Computational Techniques For
More Accurate and Diverse Recommendations

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

YoungOk Kwon

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Gediminas Adomavicius, Advisor

August 2011

Acknowledgements

I would like to express my sincere gratitude to those who helped me complete this dissertation. Foremost, I am deeply grateful to my advisor, Prof. Gedas Adomavicius, for his excellent guidance and support throughout the dissertation process. His enthusiasm and passion in research always inspired me, and I appreciate him being such a great mentor with patience and constant encouragement.

I would also like to thank my committee members, Prof. Alok Gupta, Prof. Ching Ren, and Prof. Jaideep Srivastava, who provided me with helpful advice and insight towards the completion of my dissertation. I also thank the other faculty members in the Information and Decision Sciences department for their willingness to help me. A special thanks to Prof. Shawn Curley and Prof. Paul Johnson for helping me broaden the scope of my research and expand my research experience.

I also convey my heartfelt thanks to my colleagues in the department of Information and Decision Sciences as well as graduates Kunsoo Han and Dongwon Lee. I feel greatly privileged to be a part of this wonderful group and want to extend my special appreciation to my office mates, Gregory Ramsey and Linda Wang, for their friendship and support. Many thanks also go to my friends, Sangeun Lee, Jungeun Moon, Yunyoung Kim, Jieun Kim, Hyunsun Kwak, Sunhyung Im, Jungmin Chough, Jiyoung Lee, Kiyeon Lee, and Lydia Liu, who always encouraged me during my PhD program, and my church friends, Lucy Lee, Stacey Oh, and Soyon Woo, who supported me spiritually.

Last but most importantly, I thank God, who made me strong enough to continue the journey of completing this dissertation and who showed that anything is possible, even with faith as small as a mustard seed. I am also greatly indebted to my beloved parents and brother for their unconditional love, support, and prayers.

To my parents and brother

Abstract

Recommender systems are becoming an increasingly important research area due to the growing demand for personalized recommendations. The volume of information available to each user and the number of products carried in e-commerce marketplaces have grown tremendously. Thus, recommender systems are needed to help individual users find the most relevant items from an enormous number of choices and eventually increase sales by exposing users to what they may like, but may not have considered otherwise. Despite significant progress in developing new recommendation techniques within both industry and academia, most research, to date, has focused on improving recommendation accuracy (i.e., the accuracy with which the recommender system predicts users' ratings for items they have not yet rated). While recommendation accuracy is undoubtedly important, there is a growing understanding that accuracy does not always imply usefulness to users. Therefore, in addition to investigating the *accuracy* of recommendations, my dissertation also considers the *diversity* of recommendations as another important aspect of recommendation quality and explores the relationship between accuracy and diversity. The diversity of recommendations can be expressed by the number of unique items recommended across all users, which reflects the ability of recommender systems to go beyond the obvious, best-selling items, and to generate more idiosyncratic, personalized, and long-tail recommendations.

This dissertation presents four studies which propose new recommendation approaches that can improve accuracy and diversity. The first study enhances traditional recommendation algorithms by augmenting them with *multi-criteria rating* information for more accurate recommendations. The second study applies heuristic-based *ranking* approaches for more diverse recommendations. The third study develops more sophisticated *optimization* approaches for direct diversity maximization. The fourth study explores the possible *combinations* of the two types of approaches – incorporation of multi-criteria rating information and the use of different ranking methods – as a way to generate recommendations that are both more accurate and more diverse.

The new recommendation approaches proposed in this dissertation enrich the body of knowledge on recommender systems by extending single-rating recommendation problems to address multi-criteria recommendation problems and exploring new ways to tackle the accuracy-diversity tradeoff issue. Individual users and online content providers will also benefit from the proposed approaches, in that each user will find more relevant and personalized items from more accurate and diverse recommendations provided by recommender systems. These approaches could potentially lead to increased loyalty and sales, thus, benefiting the providers as well.

Table of Contents

List of Tables

List of Figures

Chapter 1. Introduction

1.1 Research Background

Recommender systems have played a significant role in helping customers find relevant items from an enormous number of choices, particularly in e-commerce applications such as Amazon and Netflix. Recommender systems are also advantageous to online content providers. For example, according to Forrester Research (Schonfeld 2007), it is estimated that recommender systems account for 10% to 30% of an online retailer's sales. Netflix also reported that roughly two-thirds of their movies rented were ones that users may never have considered otherwise, but were recommended by their recommender system (Flynn 2006).

With the growing demand for personalized recommendations, much work has been done over the last decade on developing new recommendation techniques, both in industry and academia. The Netflix Prize competition[1] (Greene 2006) is a good example of the growing attention for these recommendation techniques. However, despite significant progress, current recommendation techniques still have a number of major challenges to be addressed (Adomavicius and Tuzhilin 2005). In particular, a multitude of recommendation algorithms proposed in the recommender systems literature as well as in industry have focused on improving recommendation *accuracy*, i.e., the accuracy with which the recommender system predicts users' ratings for the items that they have not yet rated. While recommendation accuracy is undoubtedly important, there is a growing understanding that accuracy does not always imply usefulness to users, and relying on the accuracy of recommendations alone may not be enough to find the most relevant items for each user (Herlocker et al. 2004, McNee et al. 2006, Shani and Gunawardana 2011). Therefore, in addition to the *accuracy* of recommendations, this dissertation investigates the *diversity* of recommendations that can reflect the ability of recommender systems to go beyond the obvious, best-selling items, and generate more idiosyncratic, personalized,

[1] The Netflix Prize competition was an open competition held in 2006-2009 for the best recommendation algorithm that could improve the accuracy of users' rating predictions by 10% over Netflix's own recommendation engine (www.netflixprize.com).

and long-tail recommendations. This dissertation consists of four studies, and the challenge that each study seeks to address is described below.

A typical recommender system provides recommendations to a user by estimating the ratings of items yet to be consumed by the user, based on the ratings of items already consumed. Recommendations to users are made based on the predicted ratings of each item for each user (i.e., the items with the most highly predicted ratings are the ones recommended to the user). While the majority of current recommender systems use a single numerical rating to represent each user's preference for a given item, recommender systems in some e-commerce settings have recently started adopting *multi-criteria* ratings that capture more precise information about user preferences with respect to different aspects of an item (e.g., capturing user ratings for the story and acting components of each movie in a movie recommender system). Some content-based recommender systems (Balabanovic and Shoham 1997) use multiple content attributes for item comparisons and similarity calculation, the subjective user preferences for an item are still captured by a simple overall rating. In contrast, my dissertation focuses on multi-criteria rating systems that allow users to rate a number of different aspects of an item. The single-criterion rating that a user gives to an item provides information regarding *how much* the user liked the item, while multi-criteria ratings provide some additional insights regarding *why* the user liked the item as much as she did. However, multi-criteria rating recommendation problems have remained largely untouched in the recommender system literature until recently. Consequently, to take full advantage of this additional information (i.e., multi-criteria ratings) in personalization applications, new recommendation techniques are required. The first challenge, therefore, is to *extend traditional single-rating recommendation techniques to incorporate multi-criteria ratings*, which may help improve recommendation *accuracy*.

Although recommendation accuracy is one of the most important measures of evaluating the performance of recommender systems, there exist other important measures, such as recommendation *diversity* (Herlocker et al. 2004, McNee et al. 2006, Shani and Gunawardana 2011). Since one of the implied goals of recommender systems is to provide a user with personalized or idiosyncratic items, more diverse recommendations provide more opportunities for users to obtain such recommended items. It is also important to note that diverse recommendations can be helpful not only for individual users, but also for businesses. For example, it is well-understood that it would be more profitable to Netflix if recommender systems could encourage users to rent a more diverse range of movies, especially from the "long-tail" (i.e., obscure items located in the tail of the sales distribution) because they are less costly to license and acquire from distributors, as opposed to providing a less diverse set of recommendations containing mainly new releases or highly-popular movies from well-known studios (Goldstein and Goldstein 2006). Therefore, the second challenge is to *develop new techniques that can improve recommendation diversity*. In particular, I focus on the notion of *aggregate* recommendation diversity, which can be measured as the total number of distinct items recommended across all users and represents a possible indicator of the level of personalization provided by the recommender system (e.g., whether all users receive the same exact recommendations, or whether each user receives unique recommendations tailored specifically to that user, or something in between). Traditional recommendation techniques typically recommend the most highly predicted items for each user, often resulting in a less diverse set of mostly popular items (Fleder and Hosanagar 2009). In contrast, I propose new approaches that *re-rank* candidate recommendations by factors other than the predicted rating value for better aggregate diversity, while still maintaining acceptable levels of accuracy.

While these re-ranking approaches are simple and efficient because they are based on scalable sorting-based heuristics, they do not provide direct control over diversity and, therefore, typically cannot achieve maximum possible diversity. As an extension of the work proposed from the second challenge, the third challenge is to *apply more sophisticated and systematic techniques to maximize diversity* improvements. In particular, I introduce *optimization*-based approaches for direct diversity maximization, including the following: a greedy maximization heuristic, a graph-theoretic approach based on maximum flow or maximum bipartite matching computations, and an integer programming approach.

Despite the need for higher recommendation diversity, it often comes at the expense of recommendation accuracy, or vice-versa, and the *simultaneous* improvement of both performance measures represents a non-trivial task. For example, high accuracy may often be obtained by safely recommending the most popular items, which can lead to a reduction in diversity (i.e., less personalized recommendations). Conversely, higher diversity can be achieved by uncovering and recommending highly personalized, idiosyncratic, and less popular items for each user, but these items are inherently more difficult to predict due to lack of data, and may lead to a decrease in recommendation accuracy. Therefore, the fourth challenge is to *explore the possibilities of overcoming the accuracy-diversity tradeoff*, thereby generating recommendations that are both more accurate and more diverse. I address this challenge by augmenting traditional recommender systems with techniques resulting from the first two challenges (i.e., incorporating multi-criteria ratings into traditional single-criterion recommender systems for better accuracy and by employing different recommendation ranking approaches for better diversity).

1.2 Research Objectives

The overarching research objective of my dissertation is to develop computational techniques to improve traditional single-rating recommender systems in terms of both accuracy and diversity of recommendations. Because of the inherent relationship between recommendation accuracy and diversity, it is often possible to increase one of these measures only at the expense of the other measure, so improving performance along both dimensions is not trivial. Therefore, in my dissertation, I first propose new approaches for each of the measures (i.e., accuracy or diversity) and then explore the possibility of combining these different approaches.

The specific research objectives of the dissertation are four-fold: [Study 1] developing new recommendation approaches that can incorporate multi-criteria rating information for more accurate recommendations; [Study 2] applying heuristic-based ranking approaches for more diverse recommendations; [Study 3] developing more sophisticated optimization approaches for direct diversity maximization; and [Study 4] combining the first two types of approaches, i.e., multi-criteria rating techniques and ranking approaches, in order to overcome the tradeoff between accuracy and diversity, thereby generating both more accurate and more diverse recommendations, as compared to traditional single-rating counterparts.

1.3 Overview of the Four Studies

1.3.1 Study 1: Incorporating Multi-Criteria Rating Information to Improve Recommendation Accuracy[2]

While traditional single-rating recommender systems have been successful in a number of personalization applications, the research area of *multi-criteria* recommender systems has been largely untouched. To take full advantage of multi-criteria ratings in various applications, new recommendation techniques are required. We propose two new approaches – a *similarity*-based approach and an *aggregation* function-based approach – to incorporate and leverage multi-criteria rating information in recommender systems. We also discuss multiple variations of each proposed approach, and perform an empirical analysis of these approaches using a real-world dataset. Our experimental results show that multi-criteria ratings can be successfully leveraged to improve recommendation accuracy, as compared to traditional single-rating recommendation techniques.

1.3.2 Study 2: Heuristic-Based Ranking Approaches to Improve Aggregate Recommendation Diversity[3]

Recommender systems are becoming increasingly important to individual users and businesses for providing personalized recommendations. However, while the majority of algorithms proposed in the recommender systems literature have focused on improving recommendation *accuracy* (as exemplified by the recent Netflix Prize competition), other important aspects of recommendation quality, such as the *diversity* of recommendations, have often been overlooked. We introduce and explore a number of item *ranking*

[2] Parts of this study have been published at *IEEE Intelligent Systems* (Adomavicius and Kwon 2007) as well as in a chapter of a book entitled "Recommender Systems Handbook: A Complete Guide for Research Scientists and Practitioners" (Adomavicius et al. 2011).

[3] Parts of this study were accepted for publication in *IEEE Transactions on Knowledge and Data Engineering* and are forthcoming (Adomavicius and Kwon 2011).

techniques that can generate substantially more diverse recommendations across all users, while maintaining comparable levels of recommendation accuracy. Comprehensive empirical evaluation consistently shows the diversity gains of the proposed techniques using several real-world rating datasets and different rating prediction algorithms.

1.3.3 Study 3: Optimization-Based Approaches to Maximize Aggregate Recommendation Diversity

Recommender systems help users find relevant items from a large set of alternatives in many online applications. Most existing recommendation techniques have focused on improving recommendation accuracy; however, diversity of recommendations has also been increasingly recognized in research literature as an important aspect of recommendation quality. This study proposes several optimization-based approaches for improving diversity of top-*N* recommendations, including: a greedy maximization heuristic, a graph-theoretic approach based on maximum flow or maximum bipartite matching computations, and an integer programming approach. The proposed approaches are evaluated using real-world movie rating datasets and demonstrate substantial improvements in both diversity and accuracy, as compared to the recommendation re-ranking approaches, which have been introduced in prior literature to improve diversity and are used for baseline comparisons in our study. The study also discusses the computational complexity and the scalability of the proposed approaches, as well as the potential directions for future work.

1.3.4 Study 4: Exploring Combined Approaches to Overcome the Accuracy-Diversity Tradeoff

Numerous recommendation algorithms have been developed to improve recommendation accuracy; however, recommendation diversity, which is another important aspect in evaluating recommender systems, has often been overlooked. Intuitively, it may be possible to achieve improvements in one of these two metrics at the expense of the other. For example, higher accuracy may be obtained by safely recommending the most popular ("bestselling") items, but this can lead to reduced aggregate recommendation diversity, i.e., less personalized recommendations. Conversely, higher diversity can be achieved by trying to uncover and recommend highly personalized, idiosyncratic items for each user, but these recommendations are inherently more difficult to predict, and thus may lead to a decrease in recommendation accuracy. To overcome this accuracy-diversity tradeoff, in this work, we build on the following two ideas: (a) that the incorporation of *multi-criteria* rating information into standard recommendation techniques can lead to improvements in recommendation accuracy; and (b) that employing various item *re-ranking* approaches can improve the aggregate diversity of recommendations. In particular, we propose a number of combined techniques that can build on any traditional recommendation algorithm by augmenting it with multi-criteria rating information and ranking-based approaches. We also empirically demonstrate how these techniques can generate both more accurate and diverse recommendations, as compared to their traditional counterparts.

The dissertation is organized in the following manner. Chapter 2 reviews relevant literature on traditional recommender systems and the evaluation of recommendation quality. The following four chapters consist of four studies. In Chapter 3, Study 1 presents two computational approaches that can take advantage of multi-criteria ratings to reveal more information on individual users' tastes and demonstrate improvements in accuracy, compared to typical recommendation algorithms with single-criterion ratings. Chapters 4 and 5 introduce Studies 2 and 3 which propose several heuristic-based ranking and optimization-based approaches, respectively, that can improve aggregate diversity, and provide their performance results in terms of both accuracy and diversity. Based on the findings from the first two studies, Chapter 6 presents Study 4 which explores new ways to overcome the accuracy-diversity tradeoff. Each of the four studies is presented in a self-contained manner, i.e., with its own introduction, related work, and conclusion. Chapter 7 summarizes the results and contributions of the four studies with a discussion of implications for future research.

Chapter 2. Literature Background

This chapter presents a brief overview of related topics in recommender systems literature. A typical recommendation process and the two popular recommendation algorithms that are used for empirical experiments in the dissertation are reviewed. While recommendation quality can be evaluated according to multiple dimensions (Herlocker et al., 2004; Shani and Gunawardana, 2011), the focus of this dissertation is on recommendation accuracy and diversity. Thus, prior literature on the two important dimensions in the evaluation of recommendation quality − accuracy and diversity − as well as the tradeoff between the two dimensions are discussed.

2.1 Recommendation Process

Recommender systems generally perform the following two tasks to provide recommendations to individual users: (1) unknown rating prediction and (2) recommendation generation. In most online applications, users often provide feedback using numeric rating values on the items that they have purchased or consumed. In the prediction phase, given the ratings that users have submitted for a subset of consumed items and possibly also information about item content or user demographics, a recommender system estimates ratings of items that the users have not yet consumed, using a recommendation algorithm. In the recommendation phase, the system then finds items that maximize the user's utility based on the predicted ratings, and recommends them to the user.

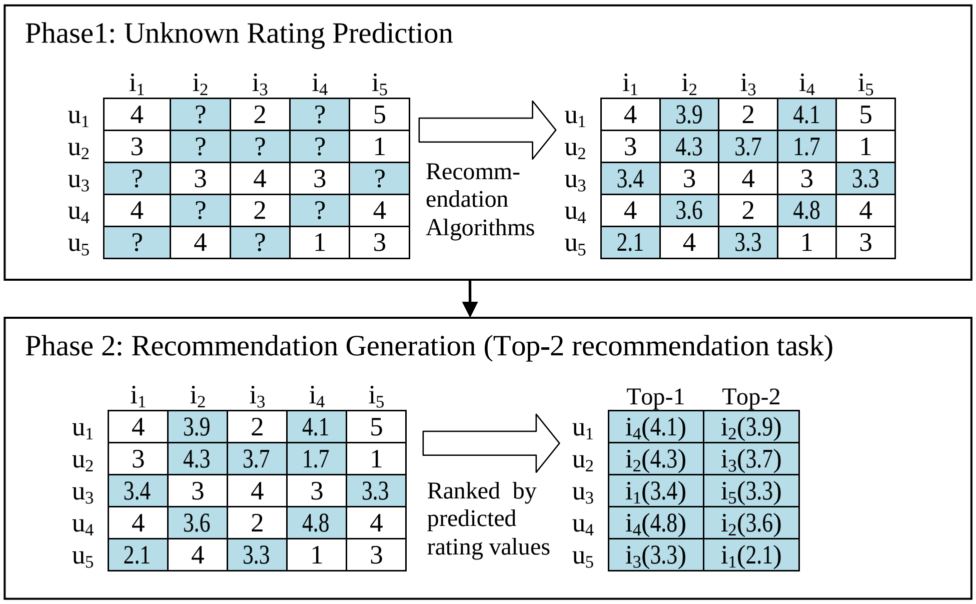
We formally define the two phases of typical recommender systems as follows. Let U be the set of users and I be the set of items available in the recommender system. Then, the utility function that measures the usefulness or utility of an item to a user can be defined as R: *Users*× *Items* → *Rating*, where *Rating* usually represents some numeric scale used by the users to evaluate each item (Adomavicius and Tuzhilin 2005). Thus, the job of the recommender system in the prediction phase is to estimate unknown ratings − $R^*(u,i)$, based on the known ratings − $R(u,i)$. Here, $R(u,i)$ represents the actual rating that user u gave to item i, and $R^*(u,i)$ represents the system-predicted rating for item i

that user u has not rated before. Given all of the predictions for each user, in the recommendation generation phase, the system selects the most relevant items, i.e., items that maximize a user's utility, according to a certain ranking criterion. Formally, item i_x is ranked ahead of item i_y (i.e., $i_x \prec i_y$) if $rank(i_x) < rank(i_y)$, where $rank: I \rightarrow \mathbf{R}$ is a function representing the ranking criterion. Typical recommender systems rank the candidate items by their *predicted rating values* and recommend the most highly predicted N items to each user because users are typically only interested in several of the most relevant recommendations. This is referred to as the *standard ranking approach* in this dissertation, and we define the ranking function as

$$rank_{\text{Standard}}(i)=R^*(u, i)^{-1}.$$

The power of -1 in the above expression indicates that the items with the *highest*-predicted (as opposed to the lowest-predicted) ratings $R^*(u, i)$ are the ones being recommended to the user.

The two-phase process for recommendations is depicted with an example in Figure 2.1. This dissertation proposes new recommendation approaches for both the prediction and recommendation generation phases (Chapter 3 for Phase 1, Chapters 4 and 5 for Phase 2, Chapter 6 for both Phase 1 and 2).



Phase1: Unknown Rating Prediction

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

	i_1	i_2	i_3	i_4	i_5
u_1	4	3.9	2	4.1	5
u_2	3	4.3	3.7	1.7	1
u_3	3.4	3	4	3	3.3
u_4	4	3.6	2	4.8	4
u_5	2.1	4	3.3	1	3

Phase 2: Recommendation Generation (Top-2 recommendation task)

	i_1	i_2	i_3	i_4	i_5
u_1	4	3.9	2	4.1	5
u_2	3	4.3	3.7	1.7	1
u_3	3.4	3	4	3	3.3
u_4	4	3.6	2	4.8	4
u_5	2.1	4	3.3	1	3

	Top-1	Top-2
u_1	i_4(4.1)	i_2(3.9)
u_2	i_2(4.3)	i_3(3.7)
u_3	i_1(3.4)	i_5(3.3)
u_4	i_4(4.8)	i_2(3.6)
u_5	i_3(3.3)	i_1(2.1)

Figure 2.1 Two-Phase Recommendation Process

2.2 Recommendation Algorithms for Rating Prediction

Recommender systems are usually classified according to their recommendation approach (Adomavicius an Tuzhilin 2005, Balabanovic and Shoham 1997):

- *Content-based* approaches recommend items similar to those the user preferred in the past.
- *Collaborative-filtering* (CF) approaches recommend items that users with similar preferences (i.e., “neighbors”) have liked in the past.
- *Hybrid* approaches combine content-based and collaborative-filtering methods in several different ways.

We can also classify recommender systems according to their algorithmic technique (Adomavicius an Tuzhilin 2005, Breese et al. 1998):

- *Heuristic-based* (or *memory-based*) techniques usually represent heuristics that calculate recommendations on the fly based directly on previous user activities, e.g., transactional data or rating values. One of the commonly used heuristic techniques is a neighborhood-based approach that finds nearest neighbors that have tastes similar to those of the target user (Breese et al. 1998, Delgado and Ishii 1999, Hill et al. 1995, Nakamura and Abe 1998, Resnick et al. 1994, Sarwar et al. 2001, Shardanand and Maes 1995).

- *Model-based* techniques use previous user activities to first learn a predictive model, typically using a statistical or machine-learning method, which the system then uses to make recommendations. Examples of model-based techniques include Bayesian clustering, aspect model, flexible mixture model, matrix factorization, and other methods (Breese et al. 1998, Hofmann 2003, Koren 2008, Koren et al. 2009, Si and Jin 2003, Su and Khoshgoftaar 2006, Takács et al. 2009).

While the recommendation approaches proposed in this dissertation can be used in conjunction with any recommendation algorithms, for the empirical tests, we use the two most popular and widely employed CF techniques for rating prediction: a heuristic neighborhood-based technique and a model-based matrix factorization technique, which will be explained in more detail in the following two subsections.

2.2.1 Neighborhood-Based Collaborative Filtering Techniques

There are multiple variations of neighborhood-based CF techniques (Breese et al. 1998, Resnick et al. 1994, Sarwar et al. 2001). Specifically, there are several ways to compute the similarity *sim*(*u*, *u'*) between two users, including *cosine-based* and *correlation-based* computations (Breese et al. 1998) and a number of other approaches. I use the cosine-based similarity measure in this dissertation since it is one of the most commonly used measure for determining how similar two users are in memory-based collaborative filtering algorithms. Assuming *I*(*u*, *u'*) represents the set of all items rated by both users *u* and *u'*, the cosine-based similarity can be calculated as follows (Breese et al. 1998, Sarwar et al. 2001):

$$sim(u,u') = \frac{\sum_{i \in I(u,u')} R(u,i) \cdot R(u',i)}{\sqrt{\sum_{i \in I(u,u')} R(u,i)^2} \sqrt{\sum_{i \in I(u,u')} R(u',i)^2}} \, . \qquad (2.1)$$

Based on the similarity calculation, set *N*(*u*) of the nearest neighbors of user *u* is obtained. The size of set *N*(*u*) can range anywhere from 1 to |*U*|-1, i.e., all other users in the dataset. Furthermore, *R**(*u*, *i*) – the rating that user *u* would give to item *i* – can be computed as the weighted average of all known ratings *R*(*u'*, *i*), where *u'*∈ *N*(*u*) (i.e., user *u'* is “similar” to *u*). Two popular ways to compute this weighted average are as follows (Delgado and Ishii 1999, Nakamura and Abe 1998):

- Weighted sum approach, i.e.,

$$R^*(u,i) = \frac{\sum_{u' \in N(u)} sim(u,u') \cdot R(u',i)}{\sum_{u' \in N(u)} | sim(u,u') |} \; ; \qquad (2.2)$$

- Adjusted weighted sum approach, i.e.,

$$R^*(u,i) = \overline{R(u)} + \frac{\sum_{u' \in N(u)} sim(u,u') \cdot \left(R(u',i) - \overline{R(u')}\right)}{\sum_{u' \in N(u)} | sim(u,u') |} \, . \qquad (2.3)$$

Here $\overline{R(u)}$ represents the average rating of user *u*. Since the value of rating *R*(*u'*, *i*) is weighted by the similarity of user *u'* to user *u*, the more similar the two users are, the more weight $R(u', i)$ will have in the computation of rating $R(u, i)$. Limiting the neighborhood size $N(u)$ to a specific number will determine how many similar users will be used in the computation of rating $R(u, i)$.

In addition, because of the inherent symmetry between users and items in the traditional memory-based collaborative filtering setting, the neighborhood-based approach can be either *user-based* or *item-based*, depending on whether we want to calculate the similarity between users or items. Equations (2.1), (2.2), and (2.3) represent the user-based approach, but they can be rewritten in a straightforward way for the item-based approach. For example, the item-based adjusted weighted sum can be calculated as follows (Sarwar et al. 2001):

$$R^*(u,i) = \overline{R(i)} + \frac{\sum_{u' \in N(u)} sim(i,i') \cdot \left(R(u,i') - \overline{R(i')}\right)}{\sum_{i' \in N(i)} | sim(i',i') |} \quad . \tag{2.4}$$

Here $\overline{R(i)}$, $sim(i, i')$, and $N(i)$ are analogous to their user-based counterparts. Both user-based and item-based approaches are used for rating estimation in empirical experiments.

2.2.1 Matrix Factorization Collaborative Filtering Techniques

Matrix factorization techniques have been widely used in numerical linear algebra, dating back to the 1970s (Gabriel and Zamir 1979, Golub and Reinsche 1970, Klema and Laub 1980) and have recently gained popularity in recommender system applications because of their effectiveness in improving recommendation accuracy (Sarwar et al. 2000b, Srebro and Jaakkola 2003, Wu 2007, Zhang et al. 2005). Many variations of matrix factorization techniques have been developed to solve the problems of data sparsity, overfitting, and convergence speed, and they turned out to be a crucial component of many well-performing algorithms in the popular Netflix Prize competition (Funk 2006, Greene 2006, Koren 2008, Koren et al. 2009, Takács et al. 2009).

With the assumption that a user's rating for an item is composed of a sum of preferences about the various features of that item, this model is induced by Singular Value Decomposition (SVD) on the user-item ratings matrix. In particular, using K features (i.e., rank-K SVD), user u is modeled as a user-factors vector p_u (the user's preferences for *K* features), and item *i* is associated with an item-factors vector q_i (the item's importance weights for *K* features). The preference of how much user *u* likes item *i*, denoted by *R**(*u*, *i*), is predicted by taking an inner product of the two vectors, i.e.,

$$R^*(u,i) = p_u^T q_i \,. \qquad (2.5)$$

Among the several methods to compute user- and item-factor vectors, we use a simple gradient descent technique, as presented in Funk 2006, in empirical experiments. With this technique (2.6), all values in user- and item-factor vectors are initially assigned arbitrary numbers, and the prediction error (*err*) is computed for each rating in the training set. Then, user- and item-factor vectors are iteratively updated with the learning rate parameter (θ) as well as the regularization parameter (λ, which is used to minimize overfitting), until the minimum improvement in predictive accuracy or a pre-defined number of iterations is reached. One learning iteration is defined as:

```
For each rating $R(u, i)$
    $err = R(u,i) - p_u^T q_i$
    $p_u = p_u + \theta(err \times q_i - \lambda \times p_u)$        (2.6)
    $q_i = q_i + \theta(err \times p_u - \lambda \times q_i)$
End For
```

Finally, unknown ratings are estimated using the final two vectors p_u and q_i, as stated in (2.5). More details on variations of matrix factorization techniques used in recommender systems can be found in (Funk 2006, Koren 2008, Koren et al. 2009, Takács et al. 2009, Wu 2007, Zhang et al. 2005).

2.3 Accuracy of Recommendations

Numerous recommendation techniques have been developed over the last few years, and various metrics have been employed for measuring the accuracy of recommendations, including statistical accuracy metrics and decision-support measures (Herlocker et al. 2004, Shani and Gunawardana 2011).

As discussed in Section 2.1, recommender systems typically attempt to predict the ratings of unknown items for each user, often using other users' ratings, and then they recommend the top N items with the highest predicted ratings. Accordingly, many studies have developed new algorithms that can improve the predictive accuracy of recommendations. Several *statistical accuracy* metrics, such as mean absolute error (MAE) and root mean squared error (RMSE), are currently used to measure predictive accuracy, i.e., how well a system can predict an exact rating value for a specific item.

The focus of this dissertation is to generate good top-N recommendation lists in terms of both accuracy and diversity, and we chose to use *decision-support* metrics to evaluate how effective a recommender system is at helping users select the most relevant items from the set of all items. Examples of decision-support metrics include precision (the percentage of truly "relevant" items among those that were predicted to be "relevant" by the recommender system), recall (the percentage of correctly predicted "relevant" items among all the ratings known to be "relevant"), and F-measure, which is a harmonic mean of precision and recall. In our experiments, we evaluate the accuracy of top-N recommendation lists using one of the most popular decision-support metrics, *precision*. Simply put, precision can be measured as a proportion of "relevant" items among the recommended items across all users.

Note that the decision-support metrics typically work with binary outcomes; therefore, here the notion of "relevance" is used to convert a numeric rating scale into a binary scale (i.e., relevant vs. irrelevant). More specifically, our empirical data ratings are provided on either a 13-point (A+ to F) or a 5-point (or 5-star) scale, and the natural assumption is that users provide higher ratings for items that are the most relevant to their desires. As a consequence, in our experiments, we treat items with ratings between 11 and 13 (A+, A, A- on a 13-point scale) or 4 and 5 (one a 5-point scale) as relevant, and items with lower ratings as irrelevant, or more precisely, we use the threshold between relevant and irrelevant items as 10.5 or 3.5 (relevance threshold, denoted by T_H).

The list of N items recommended for user u should include only items predicted to be relevant and can be formally defined as $L_N(u) = \{i_1, i_2, \ldots, i_N\}$, where $R^*(u, i_k) \geq T_H$ for all

$k \in \{1, 2, \ldots, N\}$. The precision of such top-N recommendation lists, often referred to as *precision-in-top-N*, is calculated as the percentage of truly "relevant" items, denoted by $correct(L_N(u)) = \{i \in L_N(u) \mid R(u, i) \geq T_H\}$ among the items recommended across all users, and can be formally written as follows:

$$precision\text{-}in\text{-}top\text{-}N = \sum_{u \in U} |correct(L_N(u))| \Big/ \sum_{u \in U} |L_N(u)| \cdot$$

In real-world settings, obviously, a recommender system has to be able to recommend items that users have not yet rated since the ratings for those items typically become available to the system only after item consumption, i.e., the true precision of the generated recommendation lists is not known at the time of recommendation. However, not surprisingly, we found that precision is highly correlated with average predicted rating values of recommended items, as shown in Figure 2.2. In particular, we ran three popular CF recommendation algorithms discussed in Section 2.2 on two real-world datasets, MovieLens and Netflix datasets (details about these datasets are provided in Section 4.4.1), using standard cross-validation techniques from machine learning and data mining (Mitchell 1997). Recommending items with higher predicted rating values results in higher precision, i.e., higher likelihood that the user would actually like the item, which also provides further empirical support for using the standard ranking approach (as defined in Section 2.1) if the goal is just to maximize recommendation accuracy. An important consequence of this relationship is that we can use the average predicted rating value of the top-N recommendation lists, which can always be computed at the time of recommendation, as a simple proxy for the precision metric. In addition, this metric is extremely simple to compute and easily scales to large-scale, real-world applications. In particular, this metric is used for large-scale data experiments in Chapter 5. We refer to this metric as *prediction-in-top-N* and formally define it as follows:

$$prediction\text{-}in\text{-}top\text{-}N = \sum_{u \in U} \sum_{i \in L_N(u)} R^*(u,i) \Big/ \sum_{u \in U} |L_N(u)| \, .$$

Relying on the accuracy of recommendations alone, however, may not be enough to find the most relevant items for a user. It has often been suggested that recommender systems must be not only accurate, but also useful (Herlocker et al. 2004, McNee et al. 2006). For example, McNee et al. (2006) suggests new user-centric directions for evaluating recommender systems beyond the conventional accuracy metrics. They claim that serendipity in recommendations or user experiences and expectations should also be considered in evaluating the recommendation quality. Among many different aspects that cannot be measured by accuracy metrics alone, we focus on the notion of *diversity* of recommendations, which is discussed next.

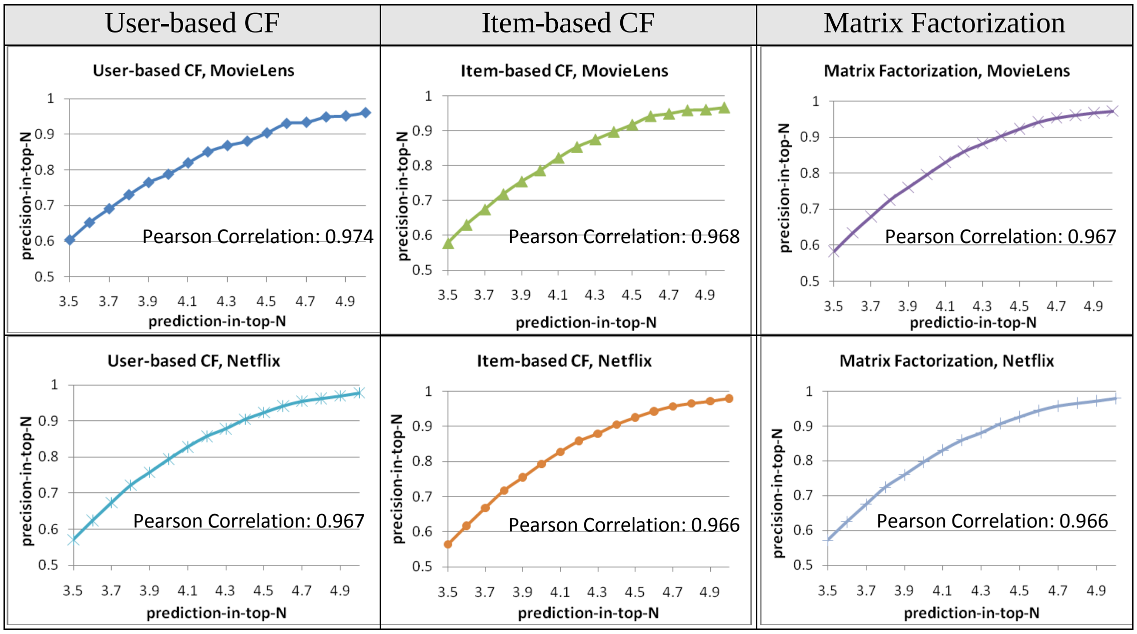


Figure 2.2 Precision versus Average Predictive Rating Values

2.4 Diversity of Recommendations

The importance of diverse recommendations has been emphasized in several studies (Bradley and Smyth 2001, Brynjolfsson et al. 2007, 2010, Fleder and Hosanagar 2009, Hu and Pu 2011, McSherry 2002, Oestreicher-Singer and Sundararajan 2011, Smyth and McClave 2001, Zhang 2009, Zhang and Hurley 2008, Ziegler et al. 2005). In these studies, the diversity of recommendations has been assessed at either the individual or the aggregate level.

2.4.1 Individual Diversity

Most previous studies have focused on *individual* diversity (Bradley and Smyth 2001, Hu and Pu 2011, McSherry 2002, Smyth and McClave 2001, Zhang and Hurley 2008, Zhang 2009, Ziegler et al. 2005). These techniques aim to avoid providing too similar recommendations for the same user. For example, some studies (Bradley and Smyth 2001, Smyth and McClave 2001, Ziegler et al. 2005) have used an intra-list similarity metric, i.e., calculating the average similarity between all pairs of items recommended to a user (e.g., based on item attributes), to determine the individual diversity. Hu and Pu (2011) measured users' diversity perceptions of the recommendation lists by asking two questions: whether the items recommended to a user are of various kinds (categorical diversity) and whether the items recommended to a user are similar to each other (item-to-item diversity). Alternatively, Zhang and Hurley (2008) used a new evaluation metric, item novelty, to measure the amount of additional diversity that one item brings to a list of recommendations. Moreover, the loss of accuracy, resulting from the increased diversity, is controlled by changing the granularity of the underlying similarity metrics in the diversity-conscious algorithms (McSherry 2002).

2.4.2 Aggregate Diversity

In contrast to *individual* diversity, *aggregate* diversity of recommendations across all users has been relatively less studied, and a recent interest in the impact of recommender systems on product variety and sales concentration patterns (Brynjolfsson et al. 2007, 2010, Fleder and Hosanagar 2009, Oestreicher-Singer and Sundararajan 2011) has sparked a renewed interest in this topic.

As observed by Brynjolfsson et al. (2010), recommender systems can play a key role in increasing both "long-tail" and "superstar" effects in real-world e-commerce applications. In particular, the "long-tail" literature argues that recommendations on the Internet help increase users' awareness of niche products and create a long tail in the distribution of product sales (Anderson 2006, Brynjolfsson et al. 2007, Fleder and Hosanagar 2009, Oestreicher-Singer and Sundararajan 2011). For example, one study, using data from an online clothing retailer, demonstrated that recommendations increase sales of items in the long tail, resulting in improved aggregate diversity (Brynjolfsson et al. 2007). In contrast, the “superstar” literature indicates that recommender systems may promote the so-called “rich get richer” phenomenon, where more popular/bestselling items are recommended compared to idiosyncratic/personalized ones. One explanation for this is that niche products often have limited historical data and, thus, are more difficult to recommend to users, whereas popular products typically have more ratings and, thus, can be recommended to more users (Fleder and Hosanagar 2009, Leonard, 2010, Thompson 2008). For example, some results from the Netflix Prize competition show that recommender systems inherently tend to generate relevant but safe recommendations and avoid extremes, by choosing the items among the common items recommended to many users (Thompson 2008).

While the debate has focused on the potential impact of recommender systems on product demand and sales revenue by examining sales data on sites such as Amazon.com, recent design science research has studied how to design recommender systems to take advantage of the long-tail phenomenon, i.e., intentionally generating more niche or idiosyncratic recommendations across all users and, in turn, increasing sales of long-tail items (Kim et al. 2010, Levy and Bosteels 2010, Park and Tuzhilin 2008).

More diverse recommendations, presumably leading to more sales of long-tail items, could be beneficial for both individual users and some business models (Brynjolfsson et al. 2003, 2006, 2009, 2010, Goldstein and Goldstein 2006). For example, diverse recommendations can help to cultivate consumers’ tastes for niche products (Brynjolfsson et al. 2006). Another consequence of this long-tail phenomenon is that an increase in the product variety available to consumers enhanced consumer surplus, and this gain has become even larger over time, as shown by the comparison of Amazon’s book sales in 2000 vs. 2008 (Brynjolfsson et al. 2003, 2009). Exposing individual consumers to more long-tail recommendations can intensify this effect. Thus, more consumers would be attracted to companies that carry a large selection of long-tail items and have long-tail strategies, such as providing more diverse recommendations (Brynjolfsson et al. 2010).

While more diverse recommendations would be helpful for individual users, they could be beneficial for some business models, such as the one used by Netflix, because more diverse recommendations would encourage users to rent more long-tail movies, which are less costly to license and acquire from distributors than new releases or extremely popular movies of big studios (Goldstein and Goldstein 2006). In fact, 70 percent of Netflix rentals come from the company's back catalog rather than recent releases (Flynn 2006), and it would be critical to allow users to access more long-tail items in this type of business model.

Taking into consideration the potential benefits of *aggregate* diversity to individual users and businesses, several studies have explored new methods that can increase the diversity of recommendations (Kim et al. 2010, Levy and Bosteels 2010, Park and Tuzhilin 2008). In particular, prior work can be divided into two lines of research since recommender systems typically compute recommendations for users in two phases: (1) estimating ratings of items that the users have not consumed yet and (2) generating top-*N* items for each user. Most current research (Kim et al. 2010, Levy and Bosteels 2010, Park and Tuzhilin 2008) aims to enhance the estimation phase (mainly for long-tail items). For example, Park and Tuzhilin (2008) propose new clustering methods to improve predictive accuracy of long-tail items that have few ratings, which can also increase the recommendation of long-tail items. In addition, Levy and Bosteels (2010) designed long-tail music recommender systems, simply by removing popular artists (i.e., those with more than 10,000 listeners) in the rating prediction phase. In addition, a local scoring model, proposed by Kim et al. (2010), was developed to alleviate the scalability and sparsity problems by suggesting a more efficient way to select the best neighbors for neighborhood-based recommendation techniques; however, as a by-product, it was shown to improve aggregate recommendation diversity. In contrast to these studies, this dissertation focuses on finding the best set of recommendations in the recommendation generation phase, i.e., proposes a flexible solution to improve recommendation diversity in conjunction with a number of different rating prediction techniques because they are applied *after* the unknown item ratings have been estimated.

Several metrics can be used to evaluate various aspects of aggregate diversity, including absolute long-tail metrics that measure the change in the absolute number of items recommended (e.g., recommendation frequency of items above a certain popularity rank), relative long-tail metrics to measure the relative share of recommendations above or below a certain popularity rank percentile, and the slope of the log-linear relationship between item popularity rank and recommendations (or sales) that can indicate the relative importance of the head versus the tail of the distribution (Brynjolfsson et al. 2010). In the recommender systems literature, both absolute and relative long-tail metrics have been used to measure the aggregate diversity of recommendations (Herlocker et al. 2004, Kim et al. 2010, Levy and Bosteels 2010). In this dissertation, a simple, absolute long-tail metric is used to measure aggregate diversity, using the total number of distinct items among the top-*N* items recommended across all users, referred to as the *diversity-in-top-N*. This metric is formally defined as

$$diversity\text{-}in\text{-}top\text{-}N = \left| \bigcup_{u \in U} L_N(u) \right| .$$

Note that high individual diversity of recommendations does not necessarily imply high aggregate diversity. For example, if the system recommends to all users the same five best-selling items that are not similar to each other, the recommendation list for each user is diverse (i.e., high individual diversity), but only five distinct items are recommended to all users and purchased by them (i.e., resulting in low aggregate diversity or high sales concentration). Furthermore, this diversity metric could also potentially be viewed as a crude indicator of the system's level of personalization, because high diversity implies that each user receives a very different and unique set of recommendations (potentially indicating a high level of personalization), but low diversity indicates that mostly the same items (possibly bestsellers) are recommended to all users (i.e., a low level of personalization). We also show that this simple and easy-to-compute metric exhibits a high correlation with more sophisticated, distributional diversity metrics in Section 4.5.3, i.e., this metric is able to capture the same diversity dynamics as some of the relative long-tail metrics on several real-world rating datasets.

Although several approaches proposed in this dissertation aim to improve aggregate recommendation diversity, their accuracy is also given the proper attention, because diverse but inaccurate recommendations may not provide significant value to users.

2.5 Tradeoff between Accuracy and Diversity

As discussed earlier, there is a tradeoff between accuracy and diversity because recommending only popular items, such as blockbuster movies that many users tend to like, could obtain relatively high accuracy, but could also lead to a decline of other aspects of recommendations, including recommendation diversity. Conversely, higher diversity can be achieved by trying to uncover and recommend highly idiosyncratic or personalized items for each user, which often have less data and whose ratings are inherently more difficult to predict, and, thus, may lead to a decrease in recommendation accuracy. This inherent tradeoff between accuracy and diversity has been observed in previous studies (McSherry 2002, Zhang and Hurley 2008, Ziegler et al. 2005) indicating that maintaining accuracy while improving diversity constitutes a difficult task.

We provide an example of the accuracy and diversity tradeoff in two extreme cases where only popular items or long-tail type items are recommended to users, using the MovieLens rating dataset (which will be described in Section 4.4.1). In this example, the item-based collaborative filtering technique (refer to Section 2.2.1) is used to predict unknown ratings. Then, as candidate recommendations for each user, we considered only the items that were predicted above the pre-defined relevance threshold, to assure an acceptable level of accuracy, as is typically done in recommender systems. Among these candidate items for each user, we identified items that were rated by most users (i.e., items with the largest number of known ratings) as popular items, and items that were rated by the least number of users (i.e., items with the smallest number of known ratings) as long-tail items. As illustrated by the results of the top-1 recommendation task in Table 2.1, if the system recommends the most popular item (among the ones that had a sufficiently high predicted rating), it is far more likely for many users to get the same recommendation (e.g., the best-selling item). The accuracy measured by the *precision-in-top*-1 metric (i.e., the percentage of truly "high" ratings among those that were predicted to be “high” by the recommender system) is 82%, but only 49 popular items out of approximately 2,000 available distinct items were recommended across all users (2,828 users in total). The system can improve the diversity of recommendations from 49 up to 695 items (a 14-fold increase) by recommending long-tail items to each user (i.e., the least popular items among highly predicted items for each user) instead of popular items. However, high diversity in this case is obtained at a significant expense to accuracy, i.e., a drop from 82% to 68%.

Table 2.1 Accuracy-Diversity Tradeoff: Empirical Example

Quality Metric: Top-1 recommendation of:	Accuracy	Diversity
Popular Items (items with the largest number of known ratings)	82%	49 distinct items
“Long-Tail” Items (items with the smallest number of known ratings)	68%	695 distinct items

Note: Recommendations (top-1 item for each user) are generated for 2,828 users among the items that are predicted above the acceptable threshold 3.5 (out of 5), using a standard item-based collaborative filtering technique with 50 neighbors on the MovieLens Dataset.

The above example shows that it is possible to obtain higher diversity simply by recommending less popular items; however, the loss of recommendation accuracy in this case can be substantial. Therefore, more exploration of new recommendation approaches is necessary to increase the diversity of recommendations with only minimal (negligible) accuracy loss or to increase both diversity and accuracy. This perspective describes the motivation of proposing new recommendation approaches in this dissertation (particularly for Chapters 4, 5, and 6).

Chapter 3. Incorporating Multi-Criteria Rating Information to Improve Recommendation Accuracy

3.1 Introduction and Motivation

The vast majority of current recommender systems typically use a single criterion (i.e., a single numerical rating) to represent the utility of an item to a user in a two-dimensional $\mathit{Users} \times \mathit{Items}$ space. Single-rating recommender systems have proved successful in several applications, but many industries have begun employing multi-criteria systems. For example, restaurant guides, such as Zagat's Guide, provide three criteria for restaurant ratings (e.g., food, décor, and service). Online shopping malls, such as Bestbuy.com and Buy.com, use multi-criteria ratings for consumer electronics (e.g., display, performance, battery life, and cost). However, these rating systems are not used in the context of personalization in the context of personalization; the rating on each criterion—for example, Zagat's "food" rating for a specific restaurant—is the same for all users. In contrast, Yahoo! Movies provides a movie recommendation service that employs user-specific multi-criteria ratings for each movie. This move indicates that multi-criteria data provides value to online content providers and consumers as a component in personalization applications. Taking full advantage of multi-criteria ratings in personalization applications requires new recommendation techniques. In this chapter, we propose several new techniques for extending recommendation technologies to incorporate and leverage *multi-criteria rating* information.

The remainder of the chapter is organized as follows. Section 3.2 reviews literature on multi-criteria problems in operation research, decision science, and some other fields, as well as personalization literature. Several new multi-criteria rating techniques are proposed in Section 3.3, followed by the main empirical results in Section 3.4. Section 3.5 concludes the chapter by summarizing the contributions.

3.2 Related Work

Multi-criteria problems have been studied extensively in operation research and the decision science fields. The majority of engineering problems are essentially multi-criteria optimization problems (Statnikov and Matusov 1995). For example, when an airplane is being designed, its reliability, longevity, efficiency, cost, and the combination of other utilization factors need to be considered. Typical methods to solve the multi-criteria optimization problems include: finding Pareto optimal solutions; optimizing the most important criterion and converting other criteria to constraints; consecutively optimizing one criterion at a time, converting an optimal solution to constraints, and repeating the process for other criteria.

The decision science field treats organizational decision making as a multi-criteria problem, i.e., it considers various points of view, such as financial, human resources-related, and environmental aspects in making a decision (Figueria et al. 2005). The objective of multi-criteria decision analysis is to assist a decision maker in choosing the best alternative when there is multiple criteria conflict that compete with each other. Most commonly used decision aiding methods, such as outranking methods and the analytical hierarchy process, are based on multi-criteria aggregation procedures. Outranking methods determine which alternatives are preferred to others by systematically comparing possible alternatives for each criterion. The analytical hierarchy process structures multi-criteria into a hierarchy and calculates the score of each criterion as a weighted sum of its sub-criteria.

Similarly, in marketing research literature, buying a product can also be regarded as a multi-criteria decision problem. For example, when we purchase a car, we consider its multiple attributes, such as price, brand, and color. The conjoint model is the most commonly used technique for solving multi-criteria problems in this field (Green et al. 2001). This model determines the importance weights of product attributes and the values of the attributes. The customers' preference for the product can then be calculated as a linear combination of weights and values.

Multi-criteria information is also used in certain electronic market mechanisms, such as multi-attribute auctions (Bichler 2000). Multi-attribute auctions are typically used in procurement settings and enable auction participants to negotiate not only on price, but also on other attributes of a deal such as quality level, style, delivery date. It has been demonstrated that multi-attribute auctions have several advantages over their single-attribute (i.e., price-only) counterparts, including improvements in the overall utility and suitability for various application domains (Bichler 2000).

The multi-criteria problems addressed in the above-mentioned fields, however, typically are not intended for personalization and recommendation settings. These problems find the solutions or items that are optimal, in general, for all users, and differences in individual user preferences are not explicitly considered. Multi-criteria rating problems have started receiving attention in recommender systems research and are regarded as an important issue for the next generation of recommender systems (Adomavicius and Tuzhilin 2005). In recommender systems literature, the roots of multi-criteria ratings could be traced to the approaches that started incorporating content-based features into collaborative filtering recommendation techniques. This allowed the recommender systems to identify favorite content attributes (e.g., “comedy” movies) based on the content analysis of the previously rated items, and then to also recommend items to a user based not only on the ratings of similar users, but also based on these favorite content attributes (Balabanovic and Shoham 1997). However, the users could submit only a single rating for each item, and could not specify their individualized feedback about a specific movie component/aspect (such as the movie’s visual effects).

There has also been some research on providing recommendation *filtering* capabilities based on item content information. For example, Schafer (2005) implemented a meta-recommendation system that allows users to indicate the preference for each content attribute (e.g., movie genre, Motion Picture Association of America rating, or film length) and rate the importance of these attributes. For example, users can indicate that they want only “comedy” movies, and that it is the most important condition for recommendations. Thus, the users’ requirements will filter the potential recommendations towards what the users really want. Note, however, that this does not represent a multi-criteria rating environment, since the users are specifying general filtering requirements for all movies, such as specifying the preferred value and weight for movie genre attribute. Similarly, Lee et al. (2002) also obtained the importance weights of content attributes directly from users. They used each attribute's rank to compare the items, but the value or rank of each attribute was assumed to be the same for all users.

In contrast to Schafer (2005) and Lee et al. (2002), in multi-criteria rating environments users would be able to specify subjective ratings for various components of *individual* items (e.g., to rate the visual effects component for the "Star Wars" movie), which could then be leveraged for prediction and personalization purposes. One example of such system is the Intelligent Travel Recommender system (Ricci et al. 2002), where users can rate multiple travel items within a "travel bag" (e.g., location, accommodation, etc.) as well as the entire travel bag. Then, candidate travel plans are ranked according to these user ratings, and the system finds the best match between recommended travel plans and the current needs of a user.

In summary, although the personalization literature includes several approaches that are somewhat related to the issue of incorporating and leveraging multi-criteria ratings in recommender systems (Adomavicius et al. 2011), it is fair to say that this issue has remained largely unexplored and needs to be further investigated.

3.3 Extending Recommender Systems to Incorporate Multi-Criteria Ratings

Before proceeding with a discussion on new recommendation techniques for multi-criteria rating settings, we briefly describe one of the traditional and commonly used single-rating collaborative recommendation techniques. The recommendation process starts with the specification of the initial set of ratings that is either explicitly provided by the users or is implicitly inferred by the system. For example, in case of a movie recommender system, user John Doe may assign a rating of 11 (out of 13) for the movie

"Black Swan," so it becomes set R(John Doe, Black Swan) = 11. Once these initial ratings are specified, a recommender system attempts to estimate the rating function R

$$R\text{: } \mathit{Users} \times \mathit{Items} \rightarrow R_0 \qquad (3.1)$$

for the (user, item) pairs that have not been rated yet. R_0 is usually represented by a totally ordered set such as integers or real numbers within a certain range. Once function R is estimated, a recommender system can recommend the highest-rated item (or a set of N highest-rated items) for each user.

Besides the overall rating, multi-criteria ratings provide additional information about user preferences regarding several important aspects/components of an item. Therefore, leveraging this additional information in recommender systems should be beneficial, since it can potentially increase the accuracy of the recommendations. The goal of multi-criteria recommender systems is to find items that maximize each user's utility, just as single-rating recommender systems. Therefore, the systems should also be able to predict the overall rating of each item for each user, because the system ultimately needs to compare the items based on their overall ratings and recommend the best items to users. The difference between single-rating and multi-criteria rating systems is that the latter have more information about the users and items, which can be effectively used in the rating prediction (the first phase of the recommendation process, as described in Section 2.1). More formally, the general form of a rating function in a multi-criteria recommender system is:

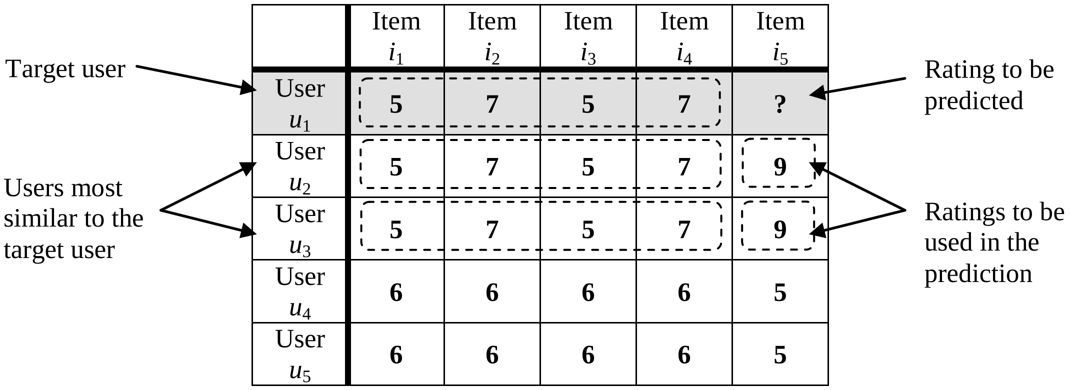
$$R\text{: } \mathit{Users} \times \mathit{Items} \rightarrow R_0 \times R_1 \times \ldots \times R_k \qquad (3.2)$$

where R_0 is the set of possible overall rating values, and R_i represents the possible rating values for each individual criterion i (i = 1, …, k), typically on some numeric scale (e.g., from 1 to 13).

In the remainder of this section we propose two new recommendation approaches and present several different variations of each. The first approach extends the traditional single-criterion neighborhood-based collaborative filtering algorithm, while the second approach has no restriction to any specific algorithm. In other words, it can use any existing single-criteria recommendation algorithm—content-based, collaborative, or hybrid.

3.3.1 Similarity-Based Approach to Extending Standard Collaborative Filtering Techniques

Consider a movie recommendation application, where users provide the recommender system with a single rating (between 1 and 13) for each movie they have seen. Suppose also that this recommender system uses a traditional user-based collaborative filtering approach for rating prediction. In particular, among the collaborative filtering algorithms described in Section 2.2, we use *user-based adjusted weighted sum* approach (2.3) along with the *cosine-based* similarity function (2.1), which will be referred to as the "standard collaborative filtering approach" throughout the chapter. Using Equations (2.1) and (2.3), the system would estimate any rating that user *u* would give to yet-unseen movie *i* according to how users *u'* who are similar to target user *u* rated movie *i*. In other words, the system calculates unknown rating $R^*(u, i)$ on the basis of ratings $R(u', i)$. So, the more accurately the system determines who the "true peers" (or "nearest neighbors") of *u* are, the more accurate the rating prediction should be. The traditional two-dimensional collaborative filtering system calculates the similarity between users *u* and *u'* on the basis on how similar their ratings are for the movies they have *both* seen.


Target user
Users most similar to the target user
Rating to be predicted
Ratings to be used in the prediction

	Item i_1	Item i_2	Item i_3	Item i_4	Item i_5
User u_1	**5**	**7**	**5**	**7**	**?**
User u_2	**5**	**7**	**5**	**7**	**9**
User u_3	**5**	**7**	**5**	**7**	**9**
User u_4	**6**	**6**	**6**	**6**	**5**
User u_5	**6**	**6**	**6**	**6**	**5**

Figure 3.1 Collaborative Filtering in a Single-Rating Setting

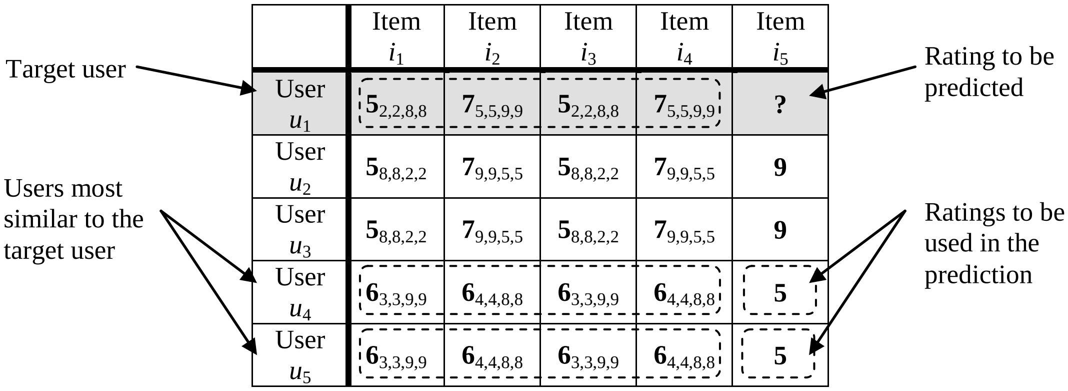
Figure 3.1 illustrates this estimation process with a simple example. Assume that we have five users u_1, …, u_5 and five movies i_1, …, i_5, and suppose that the recommender system needs to estimate how much the target user u_1 would like movie $i_{5.}$ Furthermore, as Figure 3.1 indicates, suppose that all other ratings of different users to different movies are known. The traditional collaborative filtering approach finds the users that are closest to u_1 and that have seen movie i_5. In this case, u_2 and u_3 seem to be "perfect matches" for user u_1, since all of them rated the common movies exactly the same (see Figure 3.1). Since both u_2 and u_3 rate movie i_5 as 9, the value of target rating $R^*(u_1, i_5)$ will be predicted as 9.

Now let's consider the same scenario as above, but in a multi-criteria setting. Specifically, let's assume that we have the same five users u_1, …, u_5 and five movies i_1, …, i_5, and unknown rating $R^*(u_1, i_5)$ that must be predicted, and known overall ratings of all users to different movies that are exactly the same as in Figure 3.1. In addition, however, assume that each user is also asked to provide feedback about a movie on four specific criteria—story, acting, direction, and visuals,[4] and that the overall rating in this case is a simple average of the four individual criteria ratings.

Following the idea behind the standard collaborative filtering approach, in order to predict $R^*(u_1, i_5)$ the recommender system should find the users that are closest to u_1 and that have seen movie i_5. However, the additional information available in the multi-criteria ratings shown in Figure 3.2 makes it clear that users u_2 and u_3 are quite different in their tastes and preferences from user u_1, even though their overall ratings for each movie match perfectly. In particular, user u_1 disliked the movie aspects (story and acting) that u_3 and u_3 liked and liked the aspects (direction and visuals) they disliked. However, in recommender systems that are based on single-criteria ratings, this information would be "hidden" within the aggregate overall rating, which may lead to inaccurate insights about the true similarity between user preferences (as in this example). Users u_4 and u_5 seem to be much better matches for user u_1 in this example. Not only their overall ratings are similar but also are their preferences for different movie aspects. Both u_4 and u_5 rate

[4] As is done on some movie review websites, such as Yahoo! Movies (http://movies.yahoo.com).

movie i_5 as 5, so the system would predict a value of 5 for the target rating $R^*(u_1, i_5)$. This outcome is very different from the one obtained in a single-rating scenario.



	Item i_1	Item i_2	Item i_3	Item i_4	Item i_5
User u_1	$\mathbf{5}_{2,2,8,8}$	$\mathbf{7}_{5,5,9,9}$	$\mathbf{5}_{2,2,8,8}$	$\mathbf{7}_{5,5,9,9}$	**?**
User u_2	$\mathbf{5}_{8,8,2,2}$	$\mathbf{7}_{9,9,5,5}$	$\mathbf{5}_{8,8,2,2}$	$\mathbf{7}_{9,9,5,5}$	**9**
User u_3	$\mathbf{5}_{8,8,2,2}$	$\mathbf{7}_{9,9,5,5}$	$\mathbf{5}_{8,8,2,2}$	$\mathbf{7}_{9,9,5,5}$	**9**
User u_4	$\mathbf{6}_{3,3,9,9}$	$\mathbf{6}_{4,4,8,8}$	$\mathbf{6}_{3,3,9,9}$	$\mathbf{6}_{4,4,8,8}$	**5**
User u_5	$\mathbf{6}_{3,3,9,9}$	$\mathbf{6}_{4,4,8,8}$	$\mathbf{6}_{3,3,9,9}$	$\mathbf{6}_{4,4,8,8}$	**5**

Target user
Users most similar to the target user
Rating to be predicted
Ratings to be used in the prediction

Figure 3.2 Collaborative Filtering in a Multi-Criteria Setting

In summary, while the overall rating that a user gives to an item provides information regarding *how much* the user liked the item, multi-criteria ratings provide some insights regarding *why* the user liked the item as much as he or she did. Therefore, having multi-criteria ratings provides the possibility of estimating the similarity between two users more accurately.

Based on this idea, we propose extending the standard collaborative filtering algorithm to include multi-criteria ratings. Specifically, we propose several different ways to include multi-criteria rating information in the calculation of the similarity between two different users $sim(u, u')$ or two different items $sim(i, i')$. Then, given the newly calculated similarity, a rating prediction can then be done using the weighted sum or adjusted weighted sum in the same way as with a standard collaborative filtering algorithm, using Equations (2.2) or (2.3). Below we describe two different approaches to leverage multi-criteria ratings in the similarity computation. These approaches change only the similarity function in the traditional collaborative-filtering technique to reflect multi-criteria rating information.

Aggregating traditional similarities that are based on each individual rating

This approach can use any standard similarity metric, such as cosine-based (2.1), and calculates the similarity between users (or items) based on each individual criteria. Let's assume that each rating that user *u* gives to item *i* consists of an "overall" rating r_0, and *k* multi-criteria ratings $r_1, \ldots, r_k$, i.e.,

$$R(u, i) = (r_0, r_1, \ldots, r_\mathrm{k}). \tag{3.3}$$

Then, we can obtain *k*+1 different similarity estimations by using some standard metric to measure the similarity between users *u* and *u'*: $sim_0(u, u')$ represents the similarity between *u* and *u'* based on the overall rating; $sim_1(u, u')$ – similarity based on the first criteria rating; $sim_2(u, u')$ – similarity based on the second criteria rating; and so on. The overall similarity then can be computed by aggregating the individual similarities in several ways:

- [Average similarity] By averaging all individual similarities, i.e.,

$$sim_{avg}(u,u') = \frac{1}{k+1}\sum\nolimits_{i=0}^{k} sim_i(u,u'), \tag{3.4}$$

- [Worst-case similarity] By using the smallest of similarities, i.e.,

$$sim_{min}(u,u') = \min_{i=0,\ldots,k} sim_i(u,u'). \tag{3.5}$$

Calculating similarity using multidimensional distance metrics

In a multi-criteria rating scenario, each rating $R(u, i) = (r_0, r_1, \ldots, r_\mathrm{k})$ represents a point in the *k*+1-dimensional space. Therefore, one natural approach to compute the similarity between different users is to use multidimensional distance metrics. Such metrics are easy to understand and straightforward to implement. Note that the metrics of distance and similarity are inversely related: the smaller the distance between two users, the higher the similarity. We calculate the similarity between two users in three steps.

First, we have to be able to calculate the distance between two users' ratings for the same item, i.e., $d_{\text{rating}}\left(R(u,i), R(u',i)\right)$, where $R(u,i) = (r_0, r_1, \ldots, r_k)$ and

$R(u',i) = (r'_0, r'_1, \ldots, r'_k)$. For this purpose, any of the standard multidimensional distance metrics can be used:

- Manhattan distance: $\sum_{i=0}^{k} |r_i - r'_i|$; (3.6)
- Euclidean distance: $\sqrt{\sum_{i=0}^{k} |r_i - r'_i|^2}$; (3.7)
- Chebyshev (or maximal value) distance: $\max_{i=0,\ldots,k} |r_i - r'_i|$. (3.8)

Additionally, we tried Mahalanobis distance (Mahalanobis 1936) that takes into account the correlation of data and is scale-invariant, but it was computationally more complex to compute and also did not perform better than the other distance metrics.

Second, the overall distance between two users *u* and *u'* is simply:

$$d_{\text{user}}(u,u') = \frac{1}{|I(u,u')|} \sum_{i \in I(u,u')} d_{\text{rating}}\left(R(u,i), R(u',i)\right), \qquad (3.9)$$

where *I*(*u*, *u'*) denotes the set of items that both *u* and *u'* have rated. In other words, the overall distance between two users *u* and *u'* is the average distance between their ratings for all their common items.

Finally, because the collaborative filtering techniques operate with the metric of user similarity (and not user distance), and the distance and similarity are inversely related, we use the simple transformation between the two metrics:

$$sim(u,u') = \frac{1}{1 + d_{\text{user}}(u,u')}. \qquad (3.10)$$

Note that this definition of similarity has desired range properties, i.e., the similarity will approach 0 as the distance between two users becomes larger, and it will be 1 if the distance is zero (users are identical).

In summary, both of the approaches presented in this section change only the similarity function in the traditional collaborative filtering technique in order to reflect multi-criteria rating information, which should result in a more accurate identification of similar users and, consequently, in better recommendation quality.

3.3.2 Aggregation Function-Based Approach

The approaches we have just described apply primarily to similarity-based recommenders, such as traditional collaborative filtering systems. We now present a different approach that is not limited to any specific recommendation algorithm. The intuition behind this approach comes from the assumption that multi-criteria ratings represent user preferences for different components of an item, such as story, acting, direction, and visuals in the case of movies. So, an item's overall rating is not just another rating that is independent of others; rather, it serves as some aggregation function *f* of the item's multi-criteria ratings:

$$r_0 = f(r_1, \ldots, r_k)\,. \tag{3.11}$$

In other words, this approach assumes that the overall rating has a certain relationship with multi-criteria ratings. For instance, in a movie recommendation application, the story criteria rating may have a very high priority, that is, movies with high story ratings are well liked overall by some users, regardless of other criteria ratings. So, if a system predicts that a movie's story rating will be high, it must also predict that the overall rating will be high in order to be accurate.

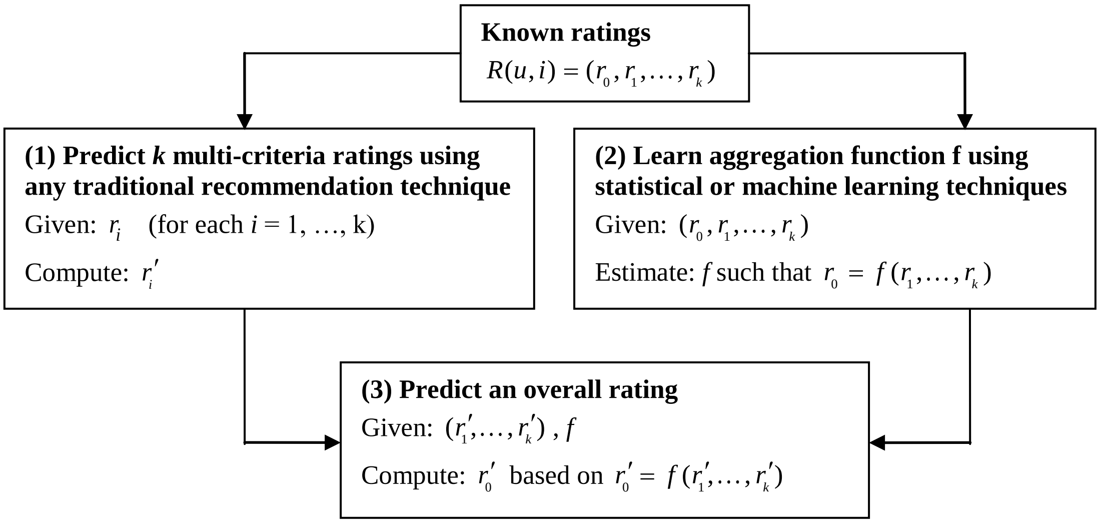

Known ratings
$R(u,i) = (r_0, r_1, \ldots, r_k)$
(1) Predict k multi-criteria ratings using any traditional recommendation technique
Given: r_i (for each i = 1, …, k)
Compute: r_i'
(2) Learn aggregation function f using statistical or machine learning techniques
Given: $(r_0, r_1, \ldots, r_k)$
Estimate: f such that $r_0 = f(r_1, \ldots, r_k)$
(3) Predict an overall rating
Given: $(r_1', \ldots, r_k')$, f
Compute: r_0' based on $r_0' = f(r_1', \ldots, r_k')$

Figure 3.3 Overview of an Aggregation Function-Based Approach

Figure 3.3 illustrates the proposed approach to rating estimation consists of the following three steps. First, we decompose the *k*-dimensional multi-criteria rating space into *k* single-rating recommendation problems and use *any* traditional single-criteria recommendation technique to estimate the ratings for each individual criterion. Second, we use statistical or machine learning techniques to estimate aggregation function *f* based on the known ratings. Third, using the multi-criteria ratings estimated in step 1 and function *f* estimated in step 2, we directly calculate the predicted overall rating. Below we discuss each of these steps in more detail.

Step 1: Predicting multi-criteria ratings

We decompose the *k*-dimensional multi-criteria rating space into *k* single-rating recommendation problems, where each problem can be represented with a traditional *Users*×*Items* matrix (like the one in Figure 3.1) and addresses the rating prediction for one of the individual criteria. In other words, instead of the multi-criteria recommendation problem $R{:}Users \times Items \rightarrow R_0 \times R_1 \times \ldots \times R_k$ we are dealing with *k* single-rating recommendation problems $R{:}Users \times Items \rightarrow R_i$ (where $i = 1, \ldots, k$). This approach provides a lot of flexibility. Unlike similarity-based approaches mentioned in the previous section, in can use *any* existing single-rating recommendation technique — collaborative, content-based, or hybrid— to estimate unknown ratings for individual criteria.

Step 2: Learning the aggregation function

This step aims to estimate relationship *f* between the overall rating and the underlying multi-criteria ratings of items, such that $r_0 = f(r_1, \ldots, r_k)$. We can already predict the individual multi-criteria ratings (Step 1), but the ability to predict the overall rating of each item for each user is also useful in many situations. For example, with the overall rating for each item, a system can *rank* all items for each user and recommend only the most relevant items. To determine the most relevant items without an overall rating, the system would have to deal with a much more complex multi-criteria optimization problem (Statnikov and Matusov 1995). Thus, finding the aggregation function is crucial for recommender systems, and there are several ways to obtain it:

- *Domain expertise*. On the basis of prior experience and knowledge of the domain, a domain expert can suggest an appropriate aggregation function. For example, the overall rating might be a simple *average* of the underlying multi-criteria ratings for each item, i.e., $r_0 = (r_1 + \ldots + r_k) / k$.
- *Statistical techniques*, including various linear and non-linear regression analysis techniques. For example, in *linear regression*, the aggregation function for the overall rating would be a linear combination of the multi-criteria ratings, i.e., $r_0 = w_1 r_1 + \ldots + w_k r_k + c$, where we can interpret weight w_i associated with criterion i as this criterion's importance in determining the overall rating. We can estimate the weights w_i (i = 1, …, k) and constant c based on the set of known ratings.
- *Machine learning techniques*. We can also obtain the function using various sophisticated computational learning techniques such as *artificial neural networks* (Mitchell 1997).

Besides supporting different learning techniques, the aggregation function can also be of three different *scopes*: total, user-based, or item-based. In particular, f is a *total* aggregation function if it is used to predict all unknown ratings, for example, if the criteria weights w_i in a regression-based function mentioned above are the same for all users and items. However, depending on the domain specifics, *user-based* or *item-based* aggregation functions can also be useful in some applications. For example, in a movie recommender system, user u might consistently give greater weight to the "story" component of all movies, whereas user u' might give significant weight to the "visuals" component. In this case, it would be advantageous for user u to have his or her own *user-based* aggregation function f_u, which the system would learn exclusively from the known ratings of user u (as opposed to all known ratings). Similarly, with the *item-based* aggregation function f_i we would assume that each item i will have its own aggregation function that is based on all the ratings involving this item.

Finally, note that a variety of different techniques are available for testing the fitness or accuracy of the predicted aggregation function(s). For example, in the case of linear regression, we can estimate the predictive power using its R-squared value. Or, more generally, we could use standard n-fold cross validation techniques to estimate the predictive accuracy of the aggregation function (Mitchell 1997). This means we can restrict the use of user-based (or item-based) aggregation functions only to the ones that exhibit sufficient predictive performance, e.g., whose accuracy is greater than some pre-specified threshold. The remaining users (or items) could use other techniques, such as the total aggregation function. As with every data-driven computational learning technique, there will be application domains where this approach will work well (i.e., domains where users or items exhibit consistent preferences on each criterion) and domains where other techniques will be more advantageous.

Step 3: Predicting overall ratings

Finally, as mentioned earlier, we compute each unknown overall rating r_0' directly by using the multi-criteria ratings estimated in step 1 and function *f* estimated in step 2:

$$r_0' = f(r_1', \ldots, r_k') .$$

3.4 Experimental Results

To evaluate the proposed approaches, we have collected a set of user-submitted movie ratings from the Yahoo! Movies website (http://movies.yahoo.com) for several hundred randomly chosen movies from the last decade. When users submit movie ratings to Yahoo! Movies, in addition to the overall rating, they are asked to provide information about four criteria for each movie: story, acting, direction, and visuals. All ratings have 13 possible values and are based on a standard grading scale from A+ to F. For the analysis purposes, we changed them to numerical values from 13 to 1. In the data preprocessing stage, we invoked two constraints on the dataset to ensure that the dataset is not extremely sparse and has enough data for rating prediction: that each user rated at least 10 movies and that each movie had at least 10 user ratings.

The resulting dataset included 155 users, 50 movies, and 2,216 known ratings in total. The dataset's sparsity is 28.6 percent—that is, 28.6 percent of ratings are known. Each user has rated 14.3 movies on average, and the average number of common movies between two users is 5.2. Each movie has been rated on average by 44.3 users, and the average number of common users between two movies is 13.6. The average rating on each criterion is approximately 9 (or "B"). Furthermore, to obtain reliable results with a relatively small amount of data, we used a standard 10-fold cross validation technique (Mitchell 1997), where we randomly divided the dataset into 10 disjoint subsets. We use nine-tenths of the data for training, and the remaining one-tenth for testing the rating prediction. Then we repeated this process 10 times (each time with a different test dataset) and performed the evaluation on all predicted ratings.

As discussed in Section 2.3, the accuracy of recommendations is measured by *precision-in-top-N*, which represents the percentage of truly relevant items among those that were predicted to be *N* most relevant items for each user. This metric was chosen because of its practicality, since many users in real-life personalization and recommendation applications are typically interested in looking at only a few highest-ranked item recommendations. Since Yahoo! Movies' rating scale (from A+ to F) was not binary, we translated the overall movie ratings into a binary scale by treating the ratings greater than 10.5 (A+, A, A-) as "relevant" and ratings less than 10.5 as "non-relevant." The threshold of 10.5 was chosen with the assumption that the users would really want to focus on the recommendations about movies that are most relevant to them (movies they would rate as A+, A, A-), and therefore the correctness of recommendations for such movies is most desirable.

In our dataset, 35.6 percent of the overall ratings were above 10.5, which means that simply recommending items at random could achieve the precision level of 35.6 percent. Any recommender system that does not achieve 35.6 percent precision would be worse than a random guess and, therefore, essentially useless.

To illustrate the performance of the proposed multi-criteria recommendation techniques on real-life data, we performed an empirical analysis of the following five approaches using the movie data (as summarized in Table 3.1):

- *standard CF* – a traditional single-rating user-based CF approach, which uses a cosine similarity metric and adjusted weighted sum, as described in (2.1) and (2.3). This approach is used as a baseline to illustrate the performance of multi-criteria recommendation approaches, as compared to a single-rating recommender system. In our implementation, we calculate the similarity between two users, only if they have rated at least three movies in common, in order to obtain useful similarity information. Otherwise, computing the similarity with, say, only one rating in common, the two users would always be considered the closest neighbors even when they show extremely different preferences on the common movie (e.g., 1 vs. 13), because the *cosine-based* similarity would be 1in such cases.
- Two similarity-based techniques (as described in Section 3.3.1) implemented with the traditional user-based CF approach:
  - *cos-min* –a technique that aggregates traditional cosine-based similarities for each individual rating.
  - *Chebyshev* – a technique that uses the Chebyshev multidimensional distance metric. In particular, we compute the multidimensional distance between two users when they have at least one movie in common, because the multi-criteria rating information of even one movie already provides 5 data points (on story, action, etc.) in our experiments to compare their rating patterns.
- Two aggregation-function-based techniques (as described in Section 3.3.2), where individual multi-criteria ratings are estimated using the traditional user-based CF approach:
  - *total-reg* – a total aggregation function that is based on linear regression.

- *movie-reg95* – an item-based aggregation function that is generated separately for each movie and restricted to movies that have the best regression fit– specifically, where R-squared ≥ 95 percent.

We used the standard user-based collaborative filtering approach as an integral part of every technique to minimize the non-essential differences between the techniques as much as possible and, thus, to maximize the possibility that any differences in performance between the *standard CF* and multi-criteria recommender systems are due to the newly introduced multi-criteria rating information.

Table 3.1 Experimental Results of Multi-Criteria Recommendation Approaches

Recommendation Approach: user-based CF		**Precision in top 3 (%)**	**Precision in top 5 (%)**	**Precision in top 7 (%)**
Neighborhood size: ALL users	standard CF	70.7*	68.7	69.0
	cos-min	70.7**	*68.8*	69.1
	Chebyshev	*74.5†*	*70.3*	*70.5*
	total-reg	71.5	*70.9*	*70.4*
	movie-reg95	*71.8*	**74.0†**	**75.3**
Neighborhood size: 3 users	standard CF	64.9	64.9	66.3
	cos-min	*67.1*	*67.1*	*67.8*
	Chebyshev	*66.2*	65.5	64.6
	total-reg	65.2	*66.6*	66.5
	movie-reg95	**69.0**	**70.7**	**72.2**

*Shaded cells represent the performance of the standard collaborative-filtering baseline approach.
**Roman font indicates precision values that represent 0–1% improvement over the baseline approach.
†Italic font indicates precision values that represent 1–4% improvement over the baseline approach.
‡Bold font indicates precision values that represent >4% improvement over the baseline approach.

For the sake of completeness, Table 3.1 includes results for different collaborative-filtering neighborhood sizes (all users versus the three most similar users) and for different *precision-in-top-N* levels (*N* = 3, 5, and 7). The shaded cells in the table represent the performance of the standard CF baseline approach. As the table shows, nearly every multi-criteria technique performed at least as well as or better than the baseline: precision values in regular font represent improvement between 0 and 1 percent; italic font, from 1 to 4 percent; and boldface, greater than 4 percent. For further comparison, we calculated the *precision-in-top-N* metric for a simple popularity-based recommendation approach, in which the system recommends *N* movies (*N* = 3, 5, and 7) that are most liked by all other users, as indicated by each movie's average rating. The *precision-in-top-N* results for this simple approach were 61.3 percent for *N* = 3, 53.3 percent for *N* = 5, and 46.4 percent for *N* = 7. These values all exceed the "random guess" threshold of 35.6 percent but fall short of the performance achieved with collaborative-filtering techniques.

Among other notable results:

- We tried *precision-in-top-1* measures (as opposed to the top 3, 5, and 7 measures in Table 3.1) for various neighborhood sizes. Similarity-based techniques (such as Chebyshev and cos-min) performed best, typically exceeding both the baseline approach and the aggregation-function-based approaches by 2 to 6 percent.
- We also tried movie-based (as opposed to user-based) collaborative filtering. In this case, total-reg performed the best of all the techniques for various neighborhood sizes and typically outperformed the baseline approach by 1 to 5 percent.
- The performance differences between multi-criteria recommendation techniques and the baseline approach are even larger in a sparser environment. For instance, we tested our recommendation algorithms on a sparser dataset, which we obtained from our initial dataset by randomly removing about one-third of its ratings (700 ratings out of 2,216). The new dataset's sparsity was 19.6 percent. Most multi-criteria recommendation algorithms (including Chebyshev and aggregation-function-based techniques) outperformed the baseline approach by 5 to 10 percent on this sparser dataset.
- Combining similarity-based and aggregation-function-based multi-criteria recommendation techniques can sometimes improve the predictive performance. This result is generally consistent with similar findings in recommender systems literature about the advantages of combining different types of recommender

systems. For example, it has been widely reported that combining content-based and collaborative systems can improve recommendation accuracy.

As with most recommender systems and, more generally, computational learning techniques, the performance of a specific technique is highly domain-dependent. In other words, performance depends significantly on the underlying data's characteristics. So, although we expect the proposed techniques to do well in a variety of application domains, we do not expect them to outperform traditional single-rating techniques in all domains where multi-criteria information exists. For example, these techniques cannot be expected to perform well in domains where multi-criteria ratings do not carry meaningful information or where no inherent relationship exists between the overall rating and the multi-criteria ratings for the users or items.

3.5 Conclusion

While single-rating recommender systems have been successful in a number of personalization applications, multi-criteria rating systems are becoming more commonly deployed in many industries. However, to take full advantage of existing multi-criteria ratings in personalization applications, new recommendation techniques are required. In this chapter, we propose two new recommendation approaches – the similarity-based approach and the aggregation-function-based approach – to incorporate and leverage multi-criteria rating information. Our experimental results on a real-world dataset confirm that, when available, multi-criteria ratings can be successfully leveraged to improve recommendation accuracy. We expect that the proposed approaches will be useful in other application domains as well, where they will be able to predict overall ratings more accurately by utilizing the available multi-criteria rating information.

The area of recommender systems has made significant progress over the last few years with many techniques being proposed and many systems being developed. However, modern recommender systems still require further significant improvements to provide better recommendations and be viable in more complex personalization applications; the ability to leverage multi-criteria rating information constitutes one such improvement. We believe that this work is just the first step in studying multi-criteria recommender systems and that significant additional work is needed to further explore this issue.

Chapter 4. Heuristic-Based Ranking Approaches to Improve Aggregate Recommendation Diversity

4.1 Introduction

The importance of *diverse* recommendations has been previously emphasized in several studies (Bradley and Smyth 2001, Brynjolfsson et al. 2007, 2010, Fleder and Hosanagar 2009, Hu and Pu 2011, McSherry 2002, Oestreicher-Singer and Sundararajan, 2011, Smyth and McClave 2001, Zhang 2009, Zhang and Hurley 2008, Ziegler et al. 2005). These studies argue that one of the goals of recommender systems is to provide a user with highly idiosyncratic or personalized items, and more diverse recommendations result in more opportunities for users to receive recommendations for such items. With this motivation, some studies have proposed new recommendation methods that can increase the diversity of recommendation sets for a given individual user, i.e., *individual* diversity (from an individual user's perspective), often measured by an average dissimilarity between all pairs of recommended items. In contrast to these studies, this chapter focuses on improving the diversity of items recommended across all users, i.e., *aggregate* diversity (from a system's perspective).

While it has been shown that more diverse recommendations, presumably leading to more sales of long-tail items, could be beneficial for both individual users and some business models such as Netflix (Brynjolfsson et al. 2003, 2006, 2009, 2010, Goldstein and Goldstein 2006), the impact of recommender systems on *aggregate* diversity in real-world e-commerce applications has not been well-understood. Previous studies (Brynjolfsson et al. 2007, 2010, Fleder and Hosanagar 2009, Oestreicher-Singer and Sundararajan 2011) show that recommender systems can play a key role in increasing both "long-tail" and "superstar" effects in real-world e-commerce applications. As seen from this recent debate, there is a growing awareness of the importance of aggregate diversity in recommender systems. Therefore, this chapter aims to develop algorithmic techniques to improve aggregate diversity of recommendations (which we will simply refer to as *diversity* throughout the chapter, unless explicitly specified otherwise). These recommendations can be intuitively measured by the number of distinct items recommended across all users.

Higher diversity, however, can come at the expense of accuracy. As illustrated in the accuracy-diversity tradeoff example in Section 2.5, we learned that it is possible to obtain higher diversity simply by recommending less popular items; however, the loss of recommendation accuracy in this case can be substantial. In this chapter, we explore new recommendation approaches that can increase the diversity of recommendations with only minimal (negligible) accuracy loss using different recommendation *ranking* techniques. In particular, traditional recommender systems typically rank the relevant items in a descending order of their predicted ratings for each user and then recommend top *N* items, resulting in high accuracy (refer to Section 2.1). In contrast, the proposed approaches consider additional factors such as item popularity when ranking the recommendation list to substantially increase recommendation diversity while maintaining comparable levels of accuracy. We provide a comprehensive empirical evaluation of the proposed approaches, where they are tested with various datasets in a variety of different settings. For example, the best results show up to 20-25% diversity gain with only 0.1% accuracy loss, up to 60-80% gain with 1% accuracy loss, and even substantially higher diversity improvements up to 250% if some users are willing to tolerate higher accuracy loss.

In addition to obtaining significant diversity gains, the proposed ranking techniques have several other advantageous characteristics. In particular, these techniques are extremely *efficient*, because they are based on scalable sorting-based heuristics that make decisions based only on the "local" data (i.e., only on the candidate items of each individual user) without having to keep track of "global" information, such as which items have been recommended across all users and how many times. The techniques are also *parameterizable*, since the user has the control to choose the acceptable level of accuracy for which the diversity will be maximized. The proposed ranking techniques also provide a *flexible* solution to improve recommendation diversity because they are applied *after* the unknown item ratings have been estimated and, thus, they can achieve diversity gains in conjunction with a number of different rating prediction techniques. To illustrate the broad applicability of the proposed recommendation ranking approaches, we used these approaches in our experiments in conjunction with the most popular and widely employed CF techniques for rating prediction: a heuristic neighborhood-based technique and a model-based matrix factorization technique, as discussed in Section 2.2. Furthermore, the vast majority of current recommender systems already employ some ranking approach, thus, the proposed techniques would not introduce new *types* of procedures into recommender systems, but they would replace existing ranking procedures. The proposed ranking approaches also do not require any additional information about users (e.g., demographics) or items (e.g., content features) aside from the ratings data, which makes them applicable in a wide variety of recommendation contexts.

The remainder of the chapter is organized as follows. Section 4.2 describes our motivations for alternative recommendation ranking techniques, such as item popularity. We then propose several additional ranking techniques in Section 4.3, and the main empirical results follow in Section 4.4. Additional experiments are conducted to further explore the proposed ranking techniques in Section 4.5. Lastly, Section 4.6 concludes the chapter by summarizing the contributions and future directions. (Refer to Chapter 2 for a review of relevant literature on traditional recommendation algorithms and two metrics to evaluate recommendation quality: *precision-in-top-N* and *diversity-in-top-N*.)

4.2 Motivations for Recommendation Re-ranking

In this section, we discuss how re-ranking of the candidate items that are predicted to be relevant can affect the accuracy-diversity tradeoff and how various item ranking factors, such as item popularity, can improve the diversity of recommendations. Note that the general idea of personalized information ordering is not new since its importance has been discussed in information retrieval literature (Park and Pennock 2007, Smyth and Bradley 2003), and there have been attempts to reduce redundancy and promote the diversity of retrieved results by re-ranking them (Carbonell and Goldstein 1998, Sanderson et al. 2009, Zhai et al. 2003).

4.2.1 Standard Ranking Approach

Following the two-phase recommendation process discussed in Section 2.1, typical recommender systems predict unknown ratings based on known ratings, using any traditional recommendation technique such as neighborhood-based or matrix factorization CF techniques. The predicted ratings are then used to support the user's decision-making. In particular, the most relevant *N* items are selected according to some *ranking criterion*, typically using *predicted rating value* as the ranking criterion: $rank_{\text{Standard}}(i)=R^{*}(u, i)^{-1}$. This *standard* ranking approach shares motivation with the widely used probability ranking principle in information retrieval literature that ranks the documents in order of decreasing probability of relevance (Robertson 1997).

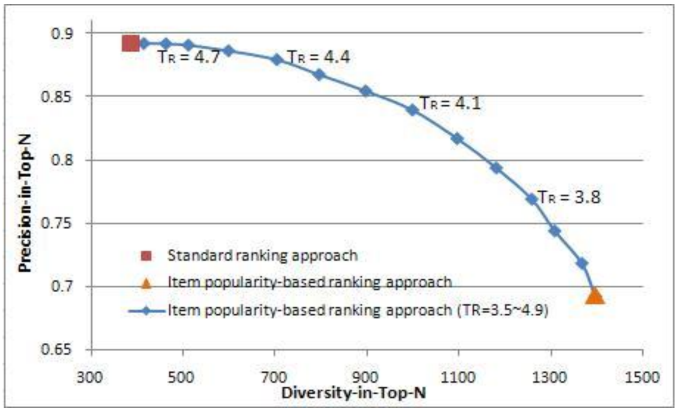
Note that, by definition, recommending the most highly predicted items selected by the standard ranking approach is designed to help improve recommendation accuracy (as empirically supported in Figure 2.2), but not recommendation diversity. Therefore, new ranking criteria are needed to achieve diversity improvement. Since recommending best-selling items to each user typically leads to diversity reduction, recommending less popular items intuitively should have an effect of increasing recommendation diversity. Following this motivation, which is supported from the example in Table 2.1, we explore the possibility of using *item popularity* as a recommendation ranking criterion, and in the next subsection we show how this approach can affect the recommendation quality in terms of accuracy and diversity.

4.2.2 Proposed Approach: Item Popularity-Based Ranking

An item popularity-based ranking approach ranks items directly based on their popularity, from lowest to highest, where popularity is represented by the number of known ratings that each item has. More formally, item popularity-based ranking function can be written as follows:

$$rank_{\text{ItemPop}}(i) = |U(i)|, \text{ where } U(i) = \{u \in U \mid \exists R(u, i)\}.$$

We compared the performance of the item popularity-based ranking approach with the standard ranking approach using the MovieLens dataset and item-based CF. This comparison using the accuracy-diversity plot is presented in Figure 4.1. In particular, the results show that, as compared to the standard ranking approach, the item popularity-based ranking approach increased recommendation diversity 3.6 times from 385 to 1395; however, recommendation accuracy dropped from 89% to 69%. Here, despite the significant diversity gain, a significant accuracy loss of 20% would not be acceptable in most real-life personalization applications. Therefore, we introduced a general technique to parameterize the recommendation ranking approaches, which allows significant diversity gains while controlling accuracy losses according to how much loss is tolerable in a given application.



MovieLens dataset, top-5 items, item-based CF (50 neighbors)

Figure 4.1 Performance of the Standard Ranking Approach and Item Popularity-Based Approach with its Parameterized Versions

4.2.3 Controlling the Accuracy-Diversity Tradeoff: Parameterized Ranking Approaches

The item popularity-based ranking approach as well as all other ranking approaches proposed in Section 4.3 below are parameterized with "ranking threshold" $T_{\mathrm{R}} \in [T_{\mathrm{H}}, T_{\mathrm{max}}]$ to allow user the ability to choose a certain level of recommendation accuracy. T_{max} is the largest possible rating on the rating scale, e.g., T_{max}=5 in our experiments where the ratings of the datasets are integers between 1 and 5, and inclusive. In particular, given any ranking function $rank_{\mathrm{X}}(i)$, ranking threshold T_{R} is used to create the parameterized version of this ranking function, $rank_{\mathrm{X}}(i, T_{\mathrm{R}})$, which is formally defined as:

$$rank_X(i, T_R) = \begin{cases} rank_X(i), & \text{if } R^*(u,i) \in [T_R, T_{\max}] \\ \alpha_u + rank_{\text{Standard}}(i), & \text{if } R^*(u,i) \in [T_H, T_R) \end{cases},$$

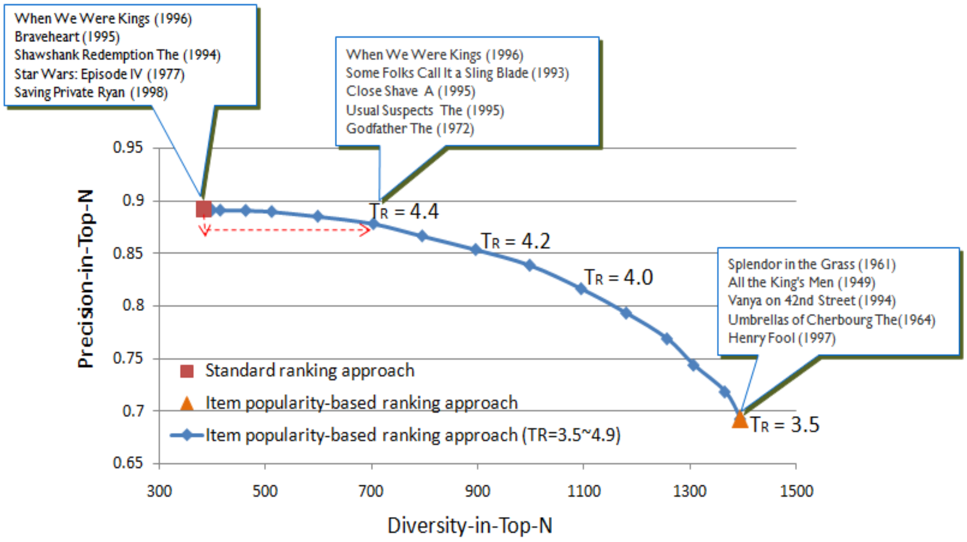
where $I_u^*(T_R) = \{i \in I \mid R^*(u,i) \geq T_R\}, \alpha_u = \max_{i \in I_u^*(T_R)} rank_X(i)$.

Simply put, items that are predicted above ranking threshold T_R are ranked according to $rank_X(i)$, while items that are below T_R are ranked according to the standard ranking approach $rank_{\text{Standard}}(i)$. In addition, all items that are above T_R are ranked ahead of all items that are below T_R as ensured by α_u in the above formal definition. Thus, increasing the ranking threshold $T_R \in [T_H, T_{\max}]$ towards $T_{\max}$ would enable the system to choose the most highly predicted items resulting in more accuracy and less diversity (becoming increasingly similar to the standard ranking approach). In contrast, decreasing the ranking threshold $T_R \in [T_H, T_{\max}]$ towards T_H would make $rank_X(i, T_R)$ increasingly more similar to the pure ranking function $rank_X(i)$, resulting in more diversity with some accuracy loss.

Choosing different T_R values in-between the extremes allows the user to set the desired balance between accuracy and diversity. In particular, as Figure 4.1 shows, the recommendation accuracy of the item popularity-based ranking approach could be improved by increasing the ranking threshold. For example, the item popularity-based ranking approach with a ranking threshold of 4.4 could minimize the accuracy loss to 1.32%, but still could obtain an 83% diversity gain (from 385 to 703), compared to the standard ranking approach. An even higher threshold of 4.7 still makes it possible to achieve 20% diversity gain (from 385 to 462) with only a 0.06% accuracy loss.

Also note that, even when there are less than N items above the ranking threshold T_R, by definition, *all* the items above T_R are recommended to a user, and the remaining top-N items are selected according to the standard ranking approach. This ensures that all the ranking approaches proposed in this chapter provide the same exact number of recommendations as their corresponding baseline techniques using the standard ranking approach, which is also very important from an experimental analysis point of view to have a fair performance comparison of different ranking techniques.

In Figure 4.1, we showed the overall performance of the item popularity-based ranking approach based on the recommendations for all users. In Figure 4.2, we also give the example recommendations that a specific user (selected at random from the users in the MovieLens dataset) would receive, using the same ranking approach. For example, based on the ratings of 110 movies that the chosen user has watched, the standard ranking approach recommends well-known popular movies to the user. However, the item popularity-based ranking approach with the minimum accuracy level (using the ranking threshold of 3.5) recommends lesser known movies to the same user, which significantly increases diversity. If we increase the ranking threshold up to 4.4, this user could receive a more accurate but still diverse set of recommendations. Thus, it is important for system designers to find an appropriate level of ranking threshold, depending on user needs and business requirements.



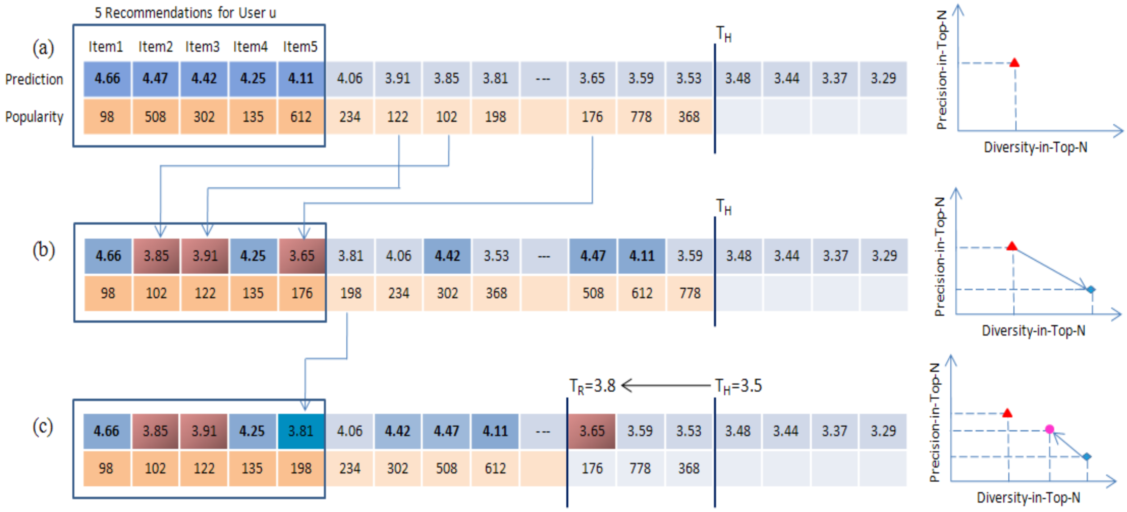
Ex. User Profile	Liked	Disliked
• male	Forrest Gump (1994)	Clueless (1995)
• age 18-24	Lethal Weapon 4 (1998)	Mr. Nice Guy (1997)
• college student	Austin Powers (1997)	Addams Family Values (1993)
• 110 ratings	Star Wars: Episode VI (1983)	Titanic (1997)
	While You Were Sleeping (1995)	Batman Forever (1995)

MovieLens dataset, top-5 items, item-based CF (50 neighbors)

Figure 4.2 Recommendation Examples of the Item Popularity-Based Ranking Approach

4.2.4 General Steps for Recommendation Re-ranking

The item popularity-based ranking approach described above is just one example of possible ranking approaches for improving recommendation diversity. A number of additional ranking functions, *rank*$_X$(*i*), will be introduced in Section 4.3. Here, based on the previous discussion in Section 4.2.3, we summarize the general ideas behind the proposed ranking approaches, as illustrated by Figure 4.3.



(a) Recommending top-*N* highly predicted items for user *u*, according to the standard ranking approach
(b) Recommending top-*N* items, according to some other ranking approaches for better diversity
(c) Confining re-ranked recommendations to the items above the new ranking threshold T_R (e.g., ≥ 3.8) for better accuracy

Figure 4.3 General Overview of Ranking Approaches to Improve Recommendation Diversity

The first step, shown in Figure 4.3a, represents the standard approach, which ranks all the predicted items for each user according to the predicted rating value and selects the top-5 candidate items, as long as they are above the highly-predicted rating threshold T_H. The recommendation quality of the overall recommendation technique is measured in terms of the *precision-in-top-N* and the *diversity-in-top-N* as shown in the accuracy-diversity plot at the right side of the example (a).

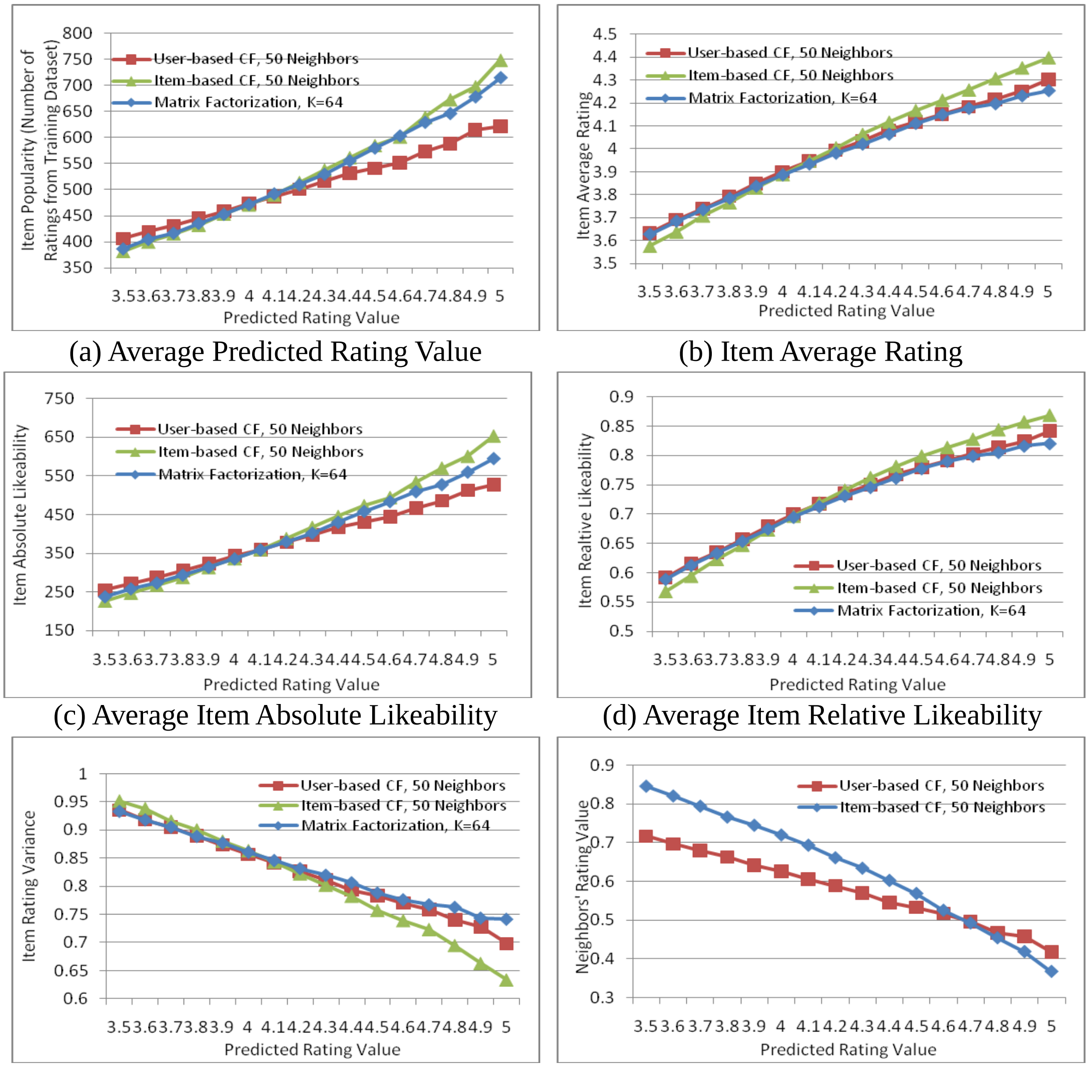
The second step, illustrated in Figure 4.3b, shows the recommendations provided by applying one of the proposed ranking functions, *rank*$_X$(*i*), where several different items that are not necessarily among the N most highly predicted, but are still above T_{H} are recommended to the user. In this way, a user can receive more idiosyncratic, long-tail recommendations, which are less frequently recommended items that may not be as widely popular but would still be very relevant to this user (as indicated by a relatively high predicted rating). Therefore, re-ranking the candidate items can significantly improve the recommendation diversity although this typically comes at some loss of recommendation accuracy. The performance graph of the second step (b) demonstrates this accuracy-diversity tradeoff.

The third step, shown in Figure 4.3c, can significantly minimize accuracy loss by confining the re-ranked recommendations to the items above the newly introduced ranking threshold T_{R} (e.g., 3.8 out of 5). In this particular illustration, note that the increased ranking threshold filters out the fifth recommended item in step (b) (i.e., the item with a predicted rating value of 3.65), and the next possible item above the new ranking threshold (i.e. the item predicted as 3.81) is recommended to user u. Averaged across all users, this parameterization helps to make the level of accuracy loss fairly small while retaining a significant diversity gain (as compared to the standard ranking approach), as shown in the performance graph of step (c).

We now introduce several additional item ranking functions, and provide empirical evidence that supports our motivation of using these item criteria for diversity improvement.

4.3 Additional Ranking Approaches

In many personalization applications such as movie or book recommendations, there are often more highly predicted ratings for a given user than can be put in the top-N list. This provides an opportunity to have a number of alternative ranking approaches, where different sets of items can be recommended to the user. In this section, we introduce six additional ranking approaches that can be used as alternatives to $rank_{\text{Standard}}$ to improve recommendation diversity. The formal definitions of each ranking approach (provided below) are illustrated in Figure 4.4 with empirical evidence that supports the use of these item ranking criteria. We only show the empirical results from MovieLens dataset; however, consistently similar patterns were found in other datasets (discussed in Section 4.4.1) as well.



(a) Average Predicted Rating Value

(b) Item Average Rating

(c) Average Item Absolute Likeability

(d) Average Item Relative Likeability

(e) Average Item Rating Variance

(f) Average Neighbors' Rating Variance

Figure 4.4 Relationships between Various Item-Ranking Criteria and Predicted Rating Value, for Highly Predicted Ratings (MovieLens Data)

In our empirical analysis we consistently observed that popular items, on average, are likely to have higher predicted ratings than less popular items, using both heuristic- and model-based techniques for rating prediction, as shown in Figure 4.4a. As discussed in Section 4.2, recommending less popular items helps improve recommendation diversity;

therefore, as can be immediately suggested from the monotonic relationship between the average item popularity and predicted rating value, recommending less highly predicted items (but still predicted to be above T_H) likely implies recommending, on average, less popular items, potentially leading to diversity improvements. Therefore, we propose using the predicted rating value itself as an item ranking criterion:

- ***Reverse Predicted Rating Value***: ranking the candidate (highly predicted) items based on their predicted rating value, from lowest to highest, and as a result choosing less popular items, according to Figure 4.4a. More formally:

$$rank_{\text{RevPred}}(i) = R^*(u,i).$$

We now propose several other ranking criteria that exhibit consistent relationships to the predicted rating value, including the average rating, absolute likeability, relative likeability, item rating variance, and neighbors' rating variance, as shown in Figures 4.4b-4.4f. In particular, the relationship between the predicted rating values and the *average actual rating* of each item (as explicitly rated by users), shown in Figure 4.4b, also supports a similar conjecture that items with a lower average rating, on average, are more likely to have lower predicted rating values which likely represent less popular items, as shown earlier. Thus, such items could be recommended for better diversity.

- ***Item Average Rating***: ranking items according to an average of all known ratings for each item:

$$rank_{\text{AvgRating}}(i) = \overline{R(i)}, \text{ where } \overline{R(i)} = \frac{1}{|U(i)|}\sum_{u \in U(i)} R(u,i) \,.$$

Similarly, the relationship between the predicted rating values and item absolute (or relative) likeability, shown in Figures 4.4c and 4.4d, also suggests that the items with lower likeability, on average, are more likely to have lower predicted rating values which likely represent less popular movies and, thus, could be recommended for better diversity.

- ***Item Absolute Likeability***: ranking items according to how many users liked them (i.e., rated the item above T_H):

$$rank_{\text{AbsLike}}(i) = |U_H(i)|, \text{ where } U_H(i)=\{u \in U(i) |\ R(u,i) \geq T_H\}.$$

- ***Item Relative Likeability***: ranking items according to the percentage of the users who liked an item (among all users who rated it):

$$rank_{\mathrm{RelLike}}(i) = |U_{\mathrm{H}}(i)| \,/\, |U(i)|.$$

We can also use two different types of rating variances to improve recommendation diversity. With any traditional recommendation technique, each item's rating variance (which can be computed from known ratings submitted for that item) can be used for re-ranking candidate items. In addition, if any neighborhood-based recommendation technique is used for prediction, we can use the rating variance of neighbors whose ratings are used to predict the rating for re-ranking candidate items. As shown in Figures 4.4e and 4.4f, the relationship between the predicted rating value and each item's rating variance and the relationship between the predicted rating value and 50 neighbors' rating variance obtained by using a neighborhood-based CF technique demonstrate that highly predicted items tend to be low in both item rating variance and neighbors' rating variance. In other words, among highly-predicted ratings (i.e., above T_{H}) there is more user consensus for higher-predicted items than for lower-predicted ones. These findings indicate that re-ranking recommendation list by rating variance and choosing the items with a higher variance could improve recommendation diversity.

- ***Item Rating Variance***: ranking items according to each item's rating variance (i.e., rating variance of users who rated the item):

$$rank_{\mathrm{ItemVar}}(i) = \frac{1}{|U(i)|}\sum_{u \in U(i)} (R(u,i) - \overline{R(i)})^2 \ .$$

- ***Neighbors' Rating Variance***: ranking items according to the rating variance of neighbors of a particular user for a particular item. The closest neighbors of user u among the users who rated the particular item i, denoted by u', are chosen from the set of $U(i) \cap N(u)$.

$$rank_{\mathrm{NeighborVar}}(i) = \frac{1}{|U(i) \cap N(u)|}\sum_{u' \in (U(i) \cap N(u))} (R(u',i) - \overline{R_u(i)})^2 \ ,$$

$$\text{where } \overline{R_u(i)} = \frac{1}{|U(i) \cap N(u)|}\sum_{u' \in (U(i) \cap N(u))} R(u',i) \ .$$

In summary, there are a number of different ranking approaches that can improve recommendation diversity by recommending items other than the ones with the top predicted rating values. In addition, as indicated in Figure 4.1, the degree of improvement and, more importantly, the degree of tolerable accuracy loss can be controlled by the chosen ranking threshold value T_R. The next section presents comprehensive empirical results demonstrating the effectiveness and robustness of the proposed ranking techniques.

4.4 Empirical Results

4.4.1 Data

The proposed recommendation ranking approaches were tested with several movie rating datasets, including MovieLens (a data file available at grouplens.org), Netflix (a data file used for the Netflix Prize competition), and Yahoo! Movies (individual ratings collected from movie pages at movies.yahoo.com). We pre-processed each dataset to include users and movies with a significant rating history, which provided a sufficient number of highly predicted items for recommendations to each user (in the test data). The basic statistical information of the resulting datasets is summarized in Table 4.1. For each dataset, we randomly chose 60% of the ratings as training data and used them to predict the remaining 40% (i.e., test data).

Table 4.1 Basic Information of Movie Rating Datasets

	MovieLens	Netflix	Yahoo!Movies
Number of users	2,830	3,333	1,349
Number of movies	1,919	2,092	721
Number of ratings	775,176	1,067,999	53,622
Data Sparsity	14.27%	15.32%	5.51%
Average number of common movies between two users	64.6	57.3	4.1
Average number of common users between two movies	85.1	99.5	6.5
Average number of users per movie	404.0	510.5	74.4
Average number of movies per user	274.1	320.4	39.8

4.4.2 Performance of the Proposed Ranking Approaches

We conducted experiments on the three datasets described in Section 4.4.1, using three widely popular recommendation techniques for rating prediction, including two heuristic-based (user-based and item-based CF) techniques and one model-based (matrix factorization CF) technique, discussed in Section 2.2. All seven proposed ranking approaches were used in conjunction with each of the three rating prediction techniques to generate top-*N* (*N*=1, 5, 10) recommendations to each user on each dataset, with the exception of neighbors' variance-based ranking of model-based predicted ratings. In particular, because there is no concept of neighbors in a pure matrix factorization technique, the ranking approach based on neighbors' rating variance was applied only with heuristic-based techniques. We set the predicted rating threshold as T_H = 3.5 (out of 5) to ensure that only relevant items were recommended to users, and ranking threshold T_R varied from 3.5 to 4.9. The performance of each ranking approach was measured in terms of *precision-in-top-N* and *diversity-in-top-N* (*N*=1, 5, 10). For comparison purposes, its diversity gain and precision loss with respect to the standard ranking approach were also calculated.

Consistent with the accuracy-diversity tradeoff discussed in Section 2.5, *all* the proposed ranking approaches improved the diversity of recommendations by sacrificing recommendation accuracy. However, with each ranking approach, as ranking threshold T_R increased, the accuracy loss was significantly minimized with smaller precision loss while still exhibiting substantial diversity improvement. These experiments show that with different ranking thresholds, one can obtain different diversity gains for different levels of tolerable precision loss, as compared to the standard ranking approach. Following this idea, in our experiments we compare the effectiveness (i.e., diversity gain) of different recommendation ranking techniques for a variety of different precision loss levels (0.1-10%).

While a comprehensive set of experiments was performed using every rating prediction technique in conjunction with every recommendation ranking function on every dataset for a different number of top-*N* recommendations, the results were very consistent across all experiments. Table 4.2 shows three results: using all possible ranking techniques on a different dataset, a different recommendation technique, and a different number of recommendations. Additional results are included as Appendix A.

Table 4.2 Diversity Gains of Ranking Approaches for Different Levels of Precision Loss

	Item Popularity		Reverse Prediction		Item Average Rating		Item Abs Likeability		Item Relative Likeability		Item Rating Variance		Neighbors' Rating Variance	
Precision Loss	Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain	
-0.1	+800	3.078	+848	3.203	+975	3.532	+897	3.330	+937	3.434	+386	2.003	+702	2.823
-0.05	+594	2.543	+594	2.543	+728	2.891	+642	2.668	+699	2.816	+283	1.735	+451	2.171
-0.025	+411	2.068	+411	2.068	+513	2.332	+445	2.156	+484	2.257	+205	1.532	+258	1.670
-0.01	+270	1.701	+234	1.608	+311	1.808	+282	1.732	+278	1.722	+126	1.327	+133	1.345
-0.005	+189	1.491	+173	1.449	+223	1.579	+196	1.509	+199	1.517	+91	1.236	+87	1.226
-0.001	+93	1.242	+44	1.114	+78	1.203	+104	1.270	+96	1.249	+21	1.055	+20	1.052
Standard: 0.892	385	1.000	385	1.000	385	1.000	385	1.000	385	1.000	385	1.000	385	1.000

(a) MovieLens dataset, top-5 items, heuristic-based technique (item-based CF, 50 neighbors)

	Item Popularity		Reverse Prediction		Item Average Rating		Item Abs Likeability		Item Relative Likeability		Item Rating Variance	
Precision Loss	Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain	
-0.1	+314	1.356	+962	2.091	+880	1.998	+732	1.830	+860	1.975	+115	1.130
-0.05	+301	1.341	+757	1.858	+718	1.814	+614	1.696	+695	1.788	+137	1.155
-0.025	+238	1.270	+568	1.644	+535	1.607	+464	1.526	+542	1.615	+110	1.125
-0.01	+156	1.177	+363	1.412	+382	1.433	+300	1.340	+385	1.437	+63	1.071
-0.005	+128	1.145	+264	1.299	+282	1.320	+247	1.280	+288	1.327	+47	1.053
-0.001	+64	1.073	+177	1.201	+118	1.134	+89	1.101	+148	1.168	+8	1.009
Standard: 0.834	882	1.000	882	1.000	882	1.000	882	1.000	882	1.000	882	1.000

(b) Netflix dataset, top-5 items, model-based technique (matrix factorization CF, K=64)

	Item Popularity		Reverse Prediction		Item Average Rating		Item Abs Likeability		Item Relative Likeability		Item Rating Variance		Neighbors' Rating Variance	
Precision Loss	Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain	
-0.1	+220	1.794	+178	1.643	+149	1.538	+246	1.888	+122	1.440	+86	1.310	+128	1.462
-0.05	+198	1.715	+165	1.596	+141	1.509	+226	1.816	+117	1.422	+72	1.260	+108	1.390
-0.025	+134	1.484	+134	1.484	+103	1.372	+152	1.549	+86	1.310	+70	1.253	+98	1.354
-0.01	+73	1.264	+92	1.332	+56	1.202	+77	1.278	+58	1.209	+56	1.202	+65	1.235
-0.005	+57	1.206	+86	1.310	+38	1.137	+63	1.227	+36	1.130	+28	1.101	+51	1.184
-0.001	+42	1.152	+71	1.256	+25	1.090	+43	1.155	+30	1.110	+19	1.069	+22	1.079
Standard: 0.911	277	1.000	277	1.000	277	1.000	277	1.000	277	1.000	277	1.000	277	1.000

(c) Yahoo dataset, top-1 items, heuristic-based technique (user-based CF, 15 neighbors)

Note: Precision Loss = [*precision-in-top-N* of proposed ranking approach] – [*precision-in-top-N* of standard ranking approach]
Diversity Gain (column 1) = [*diversity-in-top-N* of proposed ranking approach] – [*diversity-in-top-N* of standard ranking approach]
Diversity Gain (column 2) = [*diversity-in-top-N* of proposed ranking approach] / [*diversity-in-top-N* of standard ranking approach]

For example, Table 4.2a shows the performance of the proposed ranking approaches used in conjunction with the item-based CF technique to provide the top-5 recommendations on the MovieLens dataset. In particular, one can observe that, with the precision loss of only 0.001 or 0.1% (i.e., with precision of 0.891, down from 0.892 of the standard ranking approach), the item average rating-based ranking approach already increased recommendation diversity by 20% (i.e., an absolute diversity gain of 78 on top of the 385 achieved by the standard ranking approach). If users can tolerate precision loss up to 1% (i.e., precision of 0.882 or 88.2%), the diversity could increase by 81% with the same ranking technique; and a 5% precision loss (i.e., 84.2%) can provide diversity gains up to 189% for this recommendation technique on this dataset. As shown in Table 4.2, substantial diversity improvements can be observed across different ranking techniques, different rating prediction techniques, and different datasets.

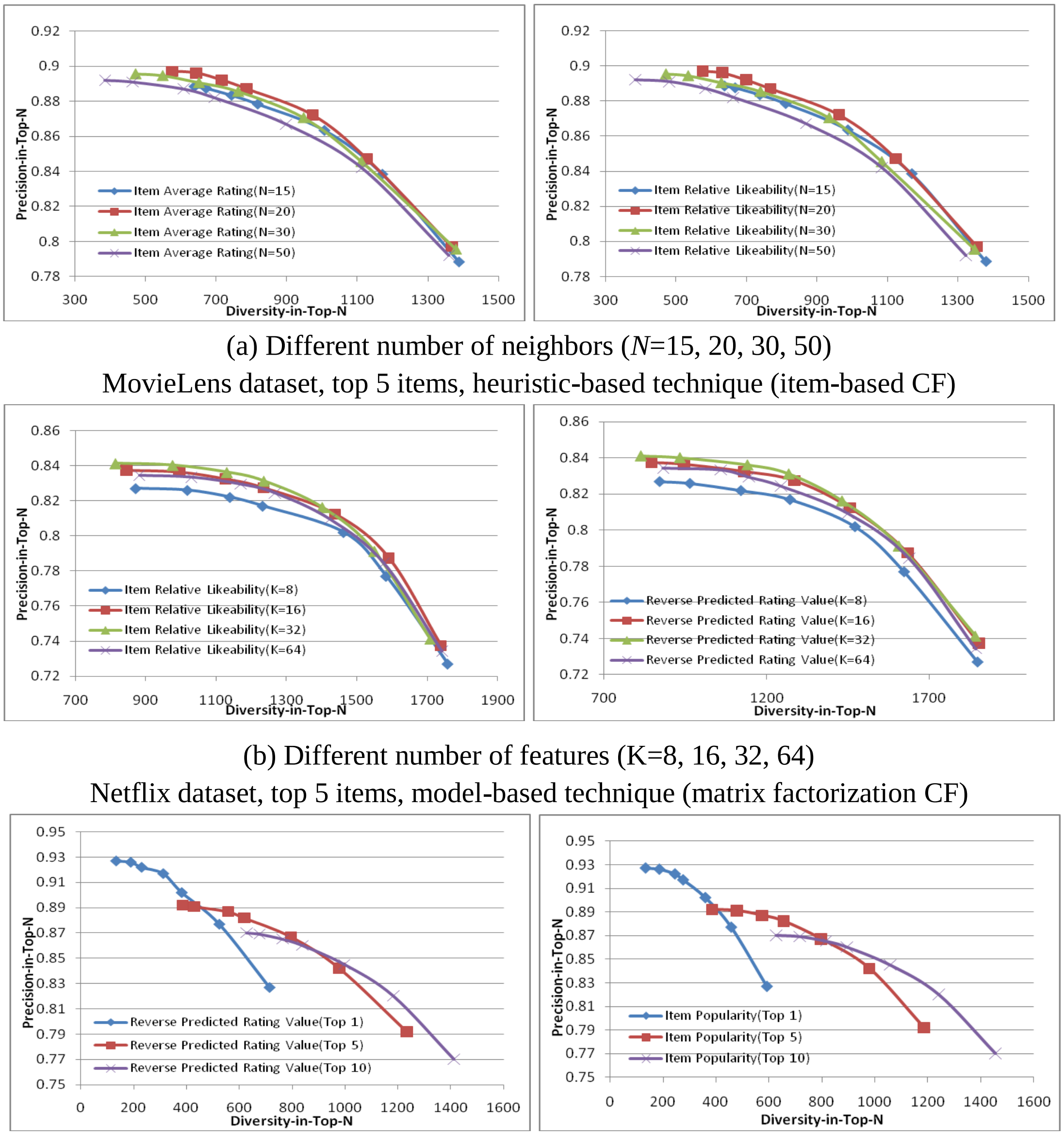
In general, all proposed ranking approaches provided significant diversity gains, and the best-performing ranking approach may be different depending on the chosen dataset and rating prediction technique. Thus, system designers have the flexibility to choose the most desirable ranking approach based on the data in a given application. We would also like to point out that since the proposed approaches are essentially implemented as sorting algorithms based on certain ranking heuristics, they are extremely scalable. For example, it took, on average, less than 6 seconds to rank all the predicted items and select top-*N* recommendations for nearly 3,000 users in our experiments with MovieLens data. Lastly, note that in our small to medium experimental recommendation settings, each user typically had a relatively small, personalized set of candidate items available for recommendations. Therefore, all proposed ranking approaches were able to provide a diverse set of recommendations. However, with large-scale datasets, where a large number of highly predicted candidate items could be available for each user, using the *non-personalized* ranking approaches (i.e., that rank based only on item-specific information, such as item popularity) will likely result in many users receiving similar recommendations. In contrast, the ranking approach based on reverse predicted rating values (i.e., based on information *personalized* to each user) can still generate truly individualized recommendations for each user and maintain high aggregate recommendation diversity, even for large-scale recommendation settings.

4.4.3 Robustness Analysis for Different Parameters

In this subsection, we present a robustness analysis of the proposed techniques with respect to several parameters: the number of neighbors used in heuristic-based CF, the number of features used in matrix factorization CF, the number of top-*N* recommendations provided to each user, and the value of predicted rating threshold T_H.

We tested the heuristic-based technique with a different number of neighbors (15, 20, 30, and 50 neighbors) and the model-based technique with a different number of features (K=8, 16, 32, and 64). For illustration purposes, Figures 4.5a and 4.5b show how two different ranking approaches for both heuristic-based and model-based rating prediction techniques were affected by different parameter values. While different parameter values may result in slightly different performance (as is well-known in recommender systems literature), the fundamental behavior of the proposed techniques remained robust and consistent, as shown in Figures 4.5a and 4.5b. In other words, using the recommendation ranking techniques with any of the parameter values, it was possible to obtain substantial diversity improvements with only a small accuracy loss.

We also vary the number of top-*N* recommendations provided by the system. Note that it is intuitively clear that top-1, top-5, and top-10 recommendations will provide different accuracy and diversity levels because it is much easier to accurately recommend one relevant item than 10 relevant items, and it is much easier to have more aggregate diversity when you can provide more recommendations. However, again, we observed that with any number of top-*N* recommendations, the proposed techniques exhibited robust and consistent behavior that allowed us to obtain substantial diversity gains at a small accuracy loss, as shown in Figure 4.5c. For example, with only a 1% precision loss, we were able to increase the diversity from 133 to 311 (134% gain) using the reverse predicted rating value-based ranking approach in the top-1 recommendation task, and from 385 to 655 (70% gain) using the item-popularity-based ranking approach in the top-5 recommendation task.



(a) Different number of neighbors (*N*=15, 20, 30, 50)
MovieLens dataset, top 5 items, heuristic-based technique (item-based CF)

(b) Different number of features (K=8, 16, 32, 64)
Netflix dataset, top 5 items, model-based technique (matrix factorization CF)

(c) Different number of recommendations (top-1, 5, 10 items)
MovieLens dataset, heuristic-based technique (item-based CF, 50 neighbors)

Figure 4.5 Performance of Ranking Approaches with Different Parameters

In addition, our finding that the proposed ranking approaches help to improve recommendation diversity is also robust with respect to the "highly predicted" rating threshold value T_H. In particular, with a different threshold, the baseline recommendation accuracy and diversity of the standard ranking approach could be very different, and the number of actual recommendations that were produced by the system could change in case there were a limited number of items that were predicted higher than the minimum threshold. However, again we observed the same consistent ability of the proposed ranking approaches to achieve substantial diversity gains with only a small accuracy loss. For example, as shown in Table 4.3, with a different predicted rating threshold (i.e., T_{H} = 4.5) and 1% precision loss, we could obtain a 68% diversity gain by ranking the recommendations based on the item average rating in the top-1 recommendation task on the MovieLens dataset using the item-based CF for rating prediction. Similar improvements were observed for other datasets and rating prediction techniques as well.

Table 4.3 Performance of Ranking Approaches with a Different Predicted Rating Threshold (T_{H} = 4.5)

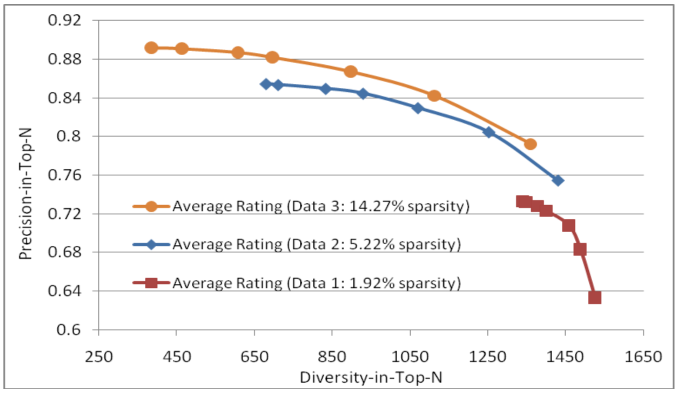
MovieLens dataset, top-1 items, item-based CF (50 neighbors)

	Item Popularity		Reverse Prediction		Item Average Rating		Item Abs Likeability		Item Relative Likeability		Item Rating Variance		Neighbors' Rating Variance	
Precision Loss	Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain	
-0.1	+187	2.928	+207	3.134	+289	3.979	+197	3.031	+262	3.701	+101	2.039	+141	2.454
-0.05	+127	2.309	+124	2.278	+189	2.948	+134	2.381	+182	2.876	+82	1.845	+83	1.856
-0.025	+72	1.742	+74	1.763	+99	2.021	+81	1.835	+101	2.041	+43	1.443	+43	1.443
-0.01	+48	1.495	+45	1.464	+66	1.680	+54	1.557	+55	1.567	+23	1.237	+18	1.186
-0.005	+41	1.420	+36	1.371	+58	1.598	+45	1.468	+47	1.485	+13	1.134	+10	1.103
-0.001	+35	1.362	+28	1.288	+52	1.536	+39	1.399	+41	1.423	+6	1.059	+4	1.039
Standard :0.775	97	1.000	97	1.000	97	1.000	97	1.000	97	1.000	97	1.000	97	1.000

Finally, the datasets we used for our experiments (see Table 4.1) were obtained using a specific sampling (pre-processing) strategy – by choosing items and users with the largest number of ratings (i.e., strategy of the top users and top items, described as Data 3 in Table 4.4). This process resulted in relatively dense rating datasets. Thus, for robustness analysis, we generated sparser datasets (Data 1 and 2 in Table 4.4) from the original MovieLens dataset by applying different sampling strategies that have been used in prior literature (Umyarov and Tuzhilin 2010). Table 4.4 summarizes the basic characteristics of these resulting datasets, including the strategies for choosing items and users. Figure 4.6 illustrates the impact of data sparsity on the recommendation results using one of the proposed re-ranking approaches, the average rating-based ranking, as an example. More importantly, as shown in Figure 4.6, the behavior of the proposed re-ranking techniques remained consistent with different data sampling approaches, i.e., making it is possible to obtain diversity improvements with only a small accuracy loss.

Table 4.4 MovieLens Datasets with Different Sampling Strategies

	Data 1	Data 2	Data 3
Number of users	2,830	2,830	2,830
Number of movies	1,919	1,919	1,919
Number of ratings	104,344	272,295	775,176
Data Sparsity	1.92%	5.22%	14.27%
(1) Choose users (among 6K users)	Random 2,830 users ranked between 1.5K and 4.5K		Top 2,830 users
(2) Choose items (among 3K items)	Random 1,919 items ranked 0.5K – 2.5K	Top 1,919 items	
Sampling strategy	Mid users Mid items	Mid users Top items	Top users Top items



Data 1-3, top-5 items, item-based CF, 50 neighbors

Figure 4.6 Diversity Gains with Sparse Datasets

4.5 Discussion and Additional Analysis

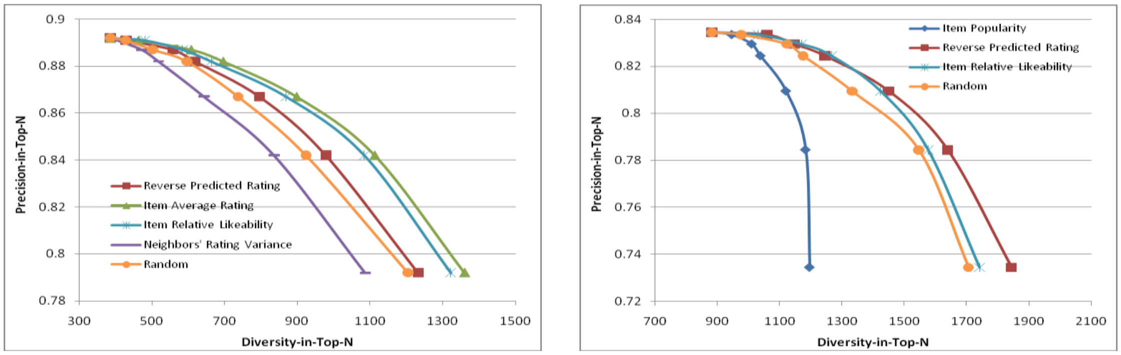
In this section, we explore and discuss several additional issues related to the proposed ranking approaches.

4.5.1 Random Ranking Approach

As mentioned earlier, the vast majority of traditional recommender systems have adopted the standard ranking approach that ranks the candidate items according to their predicted rating values and, thus, recommend the topmost highly predicted items to users. As discussed in Section 4.2, since the more highly predicted items, on average, tend to be among the more popular items, using this ranking approach often results in lower recommendation diversity. While the proposed ranking approaches improve diversity by considering alternative item ranking functions, such as item popularity, we also found that re-ranking the candidate items even at random provided diversity improvements as compared to the standard ranking approach. Here we define the random ranking as:

$$Rank_{\text{Random}}(i) = \text{Random}(0,1).$$

where Random(0,1) is a function that generates uniformly distributed random numbers in the [0, 1] interval. We compare some of the proposed ranking approaches with this random ranking approach in Figure 4.7. For example, as shown in Figure 4.7a, the random ranking approach increased diversity from 385 to 596 (55% gain) with a 1% precision loss using the heuristic-based CF technique on the MovieLens dataset. While this gain was not as large as the diversity gain of the average rating-based approach (80% gain), it outperformed the neighbors' rating variance-based approach (35% gain). As another example, as shown in Figure 4.7b, with only a 0.5% precision loss on the Netflix dataset using the model-based CF technique, the random ranking approach produced results that were almost as good (27% diversity gain) as several of the best-performing ranking approaches (with a 30% gain for the reverse predicted rating-based approach or a 33% gain for the relative likeability-based approach).



(a) MovieLens dataset, top 5 items, item-based CF, 50 neighbors

(b) Netflix dataset, top 5 items, matrix factorization CF, *K*=64

Figure 4.7 Diversity Gain of the Random Ranking Approach with Different Levels of Precision Loss

This provides a valuable insight that if the goal is to improve recommendation diversity without significant accuracy loss, even a random recommendation ranking approach can significantly outperform the traditional and widely used standard ranking approach (based on the predicted rating value). Furthermore, as illustrated in Figure 4.7, the random ranking approach works consistently well with different datasets and in conjunction with different CF techniques.

4.5.2 Improving Both Accuracy and Diversity: Recommending Fewer Items

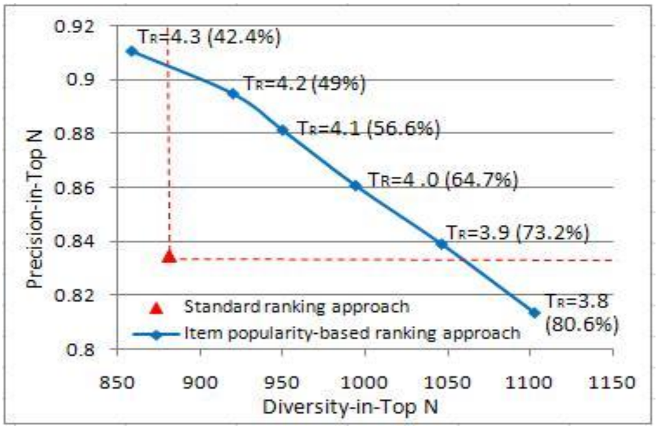
Empirical results in this chapter consistently show that the proposed ranking approaches can obtain significant diversity gains (with a small amount of accuracy loss) as compared to the standard ranking approach that ranks recommended items based on their predicted rating value. Therefore, another interesting topic for future research would be to explore the possibility of improving *both* accuracy and diversity.

Based on the findings described in this chapter, a possible approach to improving both the accuracy and diversity of the standard technique would be to modify the proposed recommendation re-ranking techniques, which are already known to produce diversity gains, in a way that increases their accuracy. Perhaps counter-intuitively, one of the possible ways involves recommending *fewer* items. In particular, the parameterized versions of the proposed ranking techniques use threshold T_R to differentiate the items that should be ranked by the proposed technique from the ones to be ranked by the standard ranking technique, as discussed in Section 4.2.3. However, T_R can be used not only for ranking, but also for filtering purposes, i.e., by updating the parameterized ranking function as follows:

$$rank_x(i, T_R) = \begin{cases} rank_x(i), & if\ R^*(u,i) \in [T_R, T_{\max}] \\ \text{Removeitem}, & if\ R^*(u,i) \in [T_H, T_R) \end{cases} .$$

This approach will recommend only items that are predicted to be not only above T_H, but also above T_R (where always $T_R \geq T_H$), consequently improving the recommendation accuracy.

While the comprehensive exploration of this phenomenon is beyond the scope of this chapter, in Figure 4.8 we illustrate how the item popularity-based ranking approach can be modified using the above-mentioned strict filtering policy to improve upon the standard approach both in terms of accuracy and diversity. As Figure 4.8 demonstrates, the item popularity-based ranking approach with $T_R = 4.1$ (out of 5) generates only 56.6% of all possible item recommendations that could be obtained from the standard ranking approach because the recommendations with predicted rating < 4.1 were removed. Interestingly, however, despite the smaller number of recommendations, this ranking approach increased the recommendation accuracy by 4.6% (from 83.5% to 88.1%) and diversity by 70 items or 7.8% (from 881 to 951). As shown in Figure 4.8, using different T_R values produces different accuracy and diversity gains.



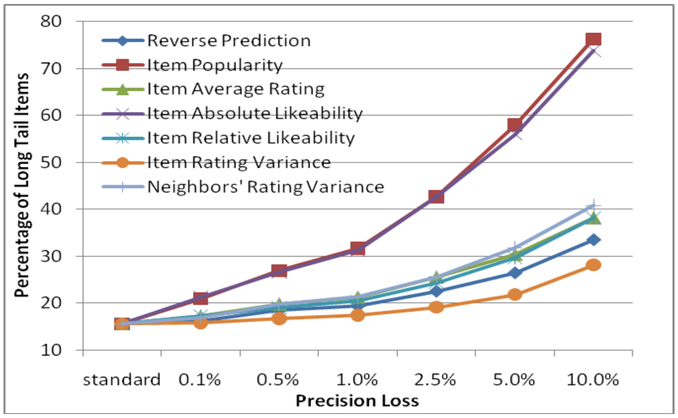
Netflix data, top-5 items, matrix factorization CF, K=64

Figure 4.8 Improving *both* Accuracy and Diversity of Recommendations

This modified popularity-based approach would not be able to provide all N recommendations for each user; nevertheless, it may be useful in cases where system designers need the flexibility to apply other recommendation strategies to fill out the remaining top-N item slots. For example, some recommender systems may want to adopt an "exploration-vs-exploitation" strategy (Hagen et al. 2003), where some of the recommendations are tailored directly towards users' tastes and preferences (i.e., exploitation), and the proposed ranking techniques with strict filtering can be used to fill out this part of the recommendation list for each user, providing both accuracy and diversity benefits over the standard approach. Meanwhile, the remaining recommendations can be designed to learn more about the user (i.e., exploration), e.g., using *active learning* techniques (Huang 2007, Zheng and Padmanabhan 2006), so that the system can make better recommendations in the future.

4.5.3 Impact of Ranking Approaches on the Distribution of Recommended Items

Since we measure recommendation diversity as the total number of distinct items that are being recommended across all users, one could possibly argue that, while the diversity can be easily improved by recommending a few new items to some users, it may not be clear whether the proposed ranking approaches would be able to shift the overall *distribution* of recommended items towards more idiosyncratic, long-tail recommendations. Therefore, in this subsection we explore how the proposed ranking approaches change the actual distribution of recommended items in terms of their popularity. Following the popular "80-20 rule" or the Pareto principle, we define the top 20% of the most frequently rated items in the training dataset as "bestsellers" and the remaining 80% of items as "long-tail" items. We calculated the percentage of long-tail items among the items recommended across all users by the proposed ranking approaches as well as by the standard ranking approach. The results are shown in Figure 4.9.



MovieLens dataset, top 5 items, item-based CF, 50 neighbors

Note: Percentage of Long-Tail Items = Percentage of recommended items that are not among the top 20% most popular items

Figure 4.9 Proportion of Long-Tail Items among Recommended Items

For example, with the standard ranking approach, the long-tail items consist of only 16% of the total recommendations (i.e., 84% of the recommendations were of bestsellers) when recommending the top-5 items to each user using the item-based CF technique on the MovieLens dataset. This confirms some findings in prior literature that recommender systems often gravitate towards recommending bestsellers and not long-tail items (Fleder and Hosanagar 2009). However, as shown in Figure 4.9, the proposed ranking approaches are able to recommend significantly more long-tail items with a small level of accuracy loss, and this distribution becomes even more skewed towards long-tail items if more accuracy loss can be tolerated. For example, with a 1% precision loss, the percentage of recommended long-tail items increased from 16% to 21% with the neighbors' rating variance-based ranking approach, or to 32% with item popularity and item absolute likeability-based approaches. In addition, with a 2.5% or 5% precision loss, the proportion of long-tail items can grow up to 43% and 58%, respectively, using the item popularity ranking technique.

This analysis provides further empirical support for the fact that the proposed ranking approaches increase not just the number of distinct items recommended, but also the proportion of recommended long-tail items, thus, confirming that the proposed techniques truly contribute to more diverse and idiosyncratic recommendations across all users.

In addition to the distributional analysis based on the simple proportion of long-tail items, we also used three more sophisticated metrics: *entropy* or *Shannon's diversity index* (Shannon 1948), *Gini coefficient* (Gini 1921), and *Simpson's diversity index* (Simpson 1949), which is also known as *Herfindahl index* (Herfindahl 1950). All of these measures provide different ways of measuring *distributional* dispersion of recommended items across all users, by showing the degree to which recommendations are concentrated on a few popular items (low diversity) or are more equally spread out across all candidate items (high diversity). In particular, the entropy-based diversity metric *Entropy-Diversity* is calculated as:

$$Entropy\text{-}Diversity = -\sum_{i=1}^{n}\left(\frac{rec(i)}{total}\right)\ln\left(\frac{rec(i)}{total}\right),$$

where *rec*(*i*) is the number of users who got recommended item *i*, *n* is the total number of candidate items that were available for recommendation, and *total* is the total number of top-*N* recommendations made across all users (*total* = *N* |*U*|). We also used the original Gini coefficient, which is a commonly used measure of wealth distribution inequality (Gini 1921), to calculate the *Gini-Diversity* metric, and the original *Simpson's diversity index* (Simpson 1949) to calculate the *Simpson-Diversity* metric. However, we reversed the scale for the two metrics for more intuitiveness so that smaller values represent lower diversity and larger values higher diversity. As a result, these metrics are calculated as:

$$Gini\text{-}Diversity = 2\sum_{i=1}^{n}\left[\left(\frac{n+1-i}{n+1}\right)\times\left(\frac{rec(i)}{total}\right)\right],$$

$$Simpson\text{-}Diversity = 1-\sum_{i=1}^{n}\left(\frac{rec(i)}{total}\right)^{2}.$$

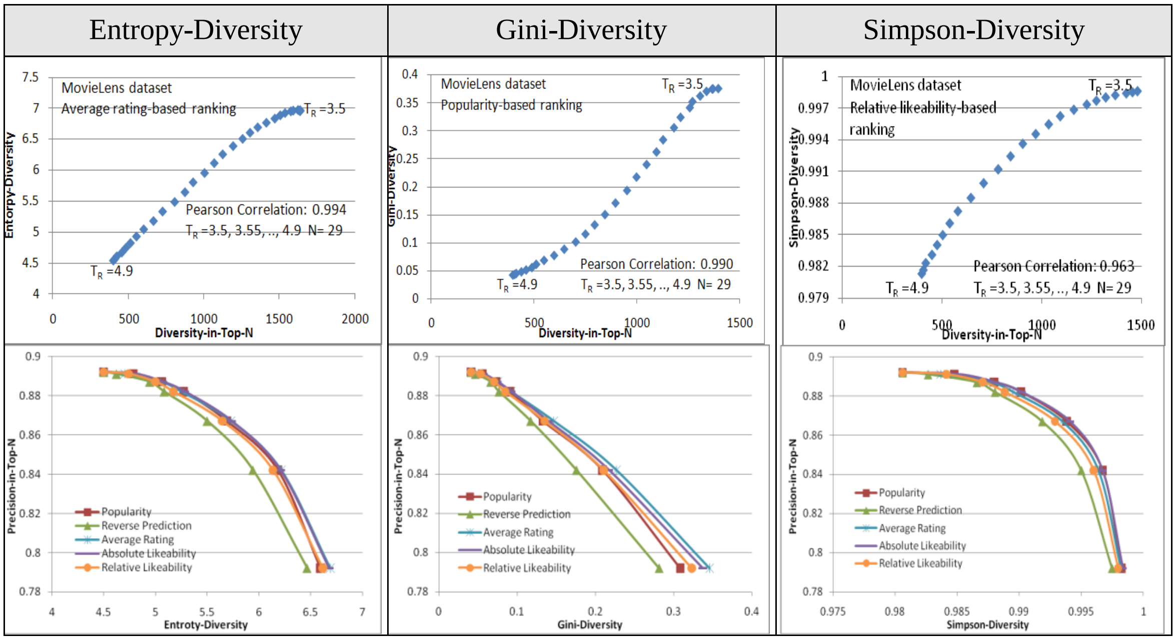


Figure 4.10 Using Distributional Diversity Metrics: Correlation with Diversity-in-top-*N* and Performance of Ranking Approaches

The top three graphs of Figure 4.10 demonstrate that all three distributional inequality metrics are very highly correlated with our *diversity-in-top-N* metric for our various ranking-based approaches. This means that our proposed ranking techniques do not just manipulate our simple *diversity-in-top-N* metric to increase the number of different items among the recommendations, but also fundamentally change the distribution of recommended items toward more evenly distributed representation. The bottom three graphs of Figure 4.10 also show that the ranking approaches exhibit similar patterns of diversity gains (or, more generally, of the accuracy-diversity tradeoff) using these more sophisticated distributional inequality metrics. This provides an additional confirmation that the proposed re-ranking techniques truly contribute to more diverse and idiosyncratic recommendations across all users.

4.6 Conclusion and Future Work

Recommender systems have made significant progress in recent years and many techniques have been proposed to improve the recommendation quality. However, in most cases, new techniques are designed to improve the accuracy of recommendations, but recommendation diversity has often been overlooked. In particular, we showed that, while ranking recommendations according to the predicted rating values, which is a *de facto* ranking standard in recommender systems, provides good predictive accuracy, it tends to perform poorly with respect to recommendation diversity. Therefore, in this chapter, we proposed a number of recommendation ranking techniques that can provide significant improvements in recommendation diversity with little accuracy loss. In addition, these ranking techniques offer flexibility to system designers, since they are parameterizable and can be used in conjunction with different rating prediction algorithms (i.e., they do not require the designer to use only a specific algorithm). They are also based on scalable sorting-based heuristics and, thus, are extremely efficient. This chapter provides a comprehensive empirical evaluation of the proposed techniques, indicating that our experiments obtained consistent and robust diversity improvements across multiple real-world datasets using different rating prediction techniques.

This work gives rise to several interesting directions for future research. In particular, additional important item ranking criteria should be explored for potential diversity improvements. This may include consumer-oriented or manufacturer-oriented ranking mechanisms (Ghose and Ipeirotis 2007), depending on the given application domain, as well as external factors, such as social networks (Lemire et al. 2008). In addition, because of the inherent tradeoff between the accuracy and diversity metrics, an interesting research direction would be to develop a new measure that captures both of these aspects in a single metric. Moreover, user studies exploring users' perceptions and acceptance of the diversity metrics as well as the users' satisfaction with diversity-sensitive recommender systems would also be an important step in this line of research. Furthermore, exploration of recommendation diversity when recommending item *bundles* (Garfinkel et al. 2006) or *sequences* (Shani et al. 2005) instead of individual items also constitute interesting topics for future research. In summary, we hope that this work will stimulate further research on improving recommendation diversity and other aspects of recommendation quality.

Chapter 5. Optimization-Based Approaches to Maximize Aggregate Recommendation Diversity

5.1 Introduction

Taking into consideration the potential benefits of *aggregate* diversity to individual users and businesses, as discussed in Section 2.4, one line of research (Kim et al. 2010, Levy and Bosteels 2010, Park and Tuzhilin 2008) aims to enhance the rating estimation phase, mainly for long-tail items, and the other focuses on finding the best set of recommendations in the recommendation generation phase (Adomavicius and Kwon 2009, 2011). The approaches proposed in this chapter fit within the latter line of research and, therefore, have the flexibility of being used in conjunction with *any* available rating estimation algorithm, as illustrated by our empirical evaluation. We build upon the recommendation re-ranking heuristics proposed by Adomavicius and Kwon (2009, 2011). While these re-ranking approaches do not provide direct control over diversity, here we develop more sophisticated and systematic optimization-based approaches for direct diversity maximization, while maintaining acceptable levels of accuracy. More specifically, our objective is to find the best top-*N* recommendation lists for all users according to two measures: accuracy and diversity. We address this by introducing three approaches of increasing complexity and sophistication: (1) a greedy heuristic for direct diversity improvement; (2) a graph-theoretic maximum-flow based approach to diversity maximization; and (3) an integer programming model for diversity maximization with explicit accuracy guarantees.

Our empirical results using real-world rating datasets show that all of the proposed optimization-based approaches consistently outperform the recommendation re-ranking approach from prior literature in terms of both accuracy and diversity. This chapter also discusses the scalability of each proposed approach in terms of their theoretical computational complexity as well as their empirical runtime based on real-world rating datasets.

The remainder of the chapter is organized as follows. In Section 5.2, we give a brief overview of a simple recommendation re-ranking approach from prior literature, which will be used as a baseline comparison technique in our experiments. Section 5.3 describes three proposed optimization-based approaches and their computational complexity, followed by empirical results in Section 5.4. While our optimization approaches are designed to increase simple diversity metric (i.e., the number of distinct items recommended across all users), we also discuss some possibilities for new approaches that can improve other diversity metrics in Section 5.5. Section 5.6 concludes the chapter with several future directions.

5.2 Related Work

This section provides an overview of prior work that shares the same objective of proposing new recommendation techniques to improve top-*N* item selection *after* the rating estimation is performed. In particular, Adomavicius and Kwon (2009, 2011) propose a heuristic approach for recommendation re-ranking, which has been shown to improve aggregate diversity with negligible accuracy loss and represents an important baseline for comparison with our proposed diversity maximization approaches. Typical recommender systems recommend to users those items that have the highest predicted ratings, using the standard recommendation ranking criterion $rank_{\text{Standard}}$, as discussed in Section 2.1. While the standard ranking approach is used to maximize the accuracy of recommendations, Adomavicius and Kwon (2009, 2011) showed that changing the ranking of items (i.e., not following the standard ranking approach) can help with other aspects of recommendation quality, in particular, with recommendation diversity. As a result, they proposed several alternative re-ranking approaches, and showed that all of them can provide substantial improvements in recommendation diversity with only negligible accuracy loss (refer to Chapter 4 for more detail). In our experiments, as a baseline for comparison, we specifically use the ranking approach based on the reverse predicted rating value. This is a personalized yet simple and highly-scalable ranking approach that can be formally defined as $rank_{\text{RevPred}}(i) = R^*(u,i)$.

While this re-ranking approach can significantly improve recommendation diversity, as might be expected, this improvement comes at the expense of recommendation accuracy, since the most highly predicted items are generally not recommended. Adomavicius and Kwon (2009, 2011) demonstrated that the balance between diversity and accuracy can be achieved by parameterizing any ranking function with "ranking threshold" $T_R \in [T_H, T_{max}]$, where T_{max} is the largest rating on the rating scale. That is, the ranking threshold allows the user to specify the level of acceptable accuracy loss while still extracting a significant portion of diversity improvement. In particular, the parameterized version $rank_{\text{RevPred}}(i, T_R)$ of ranking function $rank_{\text{RevPred}}(i)$ can be implemented as:

$$rank_{\text{RevPred}}(i, T_R) = \begin{cases} rank_{\text{RevPred}}(i), & \textit{if } R^*(u,i) \in [T_R, T_{\max}] \\ \alpha_{\text{u}} + rank_{\text{Standard}}(i), & \textit{if } R^*(u,i) \in [T_H, T_R) \end{cases},$$

$$\text{where } \alpha_{\text{u}} = \max_{i \in I_u^*(T_R)} rank_{\text{RevPred}}(i), \text{ and } I_u^*(T_R) = \{i \in I \mid R^*(u,i) \geq T_R\}$$

In particular, items with predicted ratings from $[T_R, T_{max}]$ would be ranked ahead of items with predicted ratings $[T_H, T_R)$, as ensured by α_u in the above definition. Increasing the ranking threshold T_R towards T_{max} would enable the ability to choose the most highly predicted items, with more accuracy and less diversity, which is similar to the standard ranking approach, while decreasing the ranking threshold T_R towards T_H makes $rank_{\text{RevPred}}(i, T_R)$ increasingly more similar to the pure ranking function $rank_{\text{RevPred}}(i)$, i.e., more diversity with some accuracy loss. Thus, choosing $T_R \in [T_H, T_{max}]$ values in-between the two extremes allows the ability to set the desired balance between accuracy and diversity. In our experiments, we were able to explore the accuracy-diversity tradeoff of the re-ranking approach, by varying this ranking threshold T_R.

5.3 Optimization-Based Approaches for Maximum Diversity

While the recommendation re-ranking approach can obtain a certain level of diversity gains at the expense of a small amount of accuracy loss, in this section we propose three optimization-based approaches that can directly control the diversity level, by either specifying the desired level of diversity in advance or obtaining the maximum possible diversity. Sections 5.3.1-5.3.3 describe each of the proposed approaches, and Section 5.3.4 discusses their computational complexity.

5.3.1 Greedy Approach for Diversity Improvement

The re-ranking approach, briefly discussed in Section 5.2, can improve recommendation diversity by recommending those items that have lower predicted ratings among the items predicted to be relevant, by changing ranking threshold T_R, but it does not provide direct control on how much diversity improvement can be obtained. To address this limitation, we first present a greedy diversity maximization heuristic, which attempts to directly increase the number of distinct items recommended across all users (i.e., improve the *diversity-in-top-N* measure).

The basic idea behind this iterative approach is as follows. First, the standard ranking approach is applied to each user, to obtain the initial top-*N* recommendations, typically with the best accuracy. Then, iteratively, one of the already-recommended items is replaced by another candidate item (predicted to be above the relevance threshold T_H) that has not yet been recommended to anyone, thereby increasing the diversity by one unit, until the diversity increases to the desired level, or until there are no more new items available for replacement.

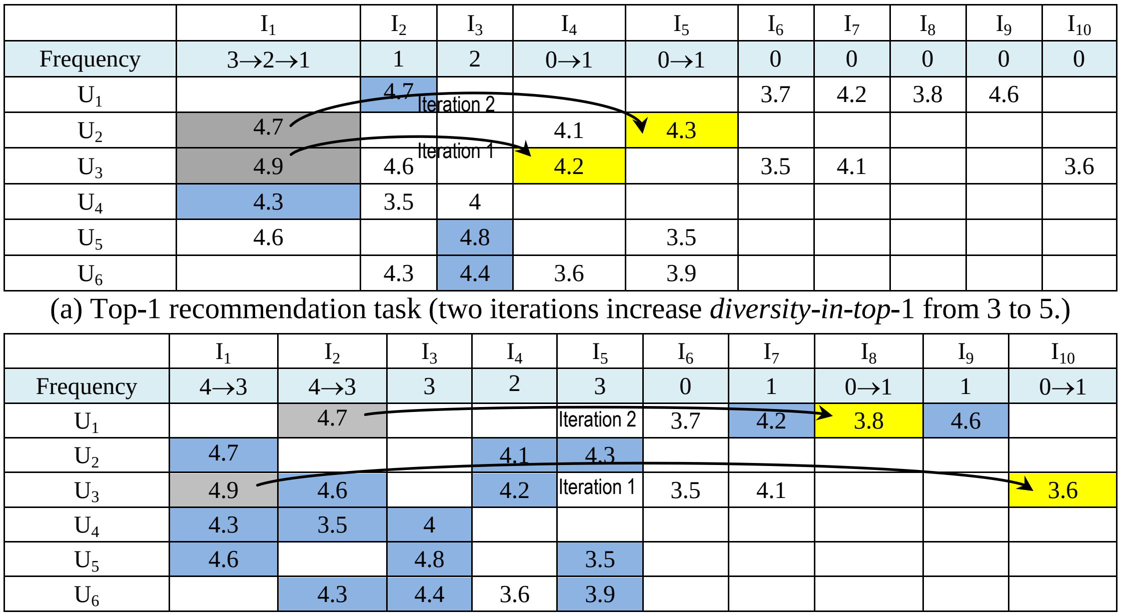
Since item replacement is made only when it results in an immediate improvement of diversity by on unit, we refer to this approach hereafter as a “greedy” approach, which is formally described in Figure 5.1. Each item replacement iteration is implemented as follows. The most frequently recommended item i_{old} is replaced by one of the never-recommended items i_{new} for the same user. Among all the users who got recommended item i_{old}, a replacement occurs for user u_{max}, who is predicted to rate item i_{old} most highly, allowing for a possibly higher predicted rating value for the replacement item i_{new} (and, therefore, for better accuracy). In other words, since any new candidate item for replacement i_{new} is predicted to be lower than item i_{old} for the chosen user, the higher the prediction of item i_{old}, the higher the possibility of obtaining a high prediction of the new item i_{new}.

As described in Figure 5.1, throughout all iterations, we keep track of a list of replaceable items that have been recommended to more than one user in array *CI*. In addition, in each iteration, potential users that have both item i_{old} and item i_{new} are stored in array *CU*. A new item is then chosen as the most highly predicted item among the candidate items available for user u_{max}, resulting in recommending diverse, but still relatively highly predicted items.

```
for each user $u \in U$ do          // initialize top-$N$ recommendation lists using the standard ranking approach
    $L_N(u)$:= top $N$ items recommended to user $u$ according to $rank_{\text{Standard}}$
    // set of new items that can replace old items
    $NL(u)$:= the remaining highly predicted items available to user $u$
end for
$CI := I$                            // initialize set of old (currently recommended) items that can be replaced
for each item $i \in I$ do
    $rec(i)$ := number of users for whom $i$ is recommended
    // cannot be a new item if already recommended
    if $rec(i) > 0$ then remove $i$ from $NL(u)$ for each user $u \in U$
    if $rec(i) \le 1$ then $CI := CI\backslash\{i\}$    // cannot be replaced if not recommended or recommended only once
end for
$diversity := | \cup_{u \in U} L_N(u) |$                 // $diversity$-$in$-$top$-$N$ based on current $L_N(u)$ lists
$target\_diversity := diversity + \theta$                // $\theta$: desirable diversity gain
// until target diversity is reached or no replaceable items exist
while ( $diversity < target\_diversity$ and $CI \neq \emptyset$ )
    found := false                                       // replacement not found yet
    // until a pair of items for replacement are found or no replaceable items exist
    while ( not found and $CI \neq \emptyset$ ) do
        $i_{\text{old}}$ := argmax$_{\text{i}}$ { $rec(i)$ | $i \in CI$ }          // old item with highest recommendation frequency
        $CU$ := { $u \in U$ | $i_{\text{old}} \in L_N(u)$ and $NL(u) \neq \emptyset$  // set of users who have $i_{\text{old}}$ and possible replacements
        // if no such users found, $i_{\text{old}}$ is removed from $CI$
        if ($CU \neq \emptyset$) then found:= true else $CI := CI \backslash\{i_{\text{old}}\}$ end if
    end while
    if (found) then
        $u_{\text{max}}$ := argmax$_{\text{u}}$ { $R^*(u, i_{\text{old}})$ | $u \in CU$ }          // user with the highest prediction of $i_{\text{old}}$
        $i_{\text{new}}$ := argmax$_{\text{i}}$ { $R^*(u_{\text{max}}, i)$ | $i \in NL(u_{\text{max}})$ }    // new item with the highest prediction for $u_{\text{max}}$
        Replace $i_{\text{old}}$ with $i_{\text{new}}$ in $L_N(u_{\text{max}})$                  // make the replacement
        $rec(i_{\text{old}}) := rec(i_{\text{old}}) - 1$
        $rec(i_{\text{new}}) := rec(i_{\text{new}}) + 1$
        if $rec(i_{\text{old}}) \le 1$then $CI := CI \backslash\{i_{\text{old}}\}$     // keep $i_{\text{old}}$ only if still recommended to more than one user
        Remove $i_{\text{new}}$ from $NL(u)$ for each user $u$ in $U$       // remove $i_{\text{new}}$ from set of new items for all users
        $diversity := | \cup_{u \in U} L_N(u) |$
    end if
end while
```

Figure 5.1 Algorithm of the Greedy Approach

Figure 5.2 illustrates the greedy approach using two small examples of top-N recommendation settings (for N = 1, 3), where there are 6 users, 10 distinct items, and 24 candidate items for recommendations that are predicted to be relevant items. In the top-1 recommendation task, as shown in Figure 5.2a, the standard ranking approach starts by recommending the items that are the most highly predicted for each user: I_1 (for U_2, U_3, U_4), I_2 (for U_1), I_3 (for U_5, U_6). Then, the greedy approach replaces the most frequently recommended item I_1 (the frequency of recommendation is 3) with I_4 for U_3, who is predicted to rate the item most highly (4.9 out of 5). Among the items that can be recommended to U_3 and have not yet been recommended to anyone else (I_4, I_6, I_7, I_{10}), I_4 is chosen for replacement because of its highest prediction (4.2). Similarly, in the next iteration, we again replace I_1, which is still the most frequently recommended item, with I_5 for U_2. After these two replacements, no other greedy diversity-increasing replacements are possible. As a result, the *diversity-in-top*-1 increases from 3 (obtained by the standard ranking approach) to 5. Similarly, for the top-3 recommendation example, shown in Figure 5.2b, the proposed approach also performs two replacements, increasing the diversity from 7 to 9. Naturally, if the desired diversity level were 8, the greedy optimization algorithm would stop after one replacement.



	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}
Frequency	3→2→1	1	2	0→1	0→1	0	0	0	0	0
U_1		4.7				3.7	4.2	3.8	4.6	
U_2	4.7			4.1	4.3					
U_3	4.9	4.6		4.2		3.5	4.1			3.6
U_4	4.3	3.5	4							
U_5	4.6		4.8		3.5					
U_6		4.3	4.4	3.6	3.9					

Iteration 2
Iteration 1

(a) Top-1 recommendation task (two iterations increase *diversity-in-top*-1 from 3 to 5.)

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}
Frequency	4→3	4→3	3	2	3	0	1	0→1	1	0→1
U_1		4.7				3.7	4.2	3.8	4.6	
U_2	4.7			4.1	4.3					
U_3	4.9	4.6		4.2		3.5	4.1			3.6
U_4	4.3	3.5	4							
U_5	4.6		4.8		3.5					
U_6		4.3	4.4	3.6	3.9					

Iteration 2
Iteration 1

(b) Top-3 recommendation task (two iterations increase *diversity-in-top*-1 from 7 to 9.)

Figure 5.2 Examples of Applying the Greedy Approach

In summary, this heuristic maximization approach can improve recommendation diversity while maintaining relatively high accuracy by replacing one of the most frequently recommended items with the most highly predicted one among the items that have not been recommended. However, being a greedy approach, it does not guarantee that the maximum possible diversity will be achieved. In the next two subsections, we propose two different optimization approaches for computing the top-*N* recommendation lists with maximum possible diversity.

5.3.2 Max-Flow Based Approach for Diversity Maximization

Graph-based algorithms have been previously used in recommender systems (Aggarwal et al. 1999, Huang et al. 2004, 2007, Liu et al. 2009), mostly for the purpose of improving predictive accuracy of CF techniques. We show that graph-based approaches can also be useful for the diversity improvement problem, by formulating it as a well-known max-flow problem in graph theory (Ahuja et al. 1993, Cormen et al. 2001).

One simple version of the general maximum flow problem, which has been extensively studied in operations research and combinatorial optimization, can be defined as follows. Assuming that V is the set of vertices (or nodes), and E is the set of directed edges, each of which connects two vertices, let G=(V, E) be a directed graph with a single source node $s\in V$ and a single sink node $t\in V$. Each directed edge $e\in E$ has capacity $c(e)\in\mathbf{R}$ associated with it. The amount of actual flow between the two vertices is denoted by $f(e)\in\mathbf{R}$. The flow of an edge cannot exceed its capacity, and the sum of the flows entering a vertex must equal the sum of the flows exiting a vertex, except for the source and the sink vertices. The maximum flow problem is to find the largest possible amount of flow passing from the source to the sink for a given graph G.

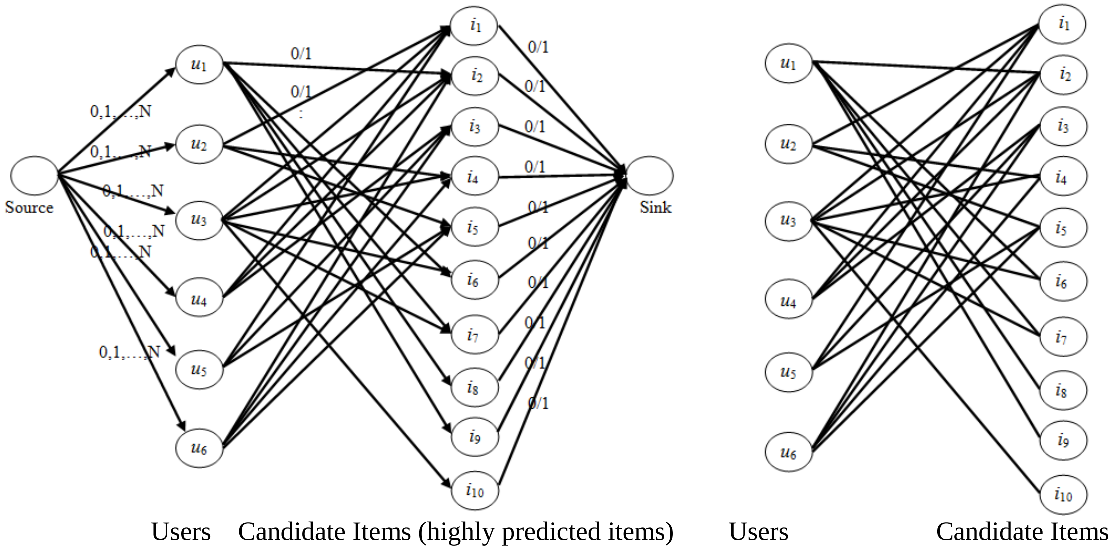
Translating the top-*N* recommendation setting into a graph-theoretic framework, let users and items be represented as vertices, and an edge from user u to item i exists if and only if item i is predicted to be relevant for user u, i.e., $R^*(u, i) \geq T_H$ or, in other words, when the item is available to the user for recommendation. Each edge has capacity $c(e)$ = 1 and can be assigned the integer flow of 1 if item i is actually recommended to user u as part of the top-*N* recommendations, and the flow of 0 otherwise. As described in the example in Figure 5.3a, we augment this directed graph by adding a source node and connecting it by directed edges to each of the user vertices. Let the capacity of each of these “source” edges be *N* and, again, only integer flows of 0, 1, …, or *N* are permitted on each of these edges. Furthermore, as shown in Figure 5.3a, we also augment this graph by adding a sink node and connecting each item vertex by a directed edge to this node. Let the capacity of each of these “sink” edges be 1, and again only integer flows (i.e., 0 or 1) are permitted for these edges.

As can be easily seen from this construction, because of the specified capacity constraints, (i.e., the “source” edges do not allow flows larger than *N* through each user node and the “sink” edges do not allow flows larger than 1 through each item node), the maximum flow value in this graph will be equal to the maximum possible number of edges from users to items that can have a flow of 1 assigned to them. In other words, the max-flow value will be equal to the largest possible number of recommendations that can be made from among the available (highly predicted) items, where no user can be recommended more than *N* items, and no item can be counted more than once, which is precisely the definition of the *diversity-in-top-N* metric.

Note that, while finding the maximum flow will indeed find the recommendations that yield maximum diversity, since the recommendation of each item is counted only once (i.e., restricted to only one user), as part of the max-flow solution, some users may have fewer than *N* recommendations. The remaining recommendations for these users can be filled arbitrarily, as they cannot further increase the maximum diversity. For the purpose of achieving better accuracy, we employ the standard ranking approach for the not-yet-recommended items for each user, so the items with the highest predictions are chosen for recommendation.

The maximum flow problem represents a simple and intuitive metaphor for computing the top-*N* recommendations with the maximum possible aggregate diversity, and there are many efficient (polynomial-time) algorithms for finding the maximum flow in a given graph (Ahuja et al. 1993, Cormen et al. 2001). Note, however, that the flow graph constructed for the diversity maximization problem is a highly specialized graph, and it may be possible to find even more effective graph-based algorithms for this problem, as compared to general-purpose max-flow algorithms.

To illustrate this, let’s consider the simplest top-*N* recommendation setting, where *N* = 1. Since each user can receive only one recommended item, *all* edges in our max-flow problem would become single-unit capacity edges, implying that the max flow in this graph will correspond to the largest possible set of edges from users to items, where no user and no item can be part of more than one such edge. Because there are no edges between two different users or between two different items for the top-1 recommendation settings (i.e., we have a bipartite user-item graph), the maximum flow problem is equivalent to the more specialized *maximum bipartite matching* problem which has more efficient algorithmic solutions. Therefore, while the max-flow approach represents a general, intuitive approach for achieving maximum diversity by implementing a single-source and single-sink flow network, we follow the equivalent yet more efficient maximum bipartite matching approach (as illustrated in Figure 5.3b) and also show how it can be extended from the top-1 to more general top-*N* recommendation settings.



(a) Maximum Flow Problem (b) Maximum Bipartite Matching Problem

Figure 5.3 Top-*N* Recommendation Task as a Graph Theory Problem

As summarized in Figure 5.4, our max-flow/matching optimization approach consists of two steps: (1) find the maximum diversity by solving the maximum bipartite matching problem, and (2) complete the top-N recommendations by applying the standard ranking approach. Since the maximum diversity in Step 1 can be obtained at some expense of accuracy, one can control the balance between accuracy and diversity with the simple parameterization of a "flow-rating threshold" $T_F \in [T_H, T_{max}]$. This allows pre-processing of the data, specifically, to include only higher predicted items above T_F among the items that can be recommended for the maximum diversity in Step 1. Similar to how the ranking threshold was used in the re-ranking approaches (Section 5.2), here the lowest T_F value provides the best diversity but a relatively lower accuracy, whereas higher values of T_F lower the diversity but provide a certain level of accuracy. Then, in Step 2, the highest predicted remaining items are used to complete the top-N recommendation lists.

More formally, let $G = (U, I; E)$ be a bipartite graph, where vertices represent users U and items I, and edges E represent the possible recommendations of items for users. A subset of edges M (i.e., $M \subseteq E$) is *matching*, if all edges in M are pairwise non-adjacent, i.e., any two edges in M share neither a user vertex nor an item vertex. A vertex is *matched* if it is adjacent to an edge in M; otherwise, the vertex is unmatched. The *maximum matching* of a bipartite graph is a match with the largest possible number of edges. The maximum bipartite matching algorithm (for top-1 recommendations) in Step 1 starts with matching $M = \varnothing$ and iteratively adds edges to M, until all users are matched or no new additional edge can be added. The edges to be iteratively added to M can be found by finding an *augmenting path* for M, which is a simple path (i.e., a sequence of alternating user and item vertices with no loops) that starts at an unmatched user and ends at an unmatched item, and its edges belong alternately to $E \backslash M$ and M. In other words, $P=(v_1, v_2, \ldots, v_{2n-1}, v_{2n})$ is an augmenting path where $v_{odd} \in U$, $v_{even} \in I$, v_1 is an unmatched user, v_{2n} is an unmatched item, $(v_{2k-1}, v_{2k}) \notin M$ where $k = \{1, \ldots, n\}$, and $(v_{2k+1}, v_{2k}) \in M$ where $k = \{1, 2, \ldots, n-1\}$. Let *edges*($P$) comprise the set of all edges of the augmenting path P. The key property of augmenting paths is that the symmetric set difference of M and *edges*(P), denoted as $M \Delta$ *edges*(P), always results in a match with cardinality one more than the cardinality of M (Ahuja et al. 1993, Cormen et al. 2001), i.e., if $M' = M \Delta$ $edges(P)$, then $|M'|=|M|+1$.

[Step 1] Find Maximum Diversity

```
$E := \{(u,i) \mid u \in U, i \in I, R^*(u, i) \in [T_F, T_{max}]\}$    // set of edges- items available for recommendation
$G := (U, I ; E)$    // bipartite graph with users, items, and edges
$CU := U$ ; $CI := \{i \in I \mid u \in U, (u,i) \in E\}$    // initialize a set of unmatched users/items
$M := \varnothing$    // set of matched edges $M \subseteq E$
```

Maximum Bipartite Matching (Top-1 Task)

```
// find augmenting paths starting from unmatched
// user $v_1$ and ending with unmatched item $v_{2n}$
$P$ := Find_AugmentingPaths($G$, C$U$, $CI$, $M$)
// until all users are matched or no augmenting path exists
while ($CU \neq \varnothing$ and $P \neq \varnothing$)
  for each $(v_1, v_2, \ldots, v_{2n-1}, v_{2n}) \in P$ do
    $edges := \{(v_{2k-1}, v_{2k}) \mid k := 1..n\} \cup$
      $\{(v_{2k+1}, v_{2k}) \mid k := 1..n-1\}$
    // flip the matched and unmatched edges
    $M := M \Delta edges$    // symmetric difference

    Remove $v_1$ from $CU$    // one match per user
    Remove $v_{2n}$ from $CI$    // one match per item
  end for
  $P$ := Find_AugmentingPaths ($G$, C$U$, $CI$, $M$)
end while
```

Extended Version for Top-N Task

```
// number of matches for each user
$\forall u \in U$, $uCnt[u] := 0$
$P$ := Find_AugmentingPaths($G$, C$U$, $CI$, $M$)

while ($CU \neq \varnothing$ and $P \neq \varnothing$)
  for each $(v_1, v_2, \ldots, v_{2n-1}, v_{2n}) \in P$ do
    $edges := \{(v_{2k-1}, v_{2k}) \mid k := 1..n\} \cup$
      $\{(v_{2k+1}, v_{2k}) \mid k := 1..n-1\}$

    $M := M \Delta edges$
    $uCnt[v_1] := uCnt[v_1] + 1$    // $N$ matches
    Remove $v_1$ from $CU$ if $uCnt[v_1] == N$
    Remove $v_{2n}$ from $CI$    // one match
  end for
  $P$ := Find_AugmentingPaths ($G$,C$U$,$CI$,$M$)
end while
```

[Step 2] Complete Top-N Recommendations

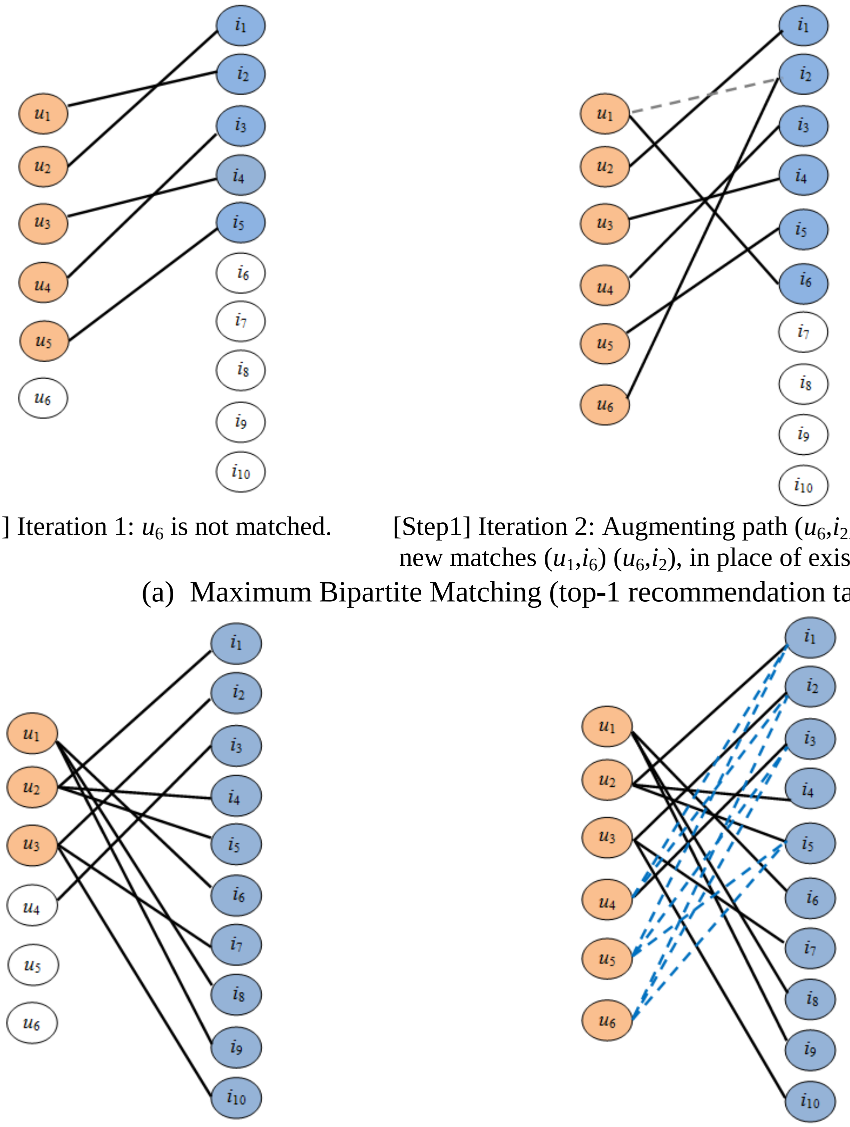
```
for each $(u, i) \in M$ do
  Add $i$ to $L_N(u)$    // assign matches as recommendations
end for
for each $u \in CU$ do    // fill the remaining items for each not-fully-matched user according to $rank_{Standard}$
  Sort items $\{i \in I \mid R^*(u, i) \in [T_H, T_{max}]$ and $i \notin L_N(u)\}$
  Add top $(N - |L_N(u)|)$ most highly predicted items to $L_N(u)$
end for
```

Figure 5.4 Algorithm of the Max-Flow/Matching Based Optimization Approach

Thus, the notion of augmenting paths allows the ability to find the maximum bipartite matching, by starting with matching $M = \varnothing$ and iteratively increasing its size one-by-one with each augmenting path, which we use in our algorithm for diversity maximization (Figure 5.4). In particular, we adopt the Hopcroft-Karp algorithm (1973), which finds a maximal set of augmenting paths during every iteration, so that multiple augmenting paths in parallel for all unmatched vertices, thereby achieving a significant reduction in time complexity. This is a well-known technique and we encapsulate it in our algorithm by using *Find_AugmentingPaths* subroutine (lines 6, 15 in Figure 5.4). The implementation details for this subroutine can be found in Burkard et al. (2009).

The original bipartite matching algorithm for the top-1 recommendations matches a user to only one item and excludes the matched user for the subsequent iterations, so the user is removed from the candidate user list *CU* (line 12 of Figure 5.4). An extended version for the top-*N* recommendations relaxes this rule by waiting to remove the user from the *CU* until the same user is matched to *N* items. We also make the extended algorithm more efficient by allowing a single user to find up to *N* item matches in the first iteration (and not just a single match per iteration), which significantly reduces the number of subsequent iterations. However, similar to the max-flow approach where an item can be recommended to only one user, some users may receive fewer than *N* recommendations. Thus, in Step 2, for accuracy considerations, the most highly predicted items among the remaining candidate items are chosen to fill the remaining top-*N* recommendations for all users. Note that this does not affect diversity, which is already guaranteed to be maximum.

Using the same example from Figure 5.2, we illustrate the first step for the top-1 recommendations, how the maximum bipartite matching algorithm can obtain the maximum diversity (Figure 5.5a). This algorithm performs two iterations: (1) finds all possible 1-edge augmenting paths between unmatched users and unmatched items, i.e., direct paths without any intermediate vertices; and (2) finds multi-edge augmenting paths, each of which increases the cardinality of matching by one unit via alternating non-matched and matched edges in the paths. After the first iteration in Figure 5.5a, the first five users are matched to one of their candidate items, but user u_6 is still unmatched because all of the candidate items (i_2, i_3, i_4, i_5) are already matched to other users. The second iteration finds an augmenting path from unmatched user u_6 to unmatched item i_6, i.e., $P= (u_6,i_2,u_1,i_6)$ and $(u_6,i_2)\notin M$, $(u_1,i_2)\in M$, $(u_1,i_6)\notin M$. As a result, user u_6 is then matched to item i_2, and user u_1, previously matched to item i_2, is now matched to new item i_6, which leads to the maximum possible cardinality for this example (i.e., max aggregate diversity of 6 items), and the iterations for searching the augmenting paths stop.



[Step1] Iteration 1: u_6 is not matched.

[Step1] Iteration 2: Augmenting path (u_6,i_2,u_1,i_6) finds two new matches (u_1,i_6) (u_6,i_2), in place of existing match (u_1,i_2).

(a) Maximum Bipartite Matching (top-1 recommendation task)

[Step1] Maximum Bipartite Matching: u_4, u_5, u_6 are matched to fewer than 3 items.

[Step2] Fill the remaining top-3 recommendations with the most highly predicted items, denoted by the dotted line.

(b) Max Bipartite Matching (top-3 recommendation task)

Figure 5.5 Max-Flow/Matching Based Optimization Approach

On the other hand, in the case of the top-3 recommendations for the same example (Figure 5.2b), while the maximum diversity (i.e., 10) is reached in Step 1, three users (u_4, u_5, u_6) are matched to fewer than 3 items. Thus, as shown in Step 2 of Figure 5.5b, the remaining top-3 recommendations are filled with the most highly predicted items among the items available for users.

Note that the sequence in which users and/or items are chosen to be evaluated in Figure 5.4 may have implications on the runtime of the algorithm. For example, finding more augmenting paths and, therefore, more matches in the first iteration may reduce the total number of iterations needed to reach maximum matching. We found that applying a simple heuristic of choosing users for matching based on the number of remaining candidate items that the users have, from smallest to highest, leads to substantial runtime improvements, because of the smaller likelihood that the items matched to those users can be replaced by other items, thus, reducing the number of iterations.

5.3.3 Integer Programming Approach for Diversity Maximization

The problem of improving both diversity and accuracy can be intuitively conceptualized as a multi-criteria optimization problem, where the system should provide recommendations that are as diverse and as accurate as possible. Due to the inherent tradeoff between these two metrics, we can use a common approach for solving multi-criteria optimization problems that optimizes one of the criteria and converts the other to a constraint. We model our problem of generating accurate and diverse recommendations as an integer programming problem that can maximize a linear objective function (maximum accuracy or maximum diversity), subject to a linear constraint (the desired user-specified level of diversity or accuracy).

An integer programming problem is specified using two sets of binary variables and three constraints, as illustrated in Figure 5.6. The integer program determines which N items are recommended to user u, denoted as $L_N(\mathrm{u})$, among several candidate items $L(u)$ that are predicted above the relevance threshold (T_H). This decision is represented as binary decision variable $\delta(u,i)$, which indicates whether candidate item i is recommended to user u. To compute the *diversity-in-top-N* measure, another binary variable $S(i)$ is introduced, which indicates whether item i is recommended to any user. Note that *diversity-in-top-N* = $\sum_{i \in I} S(i)$.

To maximize accuracy at the user-specified diversity level D, the space of possible top-N recommendation configurations for all users is searched until the optimal configuration is found. The optimal recommendation configuration maximizes the sum of predicted rating values of items (i.e., maximizing *prediction-in-top-N*), subject to three sets of constraints: (1) each user u can be recommended N items from among the candidate items $L(u)$; (2) for each item i, the binary variable $S(i)$ is assigned the value 0 if the item is not recommended to any user, otherwise 1; and (3) *diversity-in-top-N* should achieve the minimum level D. In an analogous fashion, we can also maximize recommendation diversity at a given level of accuracy A (using an average predicted rating value of all recommendations as a user-specified accuracy constraint).

$L(u) = \{ i \in I \mid R^*(u, i) \ge T_{\mathrm{H}} \}$	// set of items available for recommendation for each user
$\delta(u,i)=\begin{cases}1 & \text{if item } i \text{ is recommended to user } u\\ 0 & \text{otherwise}\end{cases}$	// binary variable for recommendation decision
$S(i)=\begin{cases}1 & \text{if item } i \text{ is recommended to at least one user}\\ 0 & \text{otherwise}\end{cases}$	// binary variable to compute item diversity
// maximize *prediction-in-top-N* (using sum of // predicted rating values of recommendations)	// maximize *diversity-in-top-N*
Maximize $\sum_{u\in U}\sum_{i\in L(u)}\left[\delta(u,i)\times R^*(u,i)\right]$,	***Maximize*** $\sum_{i\in I} S(i),$
// each user can get maximum *N* recommendations	// each user can get maximum *N* recommendations
subject to $\forall u\in U, \sum_{i\in L(u)}\delta(u,i) = min\{\lvert L(u)\rvert,N\}$	***subject to*** $\forall u\in U, \sum_{i\in L(u)}\delta(u,i) = min\{\lvert L(u)\rvert,N\}$
// if item *i* is not recommended to any user, S(*i*) =0	// if item *i* is not recommended to any user, *S*(*i*) =0
subject to $\forall i\in I, \sum_{u\in U}\delta(u,i) \ge S(i)$	***subject to*** $\forall i\in I, \sum_{u\in U}\delta(u,i) \ge S(i)$
// minimum level of diversity *D*	// minimum level of accuracy *A* (using average // predicted rating value of recommended items)
subject to $\sum_{i\in I} S(i) \ge D$	***subject to*** $\sum_{u\in U}\sum_{i\in L(u)}\left[\delta(u,i)\times(R^*(u,i)-A)\right] \ge 0$

(a) Maximize accuracy with a constraint of diversity level D

(b) Maximize diversity with a constraint of accuracy level A (average prediction)

Figure 5.6 Specification of the Top-*N* Recommendation Task as an Integer Program

To solve the proposed integer programs for diversity maximization, we use the CPLEX solver with AMPL, which is a modeling language for linear and nonlinear optimization problems. In particular, in Figure 5.7, we show optimal solutions obtained with CPLEX for the simple examples used earlier. As would be expected, the integer program with the constraint of the maximum possible diversity level, which was obtained using the max-flow/matching approach, achieved the best accuracy, meaning t the highest average prediction among the proposed optimization approaches.

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}
U_1		4.7				3.7	4.2	3.8	**4.6**	
U_2	4.7			4.1	**4.3**					
U_3	4.9	4.6		**4.2**		3.5	4.1			3.6
U_4	**4.3**	3.5	4							
U_5	4.6		**4.8**		3.5					
U_6		**4.3**	4.4	3.6	3.9					

(a) Top-1 recommendation task (max diversity: 6, average prediction: 4.417)

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}
U_1		4.7				**3.7**	4.2	**3.8**	**4.6**	
U_2	**4.7**			**4.1**	**4.3**					
U_3	**4.9**	4.6		4.2		3.5	**4.1**			**3.6**
U_4	**4.3**	**3.5**	**4**							
U_5	**4.6**		**4.8**		**3.5**					
U_6		**4.3**	**4.4**	3.6	**3.9**					

(b) Top-3 recommendation task (max diversity: 10, average prediction: 4.167)

Figure 5.7 Optimal Results from Integer Programming Approach

5.3.4 Computational Complexity of Optimization Approaches

We proposed three sophisticated approaches for diversity maximization, the increasing performance of which, however, comes at the cost of higher computational complexity. In this subsection, we briefly summarize the approaches in terms of their performance in the small recommendation example used throughout the chapter. We then discuss their flexibility to control the balance between diversity and accuracy and finally their computational complexity.

Table 5.1 shows the accuracy and diversity achieved by the standard ranking approach, the baseline recommendation re-ranking approach from prior literature, and the three proposed optimization approaches for the top-1 and top-3 recommendation tasks in the small recommendation example (6 users, 10 items) first described in Figure 5.2. By construction, the standard ranking approach provides the highest accuracy but is always lowest in terms of recommendation diversity. The recommendation re-ranking approach based on the reverse predicted rating value improves upon the standard ranking approach at the expense of accuracy loss. However, note that, as *N* increases, the accuracy losses diminish as compared to the standard ranking approach, but the diversity gains still remain. The proposed greedy approach improves upon the recommendation re-ranking approach in terms of accuracy and, at the same time, also provides some improvements in diversity as well. The max-flow/matching approach is always able to reach maximum possible diversity, but this additional diversity improvement, as expected, can come with some loss in accuracy, as compared to some of the less diverse techniques. Lastly, the integer program can produce the highest possible diversity as well as the highest possible accuracy at that diversity level (when given the maximum diversity level as a constraint).

Table 5.1 Accuracy and Diversity of Ranking and Optimization Approaches

Small Example	Top-1 recommendations		Top-3 recommendations	
	Accuracy (avg prediction)	Diversity	Accuracy (avg prediction)	Diversity
Standard Ranking	4.800	3	4.311	7
Re-Ranking (RevPred rating)	3.650	4	4.028	9
Greedy Maximization	4.450	5	4.189	9
Max-Flow/Matching	4.067	6 (max)	4.150	10 (max)
Integer Programming	4.417	6 (max)	4.167	10 (max)

As summarized in Table 5.2, in terms of their flexibility in handling the accuracy-diversity tradeoff, heuristic-based ranking and max-flow/matching approaches can use rating thresholds to ensure a certain level of accuracy, while losing some diversity; however, the exact desired accuracy and diversity levels cannot be specified in advance. The advantage of the max-flow approach is that, given any rating threshold, it will always find the maximum possible diversity level for that setting. Greedy and integer programming approaches have more control over diversity since they can specify the desired level of diversity in advance. However, the greedy approach is not able to provide any accuracy guarantees. In contrast, the integer program has direct control on both measures and, as a result, can produce maximum accuracy for the specified level of diversity or vice versa.

Table 5.2 Accuracy and Diversity Control of Ranking and Optimization Approaches

Re-Ranking Approach	Greedy Maximization	Max-Flow/Matching	Integer Program
Indirectly increases diversity (accuracy) at a loss of accuracy (diversity), by varying *ranking threshold* (T_R =3.5~4.9).	Explicitly specifies the *desired level of diversity* and keeps replacing recommendation items until the desired level is reached.	Indirectly manipulates accuracy (as well as maximum possible diversity), by varying *flow-rating threshold* (T_F=3.5~4.9).	Explicitly specifies the *desired level of accuracy* (*diversity*), and maximizes diversity (accuracy) at the given level.

We now discuss the worst-case asymptotic computational complexity of each algorithm. As a baseline approach, we used the reverse prediction-based ranking approach from prior literature (Adomavicius and Kwon 2011), which chooses the top-*N* items according to the reverse predictions of items for each user, i.e., a simple sorting algorithm. Assuming there are *m* users and *n* items, the worst case situation for this algorithm occurs when all of the *n* items are available to every user for recommendation. Then, the heuristic-based ranking does the job of sorting *n* items, $O(n\log n)$, for *m* users, and its complexity would be $O(mn\log n)$.

The greedy approach starts from the standard ranking approach with a simple sorting of items for each user according to their predicted rating value, $O(mn\log n)$. The iterative replacements between the already-recommended items and the never-recommended items (lines 13-30 of Figure 5.1) continue until the desired level of diversity is reached or no pairs of items are available for replacement. In the worst case, the standard ranking approach has diversity of *N* and replacement can occur maximum (n-N) times to reach the maximum possible diversity *n*, i.e., $O(n)$ replacements. In each iteration, we find the most frequently recommended item that can be replaced by one of the never-recommended items for a certain user in an inner loop (lines 15-19). Since we can easily find one alternative item to replace the most frequently recommended item in a realistic setting with a large number of items and users, we assume a fixed (i.e., constant) number of iterations of this inner loop. A single replacement (lines 21-28) takes $O(n+m)$, because each item's predicted rating value and recommendation frequency as well as each user's recommendation list and candidate item list are stored and maintained in hash tables, which can offer fast operations such as search, insertion, and deletion with a time complexity of $O(1)$ (Cormen et al. 2001). In summary, the worst-case time complexity of this greedy approach would be $O(mn\log n+n(n+m)) = O(mn\log n + n^2)$. The additional complexity for the greedy approach as compared to re-ranking approaches comes from the need to keep track of the "global" state of the top-*N* recommendations, i.e., which items have already been recommended, to whom, and how many times, as well as candidate users and items for replacement. In contrast, re-ranking approaches simply reorder candidate items based on "local" information, where the decision can be made for each user by performing a simple sort on a single numeric value.

The max-flow/matching optimization approach has two steps: (1) find the maximum bipartite matching for maximum diversity; and (2) apply the standard ranking approach for the remaining *N* items for each user. For Step 1, we adopted the Hopcroft-Karp algorithm (1973), which is known to be among the most efficient algorithms for maximum bipartite matching, having a complexity of $O(E\sqrt{V})$, where *E* is the number of edges in the graph and *V* is the number of vertices on the left side of the graph (i.e., the number of users in our case) (Burkard et al. 2009). In a bipartite graph with *m* user vertices, *n* item vertices, and a maximum of *mn* edges, the complexity of the Hopcroft-Karp algorithm would be $O(mn\sqrt{m})$, and by adding the standard ranking approach for Step 2, the total complexity of the max flow based approach for the top-1 recommendation tasks would be $O(mn\log n + mn\sqrt{m})$. For the top-*N* recommendation tasks, we allowed multiple edges from a single user vertex. We proposed an efficient extension of the bipartite matching algorithm for the top-*N* recommendations, as described in Section 5.3.2; however, in the worst case, the top-*N* recommendation task can be treated as the top-1 task with *Nn* users and, correspondingly, *Nmn* edges. Even this worst-case extension for the top-*N* recommendations does not change the complexity, so $O(Nmn\sqrt{Nm}) = O(mn\sqrt{m})$, assuming *N* (the number of recommendation provided to each user) is a relatively small, bounded constant. Therefore, the max-flow/matching approach is generally more complex than the greedy approach, unless the number of users is very significantly smaller than the number of items (when $m << n$); however, this may not be representative of typical real-world applications settings where recommender systems are used.[5]

The integer programming approach to diversity maximization is the most sophisticated of the approaches proposed in this chapter. The computational complexity of the general integer programming problem is known to be NP-hard (Cormen et al. 2001). We solved the integer programming problem using CPLEX, which adopts a branch-and-bound approach, one of the most widely used methods to solve large-scale NP-hard combinatorial optimization problems.

In addition to the theoretical complexity discussion, we also discuss the empirical runtimes of the proposed approaches as additional evidence of their computational complexity.

5.4 Empirical Results

The proposed optimization approaches are evaluated in terms of their diversity and accuracy (using *diversity-in-top-N* and *prediction-in-top-N* metrics) using real-world rating datasets.

5.4.1 Recommendation Performance of Optimization-Based Approaches

In our experimental evaluation, we used two movie rating datasets: MovieLens (a data file available at grouplens.org) and Netflix (a data file used for the Netflix Prize competition). Each dataset is pre-processed to include users and movies with significant rating histories, which makes it possible to have a large number of highly predicted items available for recommendations to each user, thus, potentially making the diversity maximization task more challenging. The basic statistical information of the resulting datasets can be found in Table 4.1. For each dataset, we learn from all of the known ratings and predict the unknown ratings (85.73% of the whole user-item matrix in the

[5] For example, the dataset released for the Netflix Prize competition had about 480,000 users and about 18,000 items (netflixprize.com).

MovieLens dataset and 84.68% in the Netflix dataset). As discussed in Section 2.2, we used three popular collaborative filtering techniques for rating estimation (user-based, item-based, and matrix factorization CF techniques), and the top-*N* (*N*=1, 5, 10) items are recommended for each user.

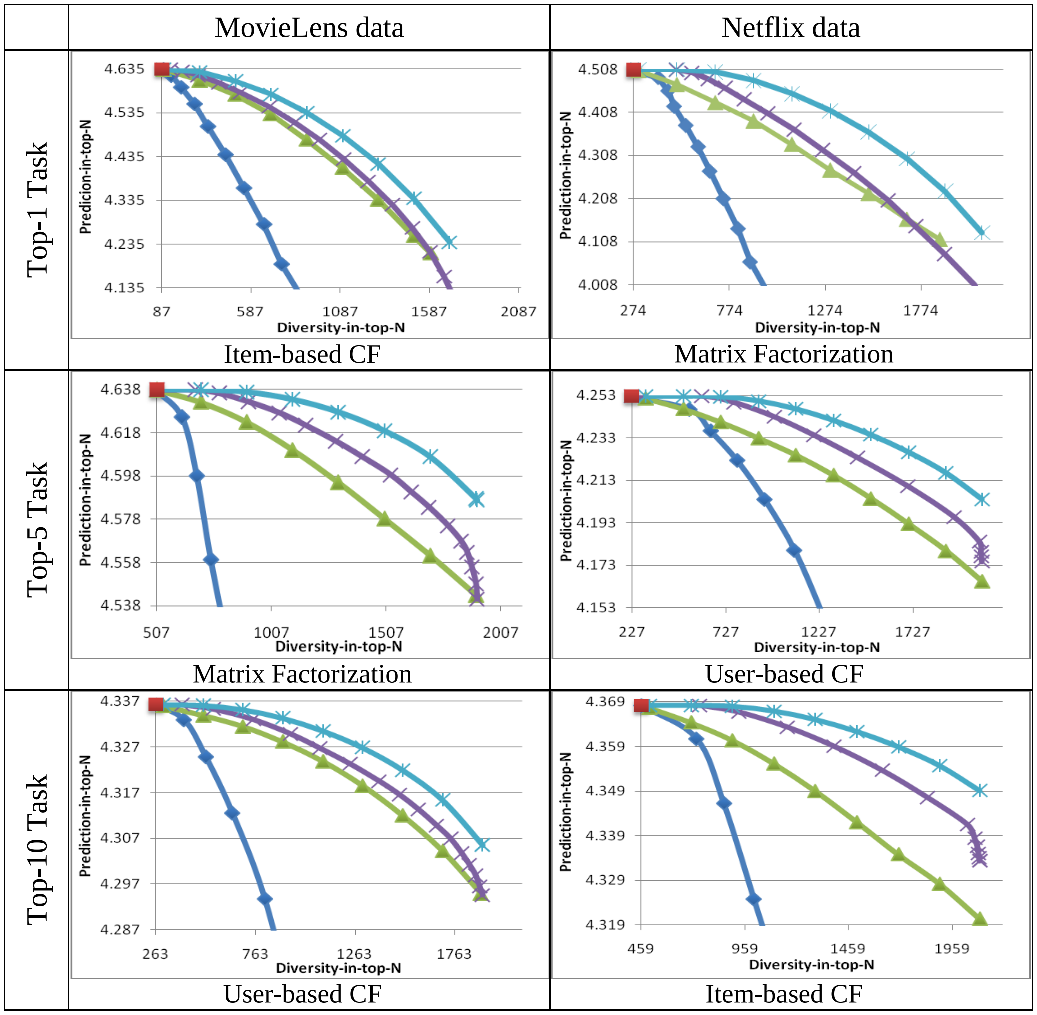


Figure 5.8 Accuracy and Diversity of Ranking and Optimization Approaches

We predicted unknown ratings based on all known ratings, where a relatively large number of highly predicted candidate items (with the predicted rating value above 3.5) were available for all users, typically around 500-800 items for each user. Figure 5.8 presents a number of representative results obtained from the empirical evaluation, which shows not only the accuracy and diversity capabilities of the three proposed approaches to the top-*N* recommendation, but also compares them with the simple recommendation re-ranking technique (Adomavicius and Kwon, 2011) used here as a baseline as well as the standard recommendation technique. As expected, the standard recommendation technique (i.e., recommending items with highest predicted ratings) represents the most accurate, but very non-diverse set of recommendations.

In Figure 5.8, the representative accuracy-diversity curves for the baseline re-ranking technique and for the max-flow/matching approach were obtained by using different ranking and flow-rating thresholds (3.5, 3.6, …, 5). Similarly, the accuracy-diversity curves for the greedy heuristic and for the integer program were obtained by measuring the accuracy at several specified levels of diversity (e.g., at 300, 500, …, up to the largest possible or maximum diversity).

One notable finding is that the proposed optimization approaches were able to obtain substantial diversity improvements at the given level of accuracy, compared to the prior recommendation re-ranking approach, across all experiments including different datasets, different recommendation techniques, and different numbers of recommendations (*N* = 1, 5, 10). Furthermore, the integer programming approach was consistently dominant , providing the highest diversity for a given level of accuracy, or highest accuracy for a given level of diversity, followed by the max-flow/matching approach and then the greedy maximization heuristic. As expected, the max-flow and integer programming approaches were able to reach the maximum diversity, and the greedy approach also performed well in that it could achieve close-to-maximum or maximum diversity in some cases of the top-5 and top-10 recommendations.

Another notable result is that, as the *N* increased, significant diversity improvements were obtained with increasingly smaller sacrifices to recommendation accuracy. For example, in the top-1 recommendation tasks, the max-flow based and integer programming approaches were able to obtain the maximum possible diversity with a decrease of about 0.5 (on scale 1-5) for an average prediction. However, for the top-5 tasks, the accuracy decrease needed for maximum diversity was about 0.1, and for the top-10 tasks it was only about 0.05.

Table 5.3 Diversity Gains of Ranking and Optimization Approaches at a Given Level of Accuracy

Accuracy level: Standard – 0.1	MovieLens data			Netflix data		
	User CF	Item CF	MF	User CF	Item CF	MF
Standard	98 accuracy:4.05	87 accuracy:4.63	247 accuracy:4.73	67 accuracy:4.31	142 accuracy:4.51	274 accuracy:4.51
Re-Ranking	409.1 (317.4%)	308.8 (254.9%)	412.3 (66.9%)	417.5 (523.1%)	420.2 (195.9%)	505.4 (84.5%)
Greedy	849.3 (766.7%)	686.5 (689.1%)	747.2 (202.5%)	759.7 (1033.8%)	714.5 (403.2%)	807.7 (194.8%)
Max Flow	826.0 (742.9%)	748.6 (760.4%)	927.5 (275.5%)	764.5 (1041.1%)	842.6 (493.4%)	967.8 (253.2%)
Integer Program	1051.4 (972.9%)	905.7 (941.0%)	1151.6 (366.2%)	1074.2 (1503.3%)	1110.4 (682.0%)	1315.9 (380.2%)

(a) Top-1 recommendation task

Accuracy level: Standard – 0.05	MovieLens data			Netflix data		
	User CF	Item CF	MF	User CF	Item CF	MF
Standard	190 accuracy:4.36	200 accuracy:4.57	507 accuracy:4.64	227 accuracy:4.25	335 accuracy:4.42	561 accuracy:4.43
Re-Ranking	648.1 (241.1%)	424.5 (112.3%)	698.1 (37.7%)	943.4 (315.6%)	754.4 (125.2%)	966.8 (72.3%)
Greedy	1478.2 (678.0%)	1214.5 (507.2%)	1385.6 (173.3%)	1533.2 (575.4%)	1387.2 (314.2%)	1540.7 (174.6%)
Max Flow	1562.9 (722.6%)	1415.9 (607.9%)	1647.7 (225.0%)	1829.1 (705.8%)	1795.8 (436.1%)	2000.9 (256.7%)
Integer Program	1728.0 (809.5%)	1611.1 (705.6%)	1895.5 (273.9%)	2092.0 (821.6%)	2092.0 (524.5%)	2091.0 (272.7%)

(b) Top-5 recommendation task

Accuracy level: Standard – 0.01	MovieLens data			Netflix data		
	User CF	Item CF	MF	User CF	Item CF	MF
Standard	263 accuracy:4.34	279 accuracy:4.53	667 accuracy:4.58	341 accuracy:4.21	459 accuracy:4.37	771 accuracy:4.39
Re-Ranking	710.4 (170.1%)	385.7 (38.3%)	794.3 (19.1%)	876.6 (157.1%)	750.0 (63.4%)	1193.2 (54.8%)
Greedy	989.4 (276.2%)	806.8 (189.2%)	1090.5 (63.5%)	984.5 (188.7%)	984.3 (114.4%)	1225.1 (58.9%)
Max Flow	1107.0 (320.9%)	978.7 (250.8%)	1408.2 (111.1%)	1528.9 (348.3%)	1426.3 (210.7%)	2456.1 (218.6%)
Integer Program	1324.3 (403.6%)	1428.1 (411.9%)	1673.2 (150.9%)	1794.7 (426.3%)	1734.9 (278.0%)	2845.3 (268.0%)

(c) Top-10 recommendation task

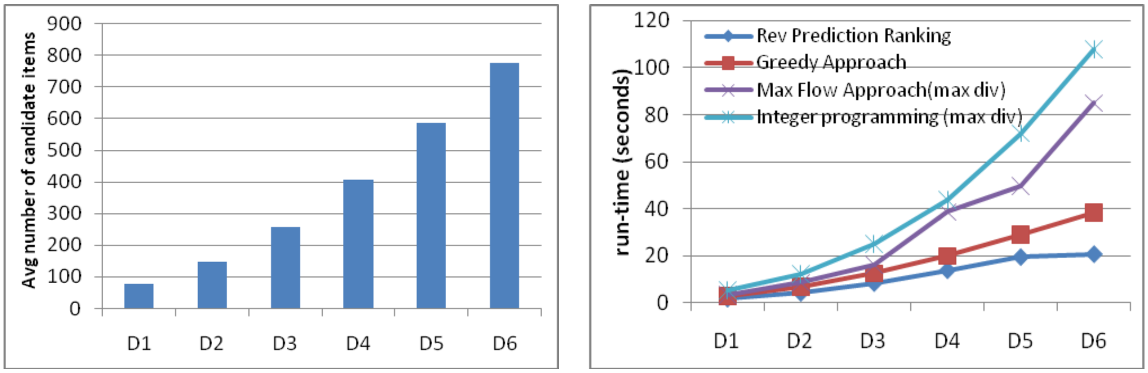
Note: For the standard ranking approach, each cell shows diversity and accuracy (*prediction-in-top-N*) measures. For the re-ranking and three optimization approaches, each cell shows the diversity performance at a given level of accuracy loss as compared to the corresponding accuracy of the standard ranking approach. Both absolute and relative (%) gains of *diversity-in-top-N*, as compared to the standard raking approach, are presented.

Table 5.3 further illustrates this point by showing the diversity gains of the recommendation re-ranking (i.e., baseline) and the three optimization approaches at three different accuracy loss levels (0.1 for the top-1 tasks, 0.05 for the top-5 tasks, 0.01 for the top-10 tasks). In summary, the three proposed optimization approaches were able to consistently provide substantial diversity improvements for all traditional recommendation algorithms (user-based, item-based, and matrix factorization CF) on different real-world recommendation datasets. Among the three approaches, the integer program achieved the best performance in terms of accuracy/diversity tradeoff, albeit at the expense of computational cost, as discussed next.

5.4.2 Scalability Analysis

The performance improvements for the proposed techniques come at a cost of computational complexity, which can become an issue as the data size increases. To complement the discussion on the theoretical computational complexity of the proposed approaches in Section 5.3.4, here we report on how the data size affects the actual runtime of each approach. We varied the size of data by changing the number of candidate items that were available for recommendations to each user. For example, for the datasets used in our experiments we treated all items that were predicted above the rating threshold T_H = 3.5 as potential candidates for recommendation. By increasing this threshold we could eliminate some candidate items across all users, thus, obtaining smaller datasets. Following this approach, we generated six datasets D_1, ..., D_6 of increasing size from the MovieLens dataset by using different rating thresholds (D_1 for T_H = 4.5, D_2 for 4.3, D_3 for 4.1, …, D_6 for 3.5), as indicated in Figure 5.9a.

We measured the runtime of each algorithm on the same computer, as each algorithm was trying to reach its largest improvement in diversity. The obtained results were consistent across different recommendation algorithms (user-based, item-based, and matrix factorization CF), different datasets (MovieLens and Netflix), and top-*N* tasks (for different *N* values). Figure 5.9b illustrates the general trends by presenting the runtimes of the simple recommendation re-ranking (i.e., baseline) and the three proposed optimization-based approaches on the MovieLens dataset, for generating diverse top-1 recommendations using the item-based CF technique. As expected, as the data size increased, the greedy heuristic approach demonstrated the best scalability, while more complex algorithms required increasingly more time, but also generated better recommendation outcomes, as discussed in Section 5.4.1. We do observe that, for our medium-size recommendation setting (with approximately 3,000 users and 2,000 items), all three proposed approaches demonstrated good computational performance; even running the most complex approach (integer program) on the largest dataset (D_6) took less than 2 minutes.



(a) Avg number of candidate items per user (b) Runtimes of ranking and proposed approaches

MovieLens dataset, top-1 items, item-based CF

Figure 5.9 Different Datasets and Algorithmic Runtimes

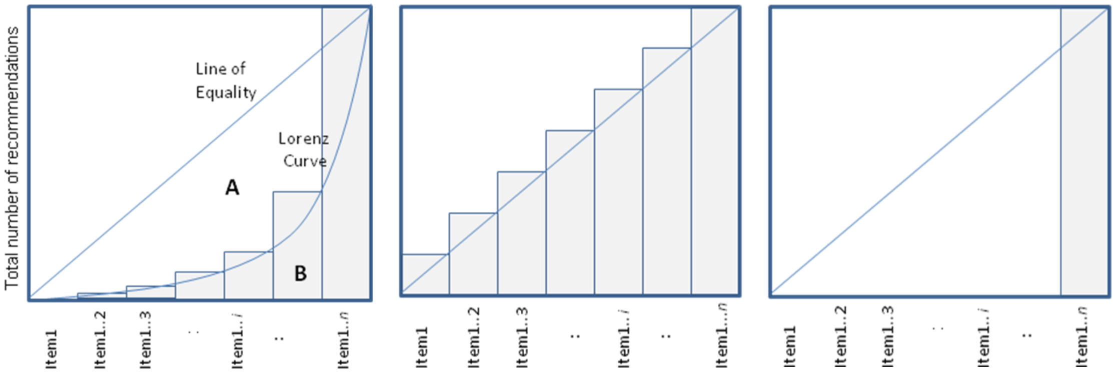
5.5 Analysis using Distributional Diversity Measures

The proposed optimization approaches were designed to improve recommendation diversity, specifically the *diversity-in-top-N* metric, which is sensitive to the additional recommendation of any new item (even for only one user). Therefore, it is not obvious whether these approaches would change the underlying distribution of recommended items, i.e., towards more evenly distributed representations. In other words, an increase in the size of a long-tail (absolute number of long-tail items) does not necessarily imply an increase in the relative share of long-tail items in total recommendations, e.g., as can be measured by entropy, the Gini coefficient, or the Simpson's diversity index.

Therefore, in this section, we evaluate the performance of the proposed approaches with respect to distributional diversity measures. In particular, we use the diversity metric based on the Gini coefficient (Gini 1921), which has recently garnered more attention as a relative long-tail metric (Brynjolfsson et al. 2007, 2010, Fleder and Hosanagar 2009, Oestreicher-Singer and Sundararajan 2011). The Gini coefficient can be used to measure the concentration of recommended items across all users, i.e., whether only a few popular items are recommended or whether all candidate items are equally recommended across all users. Here we briefly explain how the distributional diversity is computed based on the Gini coefficient, introduced as the *Gini-Diversity* metric in Section 4.5.3. The Gini coefficient can typically be calculated based on the Lorenz curve which represents a cumulative distribution function. In the recommendation settings, for example, the Lorenz curve can show what percentage of total recommendations the least popular *x*% of recommended items can have, and the Lorenz curve in the special case where all items are equally recommended across all users is known as "line of equality." The Gini coefficient can then be calculated as the ratio of the area between the line of equality and the Lorenz curve that represents a cumulative distribution of recommended items arranged in ascending order based on their popularity (area A), over the total area under the line of equality (area A+B), so G=A/(A+B) as shown in Figure 5.10a. Accordingly, the Gini coefficient ranges from 0 to 1; 0 for perfect equality (or high diversity) as shown in Figure 5.10b and 1 for perfect inequality (or low diversity) as shown in Figure 5.10c. Since we want to give a high value for diverse recommendations, with 0 for low diversity and 1 for high diversity, the original Gini coefficient is "inverted", i.e., transformed to B/(A+B). Thus, the *Gini-Diversity* metric can be formally written as:

$$\textit{Gini-Diversity} = 2\sum_{i=1}^{n}\left[\left(\frac{n+1-i}{n+1}\right)\times\left(\frac{rec(i)}{total}\right)\right],$$

where *rec*(*i*) is the number of users who were recommended item *i* as part of the top-*N*, *n* is the total number of candidate items that were available for recommendation, and *total* is the total number of top-*N* recommendations made across all users (*total* = *N*|*U*|). For example, as Figure 5.10b illustrates, when *n* different items are recommended to *n* users in the top-1 recommendation task, we can calculate the area of B as the sum of 1 to *n*, so *n*(*n*+1)/2, which is the same as the area of A, therefore setting the Gini coefficient as 1. On the other hand, as Figure 5.10c illustrates, when all the recommendations are concentrated on only one item, the area of B is calculated as *n*, whereas the area of A is the sum of 1 to *n*, thus setting the Gini coefficient as 2/(*n*+1), which approaches 0 as *n* increases.



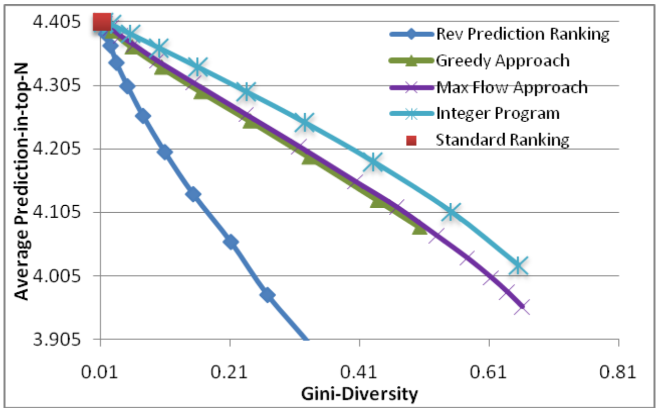
(a) Line of equality and Lorenz Curve for the distribution of recommended items

(b) Perfect equality: all items are equally recommended across all users.

(c) Perfect Inequality: one item is recommended across all users.

Figure 5.10 Distribution Equality of Recommended Items

As shown in Figure 5.11, all proposed optimization approaches as well as re-ranking approach provided significant diversity gains using the distributional diversity metric, similar to the results with the *diversity-in-to-N* metric in Figure 5.8. This result confirms that the proposed optimization approaches change the fundamental distribution of recommendations, not just the number of unique items recommended across all users, toward more diverse and long-tail recommendations. Furthermore, as was the case with the *diversity-in-top-N* metric, the integer programming approach achieves the best performance in terms of both accuracy and diversity with the *Gini-Diversity* metric as well.



MovieLens dataset, top-1 items, user-based CF

Figure 5.11 Accuracy and *Gini-Diversity* of Ranking and Optimization Approaches

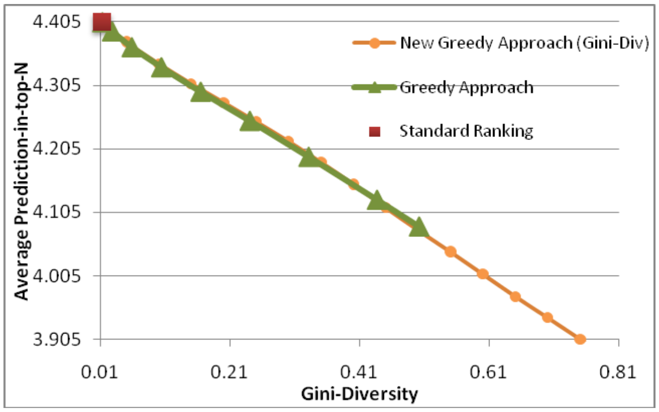
While the proposed optimization approaches solve the diversity maximization problem by directly taking into account the nature of the *diversity-in-top-N* metric, an interesting direction for future work could be to develop new approaches for directly maximizing more sophisticated distributional diversity measures. As one example, Figure 5.12 introduces a greedy maximization heuristic for improving the *Gini-Diversity*.

```
for each user $u \in U$ do       // initialize top-N recommendation lists using the standard ranking approach
    $L_N(u)$:= top $N$ items recommended to user $u$ according to $rank_{\text{Standard}}$.
    // set of new items that can replace old items
    $NL(u)$:= the remaining highly predicted items available to user $u$.
end for
for each item $i \in I$ do                       // recommendation frequency of each item
    $rec(i)$ := number of users for whom $i$ is recommended.
end for
gini-diversity:=get_gini_div(rec)              // compute Gini-Diversity using recommendation frequency
target_diversity := gini-diversity + $\theta$      // $\theta$: desirable diversity gain
done := false                                   // replacement not finished
// until target diversity is reached or no more replacements are possible
while (gini-diversity < target_diversity  and not done)
    // largest distance of two replaceable items in recommendation frequency (max Gini improvement)
    $d_{\max}$ := max{ $rec(i_1) - rec(i_2)$ | $u \in U, i_1 \in L_N(u), i_2 \in NL(u)$ }
    if ($d_{\max} \leq 1$) then done:= true            // no more replacement if the largest distance ≤ 1
    else        // if a triplet ($u_{\max}, i_{\text{old}}, i_{\text{new}}$) found - a user and two replaceable items with the largest distance
        // new item with the highest predicted rating selected when the distance is the same
        $(u_{\max}, i_{\text{old}}, i_{\text{new}})$:=$\text{argmax}_{u,i1,i2}$ { $R^*(u, i_2)$ | $(rec(i_1) - rec(i_2)) = d_{\max}, u \in U, i_1 \in L_N(u), i_2 \in NL(u)$ }
        Replace $i_{\text{old}}$ with $i_{\text{new}}$ in $L_N(u_{\max})$       // make the replacement
        Remove $i_{\text{new}}$ from $NL(u_{\max})$            // remove $i_{\text{new}}$ from set of new items for user $u_{\max}$
        $rec(i_{\text{old}}) := rec(i_{\text{old}}) - 1$
        $rec(i_{\text{new}}) := rec(i_{\text{new}}) + 1$
        gini-diversity:= get_gini_div (rec)
    end if
end while
```

Figure 5.12 Algorithm of New Greedy Approach for *Gini-Diversity* Measure

This greedy approach also starts from the standard ranking approach, and replaces items to increase the area below the Lorenz curve (area B in Figure 5.10a) until no item is available for replacement. The greatest increase in area B would result from replacement between the least and the most popular items. Adding one recommendation of the less popular item i_{new} and removing one recommendation of the more popular item i_{old} will move up the Lorenz curve by one unit across all items between the two items. Therefore, for replacement, this greedy algorithm finds a triplet (u_{max},$i_{old,}$ i_{new}), such that user u_{max} received the recommendation of item i_{old}, u_{max} has another item i_{new} for replacement, and the difference between items i_{old} and i_{new} in popularity is the largest among any pairs of items available for replacement. If there is more than one triplet that has the same largest difference in popularity, then we select the triplet that includes the replacement item with the highest predicted rating.

This approach enables further improvements in terms of the *Gini-Diversity* measure at the expense of accuracy, as compared to the original greedy approach, as shown in Figure 5.13. In particular, the new greedy approach could achieve the maximum possible *Gini-Diversity* of 0.75 with an accuracy loss of about 0.5, whereas the original greedy approach had the maximum *Gini-Diversity* of 0.5. In general, we conclude that the greedy approach for the simple *diversity-in-top-N*, described in Figure 5.1, perform as well as the new greedy approach which is designed specifically for the *Gini-Diversity* metric and is computationally more complex due to the need to compare a large number of possible combinations of item pairs for replacement and compute the *Gini-Diversity* metric after every replacement. As mentioned earlier, an interesting direction for future research would be to explore more sophisticated heuristics as well as more advanced optimization approaches for the direct maximization of distributional diversity measures.



MovieLens dataset, top-1 items, user-based CF

Figure 5.13 Accuracy and *Gini-Diversity* of New Greedy Approach

5.6 Conclusion and Future Work

Recommendation diversity has recently attracted considerable attention as an important aspect in evaluating the quality of recommendations. Traditional recommender systems typically recommend the top-*N* most highly predicted items for each user, thereby providing good predictive accuracy, but performing poorly with respect to recommendation diversity. Therefore, several heuristic-based recommendation re-ranking approaches have been proposed (Adomavicius and Kwon 2011), which can improve diversity by recommending items based on factors other than their predicted rating values. This chapter extends prior work by developing three more sophisticated optimization-based approaches that can achieve further improvements in diversity: (1) a heuristic approach of replacing popular items in recommendation lists with less popular items that have not yet been recommended to any user, (2) a graph-theoretic approach that models the diversity maximization problem as a network flow maximization or bipartite matching maximization problems, and (3) an integer programming problem that models the diversity as a linear objective function with constraints.

These three optimization approaches have several advantages over the recommendation re-ranking approaches from prior literature: (1) obtaining further improvements in diversity at the same level of accuracy; (2) achieving maximum possible recommendation diversity with a small amount of accuracy loss; and (3) providing direct control for balance in accuracy and diversity (the desired level of diversity or accuracy can be specified in advance, especially for the more sophisticated approaches, such as integer programming).

The proposed optimization approaches have been designed specifically for the *diversity-in-top-N* metric, which measures the number of distinct items among the top-*N* recommendations. The extension of the proposed optimization approaches to more sophisticated diversity metrics such as the long-tail shape parameter (e.g., the slope of the log-linear relationship between popularity and recommendations), represents a promising direction for future research and, in this chapter, we provide an example of a greedy heuristic approach to improve a relative long-tail diversity measure, the Gini coefficient, In addition, another interesting and important direction would be to investigate whether the use of the diversity-maximizing recommendation algorithms can truly lead to an increase in sales diversity and user satisfaction. In particular, as discussed in recent research (Brynjolfsson et al. 2010, Lee et al. 2011), it would be interesting to examine the impact of recommendations on long-tail phenomena in different categories of users and products and possibly propose different algorithms based on the appropriate categorization. We also believe that this work provides insights into developing new recommendation techniques that can consider multiple aspects of recommendation quality, going beyond using just the accuracy measures.

Chapter 6. Exploring Combined Approaches to Overcome the Accuracy-Diversity Tradeoff

6.1 Introduction

As discussed in Section 2.5, there is an inherent tradeoff between accuracy and diversity, because high accuracy may often be obtained by safely recommending the most popular items to users, which can lead to a reduction in diversity, with less personalized recommendations (Leonard 2010). Conversely, higher diversity can be achieved by uncovering and recommending highly personalized, idiosyncratic, and less popular items for each user, but these recommendations are inherently more difficult to predict due to lack of data, and may lead to a decrease in recommendation accuracy. Therefore, improving the performance of recommender systems along both dimensions represents a non-trivial task.

In this chapter we explore the possibilities to overcome the accuracy-diversity tradeoff, by proposing new approaches built upon two main ideas: incorporating multi-criteria ratings into traditional single-rating recommender systems for better accuracy (Adomavicius and Kwon 2007, Lakiotaki et al. 2008, Manouselis and Costopoulou 2007, Sahoo et al. 2011) and employing different ranking-based approaches for better diversity (Adomavicius and Kwon 2009, 2011). The single overall rating that has been used in traditional recommendation techniques may hide the underlying heterogeneity of users' preferences, but multi-criteria ratings (e.g., using separate ratings for story, action, direction, and visual effects components for each movie in a movie recommender system) can help to better understand each user's preferences, resulting in more accurate recommendations (Adomavicius and Kwon 2007). In addition, traditional recommendation techniques typically recommend the most highly predicted items to each user, often resulting in a less diverse set of mostly popular items (Fleder and Hosanagar 2009). Thus, ranking candidate items by factors other than the predicted rating value has been shown to increase recommendation diversity (Adomavicius and Kwon 2009, 2011).

In this chapter, we explore the possible combinations of these two types of approaches – incorporation of *multi-criteria rating* information and the use of different *ranking* methods. Through this exploration we may be able to improve the performance of several widely popular recommendation algorithms (which we use as baselines for comparison), such as neighborhood-based and matrix factorization techniques for collaborative filtering, as described in Section 2.2. We empirically demonstrate how these combinations can generate both more accurate and more diverse recommendations as compared to the baseline recommendation techniques.

The remainder of this chapter is organized as follows. Section 6.2 reviews two existing approaches for recommendation accuracy and diversity with empirical evidence. Section 6.3 describes the combined approaches and the main empirical results follow in Section 6.4. Additional experiments are conducted and discussed in Section 6.5. Section 6.6 concludes the chapter by summarizing the contributions and future directions.

6.2 Existing Approaches for Accuracy and Diversity

The main goal of new approaches proposed in this chapter is to improve traditional single- rating recommendation techniques both in terms of accuracy and diversity. Because of the inherent relationship between recommendation accuracy and diversity, it is often possible to increase recommendation accuracy at the expense of diversity and vice versa. However, improving the performance of recommender systems along both dimensions represents a non-trivial task. To address this issue, in this chapter we build upon two main ideas:

- Incorporating multi-criteria rating information into a traditional recommender system to help improve recommendation accuracy (Adomavicius and Kwon 2007);
- Applying more sophisticated ranking approaches, given the rating predictions by a traditional recommender system, to provide top-*N* item recommendations to each user to improve recommendation diversity (Adomavicius and Kwon 2009, 2011).

Note that these two ideas can be applied separately in different phases of the recommendation process and, thus, they represent excellent candidates for combination.

In particular, following the two-phase recommendation process explained in Section 2.1, multi-criteria rating information can be used to more accurately estimate unknown ratings in the rating prediction phase, whereas various ranking-based approaches can be applied in the recommendation generation phase. Therefore, in this chapter we explore the possibility of augmenting traditional recommendation techniques by combining these two ideas, resulting in recommendation techniques that provide improvements in both accuracy and diversity of recommendations.

All empirical tests in this chapter were performed using data from Yahoo! Movies (collected from movies.yahoo.com), where users provide a total of five ratings for each movie: an overall rating and four individual criteria of story, action, direction, and visual action. We pre-processed the data to include only users and movies with a certain minimal amount of rating history (i.e., users who rated at least 20 movies and movies rated by at least 20 users). The final dataset had 26,924 ratings, for 718 users and 491 movies, with data sparsity of 7.64%. In this dataset, each user rated 37.5 movies on average, and the average number of common movies between two users was 4.3. Moreover, each movie was rated, on average, by 54.8 users; the average number of common users between two movies was 6.2. We randomly split the entire ratings dataset into training data (60%) and test data (40%) for evaluation, and repeated this process 20 times to obtain 20 pairs of training and test datasets. To obtain more robust and reliable results, the performance of all techniques discussed in the chapter was calculated with an average of 20 results (corresponding to each pair of training/test data).

6.2.1 Multi Criteria Rating Information for Accurate Recommendations

While the majority of current recommender systems typically use a single numerical rating to represent a user's preference for a given item, recommender systems in some e-commerce settings have recently adopted *multi-criteria ratings* that capture more precise information about each user's preferences with respect to different aspects of an item. Examples of multi-criteria rating systems include the Yahoo! Movies website, which collects and displays each user's ratings for four criteria (e.g., story, action, direction, and visuals), in addition to the overall rating for a movie. Accordingly, there has been some work on developing new techniques that can incorporate multi-criteria rating information into the recommendation process (Adomavicius and Kwon 2007, Manouselis and Costopoulou 2007, Lakiotaki et al. 2008, Sahoo et al. 2011). Examples of new techniques for multi-criteria ratings include similarity-based, aggregation function, and probabilistic modeling approaches. In particular, the similarity-based approach incorporates multi-criteria ratings into the similarity computation in neighborhood-based CF recommendation techniques (Adomavicius and Kwon 2007, Manouselis and Costopoulou 2007). The aggregation function approach learns the relationship between the overall and multi-criteria ratings to predict the unknown overall ratings (Adomavicius and Kwon 2007). In addition, some multi-criteria recommendation approaches are being developed based on probabilistic modeling algorithms that are becoming increasingly popular in data mining and machine learning (Lakiotaki et al. 2008, Sahoo et al. 2011). In this chapter, we adopt a similarity-based approach and an aggregation function approach as representatives of multi-criteria rating techniques in the proposed combined approaches for improvement in accuracy and diversity.

Similarity-based approach. Specifically, we employ a similarity-based approach that uses a multidimensional distance metric based on the maximal value distance (or Chebyshev distance) (Adomavicius and Kwon 2007; refer to Section 3.3.1). We call this approach the *MaxDist* approach in the remainder of the chapter.

Aggregation function approach. While the overall rating is simply another criterion rating in similarity-based approaches, the aggregation function approach assumes that the overall rating serves as an *aggregation* of the multi-criteria ratings (Adomavicius and Kwon 2007; refer to Section 3.3.2). Given this assumption, this approach finds the aggregation function *f* that represents the relationship between the overall and multi-criteria ratings, e.g., a linear combination. In this chapter, as a representative of aggregation function approaches we use a total linear regression which estimates the regression coefficients based on the entire dataset, referred to as the *TotReg* approach in the remainder of the chapter.

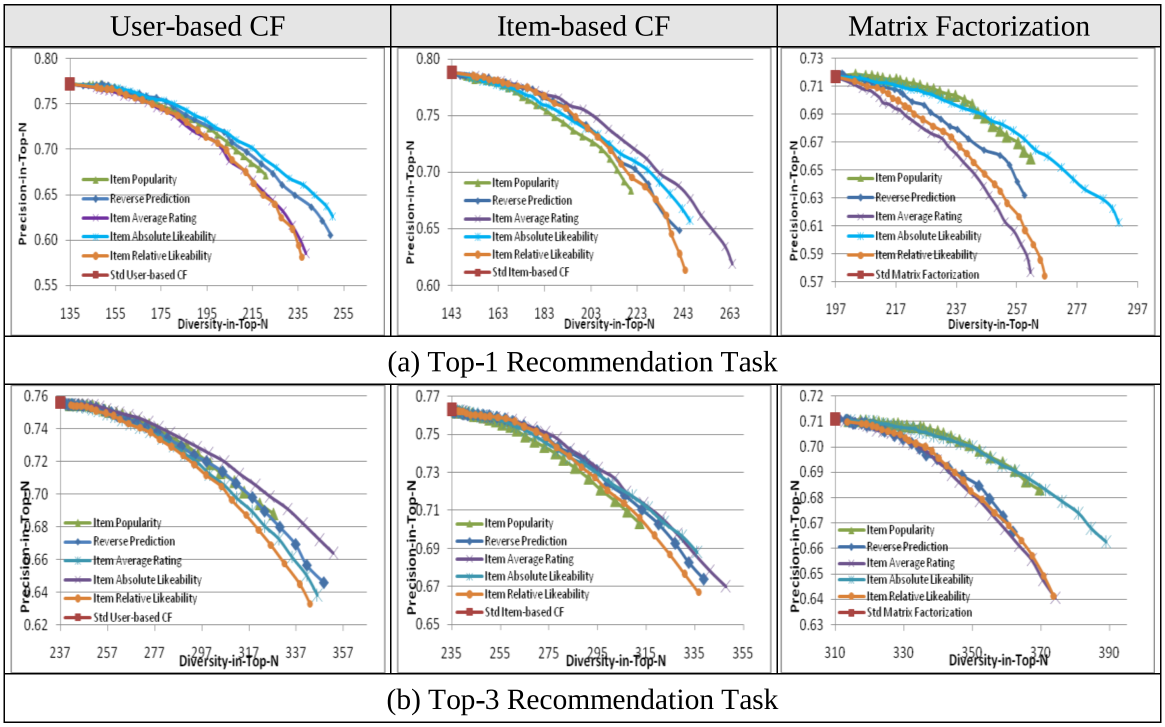
Table 6.1 Accuracy Improvements using Multi-Criteria Ratings

Algorithm	Accuracy (for fixed diversity)		Diversity (for fixed accuracy)	
	%	Gain	N	Loss (%)
Top-1 Recommendation Task				
Standard User-Based CF (single rating)	77.2	–	135	–
User CF Max Distance	81.3	4.1	107	-20.7
User CF Total Regression	78.2	1.0	126	-6.7
Standard Item-Based CF (single rating)	78.8	–	143	–
Item CF Max Distance	80.1	1.3	118	-17.5
Item CF Total Regression	79.6	0.8	134	-6.3
Standard Matrix Factorization (single rating)	71.7	–	197	–
MF CF Total Regression	78.1	6.4	154	-21.8
Top-3 Recommendation Task				
Standard User-Based CF (single rating)	75.6	–	237	–
User CF Max Distance	78.1	2.5	205	-13.4
User CF Total Regression	76.1	0.5	228	-3.8
Standard Item-Based CF (single rating)	76.3	–	235	–
Item CF Max Distance	77.3	1.0	207	-11.7
Item CF Total Regression	77.2	0.9	226	-3.8
Standard Matrix Factorization (single rating)	71.1	–	310	–
MF CF Total Regression	77.1	6.0	254	-18.0

6.2.2 Ranking-based Approaches for Diverse Recommendations

Standard recommendation ranking criterion $rank_{\text{Standard}}$, as discussed in Section 2.1, recommends the most highly predicted items for each user, typically resulting in high accuracy. However, it has been shown that in such cases, many users are likely to receive recommendations of the same popular items because some popular items tend to be predicted high across all users (Adomavicius and Kwon 2009, 2011). Therefore, more idiosyncratic or long-tail items (i.e., non-highly predicted items) need to be recommended to users to improve aggregate diversity of recommendations. As a result, a number of alternative ranking approaches to $rank_{\text{Standard}}$ have been proposed (Adomavicius and Kwon 2009, 2011). In particular, we use five recommendation ranking approaches, based on item popularity (*ItemPop*), reverse predicted rating value (*RevPred*), average rating (*AvgRating*), absolute likeability (*AbsLike*), and relative likeability (*RelLike*), as described in Section 4.3.

The items recommended by these approaches typically are not the most highly predicted ones, and are typically less popular, resulting in higher diversity, but they should still be relevant to the user (predicted as "high," i.e., above threshold T_H). It has been observed that recommending those items to users can increase the aggregate diversity to some degree (depending on the chosen rating threshold value), but at the same time the accuracy would decrease (Adomavicius and Kwon 2009, 2011). We confirm this finding using single overall ratings of our Yahoo! Movies data and three different traditional CF techniques. As shown in Figure 6.1, the performance of the five different ranking-based approaches exhibits a clear accuracy-diversity tradeoff, i.e., different ranking thresholds allow to achieve consistent and significant diversity gains at the expense of some accuracy loss. Thus, our proposed approaches aim to overcome this tradeoff to achieve performance improvements in *both* accuracy and diversity by integrating the ranking-based approaches with multi-criteria rating information, as discussed next.



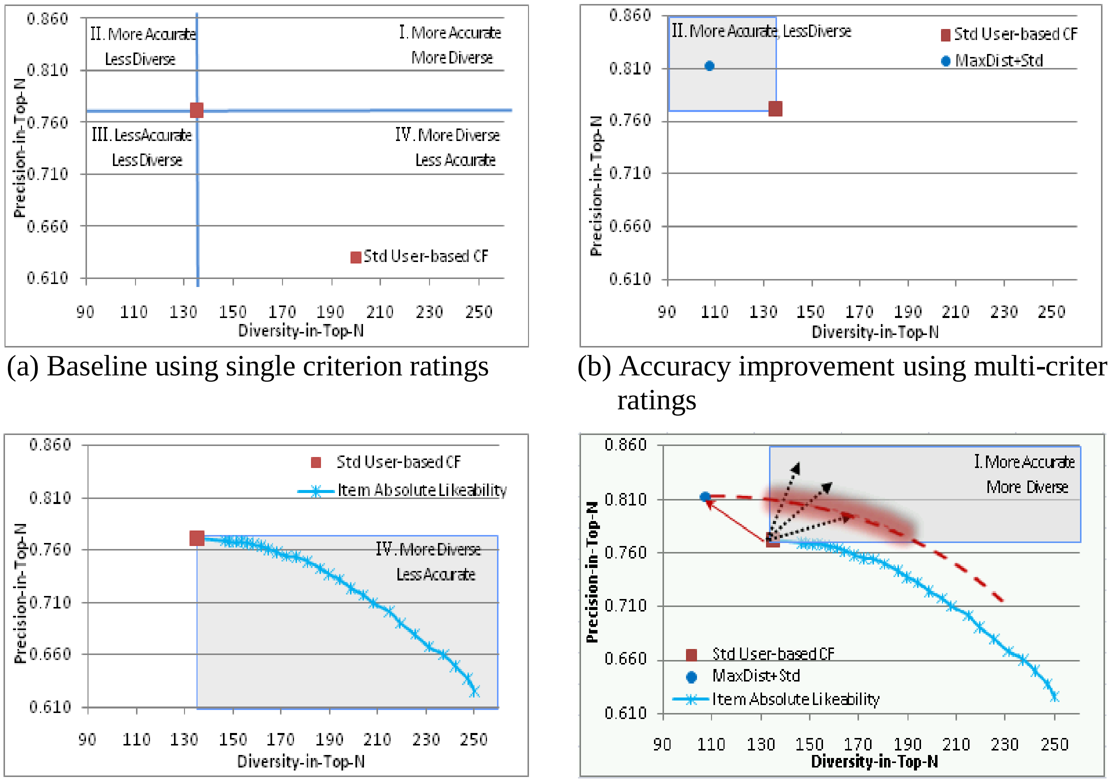
(a) Top-1 Recommendation Task

(b) Top-3 Recommendation Task

Figure 6.1 Diversity Improvements using Ranking-Based Approaches

6.3 Combined Approach

The ultimate goal of the combined approach is to generate the recommendations that are more accurate and more diverse than the ones obtained from the traditional recommender systems with single-criteria ratings. As described in Figure 6.2a, using the accuracy-diversity space, for any algorithm there are four quadrants of possible recommendation performance with respect to the standard/traditional recommendation technique as a baseline (using the actual performance of user-based CF baseline on our data as an illustration): (I) more accurate and less diverse, (II) more accurate and more diverse, (III) more diverse and less accurate, (IV) less accurate and less diverse.



(a) Baseline using single criterion ratings

(b) Accuracy improvement using multi-criteria ratings

(c) Diversity improvement using ranking-based approaches

(d) Accuracy and diversity improvement using combined approaches

Figure 6.2 Combination of Multi-Criteria Rating Information and Ranking Approaches

If the traditional recommendation techniques are augmented with multi-criteria rating approaches, as illustrated by the usage of *MaxDist* with user-based CF in Figure 6.2b, typically the accuracy of recommendations would improve at the expense of diversity (as shown earlier in Table 6.1), placing such techniques in quadrant II, as shown in Figure 6.2b. Also, if traditional recommendation techniques are augmented with ranking-based approaches, as illustrated by the usage of item absolute likeability-based ranking in Figure 6.2c, this typically results in diversity improvement at the expense of accuracy (as shown earlier in Figure 6.1), which places such techniques in quadrant IV, as shown in Figure 6.2c. Based on these observations, we conjecture that the combination of multi-criteria rating information and ranking-based approaches could simultaneously improve both accuracy and diversity with respect to the standard baseline approaches, placing such combined approaches in quadrant I, as visualized in Figure 6.2d. Since quadrant I is the most desirable performance quadrant, in the next section we focus on this quadrant when discussing the results of the proposed combined approaches.

We would also like to emphasize the flexibility of the proposed approaches. Specifically, depending on which metric (either accuracy or diversity) is more important, the user could choose different combinations of multi-criteria-rating approaches and ranking-based approaches. With the same combined approach, the user can also control the performance of recommender systems by changing the ranking threshold for the ranking approaches; for example, some users may want more diverse recommendations with the same level of accuracy as given by the standard approach, while others may want more accurate recommendations with the same level of diversity as in the standard approach. The availability of multiple techniques, that can be used in the combined approaches, and their parameterization using the ranking threshold provide the user with control and flexibility over the resulting recommendations.

6.4 Empirical Results of the Combined Approaches

We used three traditional recommendation techniques as baselines, including two neighborhood-based CF approaches (user-based and item-based) and the matrix factorization technique for rating prediction, all of which use the standard ranking approach (i.e., based on the predicted rating value) for generating the top-*N* recommendation lists. Using the combined approach idea, we have used two multi-criteria rating approaches, based on the maximum distance metric (*MaxDist*) and the total regression-based aggregation function (*TotReg*), and five recommendation ranking approaches, based on item popularity (*ItemPop*), reverse predicted rating value (*RevPred*), average rating (*AvgRating*), absolute likeability (*AbsLike*), and relative likeability (*RelLike*) to augment the baseline techniques for improved performance.

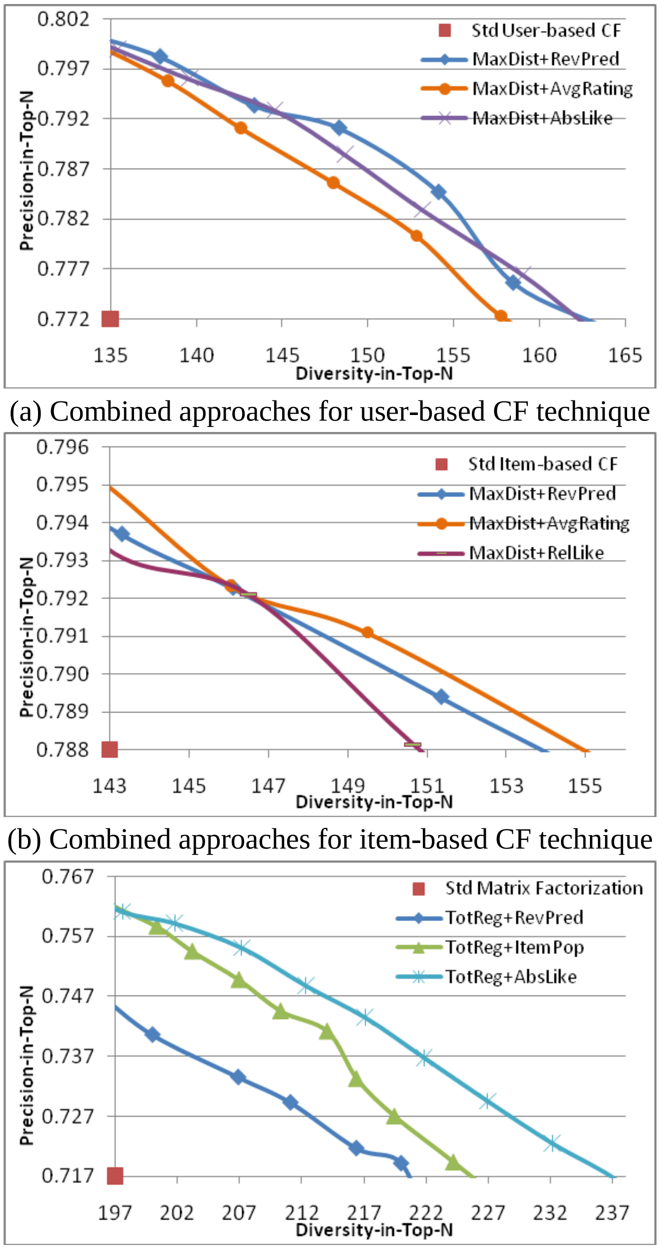
Table 6.2 Accuracy and Diversity Gains of the Combined Approaches

Algorithm	Accuracy (for fixed diversity)		Diversity (for fixed accuracy)	
	%	Gain	N	Gain (%)
Std User-Based CF	77.2	–	135	–
MaxDist + ItemPop	79.9	2.7	157	16.3
MaxDist + RevPred	80.0	2.8	163	20.7
MaxDist + AvgRating	79.9	2.7	158	17.0
MaxDist + AbsLike	79.9	2.7	162	20.0
MaxDist + RelLike	79.9	2.7	156	15.6
TotReg + Item Pop	78.1	0.9	156	15.6
TotReg + RevPred	78.0	0.8	154	14.1
TotReg + AvgRating	77.9	0.7	146	8.1
TotReg + AbsLike	78.0	0.8	159	17.8
TotReg + RelLike	77.9	0.7	147	8.9
(a) Combined approaches for user-based CF technique				
Std Item-Based CF	78.8	–	143	–
MaxDist + ItemPop	79.0	0.2	146	2.1
MaxDist + RevPred	79.4	0.6	154	7.7
MaxDist + AvgRating	79.5	0.7	155	8.4
MaxDist + AbsLike	79.2	0.4	148	3.5
MaxDist + RelLike	79.3	0.5	151	5.6
TotReg + Item Pop	79.2	0.4	150	4.9
TotReg + RevPred	79.3	0.5	164	14.7
TotReg + AvgRating	79.3	0.5	151	5.6
TotReg + AbsLike	79.2	0.4	150	4.9
TotReg + RelLike	79.4	0.6	151	5.6
(b) Combined approaches for item-based CF technique				
Std Matrix Factorization	71.7	–	197	–
TotReg + Item Pop	76.2	4.5	219	11.2
TotReg + RevPred	74.5	2.8	220	11.7
TotReg + AvgRating	73.9	2.2	207	5.1
TotReg + AbsLike	76.2	4.5	237	20.3
TotReg + RelLike	74.5	2.8	212	7.6
(c) Combined approaches for matrix factorization technique				

Also, note that only the *TotReg* multi-criteria rating approach was used with the matrix factorization baseline, because *MaxDist* was designed specifically for neighborhood-based algorithms. While there are more methods to choose from prior work, we present the results of these choices as representative ones. We measured the recommendation quality using the *precision-in-top-N* and the *diversity-in-top-N* metrics (as described in Section 2.2 and 2.3). The results in this section focus on the performance of top-1 recommendations, and we mention other top-*N* recommendations in the discussion section.

The empirical results are very encouraging: *all* proposed combinations outperform their corresponding baselines. For example, in Table 6.2a, with the same level of diversity as the standard user-based CF (135), the combined approach with *MaxDist* and *RevPred* obtained about a 2.8% improvement in accuracy over the baseline. Similarly, with the same level of accuracy as the standard user-based CF (77.2%), the same combined approach outperformed the baseline in terms of diversity by about 20.7%. Although the magnitude of the performance improvement depends on which standard technique is being augmented with the combined approach, patterns of consistent performance improvements can be found in Tables 6.2b and 6.2c as well.

So far we have shown improvements in each of the two performance dimensions (accuracy and diversity) while keeping the other dimension fixed for all the tested approaches. However, using different ranking threshold values, it is possible to adjust the performance of combined approaches to obtain recommendations that offer improvements in both accuracy and diversity, as shown in performance quadrant I. The graphs in Figure 6.3 depict quadrant I for each baseline respectively (shown in the lower left corner of each graph) with the performance of the three best combined approaches parameterized with different ranking thresholds. The three best combined approaches were chosen based on the size of the area under the curve above the diversity and accuracy levels obtained from the baseline (i.e., the area under the curve in quadrant I).



(a) Combined approaches for user-based CF technique

(b) Combined approaches for item-based CF technique

(c) Combined approaches for matrix factorization technique

Figure 6.3 Simultaneous Improvement of Accuracy and Diversity using Combined Approaches

6.5 Discussion

The idea of combining multi-criteria rating information with more sophisticated ranking approaches to augment traditional and widely used, single-rating recommendation techniques for accuracy and diversity improvement has demonstrated significant consistency and robustness. Performance improvements were obtained in a wide variety of settings: for different traditional baseline techniques, using a number of different multi-criteria and ranking methods. Since all of the experiments were done for the top-1 recommendation task, we also tested the robustness of the combined techniques with the top-3 task, even though due to the limitations of the dataset in this case all recommendation approaches were able to generate 80-87% of the possible recommendations, depending on the baseline approach used. Figure 6.4 provides an example of the representative combined approaches with the user-based CF technique, and shows that similar performance improvements –for both accuracy and diversity – are obtained in this case as well.

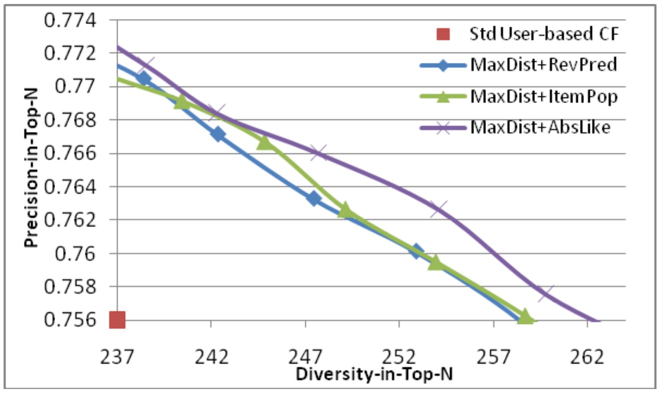


Figure 6.4 Performance Improvements in Top-3 Recommendation Task

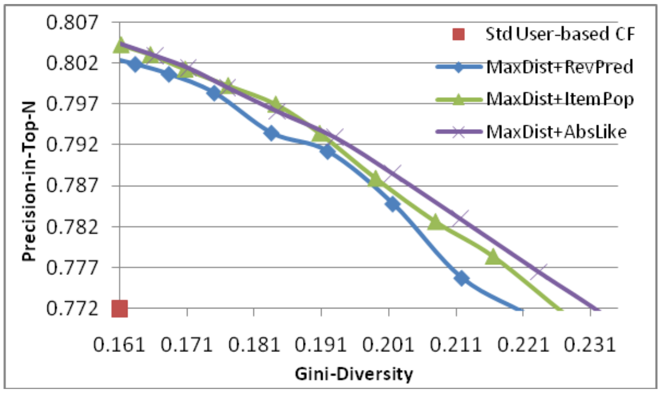
Since our diversity metric, *diversity-in-top-N*, is sensitive to the additional recommendation of a few new items (even for only one user), we also used a more sophisticated diversity metric as part of the additional robustness analysis to confirm that our proposed approaches truly contribute to more diverse and idiosyncratic recommendations across all users and not just manipulate the simple *diversity-in-top-N* metric to increase the number of different items among the recommendations. In particular, we adopt the Gini coefficient (Gini 1921), which is a commonly used metric of wealth distribution inequality, to measure the concentration of recommended items across all users, i.e., whether only a few popular items are recommended or all candidate items are equally recommended across all users. This distributional inequality measure also shows how diverse recommendations are, supporting our results with the *diversity-in-top-*

N metric. More details on the computations of this *Gini-Diversity* metric can be found in Section 5.5.

Table 6.3 and Figure 6.5 demonstrate that, as with the simpler *diversity-in-top-N* metric, the combined approaches with the user-based CF technique provide significant diversity gains using a more sophisticated, distributional *Gini-Diversity* metric. Similar performance patterns were observed with other standard baseline techniques as well. Overall, the results confirm that the proposed approaches were able to fundamentally change the distribution of recommended movies towards more evenly distributed representation, rather than increasing just the number of distinct items recommended, as well as improve recommendation accuracy.

Table 6.3 Accuracy and *Gini-Diversity* Gains of the Combined Approaches

Algorithm	Accuracy (for fixed diversity)		Diversity (for fixed accuracy)	
	%	Gain	Gini-Diversity	Gain
Std User-Based CF	77.2	–	0.161	–
MaxDist + ItemPop	80.4	3.2	0.226	0.065
MaxDist + RevPred	80.2	3.0	0.220	0.059
MaxDist + AvgRating	80.2	3.0	0.208	0.047
MaxDist + AbsLike	80.4	3.2	0.231	0.070
MaxDist + RelLike	80.3	3.1	0.207	0.046
TotReg + Item Pop	78.2	1.0	0.207	0.046
TotReg + RevPred	78.2	1.0	0.193	0.032
TotReg + AvgRating	78.0	0.8	0.181	0.020
TotReg + AbsLike	78.1	0.9	0.208	0.047
TotReg + RelLike	78.0	0.8	0.183	0.022



Yahoo!Movies, top-1 items, user-based CF technique

Figure 6.5 Performance of the Combined Approaches using the *Gini-Diversity* Metric

6.6 Conclusion and Future Work

While much research in recommender systems literature focuses on improving accuracy at the expense of diversity, in this chapter we propose new approaches for the simultaneous improvement in accuracy and diversity. While overlooked in current recommender systems, diverse recommendations can provide more personalized recommendations for users, potentially increasing customer loyalty and sales which would be a benefit for online providers as well. The proposed approaches demonstrate improvements over several widely used recommendation algorithms by augmenting them with the combination of: (1) multi-criteria rating information to improve recommendation accuracy, and (2) sophisticated recommendation ranking techniques to improve recommendation diversity. Our experimental results on a real-world dataset show that the proposed approaches outperform baseline techniques in many different configurations. In particular, the proposed approaches are general and flexible in that they can build upon a wide variety of existing recommendation techniques.

The major contribution of this chapter is as follows. This work enriches the body of knowledge on recommender systems by providing insights into addressing the tradeoff between accuracy and diversity and exploring new ways to overcome it. Furthermore, the proposed approaches offer many opportunities for further exploration. As one extension of this work, the combined approaches could be integrated with more sophisticated multi-criteria recommendation techniques such as Flexible Mixture Model (Sahoo et al. 2011) or with more advanced item selection techniques such as the optimization-based approaches proposed in Chapter 5, for possible further improvements in accuracy and diversity. In addition, the proposed approaches use multi-criteria rating information exclusively in the rating prediction phase; however, this additional information can also be potentially useful for item ranking in the recommendation generation phase. For example, some items might be rated very similarly along all criteria, while other items might be rated very differently in each criterion. Thus, a potential avenue for exploration could be to use an average rating variance of each item to recommend the items with high rating variance to recommend more controversially perceived items that are seen to be very strong on some criteria but not on others. The results might lead to increased diversity. Although multi-criteria rating systems may require a more significant level of user involvement because each user would need to rate an item on multiple criteria, the increasing availability of multi-criteria ratings opens up new opportunities for exploring various other ways to incorporate or leverage multi-criteria rating information.

We expect this work to stimulate more research on related topics which can lead to more sophisticated approaches for improving recommendation diversity and other novel aspects of recommendation quality.

Chapter 7. Conclusion

7.1 Summary of the Results

One of the important goals of recommender systems is to recommend to individual users what they would truly like (i.e., to have accurate recommendations), and many recommendation algorithms are designed to improve this recommendation accuracy. However, the goal of improving recommendation diversity, which can benefit both individual users and online content providers, has been largely ignored in recommender system literature. My dissertation is directed toward developing new techniques that can improve both accuracy and diversity by augmenting traditional recommendation techniques.

From the analysis of real-world rating datasets, useful data patterns have been found and applied to develop new recommendation approaches. In particular, the proposed approaches aim to enhance traditional recommendation algorithms by augmenting them with (1) multi-criteria rating information to improve recommendation accuracy, (2) heuristic-based ranking techniques to improve recommendation diversity, (3) more sophisticated optimization-based approaches that can achieve further improvements in diversity, and (4) a combination of the first two ideas to improve both accuracy and diversity.

Experimental results on a real-world rating dataset confirm that, when available, multi-criteria ratings can be successfully leveraged to improve recommendation accuracy. A comprehensive empirical evaluation of the proposed ranking and optimization approaches shows consistent and robust diversity improvements across multiple rating datasets and many different configurations. Lastly, the combined approaches for the simultaneous improvements in accuracy and diversity outperform baseline techniques across all experiments. All of the proposed approaches are general and flexible in that they can build upon a wide variety of existing recommendation techniques.

7.2 Contributions of Research

Three potential contributions of this dissertation are summarized as follows. First, the new recommendation approaches proposed in this dissertation will enrich *the body of knowledge* on recommender systems by providing insights into addressing multi-criteria recommendation problems and improving aggregate recommendation diversity. The multi-criteria rating techniques proposed in Chapter 3 can stimulate future research efforts towards enhancing current recommender systems with the multi-criteria ratings, which are becoming available in increasingly more online applications. Another contribution to recommender systems literature is the application of many different computational techniques, including a heuristic maximization approach, a graph-theoretic approach, and an integer programming approach, to improve the aggregate diversity of top-*N* recommendations (Chapters 4 and 5). Moreover, this dissertation is one of the first attempts in the recommender system literature to explore the possibilities to overcome the accuracy-diversity tradeoff (Chapter 6).

Second, *individual users* will benefit from the proposed approaches in that each user can find relevant and personalized items from more accurate and diverse recommendations provided by the recommender systems. Recommender systems should not only generate the items that users would like (accurate recommendations), but also provide a user with personalized or idiosyncratic items. More diverse recommendations would provide more opportunities to obtain such recommended items. As a result, diverse recommendations can also help users cultivate their taste for niche products that they may like but would never have considered without the help of a recommendation, from a large selection of long-tail items.

Third, the proposed approaches will help *online content providers* better understand their customers with more specific information on preferences—multi-criteria rating information—and improve their businesses by recommending more accurate and more personalized or "long-tail" items, enabling them to increase customer loyalty and, potentially, increase sales. In particular, more diverse recommendations could be beneficial for some business models, such as the one used by Netflix, because more sales of long-tail items (instead of new releases or extremely popular movies that are costly to license and acquire from distributors) can reduce their cost. Furthermore, taking into consideration that accurate recommendations are not always useful to users and, diverse but accurate recommendations may not provide significant value to users, it is important to control the balance between accuracy and diversity. The proposed ranking and optimization approaches offer flexibility to system designers, since they are parameterizable or the desired levels of diversity and accuracy can be specified in advance; in other words, they can have many different configurations, giving users control when setting the desired accuracy and diversity levels according to their current needs. In addition, these approaches do not require designers to use some specific algorithm since they can be used in conjunction with a wide variety of existing recommendation techniques.

7.3 Future Research

While this dissertation focuses on developing computational techniques to improve the accuracy and diversity of recommendations, future research can be directed at investigating the impact of the proposed approaches on user behaviors and the business value of diverse recommendations. One limitation of this work is that, based on previous literature, it is assumed that more personalized or diverse recommendations are beneficial to both individual users and some businesses, but the social and economic impact of the proposed approaches has not yet been thoroughly investigated. Thus, future research can study whether the use of diversity-conscious recommendation algorithms can truly lead to increased user satisfaction and sales diversity. Furthermore, as widely discussed in the prior literature (Lichtenstein and Slovic 2006), user preferences evolve over time. Another direction for future research would be to explore how user preferences change can affect recommendation accuracy and diversity.

Several additional interesting directions for future research can be suggested as follows. Multi-criteria rating systems may require a more significant level of user involvement because each user would need to rate an item based on multiple criteria. Therefore, it is important to measure the costs and benefits of adopting multi-criteria ratings and find an optimal solution to meet the needs of both users and system designers. In addition, since the multi-criteria recommendation problem can often be modeled as a decision problem with multiple criteria (i.e., as a multi-criteria decision making problem), many existing techniques in decision science literature (Figueria et al., 2005) can be applied to the recommender systems.

The proposed ranking approaches can be extended by exploring additional important item ranking criteria for potential diversity improvements. This may include consumer-oriented or manufacturer-oriented ranking mechanisms (Ghose and Ipeirotis 2007), depending on the given application domain, as well as external factors, such as social networks (Lemire et al. 2008).

While the proposed ranking and optimization approaches have been designed specifically for the *diversity-in-top-N* metric (i.e., the number of distinct items among top-*N* recommendations), the development of new recommendation approaches for other performance measures represents a promising direction for future research. Moreover, because of the inherent tradeoff between the accuracy and diversity metrics, another interesting research direction would be to develop a new measure that captures both of these aspects in a single metric. Lastly, using multi-criteria rating information in the recommendation generation phase as well as in the rating prediction phase would also be interesting, since the proposed combined approaches use this additional information exclusively in the rating prediction phase.

Bibliography

Adomavicius, G., A. Tuzhilin. 2005. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering* 17:6 734-749.

Adomavicius, G., Y. Kwon. 2007. New Recommendation Techniques for Multi-Criteria Rating Systems. *IEEE Intelligent Systems* 22:3 48-55.

Adomavicius, G., Y. Kwon. 2009. Toward More Diverse Recommendations: Item Re-Ranking Methods for Recommender Systems. *Proc. of the 19th Workshop on Information Technologies and Systems*.

Adomavicius, G., Y. Kwon. 2011. Improving Aggregate Recommendation Diversity Using Ranking-Based Techniques. *IEEE Transactions on Knowledge and Data Engineering* Forthcoming.

Adomavicius, G., N. Manouselis, Y. Kwon. 2011. Multi-Criteria Recommender Systems. in P. B. Kantor, F. Ricci, L. Rokach, B. Shapira (Eds.). *Recommender Systems Handbook: A Complete Guide for Research Scientists and Practitioners* Chapter 24. Springer.

Aggarwal, C.C., J.L. Wolf, K.L. Wu, P.S. Yu. 1999. Horting Hatches An Egg: A New Graph-Theoretic Approach to Collaborative Filtering. *Proc. of the 5th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining* (*KDD'99*). 201-212.

Ahuja, R.K., T.L. Magnanti, J.B. Orlin. 1993. *Network Flows: Theory, Algorithms, and Applications*. Englewood Cliffs, NJ: Prentice-Hall.

Anderson, C. 2006. *The Long Tail*. New York: Hyperion.

Balabanovic, M., Y. Shoham. 1997. Fab: Content-Based, Collaborative Recommendation. *Communications of the ACM* 40:3 66-72.

Bichler, M. 2000. An Experimental Analysis of Multi-Attribute Auctions. *Decision Support Systems* 29:10 249-268.

Billsus, D., M. Pazzani. 1998. Learning Collaborative Information Filters. *Proc.Int'l Conf. Machine Learning*.

Bradley, K., B. Smyth. 2001. Improving Recommendation Diversity. *Proc. of the 12th Irish Conf. on Artificial Intelligence and Cognitive Science*.

Breese, S., D. Heckerman, C. Kadie. 1998. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. *Proc. of the 14th Conf. on Uncertainty in Artificial Intelligence*.

Brynjolfsson, E., M.D. Smith, Y.J. Hu. 2003. Consumer Surplus in the Digital Economy: Estimating the value of increased product variety at online booksellers. *Management Science* 49:11 1580-1596.

Brynjolfsson, E., Y.J. Hu, M.D. Smith. 2006. From Niches to Riches: Anatomy of the Long Tail. *MIT Sloan Management Review*.

Brynjolfsson, E., Y.J. Hu, D. Simester. 2007. Goodbye Pareto Principle, Hello Long Tail: The Effect of Search Costs on the Concentration of Product Sales. *NET Institute*.

Brynjolfsson, E., Y.J. Hu, M.D. Smith. 2009. A Long Tail? Estimating the Shape of Amazon's Sales Distriution Curve in 2008. MIT Sloan School of Management.

Brynjolfsson, E., Y.J. Hu, M.D. Smith. 2010. Long Tails vs. Superstars: The Effect of Information Technology on Product Variety and Sales Concentration Patterns. *Information Systems Research* 21:4 736-347.

Burkard, R., M. Dell'Amico, S. Martello. 2009. Assignment Problems. *Society for Industrial and Applied Mathematics* (*SIAM*).

Burke, R. 2002. Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction* 12:4 331-370.

Carbonell, J., J. Goldstein. 1998. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. *Proc. of the ACM Conf. on Research and Development in Information Retrieval* (*SIGIR*). 335-336.

Cormen, T.H., C.E. Leiserson, R.L. Rivest, C. Stein. 2001. *Introduction to Algorithms*. MIT Press.

Delgado, J., N. Ishii. 1999. Memory-Based Weighted-Majority Prediction for Recommender Systems. *Proc. ACM SIGIR'99 Workshop Recommender Systems: Algorithms and Evaluation*.

Figueria, J., S. Greco, M. Ehrgott. 2005. *Multiple Criteria Decision Analysis: State of the Art Surveys*. Springer.

Fleder, D., K. Hosanagar. 2009. Blockbuster Culture's Next Rise or Fall: The Impact of Recommender Systems on Sales Diversity. *Management Science* 55:5 697-712.

Flynn, L.J. 2006. Like This? You'll Hate That. (Not All Web Recommendations Are Welcome). NYTimes,www.nytimes.com/2006/01/23/technology/23recommend.html.

Funk, S. 2006. Netflix Update: Try This At Home. http://sifter.org/˜simon/journal/20061211.html.

Gabriel, K.R., S. Zamir. 1979. Lower Rank Approximation of Matrices by Least Squares with Any Choice of Weights. *Technometrics* 21 489–498.

Garfinkel, R., R. Gopal, A. Tripathi, F. Yin. 2006. Design of a Shopbot and Recommender System for Bundle Purchases. *Decision Support Systems* 42:3 1974-1986.

Ghose, A., Ipeirotis, P. 2007. Designing Novel Review Ranking Systems: Predicting Usefulness and Impact of Reviews. *Proc. of the 9th Int'l Conf. on Electronic Commerce* (*ICEC*).

Gini, C. 1921. Measurement of Inequality and Incomes. *Economic Journal* 31 124-126.

Goldstein, D.G., D.C. Goldstein. 2006. Profiting from the Long Tail. *Harvard Business Review*.

Golub, G.H., C. Reinsche. 1970. Singular Value Decomposition and Least Squares Solution. Numer. Math 14 403-420.

Green, P.E., Krieger, A.M., Wind, Y. 2001. Thirty Years of Conjoint Analysis: Reflections and Prospects. *Interface* 31:3 56-73.

Greene, K. 2006. The $1 million Netflix challenge. Technology Review. www.technologyreview.com/read_article.aspx?id=17587&ch=biztech.

Grossman, L. 2010. How Computers Know What We Want - Before We Do. *Time*, May.

Herfindahl, O.C. 1950. Concentration in the Steel Industry. Unpublished Ph.D. Dissertation. Columbia University, New York.

Herlocker, J.L., J.A. Konstan, L.G. Terveen, J. Riedl. 2004. Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems* 22:1 5-53.

Hill, W., L. Stead, M. Rosenstein, G. Furnas. 1995. Recommending and Evaluating Choices in a Virtual Community of Use. *Proc. of Conf. on Human Factors in Computing Systems*.

Hofmann, T. 2003. Collaborative Filtering via Gaussian Probabilistic Latent Semantic Analysis. *Proc. 26th Ann. Int'l ACM SIGIR Conf.*

Hopcroft, J.E., R.M. Karp. 1973. An $n^{5/2}$ Algorithm for Maximum Matchings in Bipartite Graphs. *SIAM Journal on Computing* 2:4 225-231.

Hu, R., P. Pu. 2011. Enhancing Recommendation Diversity with Organization Interfaces. *Proc. of the 16th Int'l Conf. on Intelligent User Interfaces* (*IUI '11*). 347-350.

Huang, Z. 2007. Selectively Acquiring Ratings for Product Recommendation. *International Conference for Electronic Commerce*.

Huang, Z., W. Chung, H. Chen. 2004. A Graph Model for E-Commerce Recommender Systems. *Journal of the American Society for Information Science and Technology* 55:3 259-274.

Huang, Z., D. Zeng, H. Chen. 2007. Analyzing Consumer-product Graphs: Empirical Findings and Applications in Recommender Systems. *Management Science* 53:7 1146-1164.

Kim, H.K., J.K. Kim, Y. Ryu. 2010. A Local Scoring Model for Recommendation. *Proc. of the 20th Workshop on Information Technologies and Systems* (*WITS'10*).

Klema, V., A. Laub. 1980. The Singular Value Decomposition: Its Computation and Some Applications. *IEEE Transactions on Automatic Control* .25:2.164-176.

Knight, W. 2005. Info-mania' dents IQ more than marijuana. NewScientist.com news. URL: http://www.newscientist.com/article.ns?id=dn7298.

Koren, Y. 2008. Tutorial on Recent Progress in Collaborative Filtering. *Proc. of the 2008 ACM Conf. on recommender systems*. 333-334.

Koren, Y., R. Bell, C. Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *IEEE Computer Society* 42 30-37.

Lakiotaki, K., S. Tsafarakis, N. Matsatsinis. 2008. UTA-Rec: a Recommender System Based on Multiple Criteria Analysis. *Proc. of the 2nd ACM Conf. on Recommender Systems*.

Lee, J., J.N. Lee, H. Shin. 2011. The Long Tail or the Short Tail: The Category-Specific Impact of eWOM on Sales Distributions. *Decision Support Systems* 51:3 466-479.

Lee, W., C. Liu, C. Lu. 2002. Intelligent Agent-Based Systems for Personalized Recommendations in Internet Commerce. *Expert Systems with Applications* 22:4 275-184.

Lemire, D., S. Downes, S. Paquet. 2008. Diversity in Open Social Networks. Published online.

Leonard, D. 2010. Tech Entrepreneur Peter Gabriel Knows What You Want. *Business Week*, April.

Levy, M., K. Bosteels. 2010. Music Recommendation and the Long Tail. *Workshop on Music Recommendation and Discovery*. *ACM Int'l Conf. on Recommender Systems*.

Lichtenstein, S., P. Slovic. 2006. The Construction of Preference. New York: Cambridge University Press.

Liu, J., M. Shang, D. Chen. 2009. Personal Recommendation Based on Weighted Bipartite Networks. *Proc. of the 6th Int'l Conf. on Fuzzy Systems and Knowledge Discovery*. 134-137.

Mahalanobis, P.C. 1936. On the Generalised Distance in Statistics. *Proc. of the National Institute of Sciences of India*. 2:1 49–55.

Manouselis, N., C. Costopoulou. 2007. Experimental Analysis of Design Choices in Multi-Attribute Utility Collaborative Filtering. *International Journal of Pattern Recognition and Artificial Intelligence*.

McNee, S.M., J. Riedl, J.A. Konstan. 2006. Being Accurate is Not Enough: How Accuracy Metrics have hurt Recommender Systems. *Conf. on Human Factors in Computing Systems* 1097-1101.

McSherry, D. 2002. Diversity-Conscious Retrieval. *Proc. of the 6th European Conf. on Advances in Case-Based Reasoning* 219-233.

Mitchell, T. 1997. *Machine Learning*. McGraw-Hill.

Nakamura A., N. Abe. 1998. Collaborative Filtering Using Weighted Majority Prediction Algorithms. *Proc. of the 15th Int'l Conf. Machine Learning*.

Oestreicher-Singer, G., A. Sundararajan. 2011. Recommendation Networks and the Long Tail of Electronic. *MIS Quarterly* Forthcoming.

Park, S.T., D.M. Pennock. 2007. Applying Collaborative Filtering Techniques to Movie Search for Better Ranking and Browsing. *Proc. of the 13th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*. 550-559.

Park, Y.J., A. Tuzhilin. 2008. The Long Tail of Recommender Systems and How to Leverage It. *Proc. of the 2nd ACM Conf. on Recommender Systems.* 11-18.

Resnick, P., N. Iakovou, M. Sushak, P. Bergstrom, J. Riedl. 1994. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. *Proc. of Computer Supported Cooperative Work Conf.*

Ricci, F., B. Arslan, N. Mirzadeh, A. Venturini. 2002. ITR: A Case-Based Travel Advisory System. *Advances in Case-Based Reasoning*. LNAI 2416 613–627. Springer-Verlag: Berlin Heidelberg.

Robertson, S.E. 1997. The Probability Ranking Principles in IR. *Readings in Information Retrieval*. 281-286.

Sahoo, N., R. Krishnan, G. Duncan, J. Callan. 2011. The Halo Effect in Multicomponent Ratings and Its Implications for Recommender Systems: The Case of Yahoo! Movies. *Information Systems Research.*

Sanderson, M., J. Tang, T. Arni, P. Clough. 2009. What Else Is There? Search Diversity Examined. *European Conf. on Information Retrieval*. 562-569.

Sarwar, B.M., G. Karypis, J.A. Konstan, J. Riedl. 2000a. Analysis of Recommender Algorithms for E-Commerce. *ACM E-Commerce 2000 Conf.*, 158-167.

Sarwar, B.M., G. Karypis, J.A. Konstan, J. Riedl. 2000b. Application of Dimensionality Reduction in Recommender Systems—A Case Study. *Proc. ACM WebKDD Workshop*.

Sarwar, B.M., G. Karypis, J.A. Konstan, J. Riedl. 2001. Item-Based Collaborative Filtering Recommendation Algorithms. *Proc. of the 10th Int'l World Wide Web Conf.*

Schafer, J.B. 2005. DynamicLens: A Dynamic User-Interface for a Meta-Recommendation systems. Beyond personalization 2005: A workshop on the next stage of recommender systems research at the *ACM Intelligent User Interfaces Conf.*

Schonfeld, E. 2007. Click here for the upsell. *CNNMoney.com*, money.cnn.com/ magazines/business2/business2_archive/2007/07/01/100117056/index.htm, Jul.

Shani, G., A. Gunawardana. 2011. Evaluating Recommendation Systems. in P. B. Kantor, F. Ricci, L. Rokach, B. Shapira (Eds.). *Recommender Systems Handbook: A Complete Guide for Research Scientists and Practitioners* Chapter 8. Springer.

Shani, G., D. Heckerman, R. Brafman. 2005. An MDP-based Recommender System. *Journal of Machine Learning Research* 6 1265-1295.

Shannon, C.E. 1948. A Mathematical Theory of Communication. *Bell System Technical Journal* 27 379–423 & 623–656.

Shardanand, U., P. Maes. 1995. Social Information Filtering: Algorithms for Automating 'Word of Mouth'. *Proc. of Conf. on Human Factors in Computing Systems*.

Simpson, E.H. 1949. Measurement of Diversity. *Nature* 163:688.

Smyth, B., K. Bradley. 2003. Personalized Information Ordering: A Case-Study in Online Recruitment. *Journal of Knowledge-Based Systems* 16:5-6.269-275.

Smyth, B., P. McClave. 2001. Similarity vs. Diversity. *Proc. of the 4th Int'l Conf. on Case-Based Reasoning*: *Case-Based Reasoning Research and Development*.

Si, L., R. Jin. 2003. Flexible Mixture Model for Collaborative Filtering. *Proc. of the 20th Int'l Conf. on Machine Learning*.

Srebro, N., T. Jaakkola. 2003. Weighted Low-Rank Approximations. *In T. Fawcett and N. Mishra, editors. ICML*. AAAI Press. 720–727.

Statnikov, R.B., J.B. Matusov. 1995. *Multicriteria Optimization and Engineering*. Chapman & Hall.

Su, X., T.M. Khoshgoftaar. 2006. Collaborative Filtering for Multi-class Data Using Belief Nets Algorithms. *Proc. of the 8th IEEE Int'l Conf. on Tools with Artificial Intelligence*. 497-504.

Takács, G., I. Pilászy, B. Németh, D. Tikk. 2009. Scalable Collaborative Filtering Approaches for Large Recommender Systems. *Journal of Machine Learning Research* 10 623-656.

Thompson, C. 2008. If You Liked This, You're Sure to Love That. *The New York Times*. http://www.nytimes.com/2008/11/23/ magazine/23Netflix-t.html.

Wu, M. 2007. Collaborative Filtering via Ensembles of Matrix Factorization. *In KDDCup 2007*. 43-47.

Zhai, C., W.W. Cohen, J. Lafferty. 2003. Beyond Independent Relevance: Methods and Evaluation Metrics for Subtopic Retrieval. *Proc. of the ACM Conf. on Research and Development in Information Retrieval* (*SIGIR*).

Zhang, M. 2009. Enhancing Diversity in Top-*N* Recommendation. *Proc. of the 3rd ACM Conf. on Recommender Systems* 397-400.

Zhang, M., N. Hurley. 2008. Avoiding Monotony: Improving the Diversity of Recommendation Lists. *Proc. of the 2nd ACM Conf. on Recommender Systems*. 123-130.

Zhang, S., W. Wang, J. Ford, F. Makedon, J. Pearlman. 2005. Using Singular Value Decomposition Approximation for Collaborative Filtering. *Proc. of the 7th IEEE International Conf. on E-Commerce Technology* (*CEC'05*). 257-264.

Zheng, Z., B. Padmanabhan. 2006, Selectively Acquiring Customer Information: A New Data Acquisition Problem and an Active Learning-Based Solution. *Management Science* 50:5 697-712.

Ziegler, C.N., S.M. McNee, J.A. Konstan, G. Lausen. 2005. Improving Recommendation Lists Through Topic Diversification. *Proc. of the 14th Int'l World Wide Web Conf.* 22-32.

Appendix

Appendix A. Additional Results in Chapter 4

Table A1. Diversity gains of proposed ranking approaches for different levels of precision loss

	Item Popularity		Reverse Prediction		Item Average Rating		Item Abs Likeability		Item Relative Likeability		Item Rating Variance		Neighbors' Rating Variance	
Precision Loss	Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain	
-0.1	+805	2.851	+921	3.117	+935	3.149	+910	3.092	+910	3.092	+388	1.892	+528	2.214
-0.05	+613	2.409	+709	2.630	+708	2.628	+694	2.595	+682	2.568	+306	1.703	+379	1.871
-0.025	+443	2.018	+467	2.074	+488	2.122	+501	2.152	+481	2.106	+216	1.497	+249	1.572
-0.01	+301	1.692	+322	1.740	+307	1.706	+324	1.745	+292	1.671	+131	1.301	+141	1.324
-0.005	+234	1.538	+226	1.520	+229	1.526	+244	1.561	+221	1.508	+88	1.202	+99	1.228
-0.001	+126	1.290	+94	1.216	+79	1.182	+137	1.315	+79	1.182	+42	1.097	+34	1.078
Standard: 0.885	435	1.000	435	1.000	435	1.000	435	1.000	435	1.000	435	1.000	435	1.000

(a) MovieLens dataset, top-5 items, heuristic-based technique (user-based CF, 50 neighbors)

	Item Popularity		Reverse Prediction		Item Average Rating		Item Abs Likeability		Item Relative Likeability		Item Rating Variance	
Precision Loss	Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain	
-0.1	+760	2.538	+773	2.565	+768	2.555	+872	2.765	+749	2.516	+292	1.591
-0.05	+592	2.198	+543	2.099	+589	2.192	+663	2.342	+565	2.144	+217	1.439
-0.025	+420	1.850	+355	1.719	+398	1.806	+461	1.933	+383	1.775	+155	1.314
-0.01	+253	1.512	+199	1.403	+260	1.526	+273	1.553	+246	1.498	+88	1.178
-0.005	+154	1.312	+124	1.251	+146	1.296	+171	1.346	+149	1.302	+66	1.134
-0.001	+68	1.138	+38	1.077	+70	1.142	+75	1.152	+66	1.134	+33	1.067
Standard: 0.915	494	1.000	494	1.000	494	1.000	494	1.000	494	1.000	494	1.000

(b) MovieLens dataset, top-5 items, model-based technique (matrix factorization CF, *K*=64)

	Item Popularity		Reverse Prediction		Item Average Rating		Item Abs Likeability		Item Relative Likeability		Item Rating Variance		Neighbors' Rating Variance	
Precision Loss	Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain	
-0.1	+372	1.515	+1155	2.598	+1079	2.492	+745	2.030	+1131	2.564	+192	1.266	+564	1.780
-0.05	+333	1.461	+957	2.324	+909	2.257	+636	1.880	+974	2.347	+189	1.262	+433	1.599
-0.025	+264	1.365	+712	1.985	+690	1.954	+455	1.629	+750	2.037	+175	1.242	+318	1.440
-0.01	+186	1.257	+512	1.708	+526	1.728	+323	1.447	+568	1.786	+130	1.180	+231	1.320
-0.005	+133	1.184	+346	1.479	+324	1.448	+243	1.336	+361	1.499	+101	1.140	+158	1.219
-0.001	+53	1.073	+114	1.158	+136	1.188	+135	1.187	+124	1.172	+33	1.046	+50	1.069
Standard: 0.850	723	1.000	723	1.000	723	1.000	723	1.000	723	1.000	723	1.000	723	1.000

(c) MovieLens dataset, top-5 items, model-based technique (matrix factorization CF, *K*=64)

	Item Popularity		Reverse Prediction		Item Average Rating		Item Abs Likeability		Item Relative Likeability		Item Rating Variance		Neighbors' Rating Variance	
Precision Loss	Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain	
-0.1	+483	1.716	+1179	2.747	+1272	2.884	+886	2.313	+1112	2.647	+309	1.458	+895	2.326
-0.05	+415	1.615	+971	2.439	+1103	2.634	+728	2.079	+950	2.407	+321	1.476	+639	1.947
-0.025	+358	1.530	+746	2.105	+920	2.363	+594	1.880	+756	2.120	+298	1.441	+434	1.643
-0.01	+249	1.369	+528	1.782	+650	1.963	+379	1.561	+564	1.836	+217	1.321	+199	1.295
-0.005	+175	1.259	+390	1.578	+508	1.753	+283	1.419	+436	1.646	+145	1.215	+125	1.185
-0.001	+83	1.123	+139	1.206	+258	1.382	+139	1.206	+242	1.359	+82	1.121	+18	1.027
Standard: 0.813	675	1.000	675	1.000	675	1.000	675	1.000	675	1.000	675	1.000	675	1.000

(d) Netflix dataset, top-5 item, heuristic-based technique (item-based CF, 15 neighbors)

	Item Popularity		Reverse Prediction		Item Average Rating		Item Abs Likeability		Item Relative Likeability		Item Rating Variance		Neighbors' Rating Variance	
Precision Loss	Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain	
-0.1	+226	1.843	+171	1.638	+180	1.672	+260	1.970	+138	1.515	+76	1.284	+173	1.646
-0.05	+189	1.705	+153	1.571	+177	1.660	+216	1.806	+131	1.489	+88	1.328	+116	1.433
-0.025	+148	1.552	+119	1.444	+155	1.578	+152	1.567	+113	1.422	+80	1.299	+97	1.362
-0.01	+82	1.306	+43	1.160	+71	1.265	+87	1.325	+42	1.157	+44	1.164	+55	1.205
-0.005	+47	1.175	+25	1.093	+48	1.178	+52	1.194	+27	1.100	+16	1.060	+43	1.160
-0.001	+33	1.123	+23	1.086	+29	1.110	+37	1.138	+14	1.053	0	1.000	0	1.000
Standard: 0.913	268	1.000	268	1.000	268	1.000	268	1.000	268	1.000	268	1.000	268	1.000

(e) Yahoo!Movies dataset, top-1 item, heuristic-based technique (item-based CF, 15 neighbors)

	Item Popularity		Reverse Prediction		Item Average Rating		Item Abs Likeability		Item Relative Likeability		Item Rating Variance	
Precision Loss	Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain		Diversity Gain	
-0.1	+254	2.578	+253	2.571	+169	2.050	+285	2.770	+131	1.814	+99	1.615
-0.05	+244	2.516	+218	2.354	+162	2.006	+255	2.584	+124	1.770	+86	1.534
-0.025	+199	2.236	+148	1.919	+102	1.634	+200	2.242	+75	1.466	+79	1.491
-0.01	+99	1.615	+40	1.248	+26	1.161	+111	1.689	+20	1.124	+11	1.068
-0.005	+39	1.242	+26	1.161	+21	1.130	+41	1.255	+14	1.087	+7	1.043
-0.001	+19	1.118	+14	1.087	+13	1.081	+21	1.130	+8	1.050	+2	1.012
Standard: 0.947	161	1.000	161	1.000	161	1.000	161	1.000	161	1.000	161	1.000

(f) Yahoo!Movies dataset, top-1 item, model-based technique (matrix factorization CF, K=64)

Note: Precision Loss = [*precision-in-top-N* of proposed ranking approach] – [*precision-in-top-N* of standard ranking approach]
Diversity Gain (column 1) = [*diversity-in-top-N* of proposed ranking approach] – [*diversity-in-top-N* of standard ranking approach]
Diversity Gain (column 2) = [*diversity-in-top-N* of proposed ranking approach] / [*diversity-in-top-N* of standard ranking approach]