 to u_5 will be: $d(u_1, u_5) = d(u_1, u_4) + d(u_4, u_5) = 3$, and the distance to u_6 can be computed in the same manner. Note that although the trust value of user u_6 is computable, this user will not be regarded as an inferred trusted neighbours due to the constraint $d \leq 3$. Hence, a set of users $TN_{u_1} = \{u_1, u_2, u_3, u_4, u_5\}$ are identified as trusted neighbours for active user u_1.

Table 5.3: The merged rating profile for user u_1

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9
$\tilde{r}_{u_1,i_j}$	4.33	4	5	3	2.73			1.72	
c_{u_1,i_j}	0.19	0.38	1.0	0.25	0.47			0.47	

Second, the ratings of trusted neighbours will be merged using Equations (5.4), (5.5) and (5.9). For simplicity, in this example we set $\alpha = 0, \beta = 1$ for Equation (5.5), i.e., only trust values are used as user weights. The resultant merged ratings and confidences are presented in Table 5.3. In particular, since user u_1 has rated item i_3, and we presume the active user will always believe in her own ratings, hence there is no need to consider the ratings of trusted neighbours. Therefore, the merged rating on item i_3 is equal to r_{u_1,i_3} (i.e., 5), and the confidence

is the highest (i.e., 1.0). For other items that user u_1 has not rated, the ratings of trusted neighbours will be merged by Eqution (5.4) as well as the rating confidence by Equation (5.9). Take item i_1 as an instance. The ratings of users u_2 and u_4 will be averaged and weighted by their trust values, i.e.,

$$\tilde{r}_{u_1,i_1} = \frac{5 \times 1.0 + 3 \times 0.5}{1.0 + 0.5} = 4.33$$

For a rating scale from 1 to 5, the median rating is 3. According to Equation (5.8), user u_2's rating 5 is regraded as positive, while user u_4's rating 3 is negative. The confidence is derived by Equation (5.9):

$$c_{u_1,i_1} = c(1,1) = 0.19$$

This procedure continues until all the items rated by at least a trusted neighbour have been covered. A new rating profile is formed and shown in Table 5.3. Since there are only a few trusted neighbours, the computed confidence is relatively small. The merged rating profile is much more complete than the original.

Third, user similarity is computed by Equation (5.10) based on the formed rating profile (see Table 5.3), taking into account the rating confidence. The results are shown in Table 5.4. For consistency, the similarity between user u_1 and herself is 1.0. For comparison's purpose, we also show the similarity values computed by conventional PCC (see Equation (5.6)). It is noted that PCC values are less distinguishable than CPCC values, and the differences between CPCC and PCC values could be large. In other words, the confidence plays an important role in our similarity computation. A set of users $NN_{u_1} = \{u_2, u_4, u_5, u_8\}$ are selected as nearest neighbours, whose similarity is greater than the threshold $\theta_s = 0$ and who have rated the target item i_5 (noted that user u_3 did not rate item i_5).

Table 5.4: The computed similarity between user u_1 and others

	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9
CPCC	1.0	0.66	0.995	0.98	0.84	-0.78		0.99	-0.98
PCC	1.0	0.87	0.992	0.91	0.91	-0.91		1.0	-0.95

Finally, a prediction for item i_5 is generated by Equation (5.4):

$$\hat{r}_{u_1,i_5} = \frac{3 \times 0.66 + 2 \times 0.995 + 3 \times 0.84 + 2 \times 0.99}{0.66 + 0.995 + 0.84 + 0.99} = 2.43$$

and the rating confidence is:

$$c_{u_1,i_5} = c(0,4) = 0.53$$

Compared with the values (2.73, 0.47) shown in Table 5.3, the final prediction is different from the merged rating which is only based on trusted neighbours, and the final rating confidence is higher than the merged one since more ratings of similar users are used. In other words, generating a prediction only based on trusted neighbours may not be reliable, and the resultant rating confidence could be low if only few trusted neighbours can be identified. This is the situation for the cold-start users. In contrast, by merging the ratings of trusted neighbours, the ratings of similar users can be adopted to smooth the predictions and enhance the confidence. Furthermore, the only item that user u_1 has rated is i_3 which receives many ratings from system users. That is, item i_3 is a popular item. The conventional CF will treat all the users as similar users and hence the extreme ratings given by users u_6 and u_9 will bias the final prediction for item i_5. By forming a more complete rating profile for the active user u_1, the Merge method is able to identify that users u_6 and u_9 in fact have different preferences and hence they will be excluded to generate the final prediction. As a result, the prediction generated by our method is likely to be more accurate and reliable.

5.4 Insights into the Merge Method

Principally, the Merge method has two distinct advantages relative to other methods. First, it can effectively ameliorate the data sparsity and cold start problems. The essential challenge of the two issues is that the small amount of items commonly rated by two users makes it difficult to accurately compute user similarity, and hence to find reliable similar users. Even worse, two users may not have any co-rated items in common, resulting in non-computable user similarity. The Merge method copes with the cold conditions by merging the ratings of the trusted neighbours to form a new rating profile which is used to represent the preferences of the active user. Specifically, the relation $I_u \subseteq \tilde{I}_u$ can be inferred from Equation (5.3) because of $u \in TN_u$, that is, the newly formed rating profile covers more items than the original rating profile. The previous example also confirmed this point. Thus, more similar users can be identified in terms of user similarity, which is especially useful for cold-start users with only a few or none ratings. The example in Section 5.3 also showed that the computed similarity

tended to be more reliable and distinguishable by considering confidence. As a consequence, our method can alleviate the data sparsity and cold start problems.

Second, the Merge method can function well in the case of either sparse rating or sparse trust information. Previously, many trust-based approaches such as MoleTrust [16] and TidalTrust [43] predict item ratings only based on the ratings provided by the trusted neighbours. Hence these approaches may also suffer from the similar cold-start problem where some users may only specify a small number of other users as their trusted neighbours, which has been demonstrated in the three real-world data sets and shown in Figure 5.1. Hence this issue could be a common case for many online systems, especially when users are lack of incentives to proactively connect with each other. In this case, the performance will be limited since only a few neighbours can be incorporated for recommendation. In contrast, the Merge method addresses this problem by also making use of the ratings of the active users if any. In particular, when the active user has not specified any trusted neighbours but rated a certain number of items, the merged rating profile will then be exactly the same as her own and real rating profile because the only trusted neighbours is herself. The Merge method will have no differences with the conventional CF method. On the other hand, when the cold user has not rated any items but specified some trusted neighbours, then the ratings of these trusted neighbours can be merged as we described. In either case, our method is competent to form a new rating profile and hence mitigate the cold start problem. Although our method will fail to work when there is neither rating nor trust information of the active users, other kind of information may be needed to help model user preference which is beyond the discussion of this work.

In this regard, our method possess some advantages of hybrid approaches. By merging user- and item-based approaches together, hybrid methods can also alleviate the concern issues [143, 101]. However, our method differs from those methods in three-fold. Firstly, our work is only based on rating information, namely item ratings and trust ratings. Hybrid methods usually depend on more heterogenous information, such as music genre in [143] or item taxonomy in [101]. Hence, our method is more generic than the hybrid ones. Secondly, more complex information needs more computational steps to deal with, hence the hybrid methods are usually more complex and hard to be implemented than singe approaches. Lastly, as a user-based approach, our method holds the potential to be incorporated with other item-based approaches to form more powerful hybrid methods in the future.

Table 5.5: The specifications of three data sets

Data set	# Users	# Items	# Ratings	# Trust	Sparsity
FilmTrust	1986	2071	35497	1853	98.86%
Flixster	53K	18K	410K	650K	99.96%
Epinions	49K	139K	664K	478K	99.95%

5.5 Evaluation

In order to verify the effectiveness of the Merge method, we conduct experiments on three real-world data sets. Specifically, we aim to find out: (1) how is the performance of our method in comparison with other counterparts; (2) what is the effect of trust propagation to our method and the others.

5.5.1 Data Sets

Three real-world data sets are used in our experiments, namely FilmTrust[5.2], Flixster[5.3] and Epinions[5.4] as they all include the data of both explicit trust statements and user-item ratings. The specifications of the three data sets are summarized in Table 5.5.

To be specific, FilmTrust is a trust-based social site in which users can rate and review movies. Since there is no publicly available data sets due to the preservation of user privacy, we crawled the whole site in June 2011, collecting 1,986 users, 2,071 movies and 35,497 ratings. The ratings take values from 0.5 to 4.0 with step 0.5. In addition, we also gathered 1,853 trust ratings that are issued by 609 users. The average number of trusted neighbours per user is less than 1. Originally, users can specify other users as trusted neighbours with a certain level of trust from 1 to 10. However, these trust values are not available due to the sharing policy. We can only get the link information among users and hence the trust value is 1 if a link exists between two users otherwise the value is 0.

Flixster is a social movie site in which users are allowed to share their movie ratings, discover new movies and interact with others who have similar taste. We adopt the data set[5.5] collected by Jamali and Ester [63] which includes a large amount of data. The ratings are real

[5.2] http://www.librec.net/datasets.html
[5.3] http://www.flixster.com/
[5.4] http://www.epinions.com/
[5.5] http://www.cs.sfu.ca/~sja25/personal/datasets/

values ranged from 0.5 to 4.0 with an interval 0.5, and the trust statements are scaled from 1 to 10 but again not available. Hence, they are converted into binary values the same as FilmTrust, that is, trust value 1 is assigned to a user who is identified as a trusted neighbours and 0 otherwise. Note that the trust statements in this data set are symmetric. We sample a subset by randomly choosing 53K users who issued 410K item ratings and 655K trust ratings.

Epinions is a review sharing website in which users can express their opinions about items (such as movies, books, software, etc.) by assigning numerical ratings and writing text reviews. Users can specify other users as trustworthy (to the trust list) or untrustworthy (to the distrust list) according to whether the text reviews and comments of other users are consistently valuable to them. The data set[5.6] is generated by Massa and Avesani [16], consisting of 49K users who issued 664K ratings over 139K items and 478K trust statements. The ratings are integers ranged from 1 to 5, and the trust values are also binary. The rating sparsity is computed by:

$$\text{Sparsity} = \left(1 - \frac{\#Ratings}{\#Users \times \#Items}\right) \times 100\%.$$

It is noted that all the data sets are highly sparse, i.e., users only rate a small portion of items in the systems.

5.5.2 Experimental Settings

In the experiments, we compare the performance of our method Merge with a number of trust-based state-of-the-art methods as well as a conventional user-based CF method.

- **CF** computes user similarity using the PCC measure, selects the users whose similarity is above the predefined similarity threshold θ_s for Equation (5.11), and uses their ratings to generate item predictions by Equation (5.12). In this work, the threshold θ_s is set 0 for all the methods according to our analysis in Section 5.5.4.

- **MT**x ($x = 1, 2, 3$) is the implementation of the MoleTrust algorithm [16] in which trust is propagated in the trust network with the length x. Only trusted neighbours are used to predict the ratings of unknown items.

[5.6] http://www.trustlet.org/datasets/downloaded_epinions

- **RN** denotes the approach proposed by Ray and Mahanti [41] that predicts item ratings by reconstructing the trust networks. We adopt their best performance settings where the correlation threshold is 0.5, propagation length is 1, and the top 5 users with highest correlations are selected for rating predictions.

- **TCF**x ($x = 1, 2$) denotes the approach proposed by Chowdhury et al. [40] that enhances CF by predicting the ratings of the similar users who did not rate the items according to the ratings of the similar users' trusted neighbours, so as to incorporate more users for recommendation. The best performance that they report is achieved when the prediction iteration x over the trust network is 2. We adopt the same settings in our experiments.

- **Merge**x ($x = 1, 2, 3$) is our method with the trust propagation length x, aiming to investigate the impact of trust propagation on the Merge method. Besides, we denote **Merge-**α as a variant where parameter α in Equation (5.5) is set 1, meaning the importance weight is completely determined by user similarity. Further, we also denote **Merge-**β as a variant with the best performance when parameter β in Equation (5.5) is set 0, meaning that the explicitly specified or inferred trust value is not used.

In addition, we split each data set into two different views as defined in [16]: the view of **All Users** represents that all users and their ratings will be tested whereas the view of **cold-start users** denotes that only the cold-start users who have rated less than five items, and their ratings will be tested in the experiments. In particular, we focus on the performance in the view of *cold-start users* which mostly indicates the effectiveness in mitigating the data sparsity and cold start problems.

5.5.3 Evaluation Metrics

The performance of all the methods is evaluated in terms of both accuracy and coverage. The evaluation is proceeding by applying the *leave-one-out* method on the two data views. In each data view, users' ratings are hidden one by one in each iteration and then their values will be predicted by applying a certain method until all the testing ratings are covered. The errors between the predicated ratings and the ground truth are accumulated. The evaluation metrics are described as follows.

- Mean Absolute Error, or *MAE*, measures the degree to which a prediction is close to the ground truth, given by:

$$MAE = \frac{\sum_u \sum_i |\hat{r}_{u,i} - r_{u,i}|}{N}, \tag{5.13}$$

where N is the number of testing ratings. Hence, the smaller an MAE value is, the closer a prediction is to the ground truth. Inspired by Jamali and Ester [98] who define a measure *precision* based on root mean square error (RMSE), we define the inverse MAE, or *iMAE* as another metric of predictive accuracy normalized by the range of rating scale:

$$iMAE = 1 - \frac{MAE}{r_{\max} - r_{\min}}, \tag{5.14}$$

where $r_{\max}$ and $r_{\min}$ are the maximum and minimum rating value defined by a recommender systems, respectively. Higher iMAE values indicate better predictive accuracy.

- Ratings Coverage, or *RC*, measures the degree to which testing ratings can be predicted and covered relative to the whole testing ratings, defined by:

$$RC = \frac{M}{N}, \tag{5.15}$$

where M and N are the number of predictable and all the testing ratings, respectively.

- F-measure, or *F1*, measures the overall performance in considering both rating accuracy and coverage. Both metrics are important measures for the overall predictive performance. According to Jamali and Ester [98], the F-measure is computed by:

$$F1 = \frac{2 \cdot iMAE \cdot RC}{iMAE + RC}. \tag{5.16}$$

Hence, the F-measure reflects the balance between accuracy and coverage.

5.5.4 Results and Analysis

In this section, we conduct a series of experiments on three real-world data sets to demonstrate the effectiveness of our approach relative to others, and thus to answer the research questions proposed in Section 5.5. Both data set views, namely *All Users* and *cold-start users* are tested. The results are presented in Tables 5.6, 5.7 and 5.8 corresponding to the predictive performance on the FilmTrust, Flixster, and Epinions data sets, respectively. The value 'NaN' represents 'not a number', i.e., a value that cannot be computed. The best performance of our approach as well as the best of other methods is highlighted for easy comparison.

Table 5.6: The predictive performance on the FilmTrust data set, where the three rows in each testing view correspond to the MAE, RC and F1 performance metrics.

Views	Approaches measured by MAE, RC and F1											
	CF	MT1	MT2	MT3	RN	TCF1	TCF2	Merge-α	Merge-β	Merge1	Merge2	Merge3
All Users	**0.703**	0.852	0.795	0.771	0.571	0.714	0.719	**0.703**	0.704	0.705	0.707	0.708
	93.84%	21.20%	27.96%	30.38%	0.74%	94.92%	**95.19%**	94.06%	94.21%	94.77%	94.94%	**95.06%**
	0.8631	0.3312	0.4106	0.4373	0.0147	0.8658	**0.8661**	0.8640	0.8647	0.8667	0.8672	**0.8674**
Cold Users	**0.744**	0.853	0.880	0.819	NaN	0.751	0.751	**0.737**	0.764	0.768	0.772	0.768
	39.64%	17.11%	23.19%	23.85%	0.00%	39.97%	**40.79%**	39.80%	43.26%	53.45%	54.11%	**54.28%**
	0.5273	0.2791	0.3541	0.3637	NaN	0.5298	**0.5369**	0.5292	0.5569	0.6345	0.6387	**0.6404**

Table 5.7: The predictive performance on the Flixster data set, where the three rows in each testing view correspond to the MAE, RC and F1 performance metrics.

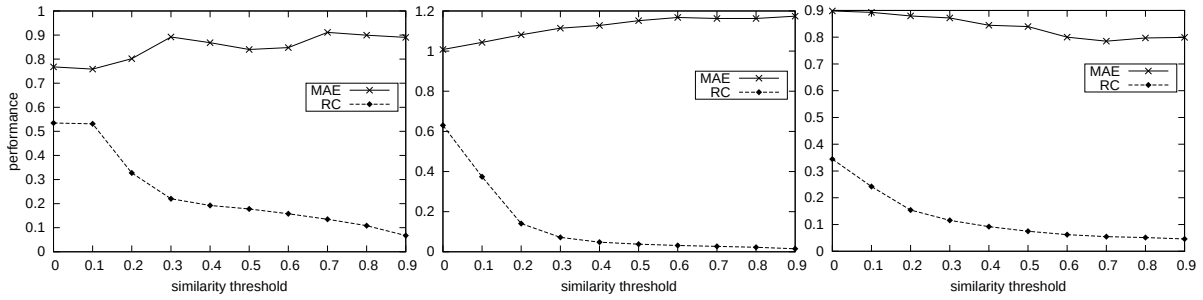
Views	Approaches measured by MAE, RC and F1											
	CF	MT1	MT2	MT3	RN	TCF1	TCF2	Merge-α	Merge-β	Merge1	Merge2	Merge3
All Users	0.928	1.060	0.932	0.862	0.858	0.870	**0.850**	0.917	0.903	0.890	0.877	**0.875**
	68.56%	12.36%	71.37%	**90.71%**	0.38%	80.92%	85.23%	69.63%	82.93%	89.64%	94.39%	**95.03%**
	0.7357	0.2128	0.7512	0.8549	0.0076	0.8079	**0.8312**	0.7429	0.8141	0.8467	0.8690	**0.8720**
Cold Users	1.153	1.127	1.005	0.934	NaN	1.047	**0.923**	1.147	1.018	1.008	0.960	0.949
	3.27%	8.11%	52.69%	**79.55%**	0.00%	12.97%	21.41%	3.30%	41.57%	63.08%	83.13%	**85.15%**
	0.0626	0.1464	0.6279	**0.7939**	NaN	0.2219	0.3373	0.0632	0.5409	0.6959	0.8083	**0.8190**

Table 5.8: The predictive performance on the Epinions data set, where the three rows in each testing view correspond to the MAE, RC and F1 performance metrics.

Views	Approaches measured by MAE, RC and F1											
	CF	MT1	MT2	MT3	RN	TCF1	TCF2	Merge-α	Merge-β	Merge1	Merge2	Merge3
All Users	0.876	0.845	0.852	0.832	**0.673**	0.867	0.864	0.851	0.841	0.839	0.824	**0.820**
	51.24%	26.34%	57.64%	71.68%	9.87%	70.28%	**77.48%**	59.72%	68.61%	73.35%	78.50%	**80.02%**
	0.6188	0.3949	0.6654	0.7525	0.1765	0.7409	**0.7794**	0.6792	0.7343	0.7608	0.7895	**0.7976**
Cold Users	1.033	**0.756**	0.916	0.890	NaN	0.982	0.941	1.038	0.913	0.898	0.876	**0.867**
	3.22%	6.57%	22.06%	**41.73%**	0.00%	7.16%	10.45%	3.37%	18.62%	34.49%	49.59%	**52.66%**
	0.0617	0.1216	0.3431	**0.5431**	NaN	0.1308	0.1839	0.0644	0.3000	0.4774	0.6066	**0.6298**

Effect of similarity threshold θ_s. The Merge method requires to select a set of nearest neighbours to make a prediction, referring to Equation (5.11) where a similarity threshold θ_s is used. In this experiment, we intend to determine the best similarity threshold for cold-start users. For simplicity, we set $\alpha = 0.5, \beta = 0.3$ (see Equation (5.5), explained in the next subsection), and vary the threshold θ_s from 0.0 to 0.9 with step 0.1. The performance of our approach in the view of *cold-start users* on the three data sets is illustrated in Figure 5.3.

Specifically, the results show that as similarity threshold increases, the rating coverage (RC) decreases dramatically. It is because less nearest neighbours are used to make predictions. Although users with greater similarities are adopted, it does not mean that the predictions



(a) FilmTrust (b) Flixster (c) Epinions

Figure 5.3: The performance of our approach Merge in the view of *cold-start users* with varying similarity thresholds on the three data sets.

generated by few highly similar users will be more reliable than those generated by many users with smaller similarities. Hence, the trends of accuracy (MAE) may vary in different data sets. We select the similarity threshold such that both the accuracy and rating coverage are high, i.e., $\theta_s = 0$ for the following experiments.

Importance weights with parameters α and β. An important step for the Merge method is to compute the importance weight of each trusted neighbours as a linear combination of rating similarity, trust value and social similarity with parameters α and β (see Equation (5.5)). When $\alpha = 1$, the weights of trusted neighbours depend completely on the rating similarity, and the performance of this variant is denoted by **Merge-α**. When $\beta = 0$, the trust values are not considered and the importance weights are totally dependent on rating and social similarities. The best performance in this case is obtained when $\alpha = 0.5$ on FilmTrust and Flixster, and $\alpha = 0.7$ on Epinions. We denote this specific variant as **Merge-β**. When $\alpha \in (0, 1)$, both rating-based value and trust-based value will be used. In fact, the experiments show that the settings of (α, β) are (0.5, 0.3) on FilmTrust, (0.5, 0.4) on Flixster and (0.2, 0.4) on Epinions achieve the best performance. These parameters are generally obtained by cross validation. It is denoted as **Merge1** when trust propagation is not used. Merge2 and Merge3 adopt the same settings except the length of trust propagation.

From Tables 5.6-5.8, it is observed that Merge-α achieves similar performance with the baseline CF method in both views of *All Users* and *cold-start users*. More specifically, both

accuracy (in terms of MAE) and coverage (in RC) are slightly improved as well as the overall performance (in F1). This is because for cold-start users (view *cold-start users*), although a number of trusted neighbours may be identified, the number of trusted neighbours with computable and positive similarity could be small due to only few ratings available. For other type of users (view *All Users*), many similar users can be identified which hence also include the trusted neighbours with computable similarity. In other words, these similar users may already cover a high ratio of items, resulting in smaller effect of the Merging process.

For the Merge-β where social similarity is considered in addition to rating similarity, the performance is greatly improved relative to Merge-α, especially in Flixster and Epinions. However in FilmTrust, the differences between Merge-α and Merge-β are marginal. This may be explained by the fact that the CF method already achieves sufficiently good performance. When social information is added, the quality of rating similarity may be decreased due to possible noise embedded, and hence the accuracy may be slightly decreased whereas the coverage is increased to a limited extent. Nevertheless, the overall performance (in F1) is improved. In contrast, the rating information is less useful in Flixster and Epinions, and using social similarity can effectively increase both the accuracy (up to 12% increments) and coverage (up to 38% increments) as well as the overall performance, especially in the view of *cold-start users* which is the main concern of this thesis.

Furthermore, by taking into account both rating and trust information, Merge1 achieves even better performance than Merge-β method in terms of accuracy and coverage. Again, the greatest improvements are observed in the view of *cold-start users*, especially in coverage (up to 20% increments). As the best parameters are set by different value combinations across three data sets, we may conclude that rating similarity (importance weight 0.5) is more important than trust value (0.3 or 0.4) which is superior to social similarity in determining user preferences. In addition, it also shows that both rating and trust information are useful and should be integrated to improve the recommendation performance.

Trust propagation in different lengths. An important factor for trust-based approaches is the use of trust transitivity. By propagating trust values through trust networks, more trusted neighbours can be identified and hence the performance of CF can be further improved. We investigate the influence of trust propagation on the performance of the Merge method. Compared with Merge1, Merge2 and Merge3 have a better accuracy and coverage. This may be

explained by that the merged ratings will be more accurate and have greater confidence due to more evidences (i.e., ratings) available.

Note that the differences between Merge2 and Merge3 are less than the differences between Merge1 and Merge2. We may conclude that trust propagation is helpful to improve recommendation performance, and for our method, it shows that a short propagation length (i.e., 2) will be good enough to achieve a satisfying performance. This is because although more trusted neighbours can be identified via trust propagation, it does not guarantee that the merged rating profile will cover a lot more items and hence increase accuracy greatly. Rather, it is possibly that adding few trusted neighbours may result in some noisy merged ratings (due to few ratings), and hence harm the predictive performance such as that in the FilmTrust data set.

Comparison with other methods. For other methods, we obtain close results on Epinions as shown in Table 5.8 relative to those reported in [16, 40]. The similar trends of results are also obtained on the other two data sets, as shown in Tables 5.6 and 5.7. More specifically, CF cannot achieve a large portion of predictable items, especially on the large-scale data sets (i.e., Flixster and Epinions) and the accuracy is usually bad. It confirms that CF suffers from cold start severely. The RN method accomplishes good accuracy but covers the smallest portion of items, since only the ratings of the users who have a large number of trusted neighbours and high rating correlations are possible to be predicted. Hence RN is not comparative with others. Comparing with CF, all other methods achieve better performance for cold-start users in all the data sets except in the FilmTrust where only our Merge methods can outperform it in either accuracy or coverage as well as the overall performance. When only direct trusted neighbours are used (MT1, Merge1), our method achieves better accuracy and coverage in FilmTrust and Flixster. In Epinions, MT1 works better than our method in accuracy for cold-start users but much worse in coverage. It shows that MT1 may have a good accuracy in some data sets, but not consistently in all the data sets. When trust is propagated in longer length, both accuracy and coverage are increased in Flixster and Epinions whereas only coverage increases in FilmTrust. Nevertheless, our method generally outperforms MTx in all the data sets. TCF methods generally obtain better coverage in the view of *All Users*. However, for cold-start users, TCF functions badly due to the limitation that it relies on CF to find similar users before it can apply trust information on them. As aforementioned, CF is not effective in cold conditions.

This fact leads to poor performance of TCF methods. In contrast, our method is not subject to the ratings of cold-start users themselves. Instead, trust information is merged to form a more concrete rating profile for the cold-start users based on which CF is applied to find similar users and hence generate recommendations. Consistently, we come to a conclusion that the Merge method outperforms the other approaches both in terms of accuracy and coverage as well as a better balance between them.

To have a better view of the overall performance that each method achieves, we further compute the percentage of relative improvements that each method obtains comparing with the CF in terms of F1. Formally, it is computed by[5.7]:

$$\text{Improvement} = \frac{\text{Method.F1 - CF.F1}}{\text{CF.F1}} \times 100\% \tag{5.17}$$

Where 'Method' refers to any one of the methods tested in our experiments except the CF approach, whose F1 performance is regarded as a reference. Hence, the greater positive changes between 'Method' and CF, the more improvements we obtain. The results are shown in Table 5.9, where *All* and *Cold* refer to the cases of *All Users* and *cold-start users* for simplicity, respectively. To explain, we take two values in Table 5.9 as an example, namely 21.45% and 1208.31% for our method $Mergex$. In the *Cold* case of FilmTrust, the best Merge method shown in Table 5.6 is Merge3 with F1 value 0.6404, while the F1 of CF is 0.5273. Hence, the improvement is (0.6404-0.5273)/0.5273*100% = 21.45%. Similarly, in the *Cold* view of Flixster, Merge3 achieves F1 value 0.8190 while CF has a poor performance with 0.0626 (see Table 5.7), leading to the improvement (0.8190-0.0626)/0.0626*100%=1208.31%. Other values can be explained and verified as well. Note that the value *NaN* indicates the improvement is not computable for the RN method in the view of *Cold*. This can be explained by the fact that RN cannot cope with cold-start users and predict item ratings (see Tables 5.6-5.8). A conclusion drawn from Table 5.9 is that our method consistently outperforms the others (in term of improvement), and significantly improve the performance of traditional collaborative filtering.

5.6 Concluding Remarks

This chapter proposed a novel method to incorporate trusted neighbours into collaborative filtering techniques, aiming to resolve the data sparsity and cold start problems of traditional

[5.7]The formula can be referred to as the *relative change* defined in http://en.wikipedia.org/wiki/Relative_change_and_difference

Table 5.9: The improvements of all methods comparing with CF in F1

Dataset	View	MTx	RN	TCFx	Mergex
FilmTrust	All	-49.33%	-98.30%	0.35%	0.50%
	Cold	-31.03%	NaN	1.82%	21.45%
Flixster	All	16.20%	-98.97%	12.98%	18.53%
	Cold	1168.21%	NaN	438.82%	1208.31%
Epinions	All	21.61%	-71.48%	25.95%	28.89%
	Cold	780.23%	NaN	198.06%	920.75%

recommender systems. Specifically, the ratings of trusted neighbours were merged to complement and represent the preferences of the active users, based on which more reliable similar users can be identified and recommendations were then generated. The quality of merged ratings was measured by the confidence considering both the number of ratings involved and the conflicts between positive and negative opinions (i.e., ratings). The rating confidence was incorporated to compute user similarity, and hence a confidence-aware similarity measure was introduced. The prediction of a given item was generated by averaging the ratings of similar users weighted by their importance. Experiments on three real-world data sets were conducted and the results showed that significant improvements against other methods were obtained both in terms of accuracy and coverage as well as the overall performance. Further, by propagating trust (with a few hops) in the trust networks, even better predictive performance can be achieved. In conclusion, we proposed a new way to better integrate both trust and similarity to improve the recommendation performance.

Chapter 6

TrustSVD: A Model-based Approach

The previous chapter has shown that integrating both social trust and user ratings can lead to superior recommendation performance, where the Merge approach is a memory-based approach. Other than memory-based approaches, model-based recommendation methods get more and more popular in the literature due to their superior performance. Although a number of trust-based recommendation models have been studied so far, it is noted that their performance may be inferior to other well-performing ratings-only recommendation models [47]. To provide an explanation, a trust analysis based on four real-world data sets is conducted in Section 6.1. We claim that one possible reason is that these trust-based models focus too much on the utility of trust whereas ignoring the value of ratings for recommendation. Hence, in this chapter we propose a new trust-based model called *TrustSVD* [51]. It takes into consideration both the explicit and implicit influence of ratings and trust simultaneously. Specifically, we build our model upon a state-of-the-art recommendation algorithm SVD++ [49] which inherently involves the explicit and implicit influence of rated items, by further incorporating both the explicit and implicit influence of trusted users on rating predictions. In addition, we further adopt a weighted-λ-regularization technique to help avoid over-fitting. The empirical results on the four real-world data sets show that our approach significantly performs better than other counterparts (ten in total). To our best knowledge, the work reported is the first to extend SVD++ with social trust information.

The rest of this chapter is organized as follows. The trust data from four real-world data sets is analyzed in Section 6.1 where two important observations are concluded. Then, our TrustSVD approach is elaborated in Section 6.2 regarding model formalization and learning,

followed by the empirical evaluation conducted in Section 6.3. Finally, Section 6.4 concludes the present work.

6.1 Trust Analysis

We first introduce the concepts of trust and trust-alike relationships, and then proceed to analyze the influence of trust for rating prediction based on real-world data sets.

6.1.1 Trust vs. Trust-alike Relationships

For ease of exposition, we first classify the social relationships for recommender systems into two categories, i.e., trust and *trust alike*, and then depict their similarities and differences. In this work, we adopt the definition of trust given by Guo et al. [144] as *one's belief towards the ability of others in providing valuable ratings*. It includes a positive and subjective evaluation about other's ability in providing valuable ratings. By contrast, we define the *trust-alike* relationships as the social relationships that are similar with, but weaker (or more noisy) than social trust. The similarities are that both kinds of relationships indicate user preferences to some extent and are thus useful for recommender systems, while the differences are that trust-alike relationships are often weaker in strength and likely to be more noisy. Typical examples are friendship and membership for recommender systems. Although these relationships also indicate that users may have a positive correlation with user similarity, there is no guarantee that such a positive evaluation always exists and that the correlation will be strong. It is well recognized that friendship can be built based on offline relations, such as colleagues and classmates, which does not necessarily share similar preferences. Trust is a complex concept with a number of properties, such as asymmetry and domain dependence [145], which trust-alike relationships may not hold, e.g., friendship is undirected and domain independent.

6.1.2 Data Sets

The four data sets used in our analysis and also our later experiments are: Epinions[6.1], FilmTrust[6.2], Flixster[6.3] and Ciao[6.4]. These four data sets are likely to be the only publicly available data sets

[6.1] trustlet.org/wiki/Epinions_datasets
[6.2] www.librec.net/datasets.html
[6.3] www.cs.sfu.ca/~sja25/personal/datasets/
[6.4] www.public.asu.edu/~jtang20/datasetcode/truststudy.htm

Table 6.1: Statistics of the four data sets

Feature	Epinions	FilmTrust	Flixster	Ciao
# users	40,163	1,508	53,213	7,375
# items	139,738	2,071	18,197	99,746
# ratings	664,824	35,497	409,803	280,391
density	0.051%	1.14%	0.04%	0.03%
# trusters	33,960	609	47,029	6,792
# trustees	49,288	732	47,029	7,297
# trusts	487,183	1,853	655,054	111,781
density	0.029%	0.42%	0.03%	0.23%

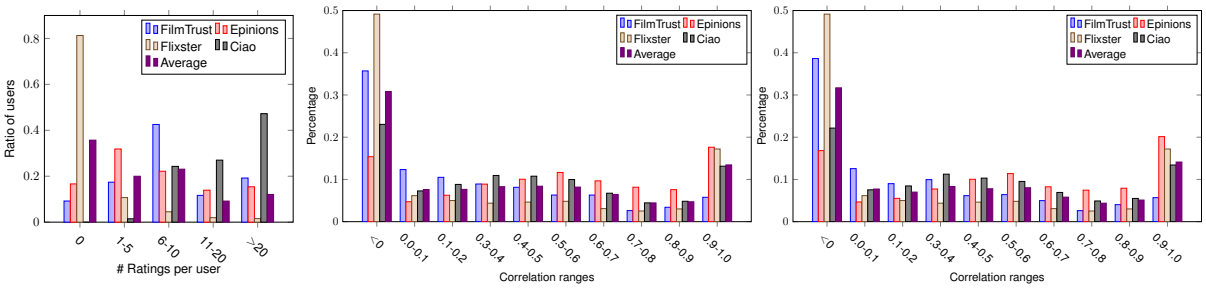
that contain both item ratings and social relationships specified by active users. They are widely used in the evaluation of previous trust-aware recommender systems. In particular, the items in Epinions and Ciao are of great variety, such as electronics, sports, computers, etc, while the items in FilmTrust and Flixster are merely movies. The ratings in Epinions and Ciao are integers from 1 to 5, while those in the other data sets are real values, i.e., [0.5, 4.0] for FilmTrust, [0.5, 5.0] for Flixster both with step 0.5. Users in these data sets can share their item ratings with each other and pro-actively connect with users of similar taste, whereby a social network can be constructed. The data set statistics are illustrated in Table 6.1.

By definition, the social relationships in Epinions and Ciao are trust relationships whereas those in Flixster and FilmTrust are trust-alike relationships. To explain, users in Epinions and Ciao specify others as trustworthy usually based on the evaluation of quality of others' ratings and text reviews. Flixster adopts the concept of friendship *per se* where user relations are symmetric and related with movies only. Although FilmTrust adopts the concept of trust (with original values from 1 to 10), the publicly available data set contains only binary values. Such degrading may cause much noise and thus we classify the relationships as trust alike rather than trust.

For clarity, in this section we refer *trust users* or *trust neighbours* to as the union set of users who trust an active user (i.e., trusters) and of users who are trusted by the active user (i.e., trustees).

6.1.3 Observations

Next we present three observations that are concluded from the four data sets, and underpin the formation of our trust-based model.



(a) (b) (c)

Figure 6.1: (a) The distribution of ratio of users who have issued trust statements w.r.t.the number of ratings that they each have given; (b) The correlations between a user's ratings and those of her out-going trusted neighbours (i.e., trustees) across all the data sets; (c) The correlations between a user's ratings and those of her in-coming trusting neighbours (i.e., trusters) across all the data sets.

Observation 1 *Trust information is very sparse, yet is complementary to rating information.*

On one hand, as shown in Table 6.1, the density of trust is much smaller than that of ratings in Epinions, FilmTrust and Flixster whereas trust is only denser than ratings in Ciao. Both ratings and trust are very sparse in general across all the data sets. In this regard, a trust-aware recommender system that focuses too much on trust (rather than rating) utility is likely to achieve only marginal gains in recommendation performance, which has been demonstrated in Section 5. As explained earlier, even the latest trust-based model cannot always beat the baseline approaches which generate predictions solely based on ratings. In fact, the existing trust-based models consider only the explicit influence of ratings. That is, the utility of ratings is not well exploited. In addition, the sparsity of explicit trust also implies the importance of involving implicit trust in collaborative filtering. Therefore, a better way may stress that both the influence of user trust and item ratings should be taken into account for rating prediction.

On the other hand, trust information is complementary to the rating information. Figure 6.1 (a) illustrates the distribution of ratio of users who have specified others as trusted friends (users) with respect to the number of ratings that each of these users has given. It shows that: (1) A portion of users have not rated any items but are socially connected with other users, e.g., 9.20% in FilmTrust, 16.65% in Epinions and up to 81.28% in Flixster[6.5]. (2)

[6.5] All the users in the Ciao data set have rated at least one item.

For the *cold-start* users who have rated few items (less than 5 in our case), trust information can provide a complementary source of information with ratio greater than 10% on average. (3) The *warm-start* users who have rated a lot of items (e.g., > 20) do not always specify many other users as trustworthy (12% on the average). As a consequence, although having distinct distributions across the different data sets, trust can be a complementary information source to item ratings for recommender systems.

This observation motivates us to consider both the explicit and implicit influence of item ratings and user trust, making better and more use of them to resolve the data sparsity and cold start problems.

Observation 2 *A user's ratings have a **weakly positive correlation** with the average of her **outgoing** social neighbours under the concept of trust-alike relationships, and a **strongly positive correlation** under the concept of trust relationships.*

Although a user's rating to a certain item is mainly determined by the intrinsic attributes (or properties, features) of the item in question and how she appreciates these features, some extrinsic attributes may also have a non-negligible influence on the user's ratings. In this work, we focus on the influence of social trust in rating prediction, i.e., the influence of trust neighbours on an active user's rating for a specific item, a.k.a.*social influence*. A graphical explanation is given in Figure (a). Briefly, user u trusts user v, and user v has rated item j by giving a rating $r_{v,j}$. Then, user u may consider the ratings of her trustees when giving her own rating $r_{u,j}$. Yang et al. [48] have also shown that trusted users will affect users' ratings in their model.

To have an intuitive comprehension of trustees' influence, we calculate the Pearson correlation coefficient (PCC) between a user's ratings and the average of her social neighbours. The results are presented in Figure 6.1 (b), indicating that: (1) A weakly positive correlation is observed between a user's ratings and the average of the social neighbours in FilmTrust (mean 0.183) and Flixster (0.063). The distributions of the two data sets are similar. (2) Under the concept of trust relationships, on the contrary, a user's ratings are strongly and positively correlated with the average of trusted neighbours. Specifically, a large portion (17.63% in Epinions, 13.14% in Ciao) of user correlations are in the range of $[0.9, 1.0]$, and (resp. 54.70%, 39.14%) of user correlations are greater than 0.5. The average correlation is 0.446 in Epinions, and 0.322 in Ciao. Since PCC values are in the range of $[-1, 1]$, values of 0.446 and 0.322 indicate decent correlations. In the social networks with relatively weak trust-alike relationships,

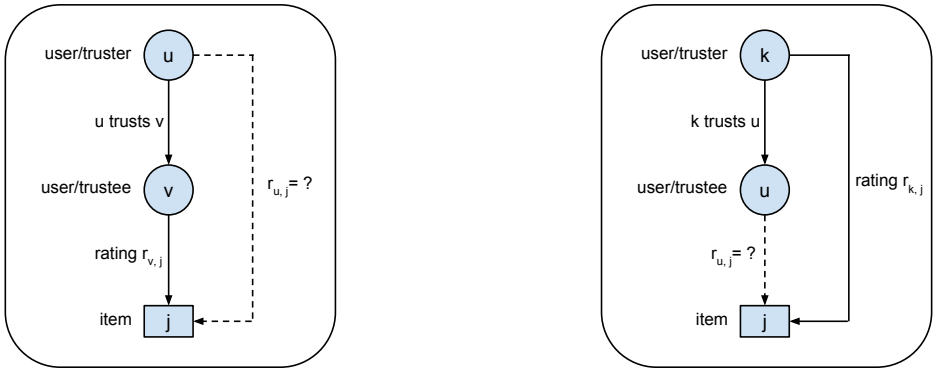
implicit influence (i.e., binary relationships) may be more indicative than explicit (but noisy) values for recommendations. In addition, most online social networks do not adopt the concept of trust relationships, but relatively weak trust-alike relationships. Hence, a trust-based model that ignores the implicit influence of item ratings and user trust may lead to deteriorated performance if being applied to such cases. We claim that a good trust-based model should function well not only for strong trust relationships, but also for relatively weak trust-alike ones.

The second observation suggests that incorporating both the explicit and implicit influence of item ratings and user trust may promote the generality of a trust-based model to both trust and trust-alike social relationships.

Observation 3 *A user's ratings have a* ***weakly positive correlation*** *with the average of her* ***in-coming*** *social neighbours under the concept of trust-alike relationships, and a* ***strongly positive correlation*** *under the concept of trust relationships.*

In social networks, a user may pro-actively connect to a number of social friends, and may also be connected by some other users. Thus, the social influence of one's ratings may flow in both directions. In other words, a user's rating is influenced not just by her trustees, but also by the users who trust her (i.e., her trusters). Yang et al. [48] have also indicated that trusting users have an impact on users' rating prediction. Yao et al. [107] design and combine both truster and trustee regularizers in a unified recommendation model, indicating the value of influence of trusters. Figure (b) gives an illustrative presentation. Specifically, user k trusts user u, and gives a rating to the target item j the action of which may further influence user u's rating on the same item. Similar with the last observation, we calculate the correlations between a user's ratings and the average of her trusters' ratings. The results are presented in Figure 6.1 (c). Note that the distribution in Flixster is the same with that in Figure 6.1 (b). The reason is that the friendships in Flixster are symmetric and undirected.

Compared with Figure 6.1 (b), similar distributions of user correlations can be observed, and the differences in each correlation range are relatively small. For example, there are 30.82% negative user correlations in Figure 6.1 (b) while the percentage in Figure 6.1 (c) is around 31.69%. Therefore, a similar observation can be drawn from the empirical results, i.e., users have a weakly (strongly) positive correlation with the average of her in-coming social neighbours under the concept of trust-alike (trust) relationships.

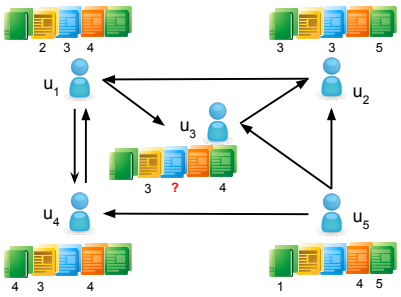


(a) (b)

Figure 6.2: The influence of (a) trustees v and (b) trusters k on the rating prediction for the active user u and target item j

user-item rating matrix

	i_1	i_2	i_3	i_4	i_5
u_1		2	3	4	
u_2	3		3		5
u_3		3	?		4
u_4	4	3		4	
u_5		1	4		5



user-user trust matrix

	u_1	u_2	u_3	u_4	u_5
u_1			1	1	
u_2	1				
u_3		1			1
u_4	1				
u_5			1	1	

Figure 6.3: A social rating network with user-item rating and user-user trust matrices

The third observation implies that the influence of trusters (in rating prediction) may be comparable with that of trustees, and thus may also provide added value to item ratings. Our approach presented next is built upon these three observations.

6.2 The TrustSVD Model

In this section, we first mathematically define the recommendation problem in social rating networks, and then introduce the TrustSVD model in detail.

6.2.1 Problem Definition

As illustrated in Figure 6.3, in social rating networks a user can label (add) other users as trusted friends and thus form a social network. Trust is not symmetric; for example, users u_1 trusts u_3 but u_3 does not specify user u_1 as trustworthy. Besides, users can rate a set of items using a number of rating values, e.g., integers from 1 to 5. These items could be products, movies, music, etc. of interest. The recommendation problem in this work is to predict the rating that a user will give to an unknown item, for example, the value that user u_3 will give to item i_3, based on both a user-item rating matrix and a user-user trust matrix. Other well-recognized recommendation problems include for example top-N item recommendation.

Suppose that a recommender system includes m users and n items. Let $R = [r_{u,i}]_{m \times n}$ denote the user-item rating matrix, where each entry $r_{u,i}$ represents the rating given by user u on item i. For clarity, we preserve symbols u, v for users, and i, j for items. Since a user only rated a small portion of items, the rating matrix R is only partially observed and oftentimes very sparse. Let $I_u = \{i | r_{u,i} \neq 0\}$ denote the set of items rated by user u. Let p_u and q_i be a d-dimensional latent feature vector of user u and item i, respectively. The essence of matrix factorization is to find two low-rank matrices: user-feature matrix $P \in \mathbb{R}^{d \times m}$ and item-feature matrix $Q \in \mathbb{R}^{d \times n}$ that can adequately recover the rating matrix R, i.e., $R \approx P^\top Q$, where $P^\top$ is the transpose of matrix P. The underlying assumption is that both users and items can be characterized by a small number of features. Hence, the rating on item j for user u can be predicted by the inner product of user-specific vector p_u and item-specific vector q_j, i.e., $\hat{r}_{u,j} = q_j^\top p_u$. In this regard, the main task of recommendations is to predict the rating $\hat{r}_{u,j}$ as close as possible to the ground truth $r_{u,j}$. Formally, we can learn the user- and item-feature matrices by minimizing the following loss (objective) function:

$$\mathcal{L}_r = \frac{1}{2} \sum_u \sum_{j \in I_u} \left(q_j^\top p_u - r_{u,j} \right)^2 + \frac{\lambda}{2} \Big(\sum_u \| p_u \|_F^2 + \sum_j \| q_j \|_F^2 \Big),$$

where $\| \cdot \|_F$ denotes the Frobenius norm, and λ is a parameter to control model complexity and to avoid over-fitting.

On the other hand, suppose that a social network is represented by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ includes a set of m nodes (users) and $\mathcal{E}$ represents the directed trust relationships among users. We can use the adjacency matrix $T = [t_{u,v}]_{m \times m}$ to describe the structure of

edges $\mathcal{E}$, where $t_{u,v}$ indicates the extent to which users u trusts v. Usually, only binary values are used, i.e., $t_{u,v} = 1$ means that user u trusts user v whereas $t_{u,v} = 0$ indicates the non-trust relationship. Similarly, the trust matrix T is also very sparse. We denote p_u and w_v as the d-dimensional latent feature vector of truster u and trustee v, respectively. We limit the trusters in the trust matrix and the active users in the rating matrix to share the same user-feature space in order to bridge them together. Hence, we have truster-feature matrix $P^{d\times m}$ and trustee-feature matrix $W^{d\times m}$. By employing the low-rank matrix approximation, we can recover the trust matrix by $T \approx P^\top W$. Thus, a trust relationship can be predicted by the inner product of a truster-specific vector and a trustee-specific vector $\hat{t}_{u,v} = w_v^\top p_u$. The matrices P and W can be learned by minimizing the following loss function:

$$\mathcal{L}_t = \frac{1}{2}\sum_u \sum_{v\in T_u^+} \left(w_v^\top p_u - t_{u,v}\right)^2 + \frac{\lambda}{2}\Big(\sum_u \|p_u\|_F^2 + \sum_v \|w_v\|_F^2\Big),$$

where T_u^+ is the set of users trusted by user u, i.e., the set of out-going trusted users.

In summary, by mapping both rating matrix and trust matrix into the same d-dimensional space, we can link the two kinds of information together and thus aim to predict an item's rating $\hat{r}_{u,j}$ as accurate as possible.

6.2.2 Model Formulation

In line with the three observations of the previous section, our TrustSVD model is built on top of a state-of-the-art model known as SVD++ proposed by Koren [49]. The rationale behind SVD++ is to take into consideration user/item biases and the influence of rated items other than user/item-specific vectors on rating prediction. Formally, the rating for user u on item j is predicted by:

$$\hat{r}_{u,j} = b_u + b_j + \mu + q_j^\top \big(p_u + |I_u|^{-\frac{1}{2}} \sum_{i\in I_u} y_i\big),$$

where b_u, b_j represent the rating bias of user u and item j, respectively; μ is the global average rating; and y_i denotes the implicit influence of items rated by user u in the past on the ratings of unknown items in the future. Thus, user u's feature vector can be also represented by the set of items she rated, and finally modeled as $\big(p_u + |I_u|^{-\frac{1}{2}} \sum_{i\in I_u} y_i\big)$ rather than simply as p_u. Koren [49] has shown that integrating implicit influence of ratings can well improve predictive accuracy. Previously, we have stressed the importance of trust influence for better

recommendations, and its potential to be generalized to trust-alike relationships. Hence, we can enhance the trust-unaware SVD++ model by incorporating both the explicit and implicit influence of trust, described as follows.

Implicit Influence of Trusted Users. Figure (a) shows that the trusted users of an active user have an effect on rating prediction for a certain item. We take into account this effect by modelling user preference in the same manner as rated items, given by:

$$\hat{r}_{u,j} = b_u + b_j + \mu + q_j^\top \Big(p_u + |I_u|^{-\frac{1}{2}} \sum_{i \in I_u} y_i + |T_u^+|^{-\frac{1}{2}} \sum_{v \in T_u^+} w_v \Big),$$

where w_v is the user-specific latent feature vector of users (trustees) trusted by user u, and thus $q_j^\top w_v$ can be explained by the ratings predicted by the trusted users, i.e., the influence of trustees on the rating prediction. In other words, the inner product $q_j^\top w_v$ indicates how trusted users influence user u's rating on item j. An intuitive understanding has been illustrated in Figure (a). Similar to ratings, a user's feature vector can be interpreted by the set of users whom she trusts, i.e., $|T_u^+|^{-\frac{1}{2}} \sum_{v \in T_u^+} w_v$. Therefore, a user u is further modeled by $\big(p_u + |I_u|^{-\frac{1}{2}} \sum_{i \in I_u} y_i + |T_u^+|^{-\frac{1}{2}} \sum_{v \in T_u^+} w_v\big)$ in the social rating networks, considering the influence of both rated items and trusted users.

Implicit Influence of Trusting Users. Figure (b) shows that the trusting users of an active user can also influence the rating prediction for a certain item. In fact, *Observation 3* has indicated that such influence may be comparable with that of trusted users. Similarly, the effect can be considered by modelling user preference, given by:

$$\hat{r}_{u,j} = \; b_u + b_j + \mu + q_j^\top \Big(p_u + |I_u|^{-\frac{1}{2}} \sum_{i \in I_u} y_i + |T_u^-|^{-\frac{1}{2}} \sum_{k \in T_u^-} p_k \Big),$$

where T_u^- is the set of users who trust user u, i.e., the set of her trusters. Thus, $q_j^\top p_k$ can be explained by the ratings predicted by the trusting users, i.e., the influence of trusters on the predictions. Similarly, the inner product $q_j^\top p_k$ indicates how trusting users k influence user u's rating on item j. An intuitive understanding has been illustrated in Figure (b). Similar to ratings, a user's feature vector can be interpreted by the set of users whom trust her, i.e., $|T_u^-|^{-\frac{1}{2}} \sum_{k \in T_u^-} p_k$. Therefore, a user u is further modeled by $\big(p_u + |I_u|^{-\frac{1}{2}} \sum_{i \in I_u} y_i + |T_u^-|^{-\frac{1}{2}} \sum_{k \in T_u^-} p_k\big)$ in the social rating networks, considering the influence of both rated items and trusting users.

Combinational Implicit Trust Influence. Therefore, the implicit influence of trust neighbours on rating prediction consists of two parts: the influence of both trustees and trusters. To consider both cases, a natural and straightforward way is to linearly combine the two kinds of implicit trust influence, given by:

$$\hat{r}_{u,j} = b_u + b_j + \mu + q_j^\top \Big(p_u + |I_u|^{-\frac{1}{2}} \sum_{i \in I_u} y_i + \alpha |T_u^+|^{-\frac{1}{2}} \sum_{v \in T_u^+} w_v + (1-\alpha) |T_u^-|^{-\frac{1}{2}} \sum_{k \in T_u^-} p_k \Big), \quad (6.1)$$

where α controls the importance of influence of trustees in rating prediction. In the case of undirected social relationships (e.g., friendship in Flixster), T_u^+ will be equivalent with T_u^-, and thus the linear combination ensures that our model can be applied to both trust and trust-alike relationships.

With the consideration of implicit trust influence, the objective function to minimize is then given as follows:

$$\mathcal{L} = \frac{1}{2} \sum_u \sum_{j \in I_u} \big(\hat{r}_{u,j} - r_{u,j} \big)^2 + \frac{\lambda}{2} \Big(\sum_u b_u^2 + \sum_j b_j^2 + \sum_u \|p_u\|_F^2 + \sum_j \|q_j\|_F^2 + \sum_i \|y_i\|_F^2 + \sum_v \|w_v\|_F^2 \Big),$$

where $\hat{r}_{u,j}$ is the prediction computed by Equation 6.1. To reduce the model complexity, we use the same regularization parameter λ for all the variables. Finer control and tuning can be achieved by assigning separate regularization parameters to different variables, though it may result in great complexity in model learning, and in comparison with other matrix factorization models.

Explicit Trust Influence. In addition, as explained earlier, we constrain that the user-specific vectors decomposed from the rating matrix and those decomposed from the trust matrix share the same feature space in order to bridge both matrices together. In this way, these two types of information can be exploited in a unified recommendation model. Specifically, we can regularize the user-specific vectors p_u by recovering the social relationships with other users. The new objective function (without the other regularization terms) is given by:

$$\mathcal{L} = \frac{1}{2} \sum_u \sum_{j \in I_u} \big(\hat{r}_{u,j} - r_{u,j} \big)^2 + \frac{\lambda_t}{2} \sum_u \Big(\alpha \sum_{v \in T_u^+} \big(\hat{t}_{u,v} - t_{u,v} \big)^2 + (1-\alpha) \sum_{k \in T_u^-} \big(\hat{t}_{k,u} - t_{k,u} \big)^2 \Big),$$

where $\hat{t}_{u,v} = w_v^\top p_u$ is the predicted trust between users u and v computed by the inner product of truster and trustee vectors, i.e., $w_v^\top p_u$. Similarly, $\hat{t}_{k,u} = w_u^\top p_k$ is the predicted trust for user k towards user u, and λ_t controls the degree of trust regularization.

Adaptive Regularization. Further, as suggested by Yang et al. [48], a technique called weighted-λ-regularization can be used to help avoid over-fitting when learning model parameters. In particular, they consider more penalties for the users who rated more items and for the items which received more ratings. However, we argue that such consideration may be of less help to avoid over-fitting since there are a large number of training examples regarding the heavy users and niche items. Instead, in this work we adopt a distinct strategy that the popular users and items should be less penalized (due to smaller chance to be over-fitted), and cold-start users and *niche items* (those receiving few ratings) should be more regularized (due to greater chance to be over-fitted). As a result, the new loss function to minimize is obtained as follows:

$$
\begin{aligned}
\mathcal{L} = & \tfrac{1}{2} \sum_{u} \sum_{j \in I_u} \left(\hat{r}_{u,j} - r_{u,j}\right)^2 \\
& + \tfrac{\lambda_t}{2} \sum_{u} \Big(\alpha \sum_{v \in T_u^+} \left(\hat{t}_{u,v} - t_{u,v}\right)^2 + (1-\alpha) \sum_{k \in T_u^-} \left(\hat{t}_{k,u} - t_{k,u}\right)^2 \Big) \\
& + \tfrac{\lambda}{2} \sum_{u} |I_u|^{-\frac{1}{2}} b_u^2 + \tfrac{\lambda}{2} \sum_{j} |U_j|^{-\frac{1}{2}} b_j^2 \\
& + \sum_{u} \Big(\tfrac{\lambda}{2} |I_u|^{-\frac{1}{2}} + \tfrac{\lambda_t}{2} \left(\delta(\alpha)|T_u^+|^{-\frac{1}{2}} + \delta(1-\alpha)|T_u^-|^{-\frac{1}{2}}\right) \Big) \|p_u\|_F^2 \\
& + \tfrac{\lambda}{2} \sum_{j} |U_j|^{-\frac{1}{2}} \|q_j\|_F^2 + \tfrac{\lambda}{2} \sum_{i} |U_i|^{-\frac{1}{2}} \|y_i\|_F^2 \\
& + \tfrac{\lambda}{2} \sum_{u} \sum_{v \in T_u^+} \delta(\alpha) |T_v^+|^{-\frac{1}{2}} \|w_v\|_F^2 \\
& + \tfrac{\lambda}{2} \sum_{u} \sum_{k \in T_u^-} \delta(1-\alpha) |T_k^-|^{-\frac{1}{2}} \|p_k\|_F^2,
\end{aligned}
\tag{6.2}
$$

where U_j, U_i are the set of users who rate items j and i, respectively; and $\delta(x)$ is an indicator function which equals 1 if $x > 0$, and 0 otherwise. Since the active user has rated a number of items and are socially connected with other trust neighbours, the penalization on user-specific vector p_u takes into account both cases.

6.2.3 Model Learning

To obtain a local minimization of the objective function given by Equation 6.2, we perform the following gradient descents on $b_u, b_j, p_u, q_j, y_i, w_v$ and p_k across all the users and items in a training data set.

$$
\begin{aligned}
&\frac{\partial \mathcal{L}}{\partial b_u} = \sum_{j \in I_u} e_{u,j} + \lambda |I_u|^{-\frac{1}{2}} b_u \\
&\frac{\partial \mathcal{L}}{\partial b_j} = \sum_{u \in U_j} e_{u,j} + \lambda |U_j|^{-\frac{1}{2}} b_j \\
&\frac{\partial \mathcal{L}}{\partial p_u} = \sum_{j \in I_u} e_{u,j} q_j + \lambda_t \alpha \sum_{v \in T_u^+} e_{u,v} w_v + \Big(\lambda |I_u|^{-\frac{1}{2}} + \lambda_t \big(\delta(\alpha) |T_u^+|^{-\frac{1}{2}} + \delta(1-\alpha) |T_u^-|^{-\frac{1}{2}} \big) \Big) p_u \\
&\frac{\partial \mathcal{L}}{\partial q_j} = \sum_{u \in U_j} e_{u,j} \Big(p_u + |I_u|^{-\frac{1}{2}} \sum_{i \in I_u} y_i + \alpha |T_u^+|^{-\frac{1}{2}} \sum_{v \in T_u^+} w_v + (1-\alpha) |T_u^-|^{-\frac{1}{2}} \sum_{k \in T_u^-} p_k \Big) + \lambda |U_j|^{-\frac{1}{2}} q_j \\
&\forall i \in I_u, \frac{\partial \mathcal{L}}{\partial y_i} = \sum_{j \in I_u} e_{u,j} |I_u|^{-\frac{1}{2}} q_j + \lambda |U_i|^{-\frac{1}{2}} y_i \\
&\forall v \in T_u^+, \frac{\partial \mathcal{L}}{\partial w_v} = \sum_{j \in I_u} e_{u,j} \alpha |T_u^+|^{-\frac{1}{2}} q_j + \lambda_t \alpha e_{u,v} p_u + \lambda \delta(\alpha) |T_v^+|^{-\frac{1}{2}} w_v \\
&\forall k \in T_u^-, \frac{\partial \mathcal{L}}{\partial p_k} = \sum_{j \in I_u} e_{u,j} (1-\alpha) |T_u^-|^{-\frac{1}{2}} q_j + \lambda_t (1-\alpha) e_{k,u} w_u + \lambda \delta(1-\alpha) |T_k^-|^{-\frac{1}{2}} p_k
\end{aligned}
\tag{6.3}
$$

where $e_{u,j} = \hat{r}_{u,j} - r_{u,j}$ indicates the rating prediction error for user u on item j, and $e_{u,v} = \hat{t}_{u,v} - t_{u,v}$ is the trust prediction error for user u towards trustee v as well as $e_{k,u} = \hat{t}_{k,u} - t_{k,u}$ for truster k towards user u.

The pseudocode for model learning and updating is given in Algorithm 2. To explain, several arguments are taken as input, including user-item rating matrix R, user-user trust matrix T, regularization parameters λ and λ_t, and the initial learning rate γ. First, we randomly initialize the decomposed vectors and matrices with small values (line 1). Then, we keep training the model until the loss function is converged (line 2). Specifically, we compute variable gradients according to Equation 6.3 (line 3), and then update variables by the gradient descent method (lines 4-10). Finally, we return the learned vectors and matrices as output (line 11).

6.2.4 Complexity Analysis

The computational time of learning the TrustSVD model is mainly taken by evaluating the objective function $\mathcal{L}$ and its gradients against feature vectors (variables). The time to compute the objective function $\mathcal{L}$ is $O(d|R| + d|T|)$, where d is the dimensionality of the feature space, and $|R|, |T|$ refer to the number of observed entries. Due to the sparsity of rating and trust matrices, the values will be much smaller than the matrix cardinality. The computational complexities for gradients $\frac{\partial \mathcal{L}}{\partial b_u}$, $\frac{\partial \mathcal{L}}{\partial b_j}$, $\frac{\partial \mathcal{L}}{\partial p_u}$, $\frac{\partial \mathcal{L}}{\partial q_j}$, $\frac{\partial \mathcal{L}}{\partial y_i}$, $\frac{\partial \mathcal{L}}{\partial w_v}$, $\frac{\partial \mathcal{L}}{\partial p_k}$ in Equation 6.3 are $O(d|R|), O(d|R|), O(d|R| + d|T|), O(d|R| + d|T|), O(d|R|k)$, $O(d|R|p^+ + d|T|p^+)$ and $O(d|R|p^- + d|T|p^-)$, where k, p^+, p^- are the average number of ratings received by an item, trust statements given and received by a user, respectively. Hence, the overall computational

Input : $R, T, d, \lambda, \lambda_t, \gamma$ (learning rate)
Output: Rating predictions $\hat{r}_{u,j}$

Initialize vectors B_u, B_j and matrices P, Q, Y, W with small and random values in $(0, 1)$;
while $\mathcal{L}$ *not converged* **do**
  compute gradients according to Equation 6.3;
  $b_u \leftarrow b_u - \gamma \frac{\partial \mathcal{L}}{\partial b_u}, u = 1 \dots m$
  $b_j \leftarrow b_j - \gamma \frac{\partial \mathcal{L}}{\partial b_j}, j = 1 \dots n$
  $p_u \leftarrow p_u - \gamma \frac{\partial \mathcal{L}}{\partial p_u}, u = 1 \dots m$
  $q_j \leftarrow q_j - \gamma \frac{\partial \mathcal{L}}{\partial q_j}, j = 1 \dots n$
  $\forall i \in I_u, \quad y_i \leftarrow y_i - \gamma \frac{\partial \mathcal{L}}{\partial y_i}, u = 1 \dots m$
  $\forall v \in T_u^+, w_v \leftarrow w_v - \gamma \frac{\partial \mathcal{L}}{\partial w_v}, u = 1 \dots m$
  $\forall k \in T_u^-, p_k \leftarrow p_k - \gamma \frac{\partial \mathcal{L}}{\partial p_k}, u = 1 \dots m$
return B_u, B_j, P, Q, Y, W;

Algorithm 2: Learning in the TrustSVD model

complexity in one iteration is $O(d|R|c + d|T|c)$, where $c = \max(p^+, p^-, k)$. Due to $c \ll |R|$ or $|T|$, the overall computational time is linear with respect to the number of observations in the rating and trust matrices. To sum up, our model has potential to scale up to large-scale data sets.

6.2.5 Insights into the TrustSVD Model

The key idea behind the TrustSVD model is to take into account both explicit and implicit influences of item ratings and social trust information when predicting users' ratings for unknown items. Specifically, for ratings, the explicit information is the rating values which are approximated by the inner product $q_j^\top p_u$ of user- and item-specific latent feature vectors, while the implicit influence is represented by $q_j^\top y_i$ regarding the effect of rated items by the active users. Similarly, for trust statements, the explicit information is the trust values that are predicted by the inner product $w_v^\top p_u$ of truster- and trustee-specific latent feature vectors. To bridge the rating and trust matrices together, we limit the user/truster vector to be the same p_u. The implicit influence of trust neighbours can be further split into two parts: trustees' influence is modelled by the inner product $q_j^\top w_v$ while trusters' influence is given by the inner product $q_j^\top p_k$. Since the state-of-the-art model SVD++ naturally incorporates the implicit influence of item ratings,

we build on top of this model by further incorporating the implicit influence of trust neighbours as well as the explicit one. Therefore, compared with other models, our TrustSVD model enables more information (both implicit and explicit) for rating prediction, and hence can lead to better recommendation performance.

In the cold-start situations where users may have only rated a few items, the decomposition of trust matrix can help to learn more reliable user-specific latent feature vectors than ratings-only matrix factorization. In the extreme case where there are no ratings at all for some users, Equation 6.3 ensures that the user-specific vector can be trained and learned from the trust matrix. In this regard, incorporating trust in a matrix factorization model can alleviate the cold start problem. By considering both explicit and implicit influence of trust rather than either one, our model can better utilize trust to further mitigate the concerned issues.

6.3 Evaluation

In this section, we conduct a series of experiments in order to investigate: (1) the impact of parameters λ_t and α on our TrustSVD model; (2) the performance of our model and its ability to cope with the cold-start situations comparing with other counterparts; (3) the performance of the proposed model on users with different trust degrees in comparison with other trust-based models.

6.3.1 Experimental Settings

Data Sets. The four data sets presented in Table 6.1 are used.

Cross-validation. We use 5-fold cross-validation for learning and testing. Specifically, we randomly split each data set into five folds and in each iteration four folds are used as the training set and the remaining fold as the test set. Five iterations will be conducted to ensure that all folds are tested. Then, the average test performance is given as the final result.

Testing Views. Two data set views are created for testing. First, the *All* view indicates that all ratings are used as the test set. Second, the *Cold Start* view means that only the users who rate less than five items will be involved in the test set. Similar testing views are also defined and used in [20, 48]. In the *Cold Start* view, 5-fold cross validation is still used but we only care about the performance for cold-start users.

Evaluation Metrics. We adopt two well-known metrics to evaluate predictive accuracy, namely mean absolute error (MAE) and root mean square error (RMSE)[6.6], defined by:

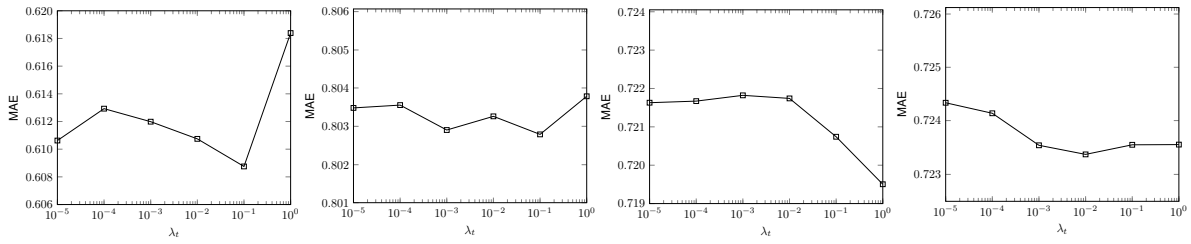
$$\text{MAE} = \frac{\sum_{u,j} |\hat{r}_{u,j} - r_{u,j}|}{N}, \qquad \text{RMSE} = \sqrt{\frac{\sum_{u,j} (\hat{r}_{u,j} - r_{u,j})^2}{N}}, \tag{6.4}$$

where N is the number of test ratings. Smaller values of MAE and RMSE indicate better predictive accuracy. Besides, since RMSE puts relatively high weights on large errors and all comparison models (except the baselines) adopt the least square errors as loss function, RMSE is more appropriate than MAE to measure the predictive performance for our work. In addition, the larger the difference between MAE and RMSE, the greater the variance of predictive errors.

Comparison Methods. Up to ten recommendation models are used and compared in our experiments listed as follows.

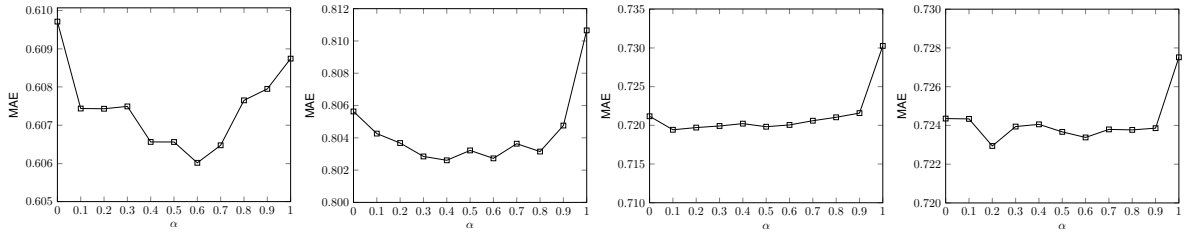
- **UAvg** and **IAvg** are baselines that predict a user's rating by the average of her historical ratings, and the average of ratings received by the target item, respectively.
- **PMF** is a probabilistic matrix factorization model proposed by [62]. It is a basic matrix factorization model.
- **RSTE** [83], **SoRec** [104] and **SoReg** [22] are earlier trust-based recommendation models as described in Section 2.3. Ma et al. [22, 146] adopt social trust as a regularizer only whereas our approach further integrates social trust into the factorization of users.
- **SocialMF** (or **SoMF** for short) [63], **TrustMF** (or **TMF** for short) [48], **Fang's** [47] are latest and state-of-the-art trust-based models that are reported to achieve better performance than simple baselines and other counterparts [48, 47].
- **SVD++** [49] is a state-of-the-art recommendation method merely based on ratings, and also adopted as a key comparison method in Fang et al. [47].
- **TrustSVD** (or **TSVD** for short) is our proposed model built upon the SVD++ model by incorporating both the explicit and implicit influence of social trust.

[6.6] All matrix factorization-based approaches can easily produce 100% rating coverage, and thus we do not adopt rating coverage and F1 to measure the recommendation performance as we did in Section 5.



(a) FilmTrust (b) Epinions (c) Flixster (d) Ciao

Figure 6.4: The effect of parameter trust regularization λ_t across all the data sets [$d = 10$]



(a) FilmTrust (b) Epinions (c) Flixster (d) Ciao

Figure 6.5: The effect of parameter trustee's importance α across all the data sets [$d = 10$]

Parameter Settings. The optimal experimental settings for all the models are determined either by our experiments or suggested by previous works. Specifically, the common settings are $\lambda = 0.001$, and the number of latent features $d = 5/10$, the same as all the previous trust-based models. The other settings are: (1) RSTE: $\alpha = 0.4$ for Epinions, and 1.0 for the others; (2) SoRec: $\lambda_c = 0.1, 1.0, 0.001, 0.01$ corresponding to FilmTrust, Epinions, Flixster and Ciao, respectively; (3) SoReg: $\beta = 1.0$ for Flixster and 1.0 for the others; (4) SocialMF, TrustMF: $\lambda_t = 1$; (5) SVD++: $\lambda = 0.1, 0.35, 0.03, 0.1$ (resp.); (6) TrustSVD: $\lambda = 0.6$ for FilmTrust, $\lambda = 1$ for Epinions, $\lambda = 0.6$ for Flixster, and $\lambda = 0.1$ for Ciao. Parameters λ_t and α will be determined in next subsection.

6.3.2 Impact of Parameters λ_t and α

Other than the parameter λ, two more parameters are used in our method, namely parameter λ_t for the importance of trust regularization and parameter α for the relative importance of influence of trustees. To determine their values for different data sets, we first fix the value

of one parameter, and then adjust the values of the other to search the best parameter settings. Specifically, we tune the parameter λ_t in the range $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}\}$ while fixing $\alpha = 0.5$, i.e., equally importance of both influence of trustees and trusters. The results are illustrated in Figure 6.5 in terms of MAE when $d = 10$. The other settings (e.g, $d = 5$ or in terms of RMSE) show similar performance trends. Nevertheless, MAE is adopted since it gives clearer changes of recommendation performance. The results clearly indicate that a proper value of λ_t for different data sets can help improve the recommendation performance. Generally, a value 0.1 would give relatively fair performance if necessary.

After determining λ_t's values, we proceed to tune the value of parameter α in [0, 1] with step 0.1. Parameter $\alpha = 1$ indicates that only the influence of trustees is taken into account while $\alpha = 0$ means that only the influence of trusters is considered in our method. The results are presented in Figure 6.5. Generally, the performance of $\alpha = 0$ and $\alpha = 1$ is much worse than the performance of other values. In other words, a proper combination of both the influence of trusters and trustees lead to better recommendation performance. Although the best value of α that reaches the superior performance may vary in different data sets, a value of 0.6 in general is a fair setting.

6.3.3 Comparison with Other Models

The experimental results are presented in Tables 6.2 and 6.3, corresponding to the testing views of *All* and *Cold Start*, respectively. For all the comparison methods in the testing view of *All*, SVD++ outperforms the other comparison methods in FilmTrust and Epinions, and UAvg performs the best in Flixster. This implies that these trust-based approaches cannot always beat other well-performing ratings-only approaches, and even simple baselines in trust-alike networks (i.e., FilmTrust and Flixster). Only in Ciao, trust-based approach (SocialMF) gives the best performance. On the contrary, our approach TrustSVD is consistently superior to the best approach among the others across all the data sets. Although the percentage of relative improvements are small (around 3.17% in RMSE on the average), Koren [76] has pointed out that even small improvements in MAE and RMSE may lead to significant differences of recommendations in practice. For example, the Netflix prize[6.7] competition offered 1M dollar for 10% improvements in term of RMSE.

[6.7] http://www.netflixprize.com/

Table 6.2: Performance comparison in the testing view of 'All', where * indicates the best performance among all the other methods, and column 'Improve' indicates the percentage of improvements that our approach TrustSVD achieves relative to the * results. For each feature dimensionality d, the first row represents MAE values while the second indicates RMSE values.

All	**UAvg**	**IAvg**	**PMF**	**RSTE**	**SoRec**	**SoReg**	**SoMF**	**TMF**	**SVD++**	**TSVD**	**Improve**
FilmTrust	0.636	0.725	0.714	0.628	0.628	0.674	0.638	0.631	**0.613***	**0.607**	0.98%
$d = 5$	0.823	0.927	0.949	0.810	0.810	0.878	0.837	0.810	**0.804***	**0.791**	1.62%
	0.636	0.725	0.735	0.640	0.638	0.668	0.642	0.631	**0.611***	**0.605**	0.98%
$d = 10$	0.823	0.927	0.968	0.835	0.831	0.875	0.844	0.819	**0.802***	**0.789**	1.62%
Epinions	0.930	0.928	0.979	0.950	0.882	0.994	0.825	0.818	**0.818***	**0.803**	1.83%
$d = 5$	1.203	1.094	1.290	1.196	1.114	1.315	1.070	1.069	**1.057***	**1.043**	1.32%
	0.930	0.928	0.909	0.958	0.884	0.932	0.826	0.819	**0.818***	**0.803**	1.83%
$d = 10$	1.203	1.094	1.197	1.278	1.142	1.232	1.082	1.095	**1.057***	**1.044**	1.23%
Flixster	**0.729***	0.858	0.814	0.751	0.750	0.820	0.770	0.890	0.794	**0.719**	1.37%
$d = 5$	**0.979***	1.088	1.076	0.975	0.974	1.087	0.994	1.146	1.062	**0.942**	3.88%
	0.729*	0.858	0.769	0.784	0.785	0.785	0.784	1.116	0.821	**0.719**	1.37%
$d = 10$	**0.979***	1.088	1.009	1.015	1.018	1.034	1.009	1.441	1.091	**0.941**	3.88%
Ciao	0.781	0.760	0.920	0.767	0.765	0.899	0.749	**0.742***	0.752	**0.723**	2.56%
$d = 5$	1.031	1.026	1.206	1.020	1.013	1.183	**0.981***	0.983	1.013	**0.956**	2.55%
	0.781	0.760	0.822	0.763	0.761	0.815	**0.749***	0.753	0.748	**0.723**	3.47%
$d = 10$	1.031	1.026	1.078	1.013	1.010	1.076	**0.976***	1.014	1.001	**0.956**	2.05%

For all the comparison methods in the testing view of *Cold Start*, SoRec and SVD++ perform respectively the best in FilmTrust and Flixster (trust-alike), while no single approach works the best in Epinions and Ciao (trust). Generally, our approach performs better than the others both in trust and trust-alike relationships. Although some exceptions are observed in Epinions in terms of MAE, TrustSVD is more powerful in terms of RMSE. Since all the trust-based models aim to optimize the square errors between predictions and real values, RMSE is more indicative than MAE, and thus TrustSVD still has best performance overall.

Besides the above-compared approaches, some new trust-based models have been proposed recently. The most relevant model is presented in Fang et al. [47]; for clarity, we denote it by *Fang's*. It is reported to perform better than other trust-based models and than SVD++ (except in Ciao). Table 6.4 shows the comparison between Fang's and our approach TrustSVD. The results of Fang's approach are reported in Fang et al. [47],[6.8] and directly re-used in our work. Note that we sampled more data for Flixster than Fang's, and thus their experimental results

[6.8] Only the results in the testing view of 'All' are reported and no results reported in the case of *Cold Start*.

Table 6.3: Performance comparison in the testing view of 'Cold Start', where * indicates the best performance among all the other methods, and column 'Improve' indicates the percentage of improvements that our approach TrustSVD achieves relative to the * results. For each feature dimensionality d, the first row represents MAE values while the second includes RMSE values.

Cold Start	**UAvg**	**IAvg**	**PMF**	**RSTE**	**SoRec**	**SoReg**	**SoMF**	**TMF**	**SVD++**	**TSVD**	**Improve**
FilmTrust	0.709	0.722	0.814	0.680	**0.670***	0.881	0.697	0.674	0.677	**0.655**	2.24%
$d = 5$	0.979	0.911	1.079	0.884	**0.857***	1.104	0.916	0.867	0.897	**0.839**	2.10%
	0.709	0.722	0.767	0.674	**0.668***	0.771	0.680	0.687	0.680	**0.659**	1.35%
$d = 10$	0.979	0.911	1.009	0.900	**0.897***	1.034	0.907	0.900	0.905	**0.847**	5.57%
Epinions	1.047	**0.852***	1.451	1.051	0.892	1.398	0.884	0.891	0.889	**0.869**	-1.99%
$d = 5$	1.430	1.127	1.770	1.266	1.138	1.735	1.133	**1.125***	1.162	**1.104**	1.87%
	1.047	0.852	1.153	0.981	**0.846***	1.139	0.857	0.853	0.891	**0.868**	-2.60%
$d = 10$	1.430	**1.127***	1.432	1.313	1.180	1.437	1.152	1.176	1.166	**1.105**	1.95%
Flixster	0.869	0.906	1.097	0.872	0.872	1.058	0.881	0.901	**0.868***	**0.844**	2.76%
$d = 5$	1.155	1.114	1.390	1.097	**1.096***	1.358	1.103	1.138	1.122	**1.056**	3.65%
	0.869*	0.906	0.949	0.889	0.892	0.951	0.884	0.976	**0.869***	**0.846**	2.65%
$d = 10$	1.155	1.114	1.206	1.137	1.144	1.218	1.112*	1.328	**1.112***	**1.059**	4.77%
Ciao	0.829	**0.735***	1.033	0.957	0.789	1.173	0.774	0.752	0.759	**0.726**	1.22%
$d = 5$	1.138	1.005	1.334	1.113	0.998	1.430	1.001	**0.954***	1.039	**0.940**	1.47%
	0.829	0.735	0.926	0.803	**0.730***	0.949	0.741	0.770	0.749	**0.725**	0.68%
$d = 10$	1.138	1.005	1.191	1.014	1.031	1.214	**0.978***	1.096	1.020	**0.939**	3.99%

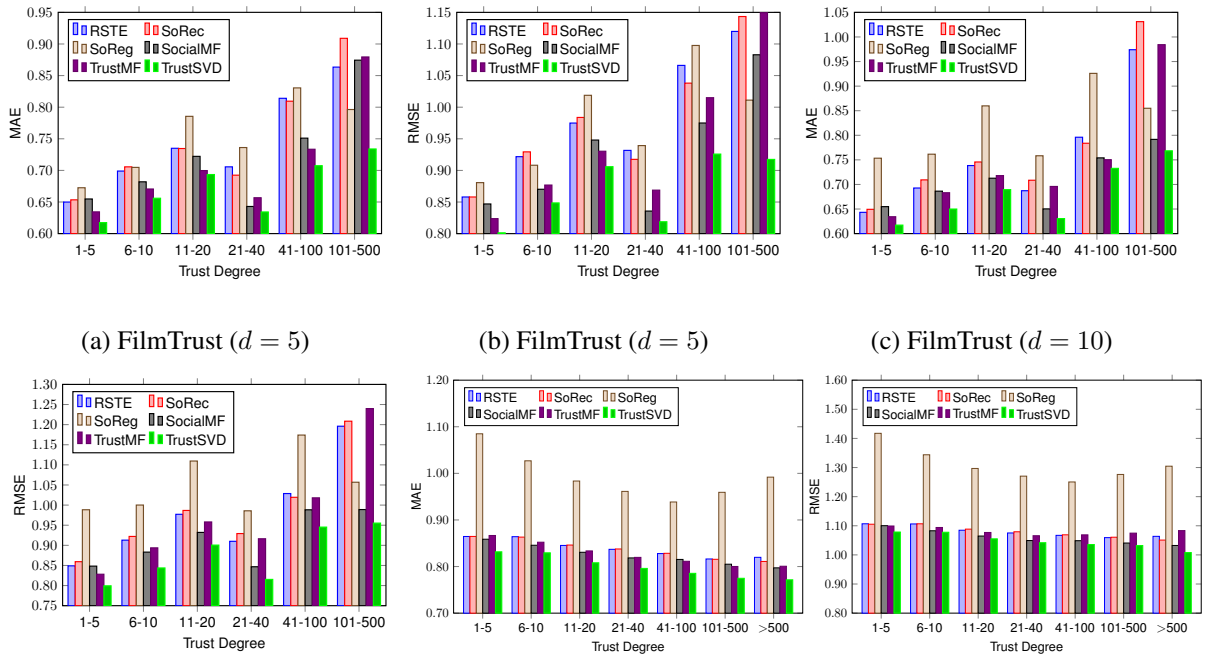
Table 6.4: Performance comparing with Fang's approach

Fang's vs.	**Epinions**		**Ciao**		**FilmTrust**	
TrustSVD	$d = 5$	$d = 10$	$d = 5$	$d = 10$	$d = 5$	$d = 10$
MAE	0.806	0.814	0.737	0.745	0.616	0.625
	0.804	**0.805**	**0.723**	**0.723**	**0.607**	**0.605**
RMSE	1.047	1.059	0.972	0.985	0.793	0.810
	1.043	**1.044**	**0.956**	**0.956**	**0.791**	**0.789**

are not comparable. Table 6.4 clearly shows that our approach performs better than Fang's in terms of both MAE and RMSE.

One more observation from Tables 6.2, 6.3 and 6.4 is that the performance of TrustSVD when $d = 5$ is very close to that when $d = 10$, indicating the reliability of our approach with respect to the feature dimensionality. We ascribe this feature to the consideration of both the explicit and implicit influence of item ratings and social trust in a unified model.

In conclusion, the experimental results indicate that our approach TrustSVD outperforms the other methods in predicting more accurate ratings, and that its performance is reliable with



(a) FilmTrust ($d = 5$) (b) FilmTrust ($d = 5$) (c) FilmTrust ($d = 10$)

(d) FilmTrust ($d = 10$) (e) Epinions ($d = 5$) (f) Epinions ($d = 5$)

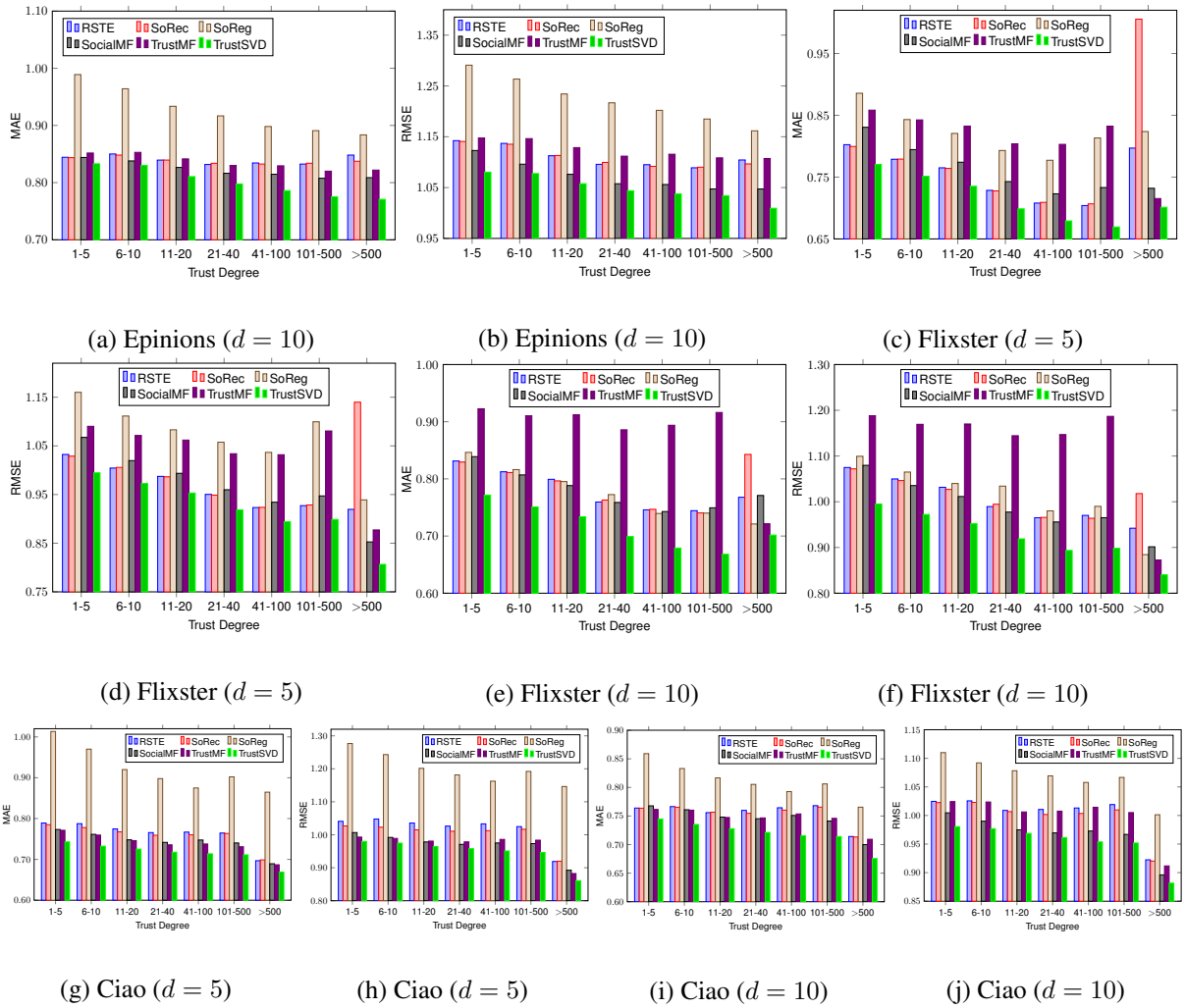
Figure 6.6: Performance comparison on users with different trust degrees across all the data sets [best viewed in color]

different number of latent features.

6.3.4 Comparison in trust degrees.

Another series of experiments are conducted to investigate the performance on users with different trust degrees, in order to further compare the performance of our approach with other trust-based counterparts, i.e., RSTE, SoRec, SoReg, SocialMF and TrustMF.[6.9] The trust degrees refer to the summation of number of trusted neighbours specified by a user (i.e., out degree) and number of trusting neighbours who trust the user (i.e., in degree). We split the trust degrees into (up to seven) categories: 1-5, 6-10, 11-20, 21-40, 41-100, 101-500, >500 as used by Yang et al. [48]. The results of trust-based models are illustrated in Figures 6.6 and 6.7. It is observed that our approach TrustSVD consistently outperforms the other trust-based models in terms of both MAE and RMSE across all the data sets. The statistic significance tests (paired t-tests, confidence 0.95) between our approach TrustSVD and other comparison

[6.9] Other trust-unaware methods (e.g., SVD++, PMF) are not used in the experiments.



(a) Epinions ($d = 10$) (b) Epinions ($d = 10$) (c) Flixster ($d = 5$)

(d) Flixster ($d = 5$) (e) Flixster ($d = 10$) (f) Flixster ($d = 10$)

(g) Ciao ($d = 5$) (h) Ciao ($d = 5$) (i) Ciao ($d = 10$) (j) Ciao ($d = 10$)

Figure 6.7: Performance comparison on users with different trust degrees across all the data sets (continued) [best viewed in color]

models are presented in Table 6.5. The table shows that our approach TrustSVD in general achieves significantly better performance than other methods. A few exceptions are observed though, which should be further handled in the future work.

6.4 Concluding Remarks

Social trust has been widely adopted to improve the performance of recommendations due in a great part to its strong and positive correlation with user similarity. Many approaches

Table 6.5: Significance tests of our TrustSVD model relative to other comparison models across all the data sets, where p-values are denoted by the significance symbols: < 0.05 with *, < 0.01 with **, < 0.001 with ***.

TrustSVD	**MAE**	**RMSE**	**MAE**	**RMSE**
FilmTrust	$d = 5$		$d = 10$	
vs. RSTE	3.574e-3**	2.39e-3**	0.02015*	7.694e-3**
vs. SoRec	8.398e-3**	3.504e-3**	0.02525$*$	6.469e-3**
vs. SoReg	5.724e-4***	5.67e-4***	1.683e-4***	1.023e-4***
vs. SocialMF	0.02827*	0.02601*	2.032e-4***	2.267e-5***
vs. TrustMF	0.06648	0.03744*	0.05085	0.02466*
Epinions	$d = 5$		$d = 10$	
vs. RSTE	4.846e-7***	5.845e-5***	1.870e-3**	1.562e-5***
vs. SoRec	9.751e-8***	2.307e-6***	1.383e-3**	4.582e-6***
vs. SoReg	3.014e-6***	1.882e-6***	3.873e-7***	2.749e-7***
vs. SocialMF	5.301e-6***	1.895e-3**	1.114e-3**	1.010e-3**
vs. TrustMF	1.343e-6***	2.499e-3**	1.236e-4***	8.249e-7***
Flixster	$d = 5$		$d = 10$	
vs. RSTE	2.719e-3**	4.650e-3**	4.143e-8***	6.248e-7***
vs. SoRec	0.06315	0.06652	2.603e-4***	5.298e-4***
vs. SoReg	4.864e-6***	2.127e-6***	7.754e-5***	2.104e-5***
vs. SocialMF	2.128e-5***	1.58e-5***	7.871e-7***	6.589e-7***
vs. TrustMF	6.092e-4***	6.92e-5***	4.577e-4***	3.069e-4***
Ciao	$d = 5$		$d = 10$	
vs. RSTE	4.925e-6***	3.241e-7***	9.398e-5***	5.795e-6***
vs. SoRec	1.565e-6***	1.091e-6***	6.842e-5***	1.530e-6***
vs. SoReg	3.256e-6***	3.825e-7***	4.011e-7***	1.596e-8***
vs. SocialMF	2.667e-6***	9.664e-5***	4.014e-6***	4.020e-4***
vs. TrustMF	4.529e-6***	4.260e-4***	4.399e-5***	9.624e-6***

(both memory-based and model-based) have been proposed in the literature. In this chapter, we continue such a research line to resolve the concerned issues. Specifically, we proposed a novel trust-based matrix factorization model by incorporating both rating and trust information, called *TrustSVD*. Our analysis of trust in four real-world data sets indicated that trust and ratings are complementary to each other, and both pivotal for more accurate recommendations. Our novel approach takes into account both the explicit and implicit influence of ratings and trust information when predicting ratings of unknown items. In particular, both the trust influence of trustees and trusters of active users are involved in our model. In addition, a weighted-λ-regularization technique was adapted and used to further regularize the generation of user-

and item-specific latent feature vectors. Comprehensive experimental results on four real-world data sets showed that our approach TrustSVD outperformed both trust- and ratings-based methods in predictive accuracy across different testing views and across users with different trust degrees. Thus, our approach can better alleviate the data sparsity and cold start problems of recommender systems. As the first work that can consistently outperforms not only other trust-based approaches but well-performing ratings-only models, our approach takes a further and important step towards better trust-aware recommender systems.

Chapter 7

Conclusions and Future Work

7.1 Summary of Contributions

This section concludes our current studies to resolve the *data sparsity* and *cold start* problems of recommender systems. These issues inherently hinder recommender systems from providing accurate recommendations, especially to the cold-start users who have not rated many items. To cope with these issues, our present research made three significant contributions from the perspectives of both ratings and trust. Firstly, we proposed a novel Bayesian similarity measure in Chapter 3 that solved the problems from which traditional similarity measures suffered, whereby existent ratings were made better use of. Many recommendation methods including both memory-based and model-based ones require a good measure to compute user similarity. Secondly, in Chapter 4 we introduced a new information source called *prior ratings* that resembled traditional user ratings but differed in the basis of virtual product experience. It opened a door to inherently handle the concerned issues by eliciting more user ratings, different kind yet useful and reliable for better recommendations. Finally, we brought light on new ways to incorporate social trust to improve the recommendation performance in Chapters 5 and 6. Two novel approaches were proposed, the effectiveness of which was verified using real-world data sets. We thus deepened the development of trust-aware recommender systems.

In more detail, the first work that we came up with in Chapter 3 was a Bayesian similarity measure with the aim of substituting the traditional ones. It took into account both the direction and length of rating vectors. The objective was to better model user correlations even if only a few ratings were available, given that previous approaches were incapable of dealing with the cold condition. The Dirichlet distribution was applied to accommodate the multi-scale rating

distances between rating pairs. A random distribution was used to form a prior probability for Bayesian inference. Then, the posterior probability of each rating distance can be consequently updated along with new rating pairs arriving. The user distance between two rating vectors was defined as the weighted average of rating distances with evidence weights. Three factors were identified to compute the evidence weight, namely rating consistency, Gaussian singularity and rating semantics. Hence, user similarity can be derived by inversely normalizing their distance. Furthermore, the correlation due to chance and potential system bias due to the computational formation were also removed to reveal user similarity more realistically. Typical examples showed that our approach can solve the four specific problems of traditional methods. We further studied the distinctions among different approaches by varying the length of rating vectors, and revealed that our method can produce more accurate and distinguishable similarity measurements. Finally, experimental results on six real-world data sets further demonstrated that our approach can result in better recommendation performance.

Instead of merely relying on existing ratings, Chapter 4 proposed a new kind of user ratings, prior ratings, in order to alleviate our research problems inherently by eliciting more ratings. Prior ratings were defined as those which were issued by users having effective interactions with virtual products, and issued usually prior to purchase in a mediated environment. We argued that users can form concrete opinions towards the quality/value of products as long as they had direct and effective interactions with them. To investigate which factors would influence users' evaluation in mediated environments, a conceptual model of prior ratings was constructed and then validated by our experiments and user study. The main observations from our experiments indicated that: (1) users were more confident and willing to give prior ratings in VR (virtual reality) than in WS (websites) as they felt more comfort and higher sense of presence as being in a real world; (2) users depended more on extrinsic attributes than intrinsic attributes to evaluate the product quality in WS while users depended more on intrinsic attributes than extrinsic attributes in VR; and (3) perceived quality has significantly positive influence on prior ratings while perceived cost has small yet positive influence on prior ratings. We also distinguished prior ratings from other kinds of information sources, such as trust and friendship used for recommender systems. Furthermore, we provided a feasible solution to demonstrate how prior ratings can be leveraged to improve recommendation performance. Experiments on the data collected from user study confirmed the usefulness of prior ratings.

Trust-aware recommender systems have attracted much attention recently, given that social trust provides an alternative view of user preferences other than item ratings [103]. Chapters 5 and 6 furthers the study of this research line by proposing two novel approaches, where one is memory-based approach and the other model-based. Specifically, Chapter 5 described a trust-based approach 'Merge' which merged the ratings of trusted neighbours to form a new and more complete rating profile for active users. That is, the ratings on a commonly-rated item were averaged according to the importance of trusted neighbours. The importance was measured by three factors, namely user similarity, social trust and social similarity. The quality (confidence) of such merged ratings was also measured in the light of both the number of ratings involved and the conflicts between positive and negative opinions. In other words, a new rating profile contained two sets: one of merged ratings and the other of rating confidence. Note that all the ratings of the active users were preserved and only the ratings on the other items were merged together, leading to more complete rating profiles. Then, an adapted collaborative filtering technique was applied to generate recommendations. This strategy was likely to help identify more reliable similar users for the active users, especially useful for the cold-start users. Experimental results based on three real-world data sets demonstrated the effectiveness of our method in terms of accuracy and coverage.

Lastly, Chapter 6 worked on a more popular matrix factorization-based model, called *TrustSVD*. Although many trust-based recommendation models have been studied previously, even the latest model may be inferior to other state-of-the-art ratings-only models such as SVD++. To give insight, we conducted a trust analysis based on four real-world data sets. The empirical results indicated that ratings and trust can be complementary to each other and should be taken into account simultaneously to enhance the generality to both trust and trust-alike relationships. Therefore, we built the TrustSVD model on top of SVD++ which inherently incorporated the influence of ratings, by further incorporating the influence of trust. Specifically, the explicit influence (of real trust values) was used to regularize the generation of user-specific rating vectors while the implicit influence (of trust connections) was integrated to predict the value of unknown items. Our work is the first to extend the SVD++ with social trust information. Experimental results on the four real-world data sets showed that our approach outperformed both trust-based and ratings-only models (ten in total) in predictive accuracy.

To summarize, four works have been proposed and studied in this thesis from the perspectives of both ratings and trust. We provided effective solutions to the concerned issues by

making better use of ratings and trust (individually and combinatorially), and by eliciting more user ratings that were useful and reliable for recommendations. Most of our approaches were verified in a number of data sets and testing views, and compared with many other counterparts, indicating their generalities and effectiveness for better recommendations.

7.2 Future Work

Based on our current research, future work can be conducted in the following two lines.

7.2.1 Further Study of Ratings

BS: an even better Bayesian similarity measure. The present work stresses the importance of three factors (rating consistency, Gaussian singularity and rating semantics) in deriving evidence weights in order to better model user similarity. However, a number of parameters are needed to be set according to the characteristics of data sets (e.g., c in Equation 3.3) or the empirical study (e.g., $\beta_1, \beta_2, \beta_3$ in Equation 3.11). We would like to further study how to better determine the values of these parameters, or ways to reduce the number of parameters. In addition, we plan to integrate more information about user ratings, such as the time when ratings were issued, in order to consider the dynamics of user interest [147]. Furthermore, the three factors we designed may be also beneficial to measure item-item similarity with proper adaptations. For example, the rating consistency can be modelled based on reliable users rather than reliable items, and the system bias is also applicable to item similarity. It would be interesting to find out how these factors can be employed to compute item similarity. Since our approach only relies on numerical ratings to model user correlation, it has the potential to be used in many other domains, such as information retrieval and social media [148].

Prior ratings: a more applicable information source. Our current research focuses on the conceptual model (and its verification) of prior ratings along with a feasible solution to leverage both prior and posterior ratings to improve recommendation performance. Further validation can be conducted by recruiting more subjects of greater variety and by designing more kinds of products in the experiments. In this way, more prior ratings can be collected for experimental evaluation to enhance the generality of our conceptual model, and to give further evidence

about the effectiveness of prior ratings. Besides, since only a feasible solution is provided in this work, in the future we would like to design more sophisticated approaches (e.g., latent factor models) to incorporate prior ratings, and investigate additional benefits such as the diversity of prior ratings. In addition, it deserves to study how prior ratings and social trust can collaborate with each other to achieve better recommendation performance since social networks are also available in virtual reality environments. Lastly, an interesting study that builds on our work would be to compare user purchase behaviours in a system with and without prior ratings, and how the differences may influence item recommendations.

Vision of ratings. Ratings provide the convenience for recommender systems to recommend users items of interest without the necessity to interpret item contents. Two limitations to our current studies of ratings-only recommender systems can be identified. Firstly, each user has only rated each item at most once. This is partially due to the fact that most publicly available data sets contain only such information. However, it is well recognized that user interest (e.g., in music) may drift over time. Hence, it would be misleading to identify similar users when a user drifts away from her preferences whereas all her past ratings are involved for similarity computation. Multiple ratings (by one user) for one and the same item may help recommender systems track how and to what extent user preferences have been changed. The rating patterns (with changes) may be worthy to generate up-to-date recommendations.

Secondly, till now we have only worked on the situations where only overall quality ratings are available for all the items. However, items may possess different properties with unequal strengthes, and users may attach varying importance to these features in item consuming. Computing user similarity solely based on the general quality ratings may produce a user bias. Besides, prior ratings can be also split into different features and allow users to give specific feature ratings. Many real systems such as Tripadvior[7.1] accept multi-criteria ratings [149]. For example, a hotel in Tripadvior can be rated in terms of location, sleeping quality, rooms, service, value and cleanliness other than a general rating. Although many multi-criteria recommender systems have been studied in the literature [150], the studies on multi-criteria similarity measure are relatively few and rare [151, 152, 153]. In the future, we would like to design a new similarity measure by incorporating multi-criteria ratings into a unified and concrete formulation.

[7.1] http://www.tripadvisor.com.sg/

7.2.2 Further Study of Trust

Merge: an easier-to-use recommendation approach. At least two directions can be imposed for future research. Firstly, we aim to adapt the settings of parameters (e.g., α and β in Equation 5.5) for the Merge method, or try to reduce the number of parameters for easier parameter tuning and adjusting in the experiments. Secondly, since an adapted version of PCC was proposed in Equation 5.10, it would be interesting to design a variant of Bayesian similarity by incorporating rating confidence. A straightforward way would be to integrate rating confidence when computing evidence weights. Finally, as a user-based approach, better performance could be achieved by hybridizing with an item-based approach.

TrustSVD: a more powerful trust-based model. For future work, we intend to further improve the proposed model by considering other formulations of trust influence, and transforming the binary trust values to real values with the consideration of structures of trust networks. It may be also beneficial to explicitly consider trust propagation in our model. Another interesting research line is the correlation of trust and ratings and how to adopt the correlation into the proposed model. Since our approach is applicable for both trust and trust-alike relationships, it deserves further investigations on (1) what kind of social relationships can be regarded as trust-alike relationships; and (2) how to adapt our TrustSVD model if two or more trust(-alike) relationships are available in the systems.

Vision of social trust. Social trust provides an alternative view of user preference in addition to user ratings, and helps resolve the data sparsity and cold start problems. To further improve the performance of trust-aware recommender systems, two research lines can be conducted in the future. The first line is to incorporate implicit trust into trust-aware recommendation approaches. Our current research relies on the explicit trust which is directly specified by users. However, such kind of information is usually very sparse or even unavailable [85]. One reason is that users may not be willing to share or expose it due to the concerns of, for example, privacy. Another reason is that only very few social networks adopt the concept of trust.

Lack of trust will greatly restrict the application (and thus the generality) of trust-aware recommender systems. Hence, it is beneficial to infer implicit trust from other kinds of information. Some research has been conducted in this area. For example, Fang et al. [154, 155]

make many efforts to infer implicit trust from user-item ratings, i.e., a kind of indirect user interactions for reputation systems. However, we show that such kind of user interactions is not effective to infer implicit trust in recommender systems [85]. Instead, Seth et al. [156, 157] design a user credibility model based on participatory media content, i.e., direct user interactions. For our research, we propose an extended trust antecedents framework (ETAF) to infer implicit trust, according to users' experience with the reviews written by other users [86] about the products that they used or purchased in the past. However, such inferred implicit trust has not been involved in our current trust-based approaches. The integration of explicit and implicit trust is not trivial and requires further understanding of their distinctions and correlations.

On the other hand, as a relevant but distinct construct, distrust has not been thoroughly studied in trust-aware recommender systems. Tang et al. [158] show that distrust is not the negation of trust and can provide added value over trust to recommender systems. Few works have considered distrust in their trust-aware recommendation approaches [43, 159, 160]. In this regard, to further our current research, one direction to explore is how to adapt our trust-based approaches to distrust information.

Lastly, we would like to investigate how trust can be utilized to resolve the other kinds of recommendation tasks while our present research focuses merely on the task of rating prediction. Other tasks such as item recommendations are also quite meaningful for recommender systems. However, not many trust-based approaches have been proposed to date. This may direct a new research interest of trust-aware recommender systems, especially considering that exclusive trust-based approaches have been proposed to predict item ratings.

References

[1] C. De Rosa, J. Cantrell, A. Havens, J. Hawk, L. Jenkins, B. Gauder, R. Limes, D. Cellentani, and OCLC, *Sharing, privacy and trust in our networked world: A report to the OCLC membership*. OCLC, 2007.

[2] X. Zhou, Y. Xu, Y. Li, A. Josang, and C. Cox, "The state-of-the-art in personalized recommender systems for social networking," *Artificial Intelligence Review*, vol. 37, no. 2, pp. 119–132, 2012.

[3] J. Pearce, S. Chang, B. Alzougool, G. Kennedy, M. Ainley, and S. Rodrigues, "Search or explore: Do you know what youre looking for?," in *Proceedings of the annual conference for the Computer-Human Interaction Special Interest Group (CHISIG) of the Human Factors and Ergonomics Society of Australia (OzCHI)*, pp. 246–249, 2011.

[4] G. Adomavicius and A. Tuzhilin, "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 (TKDE)*, vol. 17, no. 6, pp. 734–749, 2005.

[5] X. Su and T. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in Artificial Intelligence*, vol. 2009, pp. 4:2–4:2, 2009.

[6] F. Cacheda, V. Carneiro, D. Fernández, and V. Formoso, "Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems," *ACM Transactions on the Web (TWEB)*, vol. 5, no. 1, pp. 2:1–2:33, 2011.

[7] G. Lenzini, Y. van Houten, W. Huijsen, and M. Melenhorst, "Shall i trust a recommendation? towards an evaluation of the trustworthiness of recommender sites," in *Proceedings of the 13th East European Conference on Advances in Databases and Information Systems (ADBIS)*, pp. 121–128, 2010.

[8] T. Hess, M. Fuller, and J. Mathew, "Involvement and decision-making performance with a decision aid: The influence of social multimedia, gender, and playfulness," *Journal of Management Information Systems*, vol. 22, no. 3, pp. 15–54, 2006.

[9] B. Pathak, R. Garfinkel, R. Gopal, R. Venkatesan, and F. Yin, "Empirical analysis of the impact of recommender systems on sales," *Journal of Management Information Systems*, vol. 27, no. 2, pp. 159–188, 2010.

[10] J. Schafer, J. Konstan, and J. Riedl, "E-commerce recommendation applications," in *Applications of Data Mining to Electronic Commerce*, pp. 115–153, 2001.

[11] S. Komiak and I. Benbasat, "The effects of personalization and familiarity on trust and adoption of recommendation agents," *Management Information Systems Quarterly*, vol. 30, no. 4, pp. 941–960, 2006.

[12] B. K. Mohan, B. J. Keller, and N. Ramakrishnan, "Scouts, promoters, and connectors: The roles of ratings in nearest neighbor collaborative filtering," *ACM Transactions on the Web (TWEB)*, vol. 1, no. 2, p. 8, 2007.

[13] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," *Internet Computing*, vol. 7, no. 1, pp. 76–80, 2003.

[14] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th International Conference on World Wide Web (WWW)*, pp. 285–295, 2001.

[15] P. Massa and P. Avesani, "Trust metrics in recommender systems," in *Computing with social trust*, pp. 259–285, 2009.

[16] P. Massa and P. Avesani, "Trust-aware recommender systems," in *Proceedings of the 2007 ACM Conference on Recommender Systems (RecSys)*, pp. 17–24, 2007.

[17] G. Guo, J. Zhang, and N. Yorke-Smith, "A novel Bayesian similarity measure for recommender systems," in *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2619–2625, 2013.

[18] G. Guo, J. Zhang, D. Thalmann, and N. Yorke-Smith, "Prior ratings: A new information source for recommender systems in e-commerce," in *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys)*, pp. 383–386, 2013.

[19] G. Guo, J. Zhang, D. Thalmann, and N. Yorke-Smith, "Leveraging prior ratings for recommender systems in e-commerce," *Electronic Commerce Research and Applications (ECRA)*, 2014.

[20] G. Guo, J. Zhang, and D. Thalmann, "A simple but effective method to incorporate trusted neighbors in recommender systems," in *Proceedings of the 20th International Conference on User Modeling, Adaptation and Personalization (UMAP)*, pp. 114–125, 2012.

[21] Y. Ren, G. Li, J. Zhang, and W. Zhou, "The efficient imputation method for neighborhood-based collaborative filtering," in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM)*, pp. 684–693, 2012.

[22] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proceedings of the 4th ACM International Conference on Web Search and Data Mining (WSDM)*, pp. 287–296, 2011.

[23] Y. Shi, M. Larson, and A. Hanjalic, "Mining contextual movie similarity with matrix factorization for context-aware recommendation," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 4, no. 1, pp. 16:1–16:19, 2013.

[24] J. Breese, D. Heckerman, C. Kadie, *et al.*, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 43–52, 1998.

[25] H. Ma, I. King, and M. Lyu, "Effective missing data prediction for collaborative filtering," in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 39–46, 2007.

[26] H. Ahn, "A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem," *Information Sciences*, vol. 178, no. 1, pp. 37–51, 2008.

[27] A. O'Hagan, "Bayesian statistics: principles and benefits," *Frontis*, vol. 3, pp. 31–45, 2004.

[28] I. Konstas, V. Stathopoulos, and J. Jose, "On social networks and collaborative recommendation," in *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 195–202, 2009.

[29] I. Guy, I. Ronen, and E. Wilcox, "Do you know?: recommending people to invite into your social network," in *Proceedings of the 14th International Conference on Intelligent User Interfaces (IUI)*, pp. 77–86, 2009.

[30] M. Jamali and M. Ester, "A transitivity aware matrix factorization model for recommendation in social networks," in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2644–2649, 2011.

[31] M. Rymaszewski, *Second Life: The official guide*. Sybex, 2007.

[32] P. Hemp, "Avatar-based marketing," *Harvard Business Review*, vol. 84, no. 6, pp. 48–57, 2006.

[33] H. Li, T. Daugherty, and F. Biocca, "The role of virtual experience in consumer learning," *Journal of Consumer Psychology*, vol. 13, no. 4, pp. 395–407, 2003.

[34] Q. Yuan, S. Zhao, L. Chen, Y. Liu, S. Ding, X. Zhang, and W. Zheng, "Augmenting collaborative recommender by fusing explicit social relationships," in *Proceedings of ACM RecSys Workshop on Recommender Systems and the Social Web*, pp. 49–56, 2009.

[35] C. Ziegler and G. Lausen, "Analyzing correlation between trust and user similarity in online communities," in *Trust management*, pp. 251–265, 2004.

[36] A. Jøsang, W. Quattrociocchi, and D. Karabeg, "Taste and trust," in *Trust Management V*, pp. 312–322, 2011.

[37] J. O'Donovan and B. Smyth, "Trust in recommender systems," in *Proceedings of the 10th International Conference on Intelligent User Interfaces (IUI)*, pp. 167–174, 2005.

[38] A. Seth, J. Zhang, and R. Cohen, "Bayesian credibility modeling for personalized recommendation in participatory media," in *Proceedings of the 18th International Conference on User Modeling, Adaptation, and Personalization (UMAP)*, pp. 279–290, 2010.

[39] J. Golbeck, *Computing and applying trust in web-based social networks*. PhD thesis, 2005.

[40] M. Chowdhury, A. Thomo, and B. Wadge, "Trust-based infinitesimals for enhanced collaborative filtering," in *Proceedings of the 15th International Conference on Management of Data (COMAD)*, 2009.

[41] S. Ray and A. Mahanti, "Improving prediction accuracy in trust-aware recommender systems," in *Proceedings of the 43rd Hawaii International Conference on System Sciences (HICSS)*, pp. 1–9, 2010.

[42] P. Singla and M. Richardson, "Yes, there is a correlation: from social networks to personal behavior on the web," in *Proceedings of the 17th International Conference on World Wide Web (WWW)*, pp. 655–664, 2008.

[43] P. Victor, C. Cornelis, M. De Cock, and A. Teredesai, "Trust- and distrust-based recommendations for controversial reviews," in *Proceedings of the 2009 International Conference on Web Science (WebSci)*, pp. 48–55, 2009.

[44] Y. Shi, M. Larson, and A. Hanjalic, "How far are we in trust-aware recommendation?," in *Proceedings of the 33rd European Conference on Advances in Information Retrieval (ECIR)*, pp. 704–707, 2011.

[45] G. Guo, J. Zhang, and D. Thalmann, "Merging trust in collaborative filtering to alleviate data sparsity and cold start," *Knowledge-Based Systems (KBS)*, vol. 57, no. 0, pp. 57 – 68, 2014.

[46] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[47] H. Fang, Y. Bao, and J. Zhang, "Leveraging decomposed trust in probabilistic matrix factorization for effective recommendation," in *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 30–36, 2014.

[48] B. Yang, Y. Lei, D. Liu, and J. Liu, "Social collaborative filtering by trust," in *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2747–2753, 2013.

[49] Y. Koren, "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)*, pp. 426–434, 2008.

[50] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proceedings of the 25th Conference Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 452–461, 2009.

[51] G. Guo, J. Zhang, and N. Yorke-Smith, "TrustSVD: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings," in *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI)*, 2015.

[52] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl, "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)*, pp. 230–237, 1999.

[53] P. Melville, R. Mooney, and R. Nagarajan, "Content-boosted collaborative filtering for improved recommendations," in *Proceedings of the 16th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 187–192, 2002.

[54] R. Bell, Y. Koren, and C. Volinsky, "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)*, pp. 95–104, 2007.

[55] Z. Huang, D. Zeng, and H. Chen, "A comparison of collaborative-filtering recommendation algorithms for e-commerce," *IEEE Transactions on Intelligent Systems*, vol. 22, no. 5, pp. 68–78, 2007.

[56] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Modeling and User-Adapted Interaction (UMUAI)*, vol. 12, no. 4, pp. 331–370, 2002.

[57] D. Jannach, M. Zanker, M. Ge, and M. Gröning, "Recommender systems in computer science and information systems–a landscape of research," in *E-Commerce and Web Technologies*, pp. 76–87, 2012.

[58] T. Hofmann, "Latent semantic models for collaborative filtering," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 89–115, 2004.

[59] S. Vucetic and Z. Obradovic, "Collaborative filtering using a regression-based approach," *Knowledge and Information Systems*, vol. 7, no. 1, pp. 1–22, 2005.

[60] L. Xiong, X. Chen, T. Huang, J. Schneider, and J. Carbonell, "Temporal collaborative filtering with bayesian probabilistic tensor factorization," in *Proceedings of the 10th SIAM International Conference on Data Mining (SDM)*, pp. 211–222, 2010.

[61] N. Liu and Q. Yang, "EigenRank: a ranking-oriented approach to collaborative filtering," in *Proceedings of the 31st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 83–90, 2008.

[62] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Advances in Neural Information Processing Systems (NIPS)*, pp. 1257–1264, 2008.

[63] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proceedings of the fourth ACM conference on Recommender systems*, pp. 135–142, 2010.

[64] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering," in *Proceedings of the 5th International Conference on Computer and Information Technology (ICCIT)*, pp. 158–167, 2002.

[65] B. Xu, J. Bu, C. Chen, and D. Cai, "An exploration of improving collaborative recommender systems via user-item subgroups," in *Proceedings of the 21st international conference on World Wide Web (WWW)*, pp. 21–30, 2012.

[66] S. Zhang, W. Wang, J. Ford, and F. Makedon, "Learning from incomplete ratings using non-negative matrix factorization," in *Proceedings of the 6th SIAM Conference on Data Mining (SDM)*, pp. 549–553, 2006.

[67] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, A. Hanjalic, and N. Oliver, "Tfmap: Optimizing map for top-n context-aware recommendation," in *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 155–164, 2012.

[68] A. Gunawardana and C. Meek, "Tied boltzmann machines for cold start recommendations," in *Proceedings of the 2nd ACM Conference on Recommender Systems (RecSys)*, pp. 19–26, 2008.

[69] Z. Gantner, L. Drumond, C. Freudenthaler, S. Rendle, and L. Schmidt-Thieme, "Learning attribute-to-feature mappings for cold-start recommendations," in *Proceedings of the 2010 IEEE 10th International Conference on Data Mining (ICDM)*, pp. 176–185, 2010.

[70] N. N. Liu, X. Meng, C. Liu, and Q. Yang, "Wisdom of the better few: cold start recommendation via representative based rating elicitation," in *Proceedings of the 5th ACM Conference on Recommender Systems (RecSys)*, pp. 37–44, 2011.

[71] A. Ahmed, B. Kanagal, S. Pandey, V. Josifovski, L. G. Pueyo, and J. Yuan, "Latent factor models with additive and hierarchically-smoothed user preferences," in *Proceedings of the 6th ACM International Conference on Web Search and Data Mining (WSDM)*, pp. 385–394, 2013.

[72] Y. Bao, H. Fang, and J. Zhang, "TopicMF: Simultaneously exploiting ratings and reviews for recommendation," in *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI)*, 2014.

[73] R. M. Bell and Y. Koren, "Lessons from the netflix prize challenge," *ACM SIGKDD Explorations Newsletter*, vol. 9, no. 2, pp. 75–79, 2007.

[74] N. Lathia, S. Hailes, and L. Capra, "The effect of correlation coefficients on communities of recommenders," in *Proceedings of the 23rd Annual ACM Symposium on Applied Computing (SAC)*, pp. 2000–2005, 2008.

[75] A. Said, B. Jain, and S. Albayrak, "Analyzing weighting schemes in collaborative filtering: Cold start, post cold start and power users," in *Proceedings of the 27th Annual ACM Symposium on Applied Computing (SAC)*, pp. 2035–2040, 2012.

[76] Y. Koren, "Factor in the neighbors: Scalable and accurate collaborative filtering," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 4, no. 1, pp. 1:1–1:24, 2010.

[77] L. Candillier, F. Meyer, and F. Fessant, "Designing specific weighted similarity measures to improve collaborative filtering systems," in *Proceedings of the 8th Industrial Conference on Advances in Data Mining (ICDM)*, pp. 242–255, 2008.

[78] Y. Shi, M. Larson, and A. Hanjalic, "Exploiting user similarity based on rated-item pools for improved user-based collaborative filtering," in *Proceedings of the 2009 ACM Conference on Recommender Systems (RecSys)*, pp. 125–132, 2009.

[79] F. Ortega, J. Sánchez, J. Bobadilla, and A. Gutiérrez, "Improving collaborative filtering-based recommender systems results using pareto dominance," *Information Sciences*, vol. 239, no. 0, pp. 50 – 61, 2013.

[80] U. Shardanand and P. Maes, "Social information filtering: algorithms for automating "word of mouth"," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (SIGCHI)*, pp. 210–217, 1995.

[81] N. Lathia, S. Hailes, and L. Capra, "Private distributed collaborative filtering using estimated concordance measures," in *Proceedings of the 2007 ACM Conference on Recommender Systems (RecSys)*, pp. 1–8, 2007.

[82] J. Bobadilla, F. Ortega, and A. Hernando, "A collaborative filtering similarity measure based on singularities," *Information Processing & Management*, vol. 48, no. 2, pp. 204–217, 2012.

[83] H. Ma, I. King, and M. Lyu, "Learning to recommend with social trust ensemble," in *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 203–210, 2009.

[84] J. Tang, H. Gao, X. Hu, and H. Liu, "Exploiting homophily effect for trust prediction," in *Proceedings of the 6th ACM International Conference on Web Search and Data Mining (WSDM)*, pp. 53–62, 2013.

[85] G. Guo, J. Zhang, D. Thalmann, A. Basu, and N. Yorke-Smith, "From ratings to trust: an empirical study of implicit trust in recommender systems," in *Proceedings of the 29th Annual ACM Symposium on Applied Computing (SAC)*, pp. 248–253, 2014.

[86] G. Guo, J. Zhang, D. Thalmann, and N. Yorke-Smith, "ETAF: An extended trust antecedents framework for trust prediction," in *Proceedings of the 2014 International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2014.

[87] G. Carenini, J. Smith, and D. Poole, "Towards more conversational and collaborative recommender systems," in *Proceedings of the 8th International Conference on Intelligent User Interfaces (IUI)*, pp. 12–18, 2003.

[88] S. McNee, S. Lam, J. Konstan, and J. Riedl, "Interfaces for eliciting new user preferences in recommender systems," in *Proceedings of the 9th International Conference on User Modeling , Adaptation and Personalization (UMAP)*, pp. 148–148, 2003.

[89] R. Dong, K. McCarthy, M. O'Mahony, M. Schaal, and B. Smyth, "Towards an intelligent reviewer's assistant: recommending topics to help users to write better product reviews," in *Proceedings of the 2012 ACM International Conference on Intelligent User Interfaces (IUI)*, pp. 159–168, 2012.

[90] B. Xiao and I. Benbasat, "E-commerce product recommendation agents: use, characteristics, and impact," *Management Information Systems Quarterly*, vol. 31, no. 1, pp. 137–209, 2007.

[91] J. Eno, G. Stafford, S. Gauch, and C. Thompson, "Hybrid user preference models for Second Life and opensimulator virtual worlds," in *Proceedings of the 19th International Conference on User Modeling, Adaption and Personalization (UMAP)*, pp. 87–98, 2011.

[92] F. Shah, P. Bell, and G. Sukthankar, "A destination recommendation system for virtual worlds," in *Proceedings of the 21st International Florida Artificial Intelligence Research Society Conference (FLAIRS)*, pp. 475–476, 2010.

[93] X. Hu and K. Wang, "Personalized recommendation for virtual reality," in *Proceedings of the 2010 International Conference on Multimedia Technology (ICMT)*, pp. 1–5, 2010.

[94] R. Sinha and K. Swearingen, "Comparing recommendations made by online systems and friends," in *Proceedings of the DELOS-NSF Workshop on Personalization and Recommender Systems in Digital Libraries*, 2001.

[95] K. Swearingen and R. Sinha, "Beyond algorithms: An hci perspective on recommender systems," in *Proceedings of ACM SIGIR Workshop on Recommender Systems*, pp. 393–408, 2001.

[96] P. Massa and P. Avesani, "Trust-aware bootstrapping of recommender systems," in *Proceedings of ECAI 2006 Workshop on Recommender Systems*, pp. 29–33, 2004.

[97] J. Golbeck, "Generating predictive movie recommendations from trust in social networks," in *Proceedings of the 4th International Conference on Trust Management (iTrust)*, pp. 93–104, 2006.

[98] M. Jamali and M. Ester, "Trustwalker: a random walk model for combining trust-based and item-based recommendation," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pp. 397–406, 2009.

[99] F. Liu and H. Lee, "Use of social network information to enhance collaborative filtering performance," *Expert Systems with Applications*, vol. 37, no. 7, pp. 4772–4778, 2010.

[100] Q. Shambour and J. Lu, "A hybrid trust-enhanced collaborative filtering recommendation approach for personalized government-to-business e-services," *International Journal of Intelligent Systems*, vol. 26, no. 9, pp. 814–843, 2011.

[101] Q. Shambour and J. Lu, "A trust-semantic fusion-based recommendation approach for e-business applications," *Decision Support Systems (DSS)*, vol. 54, no. 1, pp. 768–780, 2012.

[102] C. Haydar, A. Boyer, A. Roussanaly, *et al.*, "Hybridising collaborative filtering and trust-aware recommender systems," in *Proceedings of hte 8th International Conference on Web Information Systems and Technologies (WEBIST)*, pp. 695–700, 2012.

[103] G. Guo, J. Zhang, and N. Yorke-Smith, "Leveraging multiviews of trust and similarity to enhance clustering-based recommender systems," *Knowledge-Based Systems (KBS)*, no. 0, pp. –, 2014.

[104] H. Ma, H. Yang, M. Lyu, and I. King, "SoRec: social recommendation using probabilistic matrix factorization," in *Proceedings of the 31st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 931–940, 2008.

[105] X. Yang, H. Steck, and Y. Liu, "Circle-based recommendation in online social networks," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 1267–1275, 2012.

[106] J. Zhu, H. Ma, C. Chen, and J. Bu, "Social recommendation using low-rank semidefinite program," in *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 158–163, 2011.

[107] W. Yao, J. He, G. Huang, and Y. Zhang, "Modeling dual role preferences for trust-aware recommendation," in *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR)*, pp. 975–978, 2014.

[108] S. Russell, P. Norvig, J. Canny, J. Malik, and D. Edwards, *Artificial intelligence: a modern approach*. Prentice hall Englewood Cliffs, NJ, 1995.

[109] D. Lin, "An information-theoretic definition of similarity," in *Proceedings of the 15th International Conference on Machine Learning (ICML)*, vol. 98, pp. 296–304, 1998.

[110] C. Desrosiers and G. Karypis, "A comprehensive survey of neighborhood-based recommendation methods," in *Recommender systems handbook*, pp. 107–144, Springer US, 2011.

[111] G. Adomavicius and J. Zhang, "Impact of data characteristics on recommender systems performance," *ACM Transactions on Management Information Systems (TMIS)*, vol. 3, no. 1, p. 3, 2012.

[112] H. Li, T. Daugherty, and F. Biocca, "Characteristics of virtual experience in electronic commerce: A protocol analysis," *Journal of Interactive Marketing*, vol. 15, no. 3, pp. 13–30, 2001.

[113] Z. Jiang and I. Benbasat, "Virtual product experience: effects of visual and functional control of products on perceived diagnosticity and flow in electronic shopping," *Journal of Management Information Systems*, vol. 21, no. 3, pp. 111–147, 2004.

[114] M. Slater, B. Spanlang, and D. Corominas, "Simulating virtual environments within virtual environments as the basis for a psychophysics of presence," *ACM Transactions on Graphics (TOG)*, vol. 29, no. 4, pp. 92:1–92:9, 2010.

[115] J. Steuer, "Defining virtual reality: Dimensions determining telepresence," *Journal of communication*, vol. 42, no. 4, pp. 73–93, 1992.

[116] A. G. Picciano, "Beyond student perceptions: Issues of interaction, presence, and performance in an online course," *Journal of Asynchronous learning networks*, vol. 6, no. 1, pp. 21–40, 2002.

[117] C. W. Phang and A. Kankanhalli, "How do perceptions of virtual worlds lead to enhanced learning? an empirical investigation," in *Proceedings of the 30th International Conference on Information Systems (ICIS)*, pp. 167:0–167:17, 2009.

[118] C. Heeter, "Being there: The subjective experience of presence," *Presence: Teleoperators and virtual environments*, vol. 1, no. 2, pp. 262–271, 1992.

[119] S. Gardial, D. Clemons, R. Woodruff, D. Schumann, and M. Burns, "Comparing consumers' recall of prepurchase and postpurchase product evaluation experiences," *Journal of Consumer Research*, vol. 20, no. 4, pp. 548–560, 1994.

[120] P. Goering, "Effects of product trial on consumer expectations, demand, and prices," *Journal of Consumer Research*, vol. 12, no. 1, pp. 74–82, 1985.

[121] J. Olson, *Cue properties of price: Literature review and theoretical considerations*. Pennsylvania State University, 1974.

[122] P. Nelson, "Advertising as information," *The Journal of Political Economy*, vol. 82, no. 4, pp. 729–754, 1974.

[123] W. Dodds, K. Monroe, and D. Grewal, "Effects of price, brand, and store information on buyers' product evaluations," *Journal of marketing research*, vol. 28, no. 3, pp. 307–319, 1991.

[124] V. Zeithaml, "Consumer perceptions of price, quality, and value: a means-end model and synthesis of evidence," *The Journal of Marketing*, vol. 52, no. 3, pp. 2–22, 1988.

[125] A. Rao and K. Monroe, "The effect of price, brand name, and store name on buyers' perceptions of product quality: An integrative review," *Journal of marketing Research*, vol. 26, no. 3, pp. 351–357, 1989.

[126] D. Lichtenstein, N. Ridgway, and R. Netemeyer, "Price perceptions and consumer shopping behavior: A field study," *Journal of Marketing Research*, vol. 30, no. 2, pp. 234–245, 1993.

[127] J. Jacoby, G. Szybillo, and J. Busato-Schach, "Information acquisition behavior in brand choice situations," *Journal of Consumer Research*, vol. 3, no. 4, pp. 209–216, 1977.

[128] H. Ha, "The effects of consumer risk perception on pre-purchase information in online auctions: Brand, word-of-mouth, and customized information," *Journal of Computer-Mediated Communication*, vol. 8, no. 1, pp. 0–0, 2002.

[129] W. Bearden and T. Shimp, "The use of extrinsic cues to facilitate product adoption," *Journal of marketing research*, vol. 19, no. 2, pp. 229–239, 1982.

[130] R. Stokes, *The effects of price, package design, and brand familiarity on perceived quality.* PhD thesis, ProQuest Information & Learning, 1974.

[131] P. Milgrom and J. Roberts, "Price and advertising signals of product quality," *The Journal of Political Economy*, vol. 94, no. 4, pp. 796–821, 1986.

[132] T. T. Nguyen, D. Kluver, T.-Y. Wang, P.-M. Hui, M. D. Ekstrand, M. C. Willemsen, and J. Riedl, "Rating support interfaces to improve user experience and recommender accuracy," in *Proceedings of the 7th ACM conference on Recommender systems (RecSys)*, pp. 149–156, 2013.

[133] J. Zhang, R. Cohen, and K. Larson, "Combining trust modeling and mechanism design for promoting honesty in e-marketplaces," *Computational Intelligence*, vol. 28, no. 4, pp. 549–578, 2012.

[134] G. Guo and M. Elgendi, "A new recommender system for 3D e-commerce: An EEG based approach," *Journal of Advanced Management Science*, vol. 1, no. 1, pp. 61–65, 2013.

[135] I. Arapakis, Y. Moshfeghi, H. Joho, R. Ren, D. Hannah, and J. M. Jose, "Enriching user profiling with affective features for the improvement of a multimodal recommender system," in *Proceedings of the ACM International Conference on Image and Video Retrieval (CIVR)*, pp. 29:1–29:8, 2009.

[136] M. Yeasin, B. Bullot, and R. Sharma, "Recognition of facial expressions and measurement of levels of interest from video," *Multimedia, IEEE Transactions on*, vol. 8, no. 3, pp. 500–508, 2006.

[137] D. Crandall, D. Cosley, D. Huttenlocher, J. Kleinberg, and S. Suri, "Feedback effects between similarity and social influence in online communities," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 160–168, 2008.

[138] D. Boyd, "Friends, friendsters, and myspace top 8: Writing community into being on social network sites," *First Monday*, vol. 11, no. 12, 2006.

[139] R. Mayer, J. Davis, and F. Schoorman, "An integrative model of organizational trust," *Academy of management review*, vol. 20, no. 3, pp. 709–734, 1995.

[140] D. J. Watts, *Six degrees: The science of a connected age*. WW Norton, 2004.

[141] Y. Wang and M. Singh, "Formal trust model for multiagent systems," in *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1551–1556, 2007.

[142] G. Xue, C. Lin, Q. Yang, W. Xi, H. Zeng, Y. Yu, and Z. Chen, "Scalable collaborative filtering using cluster-based smoothing," in *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 114–121, 2005.

[143] A. S. Lampropoulos, P. S. Lampropoulou, and G. A. Tsihrintzis, "A cascade-hybrid music recommender system for mobile services based on musical genre classification and personality diagnosis," *Multimedia Tools and Applications*, vol. 59, no. 1, pp. 241–258, 2012.

[144] G. Guo, "Integrating trust and similarity to ameliorate the data sparsity and cold start for recommender systems," in *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys)*, pp. 451–454, 2013.

[145] C. Castelfranchi and R. Falcone, *Trust Theory: A socio-cognitive and computational model*, vol. 20. Wiley, 2010.

[146] H. Ma, "An experimental study on implicit social recommendation," in *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 73–82, 2013.

[147] B. Li, X. Zhu, R. Li, C. Zhang, X. Xue, and X. Wu, "Cross-domain collaborative filtering over time," in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2293–2298, 2011.

[148] A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec, "Effects of user similarity in social media," in *Proceedings of the 5th ACM International Conference on Web Search and Data Mining (WSDM)*, pp. 703–712, 2012.

[149] C. Long, J. Zhang, M. Huang, X. Zhu, M. Li, and B. Ma, "Estimating feature ratings through an effective review selection approach," *Knowledge and Information Systems (KIS)*, vol. 38, no. 2, pp. 419–446, 2014.

[150] G. Adomavicius, N. Manouselis, and Y. Kwon, "Multi-criteria recommender systems," in *Recommender systems handbook*, pp. 769–803, Springer US, 2011.

[151] G. Adomavicius and Y. Kwon, "New recommendation techniques for multicriteria rating systems," *IEEE Transactiosn on Intelligent Systems (ToIT)*, vol. 22, no. 3, pp. 48–55, 2007.

[152] N. Manouselis and C. Costopoulou, "Experimental analysis of design choices in multiattribute utility collaborative filtering," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 21, no. 2, pp. 311–331, 2007.

[153] T. Y. Tang and G. McCalla, "The pedagogical value of papers: a collaborative-filtering based paper recommender," *Journal of Digital Information*, vol. 10, no. 2, 2009.

[154] H. Fang, Y. Bao, and J. Zhang, "Misleading opinions provided by advisors: Dishonesty or subjectivity," in *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1983–1989, 2013.

[155] H. Fang, J. Zhang, and N. M. Thalmann, "A trust model stemmed from the diffusion theory for opinion evaluation," in *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pp. 805–812, 2013.

[156] A. Seth and J. Zhang, "A social network based approach to personalized recommendation of participatory media content.," in *Proceedings of the 2008 International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2008.

[157] A. Seth, J. Zhang, and R. Cohen, "A personalized credibility model for recommending messages in social participatory media environments," *World Wide Web*, pp. 1–27, 2013.

[158] J. Tang, X. Hu, and H. Liu, "Is distrust the negation of trust? the value of distrust in social media," in *Proceedings of the 25th ACM Conference on Hypertext and Social Media (HT)*, pp. 148–157, 2014.

[159] C. Chen, Y. Wan, M. Chung, and Y. Sun, "An effective recommendation method for cold start new users using trust and distrust networks," *Information Sciences*, vol. 224, pp. 19–36, 2012.

[160] P. Victor, N. Verbiest, C. Cornelis, and M. D. Cock, "Enhancing the trust-based recommendation process with explicit distrust," *ACM Transactions on the Web (TWEB)*, vol. 7, no. 2, pp. 6:1–6:19, 2013.