

Ranking and Context-awareness in Recommender Systems

Yue Shi

Ranking and Context-awareness in Recommender Systems

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op donderdag 20 juni 2013 om 15:00 uur
door

Yue SHI

Master of Engineering in Physical Electronics,
Southeast University, China

geboren te Zhenjiang, Jiangsu, China.

Dit proefschrift is goedgekeurd door de promotoren:

Prof.dr. A. Hanjalic

Prof.dr.ir. R.L. Lagendijk

Copromotor: Dr. M.A. Larson

Samenstelling promotiecommissie:

Rector Magnificus,

Prof.dr. A. Hanjalic,

Prof.dr.ir. R.L. Lagendijk,

Dr. M.A. Larson,

Prof.dr.ir. H.J. Sips,

Prof.dr. F.M.T. Brazier,

Prof.dr. M. de Rijke,

Dr. A. Karatzoglou,

Prof.dr.ir. A.P. de Vries,

voorzitter

Technische Universiteit Delft, promotor

Technische Universiteit Delft, promotor

Technische Universiteit Delft, copromotor

Technische Universiteit Delft

Technische Universiteit Delft

University of Amsterdam, Amsterdam

Telefonica Research, Barcelona, Spain

Technische Universiteit Delft, reservelid



Portions of the research reported in this thesis were supported by the European Commission's FP7 PetaMedia project.

ISBN 978-94-6186-166-5

Copyright © 2013 by Yue Shi

All rights reserved. No part of the material protected by this copyright notice may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without written permission from the copyright owner.

Printed in the Netherlands.

Summary

In this thesis we report the results of our research on recommender systems, which addresses some of the critical scientific challenges that still remain open in this domain. Collaborative filtering (CF) is the most common technique of predicting the interests of a user by collecting preference information from many users. In order to determine which items from a collection may be favored by individual users, conventional CF approaches take the ratings previously assigned to items by a target user and use them together with ratings of users with similar preferences to predict the ratings of yet-unseen items. Then, items are recommended in a descending order according to their predicted ratings. While CF has been investigated and improved extensively over the past years, there is still room for substantial improvement. In this thesis we focus on improvement of two critical aspects of CF, namely ranking and context-awareness of the recommendations. In addition, we analyze new developments in the field of collaborative recommendation and elaborate on the challenges related to the evolution of recommender systems and their increasing impact in the future. Based on this analysis, we make recommendations for future research directions in this field.

Samenvatting

In dit proefschrift rapporteren wij de resultaten van ons onderzoek naar aanbevelingssystemen, dat een aantal van de openstaande, essentiële wetenschappelijke vraagstukken in dit onderzoeksdomein behandelt. Het collaboratief filteren (CF) is de meest gangbare techniek voor het voorspellen van interesses van een gebruiker op basis van verzamelde informatie over de voorkeuren van vele gebruikers. Om te bepalen welke items uit een collectie mogelijk worden geprefereerd door individuele gebruikers, gebruiken conventionele CF-methoden beoordelingen die eerder zijn toegekend aan items door een specifieke gebruiker. Door deze informatie te combineren met beoordelingen van andere gebruikers met vergelijkbare voorkeuren, kunnen beoordelingen voorspeld worden voor items die de specifieke gebruiker nog niet kent. Vervolgens worden de items aanbevolen in aflopende volgorde van voorspelde beoordelingsscore. Hoewel CF in de afgelopen jaren uitgebreid bestudeerd en verbeterd is, is er nog steeds ruimte voor substantiële verbeteringen. In dit proefschrift richten wij ons op het verbeteren van twee cruciale aspecten van CF, namelijk het rangschikken van aanbevelingen en het in acht nemen van de context waarin de aanbevelingen worden gedaan. Daarnaast analyseren wij nieuwe ontwikkelingen op het gebied van collaboratieve aanbevelingen en behandelen we uitvoerig de uitdagingen gerelateerd aan de evolutie van aanbevelingssystemen en hun toenemende impact in de toekomst. Op basis van deze analyse doen wij aanbevelingen voor toekomstige onderzoeksrichtingen in dit vakgebied.

Contents

Summary	v
Samenvatting	vii
1 Introduction	1
1.1 On Search and Recommendation	1
1.2 Collaborative Filtering	3
1.3 From Ratings to Rankings	4
1.4 Recommendation in a Context	6
1.5 Recommender Systems: New Developments	7
1.6 List of Publications	8
2 Unified Recommendation Model	11
2.1 Introduction	12
2.2 Related Work	14
2.3 Unified Recommendation Model	15
2.3.1 PMF: Matrix Factorization for Rating	15
2.3.2 ListRank: Matrix Factorization for Ranking	16
2.3.3 Combining PMF and ListRank	17
2.3.4 Learning Algorithm and Complexity Analysis	18
2.4 Experiments and Evaluation	19
2.4.1 Datasets	21

2.4.2	Experimental Setup and Evaluation Metrics	21
2.4.3	Impact of Tradeoff Parameter	22
2.4.4	Effectiveness and Efficiency	23
2.4.5	Performance Comparison	25
2.5	Conclusion and Future Work	27
3	Collaborative Less-is-More Filtering	29
3.1	Introduction	30
3.2	Related Work	31
3.2.1	Ranking-oriented CF	32
3.2.2	Learning to Rank	33
3.3	CLiMF	33
3.3.1	Smoothing the Reciprocal Rank	33
3.3.2	Lower Bound of Smooth Reciprocal Rank	35
3.3.3	Optimization	36
3.3.4	Discussion	37
3.4	Experimental Evaluation	39
3.4.1	Experimental Setup	39
3.4.2	Performance Comparison	41
3.4.3	Effectiveness	42
3.4.4	Scalability	43
3.5	Conclusions	44
4	Mood-specific Movie Recommendation	45
4.1	Introduction	47
4.2	Overview of the Moviepilot Challenge	48
4.2.1	Problem Statement	48
4.2.2	Characteristics of the Challenge	49
4.3	Related Work	50

4.3.1	Collaborative Filtering	50
4.3.2	Context-aware Recommendation	52
4.3.3	Tag-aware Recommendation	52
4.4	The Proposed Algorithm	53
4.4.1	Mood-specific Movie Similarity	54
4.4.2	Plot Keyword -based Movie Similarity	56
4.4.3	Joint Matrix Factorization	56
4.4.4	Complexity Analysis	58
4.5	Experimental Evaluation	60
4.5.1	Experimental Setup	60
4.5.2	Impact of Tradeoff Parameters	61
4.5.3	Effectiveness	63
4.5.4	Performance Comparison	63
4.6	Conclusion and Future Work	66
5	Non-trivial Landmark Recommendation	69
5.1	Introduction	71
5.2	Related Work	75
5.2.1	Non-trivial Recommendations	75
5.2.2	Exploiting Location Information for Recommendation	76
5.2.3	Collaborative Filtering	78
5.3	Non-trivial Landmark Recommendation	79
5.3.1	Overview	79
5.3.2	Weighted Matrix Factorization	80
5.3.3	Category-based Landmark Similarity	83
5.3.4	Weighted Matrix Factorization with Category-based Regularization	84
5.3.5	Discussion	85
5.4	Data Description	87

5.5	Experimental Framework and Results	88
5.5.1	Evaluation Framework	89
5.5.2	Impact of Parameters	91
5.5.3	Evaluation	94
5.6	Conclusion and Future Work	99
6	Optimizing MAP for Context-aware Recommendation	103
6.1	Introduction	104
6.2	Related work	106
6.3	Problem and Terminology	108
6.4	TFMAP	109
6.4.1	Smoothed Mean Average Precision	110
6.4.2	Optimization	110
6.4.3	Fast Learning	112
6.5	Experimental Evaluation	116
6.5.1	Experimental Setup	116
6.5.2	Validation: Impact of Fast Learning	118
6.5.3	Performance Comparison	121
6.5.4	Scalability	123
6.6	Conclusions and future work	123
6.A	Derivation of Eq. (10)	125
6.B	Proof of Lemma 1	125
7	Future Challenges	127
7.1	Challenges of New Conditions and Tasks	129
7.1.1	Social Recommendation	129
7.1.2	Group Recommendation	132
7.1.3	Long Tail Recommendation	136
7.1.4	Cross-domain Collaborative Filtering	138

7.2	Challenges of New Perspectives and Models	140
7.2.1	Search and Recommendation	141
7.2.2	Interaction and Recommendation	143
7.2.3	Economics and Recommendation	144
7.3	Conclusions	146
	Acknowledgements	161
	Curriculum Vitae	163

Chapter 1

Introduction

1.1 On Search and Recommendation

The amount of information available on the Internet has become immense and is still growing at an unbelievably fast rate. The emergence of social networks (e.g., Facebook¹ and Twitter²) and Internet-enabled mobile devices (e.g., smart phones and tablets) has further boosted the volume of online information resources, since these technologies enable online users to freely create, upload and share information contents, i.e., media items, such as texts, images, videos. On one hand, the abundance of online information may virtually guarantee that users are able to find what they are looking for. On the other hand, this same abundance also makes the useful information difficult to find, a problem referred to as “information overload” [47].

Two major Internet technologies, namely, information *search* and *recommendation*, have been developed to help online users handle the information overload problem. In the search case, illustrated in Fig. 1.1(a), users actively express their information needs by submitting queries to the search system (engine), and then the system tries to find the items (e.g., texts, images, videos, music) in the collection that best match the queries. In the recommendation case, the users’ information needs are expressed implicitly, which can be done in two ways, generally referred to as *content-based filtering* and *collaborative filtering*. In content-based filtering, features of previously selected items are extracted and used to identify similar unseen items to be offered to the user [120]. A typical example of a system based on this principle is Pandora³ for music recommendation, where around 400 attributes of a music piece identified in the

¹<https://www.facebook.com/>

²<https://twitter.com/>

³<http://www.pandora.com>

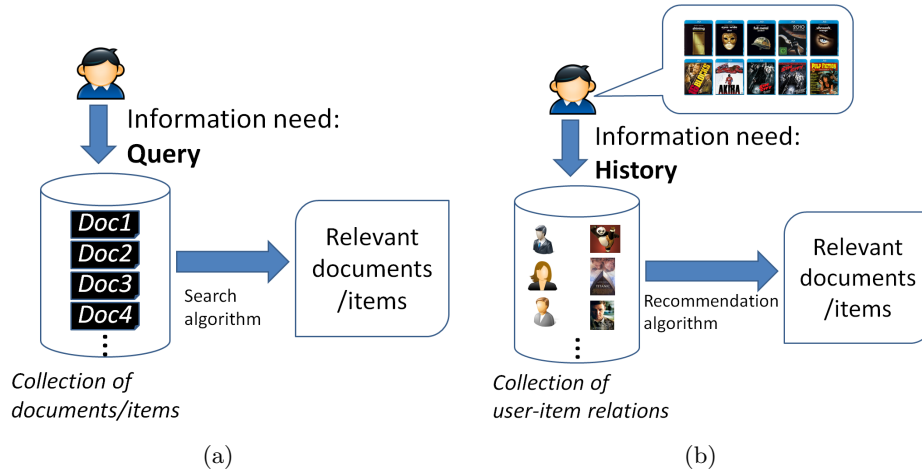


Figure 1.1: (a) Search system: The user's information need is indicated by a query, which is then matched to the collection by the search algorithm to find relevant items. (b) Recommender system based on the collaborative filtering principle: The user's information need is derived from the history of her interaction with the collection. This history is then matched with interactions patterns between the users and items of that collection to identify new items that comply with the history and to recommend them to the user.

Music Genome project⁴ are deployed for item matching. Collaborative filtering (CF) [47, 129], illustrated in Fig. 1.1(b), builds on the idea that users who share similar interests in the past may also prefer similar information items in future. Based on this idea, the information need of the user is inferred by the system from the history of user activities (e.g. download or rating of items, commenting on items) on that system. As an example, users first rate movies with a pre-defined scale after watching them, and then the movie recommender system predicts which unseen movies would be interesting for an individual user. Typical examples of CF-based recommender systems are Last.fm⁵ for music and Netflix⁶ for movies.

Compared to search systems, recommender systems provide the possibility for users to discover new items or item categories that they may not initially think of when formulating the search query. Research on recommender systems has intensified substantially over the past several years, since the function and quality of recommendation becomes more heavily in demand in a great variety of online services. In addition, a number of real-world data sets that are made available

⁴<http://www.pandora.com/about/mgp>

⁵<http://www.last.fm/>

⁶<https://signup.netflix.com/>

in the community, and a series of contests (e.g., Netflix Prize⁷, CAMRa⁸, Yahoo Music⁹) that emphasize various recommendation purposes, have further boosted the progress of research on recommender systems. In this thesis we report the results of our research on recommender systems, which addresses some of the critical open scientific challenges in this domain.

1.2 Collaborative Filtering

The two classes of recommender systems, i.e., based on the collaborative and content-based filtering principles, have their respective advantages and disadvantages. CF may suffer from the *cold start* problem, i.e., missing information on the user-item interaction history when setting up the system, based on which recommendation can be made. However, CF can provide information to individual users in a more personalized fashion, which is a direct consequence of using the user's individual activity history as input for recommendation. Compared to this, content-based filtering recommenders can become operational already based on a rather limited input (e.g., a previously seen item). However, they are also known to limit the scope of recommendation too much, namely to those items similar to the initial ones, through which the unique discovery effect mentioned above may be insufficient.

The cold start problem of CF-based recommender systems can be handled by, for instance, combining CF-based and content-based techniques into a *hybrid recommender system*. This possibility, in combination with the much higher discovery potential of CF, has made CF-based recommenders significantly more popular than the recommenders using the content-based filtering principle. It can be observed that CF has been deployed as functionalities of broader online services, e.g., product recommendation in Amazon¹⁰ [88] or video recommendation in Youtube¹¹ [34]. However, the quality of recommendations by most CF-based recommenders has been shown to be still far from satisfactory for online users [57, 71, 139, 33]. This factor has made the search for the ways to improve the effectiveness of CF-based recommendation more urgent, which motivated us to focus on CF-based recommender systems in this thesis.

Typically, the data processed by a CF-based recommender system can be illustrated as in Fig. 1.2. In order to determine which items from the collection may be favored by individual users, conventional CF approaches take the ratings of the target user on the seen items and use them to predict the ratings

⁷<http://www.netflixprize.com/>

⁸<http://www.dai-labor.de/camra2010/challenge/>

⁹<http://www.sigkdd.org/kdd2011/kddcup.shtml>

¹⁰<http://www.amazon.com/>

¹¹<http://www.youtube.com/>

					
	5	?	3	?	1
	1	?	?	?	4
	?	4	5	?	?
	?	4	?	2	5

Figure 1.2: Illustration of the data processed by a CF-based recommender system. Here, users express their preferences to the items (movies) by using a 5-scale rating. The items with a question mark are unseen for the corresponding user. CF approaches are used to predict the relevance ratings for the unseen items to an individual user. We refer to the user for whom the item ratings are predicted as *the target user*.

for this user for the unseen items. Then, items are recommended in a descending order according to their predicted ratings. While CF has been investigated and improved extensively over the past years, there is still room for substantial improvement. In this thesis we focus on improvement of two critical aspects of CF, namely the ranking and the context-awareness of the recommendations. In the following, we elaborate on each of these aspects in turn and discuss research questions that guided us in conducting our research. The results of our investigation are reported in the technical chapters of the thesis.

1.3 From Ratings to Rankings

Since the ultimate output of most recommender systems takes the form of a ranked item list, it is intuitive that the relative ranking of items inferred from the predicted ratings is much more important than the actual predicted ratings. In some use cases, users are even not able to express their preferences for items by ratings, in which cases only *implicit feedback* from users' behavior, such clicking and downloading, is recorded in the system. An illustration of such a case is given in Fig. 1.3. Such implicit feedback might only give a weak indication of which items the user might like and is therefore less informative as input to the recommendation algorithm than the ratings. This implies that the conventional CF paradigm of recommending via rating prediction is essentially not applicable in all use cases. While lots of research contributions in CF have been devoted to rating prediction, little attention was given to improve CF by modeling the ranking of items directly. Corresponding to this first open issue,




















					
		?		?	?
		?			?
	?		?	?	
		?			?

Figure 1.3: Illustration of a CF-based recommender system with implicit feedback data. Here, we have the information about which items (fruits) each user may like. However, no numerical or ordinal preferences were indicated by the users.

our first key research question to be addressed in this thesis is:

How to directly optimize the ranking of items for recommendation without first predicting individual ratings?

We approach answering this question by adopting the *learning-to-rank* paradigm, which is already well established in the domain of information retrieval, and by reformulating this paradigm in the specific case of recommender systems. We consider two specific use scenarios, i.e., the scenario in which the explicit feedback data (e.g., ratings) are available, and the other scenario in which only implicit user feedback data (e.g., clicks) are available.

- *Learning to rank from the ratings:* Here, it is likely that one user’s ratings on different items already indicate her preferences with respect to those items. For example, we can interpret the observation that *Bob* rated the movie “*Titanic*” with 5 stars and “*Matrix*” with 3 stars as that *Bob* likes “*Titanic*” better than “*Matrix*”. Following this intuition, the known ratings of each individual user for a given set of items can be transformed into training data used for learning of the ranking models. In **Chapter 2**, we propose a unified recommendation model, in which the major contribution lies in a ranking approach that directly models the ranked lists of items across all the users.
- *Learning to rank based on implicit feedback:* Here, no ratings are available for constructing the training data for developing ranking models. Furthermore, the implicit feedback is insufficiently informative as input for model learning. What can be done, however, is to measure the quality of the given list of items for a user by applying certain evaluation metrics that are defined for ranked items with binary relevance judgments. This

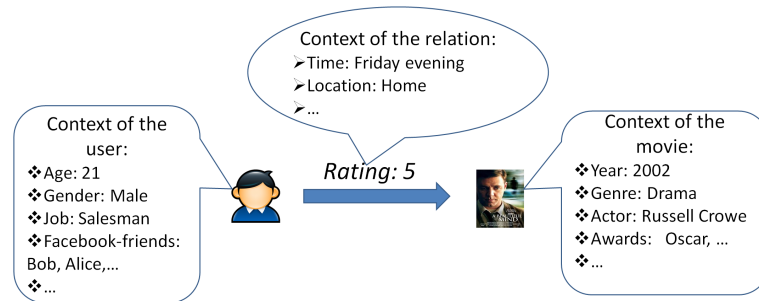


Figure 1.4: Different types of context used to improve the effectiveness of a recommender system. The information on the left and right indicates the context of the user and item, respectively. The context of the user-item interaction is indicated on the top.

observation motivates us to explore a new direction, namely, to directly model and optimize the evaluation metrics defined for assessing ranked items. In **Chapter 3**, we introduce collaborative “less-is-more” filtering (CLiMF) specifically to address the problem of recommendation in the scenarios with implicit information. CLiMF is proposed to directly model and optimize one of the most well-known ranking metrics for ranked item lists.

1.4 Recommendation in a Context

Conventional CF methods typically rely on the user-item interactions (e.g., the user-item ratings/clicks) only. In practice, various contextual information sources beyond the user-item interactions are available and have proven to be valuable for improving the effectiveness of recommender systems. For example, a user may like to watch the movie “Sleepless in Seattle” around the Valentine’s Day, but be unlikely to watch this movie on Halloween. In this example, it is obvious that the context of time plays a crucial role for determining the quality of movie recommendation. For this reason, the second open issue that inspired the research reported in this thesis can be regarded as the problem of *context-aware recommendation*. Accordingly, we establish our second key research question in this thesis as:

How to effectively incorporate the contextual information into CF for improved recommendation?

To answer this question, we first distinguish between two different types of contextual information that are increasingly available on the platforms embed-

ding the recommender functionality, which we focus on in this thesis. We also illustrate them on the example in Fig. 1.4. The first type of contextual information is *the context of the users and the items* themselves, which is not directly associated with the user-item interactions, but which can be used to enrich these interactions and improve recommendation. For example, the online social friendship links provide valuable information about the social context of the user and might point to more or different users with similar tastes and interests like the target user and better inform the interpretation of the links derived from the user-item matrix in terms of their relevance for recommendation. We investigate the mechanisms for effectively incorporating this information in two recommendation use cases, movie recommendation and landmark recommendation. We do this by formulating and evaluating the corresponding context-aware recommender algorithms as reported in **Chapter 4** and **Chapter 5**, respectively.

The second type of contextual information is *the context of user-item interactions*. For example, if a user watched a movie on Saturday evening, then this time information is the context of the interaction between the user and the movie and can be used to inform the recommendation of similar unseen movies to this user (and other users who have similar interests to this user) at this particular time in the future. With the method reported in **Chapter 6** we explore the potential of this type of contextual information to improve the effectiveness of recommendation, in a given context, but also in general.

1.5 Recommender Systems: New Developments

While the technical contributions of this thesis reported in Chapter 2-6 already address several important open challenges in the field of collaborative recommendation, many more of such challenges still wait to be pursued. Some of them have emerged from new developments on the Internet, where, for instance, rapidly growing social networks provide virtually endless information resources to learn about the users and items. Optimally exploiting this knowledge for improving the recommendation requires sophisticated new mechanisms, such as proposed in recent works in the domain of *social recommendation*. Furthermore, users are omnipresent on the Internet, uploading, downloading, rating and commenting on items simultaneously in different domains (e.g., music, books, video, news sites, and social network sites). It is intuitive that the information linking a user and an item in one domain could be informed by analyzing the relations between the users and items in other domains, which can also be referred to as *cross-domain collaborative filtering*. In addition, the spread of digital technology has increased the impact of the Internet in new societal contexts characterized by new applications, whose services may target

specific user groups, e.g., a group of seniors in assisted living environments. Recommender systems can play a critical role for this particular user group, if tuned to satisfy the specific requirements characterizing these societal contexts. For instance, they could be tailored for effective *group recommendation* for the purpose of serving the users in elderly homes and stimulating their exchange of memories.

Another category of new challenges for recommender systems can be derived from the increasing convergence between different knowledge and technology domains. The challenges building, for instance, on the synergy between search and recommendation, or between user interaction and recommendation, have a large potential not only to improve the quality of recommendation, but also to lead to new exciting paradigms of multimedia content access.

In **Chapter 7**, we analyze the new developments addressed above and elaborate in more depth on the above and other challenges related to the evolution of recommender systems and their increasing impact in the future. Based on this analysis, we make recommendations for future research directions in this field.

1.6 List of Publications

The author has published the following work during his Ph.D.. The remaining chapters in this thesis are based on the publications, as indicated.

Journals

1. Shi, Y., Larson, M. and Hanjalic, A. Collaborative Filtering beyond the User-item Matrix: Opportunities for Exploiting Context in Recommender Systems. *ACM Computing Surveys*, under review. (**Chapter 7**)
2. Shi, Y., Larson, M. and Hanjalic, A. Exploiting Social Tags for Cross-domain Collaborative Filtering. *ACM Transactions on the Web*, under review.
3. Shi, Y., Serdyukov, P., Hanjalic, A. and Larson, M. Non-trivial Landmark Recommendation Using Geotagged Photos. *ACM Transactions on Intelligent Systems and Technology*, 4(3), 2013. (**Chapter 5**)
4. Shi, Y., Larson, M., and Hanjalic, A. Unifying Rating-oriented and Ranking-oriented Collaborative Filtering for Improved Recommendation. *Information Sciences*, Elsevier, 229 (20), 29-39, 2013. (**Chapter 2**)
5. Shi, Y., Larson, M. and Hanjalic, A. Mining Contextual Movie Similarity with Matrix Factorization for Context-aware Recommendation.

ACM Transactions on Intelligent Systems and Technology, 4(1), 2013.
(**Chapter 4**)

Conferences

1. Shi, Y., Karatzoglou, A., Baltrunas, L., Larson, M., Oliver, N, and Hanjalic, A.. CLiMF: Learning to Maximize Reciprocal Rank with Collaborative Less-is-More Filtering. *Proceedings of the 6th international ACM conference on Recommender Systems*, 139-146, 2012. (**Chapter 3**)
2. Shi, Y., Karatzoglou, A., Baltrunas, L., Larson, M., Hanjalic, A. and Oliver, N. TFMAP: Optimizing MAP for Top-n Context-aware Recommendation. *Proceedings of the 35th international ACM SIGIR conference on Research and development in Information Retrieval*, 155-164, 2012. (**Chapter 6**)
3. Shi, Y., Zhao, X., Wang, J., Larson, M. and Hanjalic, A. Adaptive Diversification of Recommendation Results via Latent Factor Portfolio. *Proceedings of the 35th international ACM SIGIR conference on Research and development in Information Retrieval*, 175-184, 2012.
4. Shi, Y., Serdyukov, P., Hanjalic, A. and Larson, M. Personalized Landmark Recommendation Based on Geotags from Photo Sharing Sites. *Proceedings of the fifth international conference on weblogs and social media*, AAAI, 622-625, 2011.
5. Shi, Y., Larson, M. and Hanjalic, A. Tags as Bridges between Domains: Improving Recommendation with Tag-induced Cross-domain Collaborative Filtering. *Proceedings of the 19th international conference on User modeling, adaption, and personalization*, Springer-Verlag, 305-316, 2011.
6. Shi, Y., Larson, M. and Hanjalic, A. Reranking Collaborative Filtering with Multiple Self-contained Modalities. *Proceedings of the 33rd European conference on Advances in information retrieval*, Springer-Verlag, 699-703, 2011.
7. Shi, Y., Larson, M. and Hanjalic, A. How Far are We in Trust-aware Recommendation? *Proceedings of the 33rd European conference on Advances in information retrieval*, Springer-Verlag, 704-707, 2011.
8. Shi, Y., Larson, M. and Hanjalic, A. List-wise Learning to Rank with Matrix Factorization for Collaborative Filtering. *Proceedings of the 4th ACM conference on Recommender systems*, 269-272, 2010.

9. Shi, Y., Larson, M. and Hanjalic, A. Exploiting User Similarity Based on Rated-item Pools for Improved User-based Collaborative Filtering. *Proceedings of the 3rd ACM conference on Recommender systems*, 125-132, 2009.

Workshops

1. Said, A., Tikk, D., Shi, Y., Larson, M., Stumpf, K. and Cremonesi, P. Recommender systems evaluation: A 3D benchmark. *Proceedings of the Workshop on Recommendation Utility Evaluation: Beyond RMSE*, 2012.
2. Shi, Y., Larson, M. and Hanjalic, A. Mining Relational Context-aware Graph for Rater Identification. *Proceedings of the 2nd Challenge on Context-Aware Movie Recommendation*, 53-59, 2011.
3. Shi, Y., Larson, M. and Hanjalic, A. Towards understanding the challenges facing effective trust-aware recommendation. *Proceedings of the Workshop on Recommender Systems and the Social Web*, 2010
4. Shi, Y., Larson, M. and Hanjalic, A. Mining Mood-specific Movie Similarity with Matrix Factorization for Context-aware Recommendation. *Proceedings of the Workshop on Context-Aware Movie Recommendation*, 34-40, 2010.
5. Shi, Y., Larson, M. and Hanjalic, A. Connecting with the Collective: Self-contained Reranking for Collaborative Recommendation. *Proceedings of the 1st ACM international workshop on Connected multimedia*, 9-14, 2010.

Chapter 2

Unified Recommendation Model

We propose a novel unified recommendation model, URM, which combines a rating-oriented collaborative filtering (CF) approach, i.e., probabilistic matrix factorization (PMF), and a ranking-oriented CF approach, i.e., list-wise learning-to-rank with matrix factorization (ListRank). The URM benefits from the rating-oriented perspective and the ranking-oriented perspective by sharing common latent features of users and items in PMF and ListRank. We present an efficient learning algorithm to solve the optimization problem for URM. The computational complexity of the algorithm is shown to be scalable, i.e., to be linear with the number of observed ratings in a given user-item rating matrix. The experimental evaluation is conducted on three public datasets with different scales, allowing validation of the scalability of the proposed URM. Our experiments show the proposed URM significantly outperforms other state-of-the-art recommendation approaches across different datasets and different conditions of user profiles. We also demonstrate that the primary contribution to improve recommendation performance is contributed by the ranking-oriented component, while the rating-oriented component is responsible for a significant enhancement.

This work was first published as “List-wise learning to rank with matrix factorization for collaborative filtering” by Y. Shi, M. Larson, and A. Hanjalic, in Proc. of the fourth ACM conference on Recommender systems (RecSys ’10), Barcelona, Spain, 2010 [144]. This chapter is an extended version that has been published as “Unifying rating-oriented and ranking-oriented collaborative filtering for improved recommendation” in *Information Sciences*, 229 (20), Elsevier, 2013.

2.1 Introduction

Recommender systems attract research attention because they are able to connect users directly with consumable items, supporting them in handling the unprecedentedly large amounts of content, e.g., movies, music and books currently available online by providing personalized recommendations [2, 39]. Collaborative filtering (CF) is widely acknowledged as one of the most successful recommender techniques. Compared to content-based approaches, CF enjoys the advantage of being content-agnostic. In other words, it can recommend items without the additional computational expense or copyright issues involved with processing items directly. One of two different types of approaches can be taken by a recommender system in order to generate recommendation lists for users. Under one approach, the system predicts ratings for individual items first and then generates the ranked recommendation list. We refer to this type of CF-based recommendation as rating-oriented [55, 75, 134]. Under the other approach, the system predicts rank scores, that are not necessarily related to ratings, but rather used directly to generate the recommendation list. We refer to this type of approach as ranking-oriented [90, 92, 144, 182, 183].

To illustrate the difference between rating- and ranking-oriented CF, we consider two specific toy examples. The first example involves the ratings of a user on items i and j . We assume that the user has rated item i with a 4 and item j with a 3; these are the reference values that we use to judge the quality of the predictions of the recommender system. If two recommendation approaches give rating predictions of (3, 4), and (5, 2) on items (i, j), the rating prediction error, e.g., measured by mean absolute error or root mean square error [57] will be the same for both approaches. However, only the ranking-oriented perspective identifies the second approach as faithfully reflecting the users relatively higher preference for item i over item j . This example should not lead to the conclusion that working with absolute ratings is detrimental to recommendation performance. Quite to the contrary, successful recommender systems do use a rating-oriented approach to generate recommendation lists for users, e.g., MovieLens [55] and Netflix [75]. Our second example illustrates the usefulness of absolute ratings in capturing users preference strength. If user u and v have ratings (5, 3) and (4, 3) on items (i, j), user u is more explicit about his preference for item i over item j than user v . This information holds the potential to help resolve possible ambiguities in generating a ranked item list for the user u . Further, predicted ratings can provide the user with additional information used to inform the decision of whether or view, purchase or download the item. Taken together, these examples serve to motivate our standpoint that ranking-oriented approaches have high potential and that combining rating-oriented and ranking-oriented approaches holds promise for designing more successful recommendation algorithms.

Another source of motivation derives from the recent recommender system literature, which demonstrates a growing awareness that under ranking-oriented recommendation, the ability of the system to predict ratings is also important. This awareness is based on the insight that although users find it important to receive a high quality ranked list from the recommender system, the list will be less useful or less acceptable to the user if the ratings assigned by the system to the items fail to approximate those that the user would have assigned. The increasing emphasis on providing the user with both a high quality ranked list and accurate ratings is reflected in the recent adoption of the Normalized Discounted Cumulative Gain (NDCG) evaluation metric [90, 92, 182, 183]. As discussed in more detail in Section 2.4.2, NDCG simultaneously takes into account both the rank ordering of a list as well as the graded relevance, i.e., the magnitude of the scores of the items in the list. Somewhat unexpectedly, although recommender system research is increasingly taking both rank and rating prediction into account for evaluation, up until this point, no concerted research effort has been devoted to developing algorithms that produce recommendation lists that simultaneously optimize both rank and ratings of the recommended items. The contribution of this chapter is to combine the two types of recommendation, ranking-oriented and rating-oriented, in order to arrive at a system that generates recommendations that are more completely suited to satisfy user needs.

We accomplish the goal of generating recommendations optimized not only for ranking, but also for rating by proposing a novel unified recommendation model (URM) that enhances ranking-oriented recommendation using a rating-oriented approach. The model combines probabilistic matrix factorization (PMF) [134], i.e., rating-oriented CF, and ListRank [144], i.e., ranking-oriented CF, by exploiting common latent features shared by both PMF and ListRank. In fact, by incorporating PMF we enable ListRank to benefit from rating predictions, which contributes another basis for generating the recommendation list. We demonstrate experimentally that the URM achieves significant improvement of recommendation performance over the state-of-the-art CF approaches on various data sets. Furthermore, we analyze and empirically demonstrate that URM maintains linear complexity with the number of observed ratings in the given user-item matrix, which means that it can scale up with the increasing amount of data.

The approach presented in this chapter builds on and expands the basic finding of the effectiveness of list-wise learning-to-rank, demonstrated in [144], where we first introduced ListRank, a ranking-oriented matrix factorization approach. The expansions that are presented here extend along two dimensions. First, we combine the advantages of ranking-oriented and rating-oriented recommendation by combining ListRank with a rating-oriented component, resulting in URM, a new recommendation model. Second, we conduct experimental eval-

uations on multiple datasets of various scales to validate the usefulness of the proposed URM approach, and demonstrate its specific contributions to the state of the art.

The remainder of the paper is structured as follows. In the next section, we summarize related work and position our approach with respect to it. Then, we present the URM and validate it experimentally. Finally, we sum up the key aspects of URM and address possible directions for future work.

2.2 Related Work

Our work builds on the foundation of the large body of work that has been carried out on CF. CF approaches are generally considered to fall into one of two categories, i.e., memory-based CF and model-based CF [2, 39]. In general, memory-based CF uses similarities between users (user-based CF) or similarities between items (item-based CF) to make recommendations. User-based CF [55, 129] recommends items to a user on the basis of how well similar users like those items. Item-based CF [38, 88, 136] recommends items to a user based on the similarity between the user’s favored items and the items to be recommended. Recently, various studies have been devoted to the modification and enhancement of memory-based CF, e.g., to specifically improve user-based CF [142, 194], to specifically improve item-based CF [191], and to combine user-based CF and item-based CF [95, 176]. Although substantial improvements have been achieved, memory-based CF approaches still suffer from high computational complexity, i.e., computing similarities among the typically enormous number of users or items in recommender system applications is expensive.

In comparison, model-based CF approaches first fit prediction models based on training data and then use the model to predict users’ preferences on items. These models include latent semantic models [58], mixture models [66, 152] and fuzzy linguistic models [105]. Matrix factorization (MF) [75, 134] has been recognized as one of the most successful model-based CF approaches, due to its superior accuracy and scalability. Generally, MF models learn low-rank representations (latent features) of users and items from the observed ratings in the user-item matrix, which are further used to predict unobserved ratings. MF can also be formulated from a probabilistic perspective, i.e., PMF [134], which models the conditional probability of latent features given the observed ratings, and factors for complexity regularization encoding prior information on user and item ratings. In this chapter, we adopt PMF as the rating-oriented CF component of our proposed URM.

Compared to the large volume of research on rating-oriented CF, the research

on ranking-oriented CF is limited. The first mature ranking-oriented CF approach is *CofiRank* [182, 183], which introduces structured ranking losses and various other extensions to MF. Further studies mainly focus on exploiting pair-wise preference between items for users, e.g., *EigenRank* [90], probabilistic latent preference analysis [92] and Bayesian personalized ranking [126]. However, all these existing pair-wise approaches [90, 92, 126] require deriving pair-wise training examples from individual ratings, thus, in general all suffer from high computational complexity of pair-wise comparisons, which scale quadratically to the number of rated items in a given data collection. In contrast, *ListRank* [144] is designed to incorporate a list-wise learning-to-rank concept with MF, which is characterized by a low complexity, i.e., complexity is linear with the number of the observed ratings in a given user-item rating matrix. Preliminary experiments [144] also show *ListRank* to be competitive for recommendation in comparison to other state-of-the-art approaches, represented by *CofiRank*. One of the latest contributions on exploiting other learning-to-rank methods for CF [9] shares the same motivation of *ListRank*, and also envisioned the potential of list-wise approach for CF, which is represented by *ListRank*. The established performance and value of *ListRank* makes it a natural choice as our ranking-oriented approach, to be extended within the proposed URM.

Our work in this chapter unifies a rating-oriented CF, i.e., PMF and a ranking-oriented CF, i.e., *ListRank* in terms of the same latent features shared by PMF and *ListRank*. In view of the comparison of ranking-oriented and rating-oriented CF in the previous section, and also considering the target of generating a ranked list of recommendations for the user, we chose the ranking-oriented approach as the basis of our unified recommendation model (URM) and deploy rating-oriented PMF to expand it.

2.3 Unified Recommendation Model

In this section, we first briefly present the basic formulation of PMF and *ListRank*. Then, we combine PMF and *ListRank* by means of the URM and, finally, we present an efficient learning algorithm for solving the optimization problem in the URM and analyze the complexity of the algorithm.

2.3.1 PMF: Matrix Factorization for Rating

If we denote by R a user-item rating matrix consisting of M users ratings on N items, PMF [134] seeks to represent the matrix R by two low-rank matrices, U and V . A d -dimensional set of latent features is used to represent both users (in U) and items (in V). Note that we use U_i to denote a d -dimensional column feature vector of user i , V_j to denote a d -dimensional column feature vector of

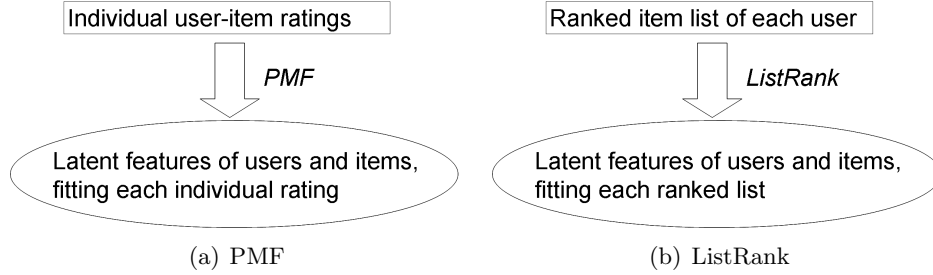


Figure 2.1: The input-output diagrams of PMF and ListRank

item j , and R_{ij} to denote the user i 's rating on item j . Usually, the rating scale is different from one dataset (application scenario) to another. To achieve generality, the ratings are normalized to the range from $[0, 1]$. The objective of PMF is now to fit each rating R_{ij} with the corresponding inner product $U_i^T V_j$, which can be formulated as follows:

$$U, V = \arg \min_{U, V} \left\{ \frac{1}{2} \sum_{i=1}^M \sum_{j=1}^N I_{ij} (R_{ij} - g(U_i^T V_j))^2 + \frac{\lambda_U}{2} \|U\|_F^2 + \frac{\lambda_V}{2} \|V\|_F^2 \right\} \quad (2.1)$$

Here, I_{ij} is an indicator function that equals 1 when $R_{ij} > 0$, and 0 otherwise. The parameters λ_U and λ_V are regularization coefficients used to reduce overfitting, while $\|U\|_F$ and $\|V\|_F$ are the Frobenius norms of the matrices U and V . For simplicity, we set $\lambda_U = \lambda_V = \lambda$. The $g(x)$ is a logistic function serving to bound the range of $U_i^T V_j$ to be also in the range $[0, 1]$, i.e., $g(x) = 1/(1 + e^{-x})$. The input-output diagram of PMF is illustrated in Fig. 2.1(a).

2.3.2 ListRank: Matrix Factorization for Ranking

In order to model the user's preference from her ranked list of rated items, we need to transform the user's ratings on different items to ranking scores, which are required to maintain two properties. First, for a given user, the ranking score of item i should be higher than (or lower than, or equal to) item j , if she rates item i higher than (or lower than, or equally to) item j . Second, the ranking scores of all the users should share the same scale/space. For this reason, we exploit the top one probability [123] for the transformation from ratings of each user to ranking scores. From the probabilistic point, the top one probability indicates the probability of a graded item being ranked in the top position from all the graded items. Note that top one probability and its similar variants are usually used to map graded scores into a probability space in the literature [22, 25]. Specifically, the top one probability (the ranking score) for item j that is rated R_{ij} by user i can be expressed as:

$$p(R_{ij}) = \frac{\exp(R_{ij})}{\sum_{k=1}^N \exp(R_{ik})} \quad (2.2)$$

in which $\exp(x)$ denotes the exponential function of x .

As opposed to PMF that aims at reproducing and extrapolating the ratings from R , the ListRank [144] has the objective to fit each user's ranked list of items with a factorization model. A regularized loss function that models the cross-entropy of top-one probabilities of the items in the training ranked item lists and the lists from the factorization model can be formulated as follows:

$$\begin{aligned}
 L(U, V) &= \sum_{i=1}^M \left\{ - \sum_{j=1}^N I_{ij} p(R_{ij}) \log p(g(U_i^T V_j)) \right\} + \frac{\lambda}{2} (\|U\|_F^2 + \|V\|_F^2) \\
 &= \sum_{i=1}^M \left\{ - \sum_{j=1}^N I_{ij} \frac{\exp(R_{ij})}{\sum_{k=1}^N I_{ik} \exp(R_{ik})} \log \frac{\exp(g(U_i^T V_j))}{\sum_{k=1}^N I_{ik} \exp(g(U_i^T V_k))} \right\} \\
 &\quad + \frac{\lambda}{2} (\|U\|_F^2 + \|V\|_F^2) \tag{2.3}
 \end{aligned}$$

Note that in ListRank we also adopt the same simplification strategy as used in PMF (see Section 2.3.1), i.e., setting an equal regularization parameter λ for penalizing the magnitudes of both U and V . While the training lists are derived from the profiles of the users, the loss function reflects the uncertainty in predicting the output lists from the factorization model using the training lists. Note that minimizing the regularized loss function 2.3 results in a factorization model, i.e., U and V that is not optimized for rating prediction, but for ranking positions of items in the users lists. This key difference between ListRank and PMF is also shown in Fig. 2.1.

2.3.3 Combining PMF and ListRank

As introduced above, PMF and ListRank learn the latent features of users and items by taking different views on the known data, i.e., PMF exploiting the individual ratings, and ListRank exploiting the ranked lists. Our motivation of URM is then straightforward so that the two different views can be exploited simultaneously, by which the knowledge encoded in individual ratings is expected to improve the latent features of users and items from ListRank to achieve better ranking performance, as the example mentioned in Section 2.1. The illustration diagram of URM is shown in Fig. 2.2. Since both the PMF and the ListRank are based on matrix factorization, we link the two by imposing common latent features for both models. Then, the URM can be formulated

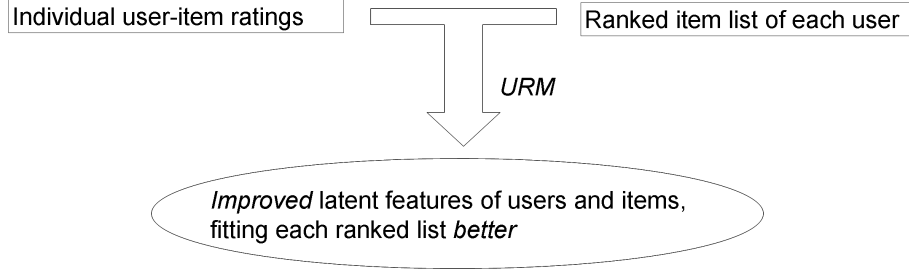


Figure 2.2: The input-output diagram of URM

by means of a new regularized loss function $F(U, V)$ as follows:

$$\begin{aligned}
 F(U, V) = & \alpha \times \frac{1}{2} \sum_{i=1}^M \sum_{j=1}^N \left(I_{ij} (R_{ij} - g(U_i^T V_j))^2 \right) \\
 & + (1 - \alpha) \times \sum_{i=1}^M \left\{ - \sum_{j=1}^N I_{ij} \frac{\exp(R_{ij})}{\sum_{k=1}^N I_{ik} \exp(R_{ik})} \log \frac{\exp(g(U_i^T V_j))}{\sum_{k=1}^N I_{ik} \exp(g(U_i^T V_k))} \right\} \\
 & + \frac{\lambda}{2} (\|U\|_F^2 + \|V\|_F^2) \tag{2.4}
 \end{aligned}$$

The trade-off parameter α is used to control the relative contribution from PMF and ListRank. As stated in the introduction, we bias the loss function towards ranking. Consequently, the value of α should be relatively small. We justify this choice in Section 2.4.3, where we experimentally investigate the impact of α on the recommendation performance. Minimizing the loss function Eq. (2.4) results in the matrices U and V that are not only optimized for item ranking, but also enhanced by the information used to predict each item's rating. This result can be used to produce a ranked recommended items list for each user i and is generated by ordering items in the collection in the descending order according to the value $U_i^T V$. Note that items already rated by a user will be removed from the list.

2.3.4 Learning Algorithm and Complexity Analysis

Since the loss function (2.4) is not convex jointly over U and V , we choose to deploy a gradient descent method by alternatively fixing U and V and searching for local minima. The gradients of $F(U, V)$ with respect to U and V can be

computed as:

$$\begin{aligned} \frac{\partial F}{\partial U_i} = & \alpha \sum_{j=1}^N I_{ij} (g(U_i^T V_j) - R_{ij}) g'(U_i^T V_j) V_j \\ & + (1 - \alpha) \sum_{j=1}^N I_{ij} \delta_{ij} g'(U_i^T V_j) V_j + \lambda U_i \end{aligned} \quad (2.5)$$

$$\begin{aligned} \frac{\partial F}{\partial V_j} = & \alpha \sum_{i=1}^M I_{ij} (g(U_i^T V_j) - R_{ij}) g'(U_i^T V_j) U_i \\ & + (1 - \alpha) \sum_{i=1}^M I_{ij} \delta_{ij} g'(U_i^T V_j) U_i + \lambda V_j \end{aligned} \quad (2.6)$$

where:

$$\delta_{ij} = \frac{\exp(g(U_i^T V_j))}{\sum_{k=1}^N I_{ik} \exp(g(U_i^T V_k))} - \frac{\exp(R_{ij})}{\sum_{k=1}^N I_{ik} \exp(R_{ik})} \quad (2.7)$$

$g'(x)$ denotes the derivative of $g(x)$. An overview of the algorithm deploying Eq. (2.5) and (2.6) for solving the minimization problem in the URM is given in Algorithm 1. The stopping parameter ϵ is used to indicate the desired level of the convergence of the algorithm. In our experiments, the value of ϵ is set to 0.01. Our experiments showed that the algorithm usually converges after no more than 200 iterations. Unlike the constant learning step size η as used for ListRank [144], we allow η in the URM to be as large as possible (maximally 1) in each iteration, as long as it leads to a decrease in the loss function Eq. (2.4). Setting η in this flexible way helps to speed up the convergence of the algorithm.

It can be easily shown that the complexity of the loss function for URM is in the order of $O(dS + d(M + N))$, where S denotes the number of observed ratings in a given user-item matrix and where d is the dimensionality of latent features. The complexity of the gradients in Eq. (2.5) and (2.6) is of the order $O(dS + dM)$ and $O(dS + p dS + dN)$, respectively, where p denotes the average number of items rated per user and usually is substantially smaller than S . Considering we also often have $S \gg M, N$, the total complexity in one iteration has the order of $O(dS)$, which is linear with the number of observed ratings in the matrix. This analysis indicates the computational efficiency and scalability of URM. This will also be illustrated quantitatively in Section 2.4.4.

2.4 Experiments and Evaluation

In this section we present a series of experiments that evaluate the proposed URM. We first give a detailed description of the setup of our experiments. Then,

ALGORITHM 1: Learning algorithm for URM

Input: Training data R , tradeoff parameter α , regularization parameter λ , stopping threshold ϵ .**Output:** Complete user-item relevance matrix \hat{R} .Initialize $U^{(0)}, V^{(0)}$ with random values;Initialize f_1 with a large value and f_2 a small value; $t = 0$;**repeat** $f_1 = F(U^{(t)}, V^{(t)})$; $\eta = 1$; Compute $\frac{\partial F}{\partial U^{(t)}}$, $\frac{\partial F}{\partial V^{(t)}}$ as in Eq. (2.5) and (2.6); **repeat** $\eta = \eta/2$; **until** $F(U^{(t)} - \eta \frac{\partial F}{\partial U^{(t)}}, V^{(t)} - \eta \frac{\partial F}{\partial V^{(t)}}) < f_1$; $U^{(t+1)} = U^{(t)} - \eta \frac{\partial F}{\partial U^{(t)}}$, $V^{(t+1)} = V^{(t)} - \eta \frac{\partial F}{\partial V^{(t)}}$; $f_2 = F(U^{(t+1)}, V^{(t+1)})$; $t = t + 1$;**until** $f_1 - f_2 \leq \epsilon$; $\hat{R} = U^{(t)T} V^{(t)}$;

we investigate the impact of tradeoff parameters in URM and demonstrate the effectiveness and efficiency of URM. Finally, we compare the recommendation performance of URM with some other baseline and state-of-the-art approaches.

We designed the experiments in order to be able to answer the following research questions:

1. Could URM as a combination of a rating-oriented and a ranking-oriented CF approach outperform each of the individual approaches? (Section 2.4.3 and 2.4.5)
2. Does the recommendation performance increase with the minimization of the loss function Eq. (2.4)? (Section 2.4.4)
3. How efficient and scalable is URM? (Section 2.4.4)
4. How does URM compare to alternative state-of-the-art approaches across different data sets and across users with different profiles? (Section 2.4.5)

Table 2.1: Statistics of datasets used in the experiments

	# users	# items	# ratings	Sparseness	Scale	Ave. # ratings/user	Ave. rating
ML1	943	1682	100000	93.7%	1-5	106.0	3.53
ML2	6040	3706	1000209	95.5%	1-5	165.6	3.58
EM	61265	1623	2811718	97.2%	1-6	45.9	4.04

2.4.1 Datasets

Our experiments are conducted on three publicly available datasets, i.e., two datasets from MovieLens¹, and the EachMovie² dataset. All of them are widely used in the field of recommender systems. The first MovieLens dataset [55], denoted as ML1, contains 100K ratings (scale 1-5) from 943 users on 1682 movies. The second MovieLens data set, denoted as ML2, contains 1M ratings (scale 1-5) from ca. 6K users on ca. 3.7K movies. Each user in both ML1 and ML2 has rated at least 20 movies. The EachMovie dataset contains ca. 2.8M ratings (scale 1-6) from ca. 61K users on ca. 1.6K movies. Note that in all of the used datasets we excluded the items (i.e., movies) that are never rated. Thus, the aforementioned statistics of the datasets may be slightly different from those in other literature. Some detailed statistics of the datasets are summarized in Table 2.1.

2.4.2 Experimental Setup and Evaluation Metrics

We choose to conduct our experiments following a standard protocol as widely used in related work [144, 182, 183]. Note that our experimental protocol is designed to demonstrate the effectiveness of URM under different conditions of user profiles. We create variants of the datasets in order to test experimental conditions involving three different user profile lengths (UPLs), i.e., 10, 20 and 50. For example, in the case of UPL=10, we randomly select 10 rated items for each user for training, and use the remaining user ratings for testing. Per UPL, users with less than 20, 30, or 60 rated items are removed in order to ensure we can evaluate on at least 10 rated items per user. For each UPL, we create 10 different versions of the dataset by sampling the user profiles to arrive at the targeted number of items in the training set. Note that in the case of UPL=50 for each dataset, we create an additional version that is used as a validation set to tune the tradeoff parameter and investigate the impact of this parameter as shown in Section 2.4.3. The data from the validation sets have not been used for the test runs, which are used to evaluate the algorithm. We report the

¹<http://www.grouplens.org/node/73>

²<http://kumpf.org/eachtoeach/eachmovie.html>

average performance attained across all users and 10 test runs in Section 2.4.5.

Following the standard evaluation strategy applied to recommender systems [90, 92, 182, 183], we measure the recommendation performance only based on the rated items from each user. We consider the performance of a recommender algorithm to be good if it ranks items with high ratings in the test set to higher positions in the ranked list than those having low ratings. The algorithm should also emphasize the accuracy of highly ranked items, since users usually expect highly relevant items to be recommended as early as possible. The evaluation metric Normalized Discounted Cumulative Gain (NDCG) satisfies the two requirements and is widely used in recommender systems research [90, 92, 182, 183]. Note that since we are not interested in rating prediction performance, metrics, such as mean average error (MAE), root mean square error (RMSE), are not considered. Also notice that since the datasets in our experiments contain graded relevance, NDCG should be more appropriate than other metrics, such as precision, recall, mean average precision (MAP), for which artificial thresholds need to be assumed to convert graded relevance to binary case. For those reasons, NDCG could be the best choice among all the metrics for our experimental evaluation. The definition of NDCG at the top- K ranked items for a user u can be given as:

$$NDCG_u@K = Z_u \sum_{k=1}^K \frac{2^{Y_u^{(k)}} - 1}{\log_2(1 + k)} \quad (2.8)$$

Here, $Y_u(k)$ denotes the grade of relevance of the item that is ranked in the k -th position for user u . Note that in this setting the rating is regarded as the grade of relevance. Z_u is a normalization factor securing that the perfect ranking list will have $NDCG_u@K$ equal to 1. In other words, $1/Z_u$ is equal to $NDCG_u@K$ when the ranked list is created by sorting the ground truth items of the users in the test set in descending order by their ratings. In this chapter, we report the recommendation performance by NDCG@5 and NDCG@10, which are averaged across all users.

We did not formally tune the dimensionality d of latent features and the regularization parameter λ for the URM in the experiments. The dimensionality d is set independent of the user-item matrix, and usually a small value of d is sufficient for acceptable recommendation performance [183]. In this work, we fix d as 10, which we adopted from the recently proposed CofiRank approach [183]. The regularization parameter λ is usually set large enough to avoid over-fitting, as demonstrated in ListRank [144]. We fix λ as 0.1 for all the experiments on different data sets, a setting from which we did not observe over-fitting.

2.4.3 Impact of Tradeoff Parameter

In this subsection we investigate the impact of tradeoff parameter α on the performance of the proposed URM. For each dataset, we conduct an experiment

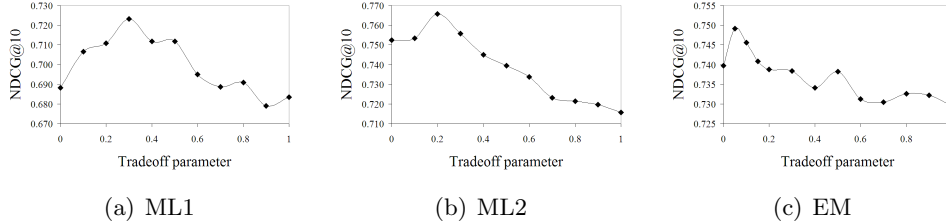


Figure 2.3: Impact of the tradeoff parameter in URM on NDCG@10

on a validation set under the condition of $UPL=50$, i.e., in the validation set we randomly select 50 rated items for each user for training and use the remaining rated items for testing. By varying the tradeoff parameter in the URM, we can evaluate its influence on the recommendation performance, i.e., NDCG@10 here, as shown in Fig. 2.3. Note that the URM is equivalent to ListRank if the tradeoff parameter $\alpha = 0$ and to PMF if $\alpha = 1$. The diagrams in Fig. 2.3 indicate that different optimal values of tradeoff parameters can be selected for different datasets. Selecting this optimal value per dataset leads to an improvement in the recommendation performance compared to either ListRank or PMF taken individually. This observation suggests the promise of combining rating-oriented and ranking-oriented approaches, providing initial evidence that our first research question can be answered positively. Additional experiments in Section 2.4.5 make further contribution to this issue. Furthermore, it can also be observed that the optimal tradeoff parameter in each dataset is below 0.5, which means that the major contribution to the recommendation performance comes from the ranking-oriented CF. This observation confirms the achievements by the recent progress in ranking-oriented CF approaches, e.g., [90, 92, 182], which usually outperform rating-oriented CF approaches, and also justifies our choice to bias the URM towards ranking prediction, as stated in Section 2.3.3. The optimal tradeoff parameters obtained from analyzing the validation sets are used subsequently on the three datasets for the test runs in all test cases as reported in Section 2.4.5.

2.4.4 Effectiveness and Efficiency

In this subsection, we investigate whether minimizing the loss function of URM in Eq. (2.4) indeed leads to an increase in recommendation performance, and whether the proposed URM is empirically an efficient algorithm. These experiments were also conducted on the validation sets. We adopt the optimal tradeoff parameters obtained from previous subsection for this investigation. The diagrams in Fig. 2.4 demonstrate the development of the loss function and NDCG@10 during the iterations of the minimization process. Here, we normal-

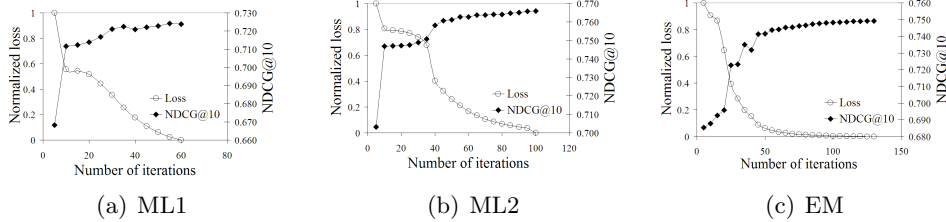


Figure 2.4: The variation of NDCG@10 and the loss in URM during the minimization

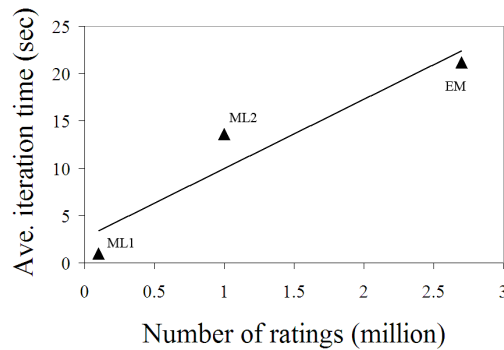


Figure 2.5: The relationship between the average iteration time in the URM and the scale of the data

ized loss for the demonstration purposes. We can see that the NDCG@10 grows steadily and converges in all of the datasets in parallel with the loss function being minimized. These observations indicate that the approach proposed in this chapter is effective in achieving improved recommendation performance, thus addressing our second research question and allowing us to give it a positive answer. Additionally, they provide evidence that URM is indeed a model of the phenomenon that it was designed to capture.

Furthermore, we can also observe in Fig. 2.4 that the NDCG@10 already becomes close to optimal after 10 iterations on ML1 dataset, after 50 iterations on ML2 dataset and EM dataset. This observation indicates that the proposed URM is efficient in reaching convergence, even for a large scale data set. We also demonstrate the relationship of average iteration time against the scale (i.e., the number of ratings) of each data set, as shown in Fig. 2.5. Note that for the smallest dataset ML1 one iteration only takes around 1 second, and for the largest data set (EM) one iteration only takes around 20 seconds in our MATLAB implementation on a PC with 1.59 GHz CPU and 2.93 GB memory. Moreover, the runtime of a single iteration increases almost linearly with the increase of the data scale, which empirically verifies that the URM can easily scale up to address datasets of any size. The conclusions from this section

lead to an answer to our third research question, namely, they demonstrate the efficiency and scalability of URM.

2.4.5 Performance Comparison

In this subsection, we compare the performance of URM and a number of representative alternative CF approaches that we list and briefly describe below. Our selection of alternative approaches covers various aspects in recommendation area, including a non-personalized approach, a widely-used memory-based approach and two state-of-the-art model-based approaches.

- **ItemAvRat**: This is a naive non-personalized recommendation approach, that recommends items to users according to the average item rating. In other words, the item that has the highest average rating in the training data will be the top recommended item for every user. Using this method, every user will get offered the same recommendation list.
- **ItemCF**: This is a traditional and widely used item-based CF approach [38, 88, 136]. Our implementation of ItemCF is based on [38].
- **PMF**: This is a state-of-the-art rating-oriented CF approach [134], which is equivalent to the proposed URM when α is set to 1. Note that we use the same dimensionality of latent features and regularization parameter as used in URM.
- **ListRank**: This is a state-of-the-art ranking-oriented CF approach [144], which is equivalent to the proposed URM when α is set to 0. Note that we also use the same dimensionality of latent features and regularization parameter as used in URM.
- **CofiRank**: This is another state-of-the-art ranking-oriented CF approach. We implemented it using publicly available software³. Regarding the parameter setting, we adopted the optimal values of most parameters from [183], and we tuned the rest of them for optimal performance using the same validation sets as for URM. Since our experimental setting is exactly same as the work of CofiRank and its extensions [183], we can compare our results directly to the best of theirs among various parameter settings if available, i.e., the CofiRank performance of NDCG@10 on ML1 and EM data sets in Table 2.2 and Table 2.4 are directly adopted from [183].

³<http://www.cofirank.org/downloads>

Table 2.2: Performance comparison in terms of NDCG between URM and other recommendation approaches on ML1 dataset.

	UPL=10		UPL=20		UPL=50	
	NDCG@5	NDCG@10	NDCG@5	NDCG@10	NDCG@5	NDCG@10
ItemAvRat	0.345	0.400	0.313	0.357	0.274	0.309
ItemCF	0.552	0.578	0.556	0.580	0.546	0.571
PMF	0.603	0.630	0.588	0.610	0.597	0.616
CofiRank	0.600	0.678	0.633	0.681	0.664	0.701
ListRank	0.672	0.693	0.682	0.691	0.687	0.684
URM	0.673*	0.694*	0.699*†	0.708*†	0.717*†	0.718*†

Table 2.3: Performance comparison in terms of NDCG between URM and other recommendation approaches on ML2 dataset.

	UPL=10		UPL=20		UPL=50	
	NDCG@5	NDCG@10	NDCG@5	NDCG@10	NDCG@5	NDCG@10
ItemAvRat	0.297	0.342	0.280	0.322	0.255	0.293
ItemCF	0.594	0.589	0.603	0.616	0.589	0.607
PMF	0.645	0.653	0.644	0.653	0.680	0.686
CofiRank	0.671	0.668	0.694	0.689	0.693	0.692
ListRank	0.647	0.654	0.683	0.688	0.751	0.751
URM	0.732*†	0.735*†	0.748*†	0.747*†	0.764*†	0.760*†

The performance of different approaches with respect to different user profile length (UPL) is shown in Table 2.2-2.4. For each dataset and each UPL we repeat experiments 10 times, i.e., with 10 random splits of training and testing data as described in Section 2.4.2. As can be seen from Table 2.2, URM outperforms other approaches significantly in most of the cases on ML1 dataset, according to Wilcoxon signed rank significance test with $p < 0.05$. Note that we use \dagger to denote the significant improvement over ListRank, and $*$ to denote the significant improvement over all the other approaches except ListRank. For the results directly available from CofiRank (Weimer et al., 2008), we did not conduct the significance test for the comparison with the corresponding results from URM, since we do not have the results of CofiRank in each run. The URM achieves large amount of improvement (ca. 20%) over the naive approach ItemAvRat and the traditional CF approach ItemCF, and over 10% improvement over PMF. Compared to the state-of-the-art CofiRank, it also achieves ca. 3-10% improvement. Note that these improvements are consistent across different user profiles, i.e., different conditions of UPL. We can also observe that URM significantly improves upon ListRank by ca. 2-5% in the cases of UPL as 20 and 50. Although the improvement over ListRank in the case of UPL as 10 is not statistically significant, we emphasize that the tradeoff parameter used in the testing runs is based on the validation set, which is formed in the

Table 2.4: Performance comparison in terms of NDCG between URM and other recommendation approaches on EM dataset.

	UPL=10		UPL=20		UPL=50	
	NDCG@5	NDCG@10	NDCG@5	NDCG@10	NDCG@5	NDCG@10
ItemAvRat	0.236	0.307	0.222	0.291	0.194	0.255
ItemCF	0.534	0.579	0.545	0.592	0.552	0.598
PMF	0.608	0.643	0.606	0.646	0.690	0.714
CofiRank	0.639	0.646	0.671	0.653	0.641	0.647
ListRank	0.567	0.607	0.642	0.674	0.721	0.740
URM	0.668* [†]	0.695* [†]	0.707* [†]	0.726* [†]	0.735* [†]	0.747* [†]

condition of UPL=50. In practice, we could tune the tradeoff parameter more tightly by considering the targeting user profile length in order to attain further performance gain. In this chapter, we only tune tradeoff parameter based on a certain condition of UPL, which allows us to show that the tuned tradeoff parameter could be robust enough to be applied to other conditions of UPL.

For the performance of the URM on ML2 and EM datasets, which are much larger than ML1, similar observations can be found, as shown in Table 2.3 and Table 2.4. Note that on these datasets URM achieves significant improvement over all the other approaches in all the conditions of UPL. Compared to the second best approach in each case, the improvement attained by the URM is of ca. 2-10%. These results allow us to give a positive answer to our first research question, namely, they show that URM could improve recommendation performance over state-of-the-art approaches across different datasets and for users with different profiles. They also make it possible to give a positive answer to our fourth and final research question: Regarding the comparison of URM with other state-of-the-art approaches, the performance of URM is clearly and consistently superior.

2.5 Conclusion and Future Work

In this chapter, we present a novel recommendation approach URM, which is capable of unifying a ranking-oriented CF approach ListRank and a rating-oriented CF approach PMF by exploiting common latent features of users and items. We qualitatively and quantitatively demonstrate that the complexity of URM is linear with the number of observed ratings in a given user-item matrix, indicating that URM can be deployed in large-scale use cases. We also experimentally verify that the recommendation performance of URM mainly derives from the ranking-oriented component, i.e., ListRank, while the rating-oriented component, i.e., PMF, contributes significant enhancement. Our experimen-

tal results indicate that URM substantially outperforms both component approaches, i.e., ListRank and PMF, and other traditional and state-of-the-art recommendation approaches. Performance improvements achieved by URM are also shown to be consistent with respect to various datasets and users with various profile lengths.

Moving forward, future work in this area will explore two interesting directions. First, we are interested in investigating other options of item-list representation, which might influence the performance of the ranking-oriented recommendation approach, thus, improve the performance of URM. Second, in this paper we established that the latent space can mediate between the rating-oriented approach and the ranking-oriented approach. We are interested in exploring the shared latent space to integrate in the framework of URM with other types of information, e.g., item content features, contextual information of users and items. Third, we are also interested in investigating the potential to develop recommendation models by directly optimizing the ranking measures.

Chapter 3

Collaborative Less-is-More Filtering

In this chapter we tackle the problem of recommendation in the scenarios with binary relevance data, when only a few (k) items are recommended to individual users. Past work on Collaborative Filtering (CF) has either not addressed the ranking problem for binary relevance datasets, or not specifically focused on improving top- k recommendations. To solve the problem we propose a new CF approach, *Collaborative Less-is-More Filtering (CLiMF)*. In *CLiMF* the model parameters are learned by directly maximizing the Mean Reciprocal Rank (MRR), which is a well-known information retrieval metric for measuring the performance of top- k recommendations. We achieve linear computational complexity by introducing a lower bound of the smoothed reciprocal rank metric. Experiments on two social network datasets demonstrate the effectiveness and the scalability of *CLiMF*, and show that *CLiMF* significantly outperforms a naive baseline and two state-of-the-art CF methods.

This work has been published as “CLiMF: Learning to maximize reciprocal rank with collaborative less-is-more filtering” by Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, N. Oliver, and A. Hanjalic, in Proc. of the sixth ACM conference on Recommender systems, Dublin, Ireland, 2012 [141]. The paper won the Best Paper Award in the conference.

3.1 Introduction

Collaborative Filtering (CF) [2] methods are at the core of most recommendation engines in online web-stores and social networks. The main underlying idea behind CF methods is that users that shared common interests in the past would still prefer similar products/items in the future [129]. While a lot of the CF literature has been devoted to recommendation scenarios where explicit user feedback is present (i.e., typically ratings), CF has also shown to be very valuable in scenarios with only implicit feedback data [60], e.g., the counts of a user watching a TV show, the counts of a user listening to songs of an artist. These counts can be interpreted as a measure of preference and thus a proxy to explicit feedback.

However, in some scenarios even the “count” information is not available, while only binary relevance data exists, e.g., the friendship between users in a Online Social Network, the follow relationship between users (or between a user and an event, etc.) in Twitter¹ or the dating history in online dating sites [122]. Specifically, in these scenarios, we use “1” for a given user-item pair to denote that the user has an interaction (e.g., friendship, follow) with the item, and “0” otherwise. Typically the observed interactions are regarded as positive signals (i.e., indicating relevant items), and although not all items without observed interactions are irrelevant it is safe to assume the vast majority of these items will be irrelevant for the user. In other words, for a given user, the signal “0” indicates an item set containing unobserved items that could be relevant, but are most likely irrelevant. One of the most typical CF methods for those scenarios is item-based CF [38, 88], in which an item-item similarity matrix is first computed, and users are recommended items that are most similar to their past relevant items. However, item-based CF approaches typically require expensive computations in order to construct the similarity matrix. They are thus not a sound solution for large scale scenarios.

Bayesian Personalized Ranking (BPR) [126] has been recently proposed as a state-of-the-art recommendation algorithm for situations with binary relevance data. The optimization criterion of BPR is essentially based on pair-wise comparisons between relevant and a sample of irrelevant items. This criterion leads to the optimization of the Area Under the Curve (AUC). However, the AUC measure does not reflect well the quality of the recommendation lists, since it is not a top-biased measure [193], i.e., the position at which the pairwise comparisons are made is irrelevant to the contribution to the loss: mistakes at the lower ranked positions are penalized equally to mistakes in higher ranked positions, which is not the desired behavior in a ranked list.

¹<http://twitter.com/>

In view of the drawbacks of previous work, we propose a new CF approach, *Collaborative Less-is More Filtering (CLiMF)*, that is tailored to recommendation domains where only binary relevance data is available. *CLiMF* models the data by means of directly optimizing the Mean Reciprocal Rank (MRR) [175], a well-known evaluation metric in Information Retrieval (IR). Given the analogy between query-document search and user-item recommendation, we can define the Reciprocal Rank (RR) for a given recommendation list of a user, by measuring how early in the list (i.e. how highly ranked) the first relevant recommended item is ranked. The MRR is the average of the RR across all the recommendation lists for individual users. MRR is a particularly important measure of recommendation quality for domains that usually provide users with only few but valuable recommendations (i.e., the *less-is-more* effect [27]), such as friends recommendation in social networks where top-3 or top-5 performance is important.

Taking insights from the area of learning to rank and integrating latent factor models from CF, *CLiMF* directly optimizes a lower bound of the smoothed RR for learning the model parameters, i.e., latent factors of users and items, which are then used to generate item recommendations for individual users.

Our contributions in this chapter can be summarized as:

- We present a new CF approach, *CLiMF*, for MRR optimization for scenarios with binary relevance data. We demonstrate that *CLiMF* outperforms other state-of-the-art approaches with respect to making recommendations that are few in number, but relevant.
- We introduce a lower bound of the smoothed RR measure, significantly reducing the computational complexity of RR optimization, and enabling *CLiMF* to scale for large datasets.

The paper is organized as follows. In Section 3.2 we discuss the related work and position our paper with respect to it. Section 3.3 presents in detail the proposed *CLiMF* model. Our experimental evaluation is described in Section 3.4, followed by a summary and conclusions in Section 3.5.

3.2 Related Work

The work presented in this chapter closely relates to the research on ranking-oriented CF and learning to rank. In the following, we briefly review related work.

3.2.1 Ranking-oriented CF

A large portion of the Recommender Systems literature has been devoted to the rating prediction problem, as defined in the Netflix prize competition². Latent factor models and in particular Matrix Factorization (MF) techniques, have been shown to be particularly effective [3, 75, 134] for this problem. The main idea underlying MF is to extract latent factor U_i, V_j vectors for each user and item in the dataset so that the inner product of these factors $f_{ij} = \langle U_i, V_j \rangle$ fits the observed ratings.

Several state-of-the-art ranking-oriented CF approaches, that extend upon MF techniques, have been recently proposed. These approaches typically use a ranking oriented objective function to learn the latent factors of users and items, e.g., CofiRank [182], collaborative competitive filtering (CCF) [189], and OrdRec [76]. The *CLiMF* model presented in this chapter can also be regarded as an extension to conventional MF, while it introduces several new characteristics that are presented in Section 3.3.4, compared to the state-of-the-art.

A ranking-oriented CF that extends memory-based (or similarity-based) approaches has been proposed in EigenRank [90]. Moreover, extensions to probabilistic latent semantic analysis [58] that optimize a ranking objective have been proposed in pLPA [92]. However, these methods are all designed for recommendation scenarios with explicit graded relevance scores from users to items.

For the use scenarios with only implicit feedback data, one of the first model-based methods was introduced in [60], where an extension of MF is proposed by weighting each factorization of user-item interaction proportionately to the count of the interactions. A similar approach, one-class collaborative filtering [114], was also proposed to exploit weighting schemes for the factorizations of missing data, which are taken as non-positive examples. However, the computational cost of that work could be inflated due to the large number of non-positive data. In this chapter, we study the problem of generating recommendations for the scenarios with only binary relevance data, i.e., where even the count of user-item interaction is not available. In addition, our work directly takes into account an evaluation metric, MRR, when developing the recommendation model, which is also substantially different from the work of [60, 114].

The most similar work to ours is Bayesian personalized ranking (BPR) [126], since it also optimizes a ranking loss (AUC) and deals with binary relevance data. The main benefits of using *CLiMF* lies in its performance in terms of top- k recommendations (i.e., the fraction of relevant items at the top k positions of the list), an issue not addressed by the BPR model. Note that we also leave the detailed discussion of the relationship between *CLiMF* and BPR to

²<http://www.netflixprize.com/>

Section 3.3.4, after the presentation of the *CLiMF* model.

3.2.2 Learning to Rank

Learning to Rank (LTR) has been an active research topic in Machine Learning, Information Retrieval [93] and Recommender Systems [9, 144, 182]. The work in this chapter is closely related to one branch of LTR that focuses on direct optimization of IR metrics, for which the main difficulty lies in their non-smoothness with respect to the predicted relevance scores [17]. The approaches proposed in this branch of LTR approximate the optimization of IR measures either by minimizing convex upper bounds of loss functions that are based on the evaluation measures [24, 182, 187], e.g., *SVM^{MAP}* [193], or by optimizing a smoothed version of an evaluation measure, e.g., *SoftRank* [169] and *generalized SoftRank* [26].

In this chapter, we also propose to approximate the Mean Reciprocal Rank (MRR) with a smoothed function. However, our work is different from aforementioned work not only in that we target the application scenario of recommendation rather than query-document search, but also in that we propose an algorithm (CLiMF) that makes the optimization of the smoothed MRR tractable and scalable. We also provide insights about the ability of *CLiMF* to recommend relevant items in the top positions of a recommendation list.

3.3 CLiMF

In this section, we present the *CLiMF*, Collaborative Less-is-More Filtering, algorithm. We first introduce a smoothed version of Reciprocal Rank by building on insights from the area of learning to rank. Then, we derive a lower bound of the smoothed reciprocal rank, and formulate an objective function for which standard optimization methods can be deployed. Finally, we discuss the characteristics of the proposed *CLiMF* model and its relation to other state-of-the-art recommendation models.

3.3.1 Smoothing the Reciprocal Rank

The definition of reciprocal rank of a ranked list for user i , as defined in information retrieval [175], can be expressed as:

$$RR_i = \sum_{j=1}^N \frac{Y_{ij}}{R_{ij}} \prod_{k=1}^N (1 - Y_{ik} \mathbb{I}(R_{ik} < R_{ij})) \quad (3.1)$$

in which N is the number of items, Y_{ij} denotes the binary relevance score of item j to user i , i.e., $Y_{ij} = 1$ if item j is relevant to user i , 0 otherwise. $\mathbb{I}(x)$ is an indicator function that is equal to 1, if x is true, otherwise 0. R_{ij} denotes the rank of item j in the ranked list of items for user i . Note that the items are ranked in a descending order according to their predicted relevance scores for user i . Clearly, RR_i is dependent on the rankings of relevant items. The rankings of the relevant items change in a non-smooth way as a function of the predicted relevance scores and thus, RR_i is a non-smooth function over the model parameters. The non-smoothness of the RR measure makes it impossible to use standard optimization methods –such as gradient-based methods– to directly optimize RR_i . Inspired by recent developments in the area of learning to rank [26], we derive an approximation of $\mathbb{I}(R_{ik} < R_{ij})$ by using a logistic function:

$$\mathbb{I}(R_{ik} < R_{ij}) \approx g(f_{ik} - f_{ij}) \quad (3.2)$$

where $g(x) = 1/(1 + e^{-x})$, f_{ij} denotes the predictor function that maps the parameters from user i and item j to a predicted relevance score. The predictor function that we use in our model is the basic and widely-used factor model, expressed as:

$$f_{ij} = \langle U_i, V_j \rangle \quad (3.3)$$

where U_i denotes a d -dimensional latent factor vector for user i , and V_j a d -dimensional latent factor vector for item j . Even though a sophisticated approximation for the item rank was proposed in [26], it has not been deployed in practice. Notice that in the case of RR_i in Eq. (3.1), only $1/R_{ij}$ is actually in use. We thus propose to directly approximate $1/R_{ij}$ by another logistic function:

$$\frac{1}{R_{ij}} \approx g(f_{ij}) \quad (3.4)$$

which makes the basic assumption that the lower the item rank, the higher the predicted relevance score, i.e., $1/R_{ij}$ would approach to 1. Substituting Eq. (3.2) and (3.4) into Eq. (3.1), we obtain a smooth version of RR_i :

$$RR_i \approx \sum_{j=1}^N Y_{ij} g(f_{ij}) \prod_{k=1}^N (1 - Y_{ik} g(f_{ik} - f_{ij})) \quad (3.5)$$

Notice that although Eq. (3.5) is a smooth function with respect to the predicted relevance scores and thus the model parameters U and V , optimizing this function could still be practically intractable, due to its multiplicative nature. For example, the complexity of the gradient of Eq. (3.5) with respect to V_j (i.e., only for one item) is $O(N^2)$: the computational cost grows quadratically with the number of items N and for most recommender systems N is typically large. In the following, we present a lower bound of an equivalent variant of Eq. (3.5), for which we derive a computationally tractable optimization procedure.

3.3.2 Lower Bound of Smooth Reciprocal Rank

Suppose that the number of relevant items for user i in the given data collection is n_i^+ . Given the monotonicity of the logarithm function, the model parameters that maximize Eq. (3.5) are equivalent to the parameters that maximize $\ln(\frac{1}{n_i^+}RR_i)$. Specifically, we have:

$$\begin{aligned} U_i, V &= \arg \max_{U_i, V} \{RR_i\} = \arg \max_{U_i, V} \left\{ \ln \left(\frac{1}{n_i^+} RR_i \right) \right\} \\ &= \arg \max_{U_i, V} \left\{ \ln \left(\sum_{j=1}^N \frac{Y_{ij}}{n_i^+} g(f_{ij}) \prod_{k=1}^N (1 - Y_{ik} g(f_{ik} - f_{ij})) \right) \right\} \end{aligned} \quad (3.6)$$

Based on Jensen's inequality and the concavity of the logarithm function, we derive the lower bound of $\ln(\frac{1}{n_i^+}RR_i)$ as below:

$$\begin{aligned} \ln \left(\frac{1}{n_i^+} RR_i \right) &= \ln \left(\sum_{j=1}^N \frac{Y_{ij}}{\sum_{l=1}^N Y_{il}} g(f_{ij}) \prod_{k=1}^N (1 - Y_{ik} g(f_{ik} - f_{ij})) \right) \\ &\geq \frac{1}{n_i^+} \sum_{j=1}^N Y_{ij} \ln \left(g(f_{ij}) \prod_{k=1}^N (1 - Y_{ik} g(f_{ik} - f_{ij})) \right) \\ &= \frac{1}{n_i^+} \sum_{j=1}^N Y_{ij} \left(\ln g(f_{ij}) + \sum_{k=1}^N \ln (1 - Y_{ik} g(f_{ik} - f_{ij})) \right) \end{aligned} \quad (3.7)$$

Note that in the derivation above we make use of the definition of n_i^+ , i.e., $n_i^+ = \sum_{l=1}^N Y_{il}$. We can neglect the constant $1/n_i^+$ in the lower bound, and obtain a new objective function as:

$$L(U_i, V) = \sum_{j=1}^N Y_{ij} \left[\ln g(f_{ij}) + \sum_{k=1}^N \ln (1 - Y_{ik} g(f_{ik} - f_{ij})) \right] \quad (3.8)$$

We can take a close look at the two terms within the first summation. The maximization of the first term contributes to learning latent factors that promote relevant items. However, given one relevant item, e.g., item j , maximizing the second term contributes to learning latent factors of all the other items (e.g., item k) in order to degrade their relevance scores. In sum, the two effects come together to promote and scatter the relevant items at the same time, the main characteristic of the proposed *CLiMF*. In other words, *CLiMF* will lead to a recommendation where some but not all relevant items are at the very top of the recommendation list for a user. We notice that this behavior of *CLiMF* corresponds to the analysis of MRR optimization for a search result list [179], i.e., optimizing MRR results in diversifying ranked documents.

Taking into account the regularization terms that usually serve to control the complexity of the model (i.e. in order to avoid overfitting), and all the M users

in the given data collection, we obtain the objective function of *CLiMF*:

$$F(U, V) = \sum_{i=1}^M \sum_{j=1}^N Y_{ij} [\ln g(U_i^T V_j) + \sum_{k=1}^N \ln (1 - Y_{ik} g(U_i^T V_k - U_i^T V_j))] - \frac{\lambda}{2} (\|U\|^2 + \|V\|^2) \quad (3.9)$$

in which λ denotes the regularization coefficient, and $\|U\|$ denotes the Frobenius norm of U . Note that the lower bound $F(U, V)$ is much less complex than the original objective function in Eq. (3.5), and standard optimization methods, e.g., gradient ascend, can be used to learn the optimal model parameters U and V .

3.3.3 Optimization

We use stochastic gradient ascent to maximize the objective function in Eq. (3.9), i.e., for each user i , we optimize $F(U_i, V)$. The gradients of the objective for user i with respect to U_i and V_j can be computed as below:

$$\frac{\partial F}{\partial U_i} = \sum_{j=1}^N Y_{ij} [g(-f_{ij})V_j + \sum_{k=1}^N \frac{Y_{ik} g'(f_{ik} - f_{ij})}{1 - Y_{ik} g(f_{ik} - f_{ij})} (V_j - V_k)] - \lambda U_i \quad (3.10)$$

$$\begin{aligned} \frac{\partial F}{\partial V_j} = & Y_{ij} [g(-f_{ij}) + \sum_{k=1}^N Y_{ik} g'(f_{ij} - f_{ik}) \left(\frac{1}{1 - Y_{ik} g(f_{ik} - f_{ij})} - \frac{1}{1 - Y_{ij} g(f_{ij} - f_{ik})} \right)] U_i \\ & - \lambda V_j \end{aligned} \quad (3.11)$$

where $g'(x)$ denotes the derivative of $g(x)$. Note that we have used a property of $g(x)$, namely, $g(-x) = g'(x)/g(x)$, in the derivation of Eq. (3.10) and (3.11) above to simplify the computation.

The learning algorithm for the *CLiMF* model is outlined in Algorithm 2. We analyze the complexity of the learning process for one iteration. By exploiting the data sparseness in Y , the computational complexity of the gradient in Eq. (3.10) is $O(d\tilde{n}^2M + dM)$. Note that \tilde{n} denotes the average number of relevant items across all the users. The complexity of computing the gradient in Eq. (3.11) is $O(d\tilde{n}^2M + d\tilde{n}M)$. Hence, the complexity of the learning algorithm in one iteration is in the order of $O(d\tilde{n}^2M)$. In the case that \tilde{n} is a small number, i.e., $\tilde{n}^2 \ll M$, the complexity is linear to the number of users in the data collection. Note that we have $\tilde{n}M = S$, in which S denotes the number of non-zeros in the user-item matrix. The complexity of the learning algorithm is then $O(d\tilde{n}S)$. Since we usually have $\tilde{n} \ll S$, the complexity is $O(dS)$ even in the case that \tilde{n} is large, i.e., being linear to the number of non-zeros (i.e., relevant observations in the data). In sum, our analysis shows that *CLiMF* is suitable for large scale use cases. Note that we also empirically verify the complexity of the learning algorithm in Section 3.4.4.

ALGORITHM 2: Learning Algorithm for *CLiMF*

Input: Training set Y , regularization parameter λ , learning rate γ , and the maximal number of iterations $itermax$.**Output:** The learned latent factors U, V .**for** $i = 1, 2, \dots, M$ **do** % Index relevant items for user i ; $N_i = \{j | Y_{ij} > 0, 1 \leq j \leq N\}$;**end**Initialize $U^{(0)}$ and $V^{(0)}$ with random values, and $t = 0$;**repeat** **for** $i = 1, 2, \dots, M$ **do** % Update U_i ; $U_i^{(t+1)} = U_i^{(t)} + \gamma \frac{\partial F}{\partial U_i^{(t)}}$ based on Eq. (3.10); **for** $j \in N_i$ **do** % Update V_j ; $V_j^{(t+1)} = V_j^{(t)} + \gamma \frac{\partial F}{\partial V_j^{(t)}}$ based on Eq. (3.11); **end** **end** $t = t + 1$;**until** $t \geq itermax$; $U = U^{(t)}, V = V^{(t)}$

3.3.4 Discussion

We discuss the relationship between the proposed *CLiMF* and other state-of-the-art recommendation models, and present the insights that highlight the contribution of *CLiMF* to the area of CF when compared to other models.

Relation to CofiRank: CofiRank [182] was the first work that introduced learning to rank to address CF as a ranking problem. CofiRank makes use of structured estimation of a ranking loss based on NDCG, and learns the recommendation model by minimizing over a convex upper bound of the loss function. The major differences between *CLiMF* and CofiRank lie in two aspects: First, due to its foundation on the measure of NDCG, CofiRank suits scenarios where graded relevance data, e.g., ratings, are available from users to items, but it might not be appropriate for the scenarios with only binary relevance data, for which *CLiMF* is tailored. Second, CofiRank and *CLiMF* root in different classes of methods to achieve learning to rank [93, 187], such as the difference between *SVM^{M_{AP}}* [193] and *SoftRank* [169]. CofiRank exploits a convex upper bound of the structured loss function based on the evaluation

metric NDCG, and then optimizes the upper bound. However, *CLiMF* first smooths the evaluation metric RR, and then optimizes the smoothed version of the metric via a lower bound.

Relation to CCF: Collaborative competitive filtering (CCF) [189] was proposed as an algorithm that not only exploits rated items from users, but also the candidate items (or *opportunities*) that were available for the users to choose. The key constraint introduced in CCF is that the utility (or relevance) of a rated item should be higher than any items that are in the candidate set but not rated/selected. *CLiMF* is similar to CCF in the sense that it also considers the relative pair-wise constraints in learning the latent factors, as shown in the second term with the summation in Eq. (3.8). However, *CLiMF* only requires relevant items, while CCF requires all the items in the candidate set, which are not usually available. In practice, CCF needs to include some unrated items together with the rated items to form the candidate set. In addition, CCF is not directly related to any evaluation metrics, while *CLiMF* is designed for MRR optimization.

Relation to OrdRec: OrdRec [76] is an ordinal model that formulates the probability that a rating predictor (a function of the model parameters, such as the latent factors) is equal to a known rating as the probability that the rating predictor falls in the interval of two parameterized scale thresholds corresponding to two adjacent rating values. OrdRec has a point-wise nature, i.e., it does not require any pair-wise computation between any rated/unrated items. Hence, it enjoys the advantage of a computational complexity that is linear to the data size, the same advantage attained by *CLiMF*. However, although OrdRec generally suits to scenarios with implicit feedback data, “count” information is necessary to extract the ordinals, i.e., the ordered preferences of users. For this reason, OrdRec may not be suitable for the scenarios with only binary relevance data. In addition, OrdRec has no direct relation to the ranking-oriented evaluation metrics.

Relation to BPR: BPR [126] models the pair-wise comparisons between positive and negative feedback data (in the scenarios with binary relevance data), and optimizes an objective that corresponds to Area Under Curve (AUC) optimization. BPR is similar to *CLiMF* in the sense that it also directly optimizes a smoothed version of an evaluation metric for binary relevance data, there are though two main differences. First, BPR requires a sampled set of negative feedback data, i.e., a set of unobserved items to be assumed as irrelevant to the users. However, *CLiMF* only requires the relevant items from the users. Second, while BPR aims at promoting all the relevant items, *CLiMF* particularly focuses on recommending items that are few in number, but relevant at top- k positions of the recommendation list, a goal which is attained by promoting and scattering relevant items at the same time, as shown in Eq. (3.8). Since

BPR shares a close relationship with *CLiMF* in terms of modeling and application scenarios, we choose BPR as the main baseline to compare against in the experiments.

3.4 Experimental Evaluation

In this section we present a series of experiments to evaluate *CLiMF*. We first describe the datasets used in the experiments and the setup. Then, we compare the recommendation performance of *CLiMF* with two baseline approaches in terms of providing only a few but relevant recommendations at the top positions of the recommendation list. Finally, we analyze the effectiveness and the scalability of the proposed *CLiMF* model.

We designed the experiments in order to address the following research questions:

1. Does the proposed *CLiMF* outperform alternative state-of-the-art algorithms, particularly when recommending just a few but relevant items at top-ranked positions?
2. Is the learning algorithm of *CLiMF* effective for increasing MRR to a local maximum?
3. Is *CLiMF* scalable for large-scale use cases?

3.4.1 Experimental Setup

Datasets. We conduct experiments using two social network datasets from Epinions³ and Tuenti⁴. The Epinions dataset is publicly available⁵, and it contains trust relationships between 49288 users. The Epinions dataset represents a directed social network, i.e., if user i is a trustee of user j , user j is not necessary a trustee of user i . Most microblogging social networks are also directed, such as Twitter. For the purpose of our experiments, we exclude from the dataset the users who have less than 25 trustees. The second dataset collected from Tuenti, one of the largest social networks in Spain, represents an undirected social network, containing friendship between 50K users. Similar to the Epinions dataset, we also exclude the users with less than 25 friends. Note that in these two datasets, friends or trustees are regarded as “items” of users. The task is to generate friend or trustee recommendations for individual users. Statistics on the two datasets used in our experiments are summarized in Table 3.1.

³<http://www.epinions.com>

⁴<http://www.tuenti.com>

⁵http://www.trustlet.org/wiki/Downloaded_Epinions_dataset

Table 3.1: Statistics of the datasets.

Dataset	Epinions	Tuenti
Num. non-zeros	346035	798158
Num. users	4718	11392
Num. friends/trustees	49288	50000
Sparseness	99.85%	99.86%
Avg. friends/trustees per user	73.34	70.06

Experimental Protocol and Evaluation Metrics. We separate each dataset into a training set and a test set under various conditions of user profiles. For example, the condition of “Given 5” denotes that for each user we randomly selected 5 out of her trustees/friends to form the training set, and use the remaining trustees/friends to form the test set. The task is to use the training set to generate recommendation lists for individual users, and the performance is measured according to the holdout data in the test set. We repeat the experiment 5 times for each of the different conditions of each dataset, and the performances reported are averaged across 5 runs. Again, we emphasize that in this work we only consider the observed items as being relevant to the user. Although this setting would underestimate the power of all the recommenders, the comparative results are still useful, since they can be regarded as the approximation of the lower limit of each recommender.

The main evaluation metric that we use in our experiments to measure the recommendation performance is MRR, the measure that is optimized in our model. In addition, we also measure the performance by precision at top-ranked items, such as precision at top-5 (P@5), which reflects the ratio of the number of relevant items in the top-5 recommended items. In order to emphasize the value of “less-is-more” recommendations, we also use the measure of 1-call at top-ranked items [27]. Specifically, 1-call at top-5 recommendations (1-call@5) reflects the ratio of test users who have at least one relevant item in their top-5 recommendation lists.

Finally, as revealed in recent studies from different recommender domains, it is possible that popular items could heavily dominate the recommendation performance [33, 149, 161]. We also notice this effect in our experiments, namely, recommending the most popular friends or trustees (i.e., those have the most friends or trusters) could already result in a high performance. For this reason, in our experiments we consider the top three most popular items as being irrelevant in order to reduce the influence from the most trivial recommendations [33, 149]. In other words, recommending any of the top three popular friends/ trustees has no contribution to any of the evaluation metrics.

Parameter Setting. We use one fold of randomly generated training-test sets of each dataset under the condition “Given 5” for the purpose of validation,

Table 3.2: Performance comparison of *CLiMF* and baselines on the Epinions dataset.

	Given 5			Given 10			Given 15			Given 20		
	MRR	P@5	1-call@5	MRR	P@5	1-call@5	MRR	P@5	1-call@5	MRR	P@5	1-call@5
PopRec	0.142	0.035	0.166	0.127	0.032	0.134	0.117	0.032	0.136	0.131	0.048	0.210
iMF	0.154	0.059	0.225	0.143	0.059	0.236	0.155	0.063	0.231	0.153	0.059	0.226
BPR-MF	0.241	0.148	0.532	0.167	0.072	0.334	0.177	0.098	0.380	0.216	0.096	0.422
CLiMF	0.292	0.216	0.676	0.233	0.092	0.392	0.248	0.127	0.496	0.239	0.110	0.448

which is used to tune parameters in CLiMF. The values of the parameters that yield the best performance on the validation set are: the regularization parameter $\lambda = 0.001$, the latent dimensionality $d = 10$ and the learning rate $\gamma = 0.0001$.

3.4.2 Performance Comparison

We compare the performance of *CLiMF* with three baselines, PopRec, iMF and BPR, which are described below:

- **PopRec.** A naive baseline that recommends a user to be a friend or trustee in terms of her popularity, i.e., the number of friends or trusters she has in the given training set. The more friends or trusters the user has, the higher her position in the recommendation list. Note that it is a non-personalized recommendation approach: for any target user, the recommendations are always the same.
- **iMF:** A state-of-the-art matrix factorization technique for implicit feedback data by Hu et al. [60], as discussed in Section 3.2. The regularization parameter is tuned to 1, based on the performance on the validation sets.
- **BPR-MF.** Bayesian personalized ranking (BPR) represents the state-of-the-art optimization framework of CF for binary relevance data [126]. BPR-MF represents the choice of using matrix factorization (MF) as the learning model with BPR optimization criterion. Note that the implementation of this baseline is done with the publicly available software MyMediaLite [42]. The relevant parameters, such as the regularization coefficients and the number of iterations, are tuned on the validation sets, which are the same sets that were used for tuning the *CLiMF* model.

The recommendation performances of *CLiMF* and the baseline approaches on the Epinions and the Tuenti datasets are shown in Table 3.2 and Table 3.3, respectively.

Three main observations can be drawn from the results: First, the proposed *CLiMF* model *significantly* outperforms the three baselines in terms of MRR

Table 3.3: Performance comparison of *CLiMF* and baselines on the Tuenti dataset.

	Given 5			Given 10			Given 15			Given 20		
	MRR	P@5	1-call@5	MRR	P@5	1-call@5	MRR	P@5	1-call@5	MRR	P@5	1-call@5
PopRec	0.096	0.029	0.138	0.074	0.017	0.080	0.074	0.019	0.088	0.074	0.019	0.086
iMF	0.064	0.020	0.090	0.065	0.017	0.076	0.065	0.021	0.098	0.076	0.023	0.108
BPR-MF	0.096	0.030	0.142	0.075	0.025	0.116	0.075	0.020	0.090	0.076	0.021	0.106
CLiMF	0.100	0.039	0.190	0.077	0.027	0.124	0.077	0.022	0.104	0.083	0.024	0.116

across all the conditions and the two datasets. Note that in our experiments, the statistical significance is measured based on the results from individual test users, according to a Wilcoxon signed rank significance test with $p < 0.01$. This result corroborates that *CLiMF* achieves the goal that was designed for and optimizes the value of the reciprocal rank for the recommendations to the individual users. Notice that it is not possible to compare the results in Table 3.2 and Table 3.3 across conditions, since different conditions involve a different set of test items, containing different numbers of items. Second, *CLiMF* also achieves a *significant* improvement over the baselines in terms of P@5 and 1-call@5 across all the conditions and the two datasets. The improvement of P@5 indicates that by optimizing MRR, *CLiMF* also improve the quality of recommendations among the top-ranked items. In addition, the improvement of 1-call@5 supports that *CLiMF* particularly contributes to providing valuable recommendations at the top- k positions, i.e., raising the chance that users would receive at least one relevant recommendation among just a few top-ranked items. Compared to BPR, where AUC is optimized, *CLiMF* succeeds in enhancing the top-ranked performance by optimizing MRR, the top-biased metric. As can be also seen from the results, iMF performs worse than both BPR and *CLiMF* in all the conditions of the Epinions dataset and in most of the conditions of the Tuenti dataset. The reason might be that iMF is particularly designed for implicit feedback datasets with the “count” information as mentioned in Section 3.2, while it may not be suitable for the scenarios with only binary relevance data. Third, in cases in which users have a lower number of friends/trustees (i.e., the case of “Given 5”) the improvement achieved by *CLiMF* over the alternative approaches is relatively larger than the improvement achieved in cases in which users have a higher number of friends/trustees (i.e., the case of “Given 20”). This result suggests that *CLiMF*’s key mechanism of scattering relevant items could be particularly beneficial for scenarios under very high data sparseness. Hence, we give a positive answer to our first research question.

3.4.3 Effectiveness

The second experiment investigates the effectiveness of the proposed learning algorithm for *CLiMF*, as presented in Section 3.3.3. Figures 3.1 (a) and (b)

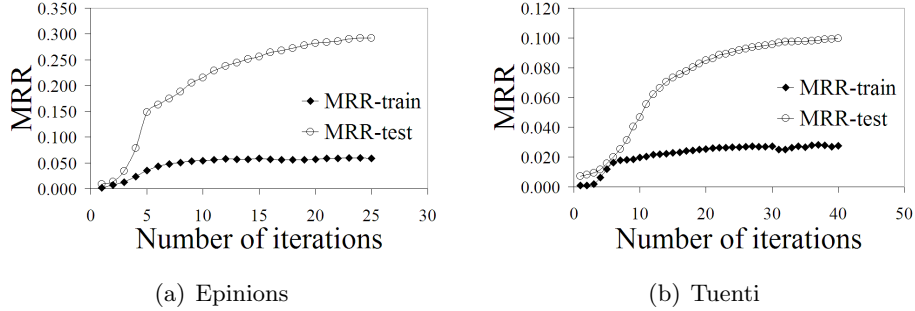


Figure 3.1: Effectiveness of the learning algorithm for *CLiMF* under the “Given 5” condition for both datasets.

show the evolution of MRR with each iteration –as measured in both the training and the test sets– under the “Given 5” condition for the Epinions and Tuenti datasets, respectively. We can see that both MRR measures gradually increase with each iteration and convergence is reached after a few iterations, i.e., nearly after 20 iterations on the Epinions dataset and 30 iterations on the Tuenti dataset. This observation indicates that *CLiMF* effectively learns from the training set latent factors of users and items that optimize reciprocal rank, which consequently also contributes to improving MRR in the test set. With this experimental result, we give a positive answer to our second research question.

3.4.4 Scalability

The last experiment investigates the scalability of *CLiMF*, by measuring the training time that is required for the training set at different scales. First, as analyzed in Section 3.3.3, the computational complexity of *CLiMF* is linear in the number of users in the training set when the average number of friends/trustees per user is fixed. To demonstrate the scalability, we use different numbers of users in the training set under each condition: we randomly select from 10% to 100% users in the training set and their known friends/trustees as the training data for learning the latent factors. The results on the Epinions dataset and the Tuenti dataset are shown in Fig. 3.2(a) and 3.2(b), respectively. We can observe that for both datasets, the computational time under each condition increases almost linearly to the increase of the number of users. Second, as also discussed in Section 3.3.3, the computational complexity of *CLiMF* could be further approximated to be linear to the amount of known data (i.e., non-zero entries in the training user-item matrix). To demonstrate this, we examine the runtime of the learning algorithm against different scales of the training sets under different “Given” conditions. For example, under the “Given 5” condi-

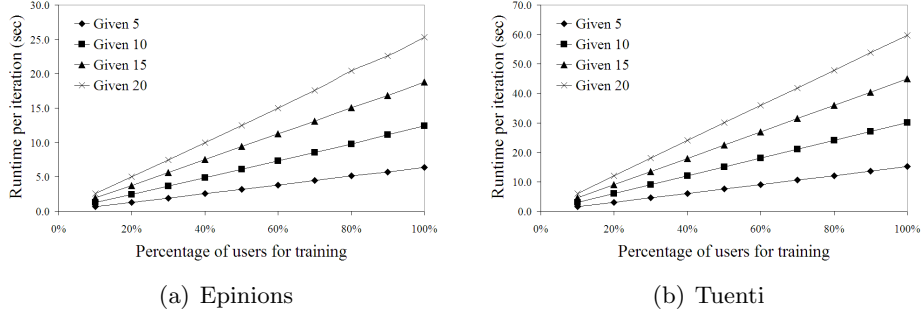


Figure 3.2: Scalability analysis of *CLiMF* in terms of the number of users in the training set

tion of the Epinions dataset, there are $5 \times 4718 = 23590$ non-zeros in the training set. The result is shown in Fig. 3.2, from which we can observe that the average runtime of the learning algorithm per iteration increases almost linearly as the number of non-zeros in the training set increases. The observations from this experiment allow us to answer our last research question positively.

3.5 Conclusions

In this chapter we have presented a new CF approach, *CLiMF*, that learns latent factors of users and items by directly maximizing MRR. *CLiMF* is designed to improve the performance of top- k recommendations for usage scenarios with only binary relevance data. We have demonstrated in our experiments that *CLiMF* offers significant improvements over a naive and two state-of-the-art baselines in two social network datasets. We have also experimentally validated that *CLiMF*'s learning algorithm is effective for MRR optimization, and has linear computational complexity to the size of the known data, and thus is scalable for large scale use cases.

Future work involves a few interesting directions. First, we would like to extend our *CLiMF* model to suit domains with explicit feedback data, e.g., ratings. Second, it is also interesting to experimentally investigate the impact of *CLiMF* on the recommendation diversity, by exploiting external information resources, such as the categories of items. Third, we are also interested in investigating recommendation models that optimize other evaluation measures, such as mean average precision [140], and in exploring the impact of optimizing different measures on various aspects of recommendation performance [179].

Chapter 4

Mood-specific Movie Recommendation

Context-aware recommendation seeks to improve recommendation performance by exploiting various information sources in addition to the conventional user-item matrix used by recommender systems. We propose a novel context-aware movie recommendation algorithm based on joint matrix factorization (JMF). We jointly factorize the user-item matrix containing general movie ratings and other contextual movie similarity matrices to integrate contextual information into the recommendation process. The algorithm was developed within the scope of the mood-aware recommendation task that was offered by the Moviepilot mood track of the 2010 context-aware movie recommendation (CAMRa) challenge. Although the algorithm could generalize to other types of contextual information, in this work, we focus on two: movie mood tags and movie plot keywords. Since the objective in this challenge track is to recommend movies for a user given a specified mood, we devise a novel mood-specific movie similarity measure for this purpose. We enhance the recommendation based on this measure by also deploying the second movie similarity measure proposed in this chapter that takes into account the movie plot keywords. We validate the effectiveness of the proposed JMF algorithm with respect to the recommendation performance by carrying out experiments on the Moviepilot challenge data set. We demonstrate that exploiting contextual information in JMF leads to

This work was first published as “Mining mood-specific movie similarity with matrix factorization for context-aware recommendation” by Y. Shi, M. Larson, and A. Hanjalic, in ACM RecSys Challenge on Context-aware Movie Recommendation, 2010 (CAMRa 2010). The paper won the Overall Winner Award in the challenge [145]. This chapter is an extended version that has been published as “Mining contextual movie similarity with matrix factorization for context-aware recommendation” in ACM Transactions on Intelligent Systems and Technology, 4(1), 2013 [148].

significant improvement over several state-of-the-art approaches that generate movie recommendations without using contextual information. We also demonstrate that our proposed mood-specific movie similarity is better suited for the task than the conventional mood-based movie similarity measures. Finally, we show that the enhancement provided by the movie similarity capturing the plot keywords is particularly helpful in improving the recommendation to those users who are significantly more active in rating the movies than other users.

4.1 Introduction

Recently, context-aware recommendation has experienced an upsurge of interest in the recommender systems community [131]. The interest has been spurred by a growing awareness of the potential of contextual information, if available, to improve the quality of recommendations [1, 18]. Such information can include, e.g., relationships among users in social media sites, tags of products, introduction about products or timestamps of user actions. One of the most promising potential contributions of contextual information is its ability to alleviate the problem of data sparseness in the original user-item matrix. Contextual information can be exploited to more reliably estimate relationships between items compensating for cases in which the information in the original user-item matrix is insufficient.

In addition to relying on information sources beyond the conventional user-item matrix, context-aware recommendation also differs from traditional recommendation in the sense that its purpose is more specific, e.g., movies are recommended specifically for the week of Christmas [43, 89], or for a specific mood that they should elicit in the user [145]. Accordingly, the research challenge in the area of context-aware recommendation involves two aspects. It is expected that the new recommendation technique/model retains the benefits of conventional recommender approaches, such as collaborative filtering (CF) [2], which infers the recommendation from a user-item matrix, while also allowing contextual information to steer the recommendation process towards results suitable for a given use case (purpose).

In this chapter, we address the recommendation task formulated in the Moviepilot mood track of the context-aware movie recommendation (CAMRa) challenge [131], henceforth referred to as the Moviepilot challenge, in which the task is to recommend movies to a user given a specific mood. For this purpose, not only the user-item matrix is provided, capturing general preferences of the users for different movies, but also the contextual information consisting of various movie metadata, such as mood tags, plot keywords, movie locations and intended audience.

In order to maximize the benefit of the given contextual information and optimally address the two aspects of context-aware recommendation defined above, we propose a novel recommendation model, based on a joint matrix factorization (JMF), that factorizes the user-item (user-movie) matrix, while also exploiting the contextual information as additional regularization terms. Specifically, for generating context-based links between movies, we propose a set of contextual movie similarities, each of which steers the recommendation process and contributes to JMF in a specific fashion. From the contextual information available within the setting of our task, we deploy mood tags and plot keywords (PK).

The potential of plot keywords to improve mood-based recommendation lies in the assumption that if mood-based movie similarity is difficult to infer reliably from the mood tags, then movies with similar moods might still be linked together if they have similar plots. Although we focus only on two types of contextual information, the proposed JMF model could be easily expanded to integrate other contextual information.

The novel contribution of this chapter can be summarized as follows. We propose a novel context-aware movie recommendation algorithm that extends the basic matrix factorization (MF) model to take into account context-induced links between movies. Furthermore, we propose a set of contextual movie similarities that evaluate the relationships between movies in view of specific contextual information, i.e., mood-specific movie similarity and PK-based movie similarity, and integrate these similarities in our recommendation model. Finally, we apply these in the setting of the Moviepilot challenge for evaluation, demonstrating that the proposed algorithm outperforms a wide range of state-of-the-art approaches for context-aware recommendation.

The remainder of the paper is structured as follows. In the next section, we present an overview of the Moviepilot challenge. Then, in section 4.3, we summarize related work and position our approach with respect to it. The proposed contextual movie similarities and JMF model are described in detail in section 4.4, after which, in section 4.5, we present experimental evaluation on the Moviepilot challenge dataset. The last section sums up the key aspects of the proposed algorithm and briefly addresses the direction for future work.

4.2 Overview of the Moviepilot Challenge

4.2.1 Problem Statement

The task of the Moviepilot challenge can be defined as follows: *Based on both the user-movie rating matrix and other provided contextual information, recommend a list of movies that have a specific mood property to each target/test user* [131]. In other words, the task is to design a model that takes the user-movie rating matrix, the contextual information and a specific mood as input, and outputs a list of movies with the specific mood for each target user. The recommendation list should contain as many relevant movies as possible, which are also ranked as high as possible. Note that within this challenge a movie is considered to be “relevant” to a user if it has been rated by that user and if it is characterized by the pre-specified mood. In addition to user-movie ratings, the data set provided for this challenge contains various types of contextual information, e.g., movie-emotion (mood) assignments, movie-PK assignments

and the release date of each movie. Note that all the data are provided in the form of identifiers and the real identities of the underlying entities are not made known to the public due to privacy considerations. For example, we do not have access to direct knowledge concerning the identity of the movie with identifier 10, or the mood with identifier 5.

4.2.2 Characteristics of the Challenge

In this section we discuss the characteristics of this challenge that make it distinctive from the traditional CF problem.

First, the evaluation of the recommendation performance is not based on the rating prediction error rate, e.g., as done in the Netflix contest (www.netflixprize.com), but on the recommendation list, the quality of which is assessed using metrics for the evaluation of ranked results lists, e.g., precision at N and mean average precision. As recently suggested in [Liu et al. 2009; Shi et al. 2010b], these evaluation criteria are more sensible, since the ultimate goal of a recommender system is to generate a list of recommended items for a user, rather than only provide the predictions of relevance scores for different items.

Second, the evaluation focuses on the recommended movies characterized by the pre-specified mood, which requires a different approach to designing recommendation mechanisms, compared to the traditional, mainly CF-based practice. In view of such a focus, one could namely first apply a known recommendation approach based on an analysis of the user-item matrix to generate the initial recommendation and then remove those movies not having the desired mood. This approach is, however, likely lead to a recommendation performance being far short of what is targeted, as is demonstrated later during our experiments. A priori, a basis for this expectation could also be drawn from the distribution of ratings in the data set, shown in Fig. 4.1. The distribution reveals that many movies in the provided validation set (i.e., a set provided for tuning the parameters) have low ratings. Note that low-rated movies are significant in the Moviepilot challenge, since if the number of movies of the target mood is limited, these movies come into consideration as the most appropriate ones to recommend to the user. As a consequence of this discrepancy between the rating predictions generated by the user-item matrix and the contextual information, the number of movies resulting from an initial traditional recommendation step and having the proper mood characteristics may be too small. Instead, one should conceptualize the recommendation process as not only involving the movies that the user would generally be interested in, but also simultaneously emphasizing the movies with the specified mood. In other words, context awareness of the recommendation is not likely to emerge from

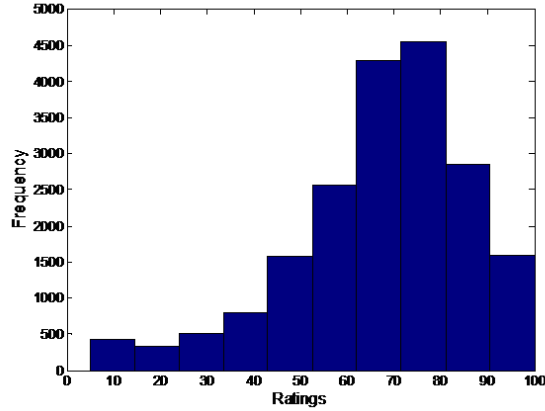


Figure 4.1: The distribution of ratings in the validation set of Moviepilot challenge.

a simple filtering step of the initial recommended item list, but rather through letting the contextual information actively steer the recommendation process.

Finally, other alternatives for approaching collaborative recommendation, such as the ranking-oriented approaches [92, 126, 144], are also expected not to perform well within this challenge, since both a pair-wise ranking approach and a list-wise ranking approach require training examples annotated with explicit or implicit item ratings. However, the value of ratings would not influence the evaluation according to the challenge setting.

4.3 Related Work

This section briefly summarizes the existing related research in CF, context-aware recommendation and tag-aware recommendation, in order to position the recommendation approach we propose in this chapter.

4.3.1 Collaborative Filtering

CF approaches are usually categorized as memory-based or model-based [2, 16, 57]. Memory-based approaches can be further categorized as user-based or item-based, depending on whether the recommendation for a user is aggregated from users with similar preference, e.g., the work of [55, 142, 194], or from items that are similar to those he already liked, e.g., the work of [38, 88, 136]. The key drawback of memory-based CF approaches lies in the expensive computation for similarities among all users or items, which does not scale with the typically large numbers of users and items in real-world recommender sys-

tems. Compared to memory-based approaches, model-based approaches first employ statistical and machine learning techniques to learn a prediction model from a training set of user-item matrix data and then apply that model to generate recommendations, such as Gaussian mixture model [66] and latent semantic model [58]. Among different model-based approaches, matrix factorization (MF) techniques have attracted much research attention, due to the advantages of scalability and accuracy [75, 134], especially for large-scale data, as exemplified by the Netflix contest. Generally, MF techniques learn latent features of users and items from the observed ratings in the user-item rating matrix. These latent features are further used to predict unobserved ratings. Rather than targeting the rating prediction problem, recent research started to exploit possibilities for ranking-oriented CF approaches that focus on the quality of recommendation lists, e.g., EigenRank [90], CoFiRank [182], probabilistic latent preference analysis [92], Bayesian personalized ranking [126], and ListRank [144].

The joint matrix factorization (JMF) is an extension of MF. The designation “joint” makes reference to the simultaneous factorization of more than one matrix. JMF is formulated similarly to relational learning, as defined by [155], which also factorizes multiple matrices from related domains. It has been widely applied e.g., for fusing document content and graph link information for document retrieval or web page classification [37, 202], or for fusing geographical location features and people activity correlation for location-based recommendation [197]. In this chapter, we exploit JMF to fuse contextual movie similarities, i.e., the mood-specific movie similarity and the plot-keyword-based movie similarity, with the user-movie rating matrix. The difference between our work and the aforementioned previous work on JMF is two-fold. First, compared to the work of [37, 155, 202], we exploit the available contextual information in the form of contextual movie similarities rather than original movie contextual information. By this means, we maintain the advantage of using contextual information for learning latent movie features, namely alleviating data sparseness in the user-movie rating matrix, while at the same time eliminating unnecessary learning for additional latent features representing other entities, e.g., movie mood and plot keywords. Second, above and beyond this body of existing work, i.e., [37, 155, 197, 202], we propose a new mood-specific movie similarity that explicitly addresses the recommendation bias of the Moviepilot challenge and propose a method to integrate this similarity into the JMF framework. We note that our work is consistent with the contemporary trend of new work in the area, falling into the category of approaches extending matrix factorization [43, 89].

4.3.2 Context-aware Recommendation

Some of the earliest work on context-aware recommendation was done by Adomavicius et al. [1], who deployed contextual information, e.g., time and place, to generate additional dimensions in the user and item rating vectors. Anand and Mobasher [6] exploited users' preference information from both long-term memory and short-term memory in the recommendation process. Baltrunas and Ricci [11] proposed to take into account the context, e.g., user age and gender, to split item ratings as a pre-processing step for CF approaches, in order to improve CF accuracy. Application of context-aware recommendation approaches in specific use cases has been explored in work exploiting contextual information for travel recommendation [23], news recommendation [21], and music recommendation [163]. Compared to this previous work, our proposed approach has the advantages of being adaptive to any application domain and being able to handle large-scale data sets.

Specifically concerning the context-aware movie recommendation tasks in the Moviepilot challenge, in addition to the mood-specific recommendation task other tasks have been addressed as well, such as recommending movies for a specific week (such as a holiday week) and recommending movies by exploiting social relationships [131]. For week-based movie recommendation, timestamps of users ratings have been made available as the contextual information. Approaching this task, [43] extended a pair-wise interaction tensor factorization model [128], which was originally designed for tag recommendation, to factorize the {user, time, movie} ternary data for movie recommendation in a given week. [89] investigated both tensor factorization and sequential matrix factorization to integrate time-dependent characteristics of users and items into the recommendation process. In addition to the work presented in the Moviepilot challenge, [74] has proposed to include temporal dynamics into neighbor-based CF and matrix factorization for improved performance in the Netflix contest.

On the other hand, regarding the task targeting the integration of social relationships into recommendation, [89] investigated both collective matrix factorization (equivalent to the work of [98]) and network-regularized matrix factorization (equivalent to the work of [97]) for this purpose. However, they found that including social relationships into MF leads to only a small improvement compared to basic matrix factorization.

4.3.3 Tag-aware Recommendation

We also point out that our work on context-aware recommendation is related to tag-aware recommendation. [170] proposed a fusion method to incorporate tags into traditional user-based CF and item-based CF for item rating prediction.

[184] proposed exploiting probabilistic latent semantic analysis [58] to unify user-item relations and item-tag relations into one model, resulting in improved item recommendation performance. A similar principle was presented by [178], but from a more fundamental perspective.

More recently, another group of state-of-the-art approaches has emerged that makes use of tensor factorization techniques, e.g., the work of [70]. Under such approaches, latent features are learned from the {user, tag, item} triplet/ternary data directly for item recommendation [65], tag recommendation [125, 128, 165], or both [166]. However, tensor factorization is known to be computationally expensive, i.e., usually being cubic in the number of latent dimensions. However, in the Moviepilot challenge, where the mood tags and plot keywords of movies are not associated with users, there is no ternary data actually available for either exploiting or comparing with tensor factorization techniques.

Another way of benefiting from the relations among users, tags and items for the purpose of recommendation is to deploy a graph-based approach. For instance, [72] proposed a recommendation framework that infers the item preferences of the users from a hyper-graph including different types of nodes and links, which captures user-user, tag-user, tag-item and user-item relations. Preferences are inferred using a random-walk-with-restarts method. While the method was proved effective conceptually, it requires the availability of rich contextual information in order to result in significant recommendation benefits in a practical use case. Graph-based approaches are not suitable for our task, which mixes binary information (mood-movie relation) with scale information (user ratings). In [72], the links in the hyper-graph are assumed to be binary, which represents a radical simplification of the relationships between the nodes. For integrating user-item ratings and other contextual information into a hyper-graph, it would also be necessary not only to impose this simplification, but also to impose it in a way that retains the balance between the mix of different information types without information loss, which could be introduced during the simplification step. For this reason, graph-based approaches are not an obvious choice for application in our work.

4.4 The Proposed Algorithm

In this section, we introduce our proposed algorithm for the context-aware movie recommendation task of the Moviepilot challenge. The flow chart of the proposed algorithm is given in Fig. 4.2. While the use case and rationale behind our general approach have been discussed in Section 4.1, we focus in this section on the analysis of three key components of the algorithm, namely

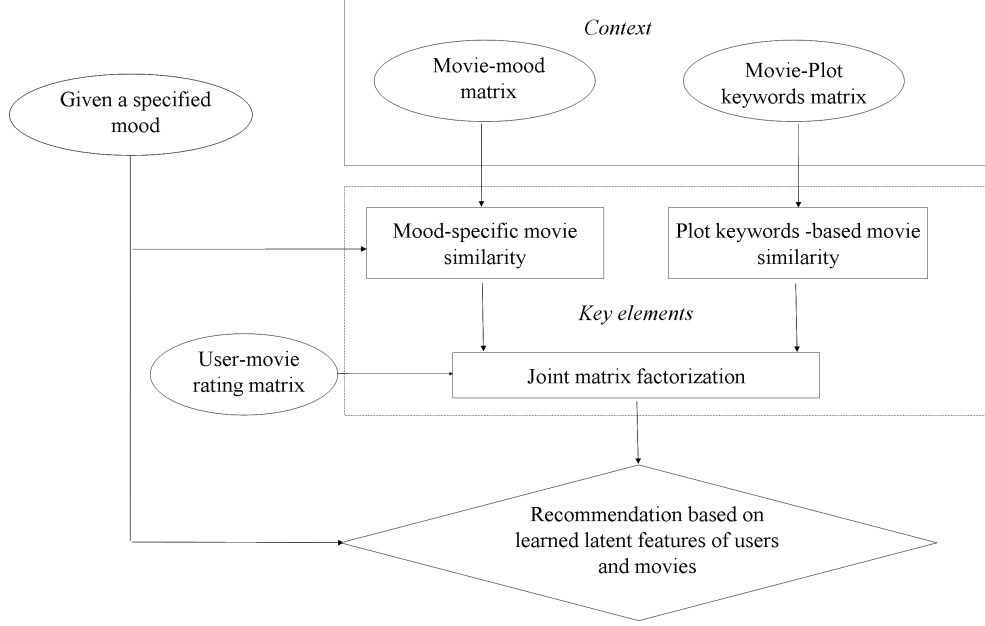


Figure 4.2: The flow chart of the proposed algorithm.

the mood-specific movie similarity and the similarity based on plot keywords, and the joint matrix factorization model. Then, we also perform a complexity analysis of the proposed algorithm.

4.4.1 Mood-specific Movie Similarity

According to the traditional item-based CF, item-to-item similarity can be computed as the cosine similarity between two item rating vectors [38]. Similarly, given the movie-mood (binary) matrix \mathbf{M} (consisting of N movies and K_1 mood tags), we can compute mood-based similarity between movie j and movie n as:

$$S_{jn}^{(Mov-mood)} = \frac{\sum_{k=1}^{K_1} M_{jk} M_{nk}}{\sqrt{\sum_{k=1}^{K_1} M_{jk}^2} \sqrt{\sum_{k=1}^{K_1} M_{nk}^2}} \quad (4.1)$$

Here, $M_{jk} = 1$ indicates that the movie j has the mood tag k , otherwise $M_{jk} = 0$. The mood-based similarity in Eq. (4.1), however, only indicates general closeness of two movies in terms of all their mood properties. For example as shown in Fig. 4.3, two movies (**A** and **B**) sharing different mood properties could be equally similar to another movie (**D**). If the required mood of a movie is specified, this similarity fails to differentiate between movies **A** and **B**.

In view of the above, we expect that an accurate approach would involve ad-

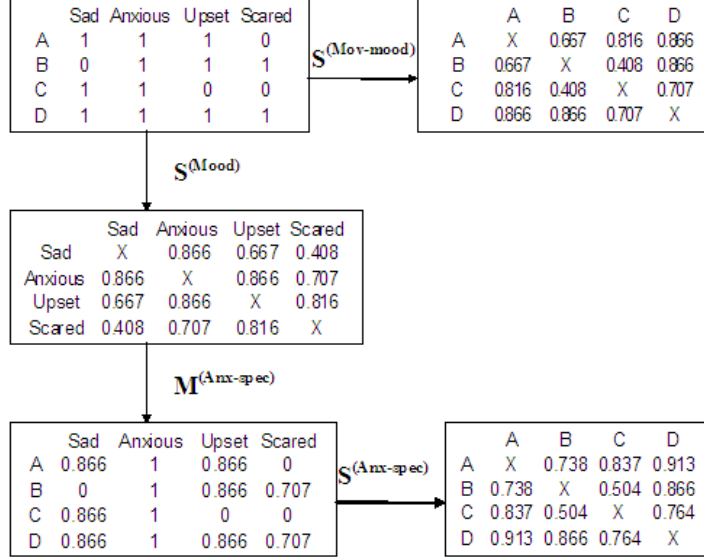


Figure 4.3: An illustrative example of mood-specific movie similarity.

justing the movie-to-movie similarity in a way to make it more biased towards a particular mood in case movies with that mood need to be recommended. We therefore propose a novel mood-specific similarity measure to address this issue. Although we focus on the Moviepilot challenge that targets mood-specific movie recommendation, the concept of mood-specific similarity can easily be generalized for recommendation with different contextual constraints.

Instead of evaluating the consistency of mood tags between two movies, like in Eq. (4.1), we can also compute the normalized co-occurrence of the mood i and mood k in the movie collection as:

$$S_{ik}^{(Mood)} = \frac{\sum_{j=1}^N M_{ji} M_{jk}}{\sqrt{\sum_{j=1}^N M_{ji}^2} \sqrt{\sum_{j=1}^N M_{jk}^2}} \quad (4.2)$$

Once the mood co-occurrence matrix $S^{(Mood)}$ is obtained, we can generate a mood-specific movie-mood matrix that is biased towards a given mood m , as expressed in Eq. (4.3):

$$M_{jk}^{(m-spec)} = \begin{cases} S_{jk}^{(Mood)}, & k \neq m \\ M_{jk}, & k = m \end{cases} \quad (4.3)$$

While preserving the original movie-mood matrix values in the column corresponding to the mood m , the values in the matrix as in Eq. (4.3) for any other mood k are replaced by the values of the similarities in Eq. (4.2), implicitly indicating to which extent mood k is informative about mood m . Note that the

mood-specific movie-mood matrix $\mathbf{M}^{(m\text{-spec})}$ is not binary. Then, we define the mood-specific movie similarity biased towards a given mood m as:

$$S_{jn}^{(m\text{-spec})} = \frac{\sum_{k=1}^{K_1} M_{jk}^{(m\text{-spec})} M_{nk}^{(m\text{-spec})}}{\sqrt{\sum_{k=1}^{K_1} M_{jk}^{(m\text{-spec})^2}} \sqrt{\sum_{k=1}^{K_1} M_{nk}^{(m\text{-spec})^2}}} \quad (4.4)$$

In order to illustrate the effect of this similarity, we again focus on the example in Fig. 4.3 and assume that there is specific demand for movies corresponding to the mood “anxious”. In the search for all movies satisfying this recommendation criterion, we can derive the mood-specific movie similarity for the mood “anxious”. In contrast with the case where general mood-based similarity in Eq. (4.1) is computed, the mood-specific movie similarity now indicates movie **D** to be more similar to movie **A** than to movie **B**. This is because movie **A** has mood tags (“sad” and “upset”) that are more informative about “anxious” than the mood tags of the movie **B**. In this way, the movie-specific movie similarity steers the movie comparison towards the target mood and helps the context-aware recommendation.

4.4.2 Plot Keyword -based Movie Similarity

Similar to the mood-based movie similarity, we also define the similarity between movies in terms of movie plot keywords (PKs). Since PKs represent the movie content, this similarity can improve the mood-based links between movies. Since both PKs and moods potentially reflect the underlying movie content, it is reasonable to expect that movies having similar plots could evoke similar emotions in users. We first create a binary movie-PK matrix \mathbf{P} consisting of N movies and K_2 PKs, where $P_{jk} = 1$ if the movie j has the PK k , and $P_{jk} = 0$ otherwise. Then, the PK-based similarity between movie j and movie n can be defined as:

$$S_{jn}^{(Mov-PK)} = \frac{\sum_{k=1}^{K_2} P_{jk} P_{nk}}{\sqrt{\sum_{k=1}^{K_2} P_{jk}^2} \sqrt{\sum_{k=1}^{K_2} P_{nk}^2}} \quad (4.5)$$

4.4.3 Joint Matrix Factorization

The basic MF [75] can be formulated as in Eq. 4.6:

$$U, V = \arg \min_{U, V} \left\{ \frac{1}{2} \sum_{u=1}^K \sum_{j=1}^N I_{uj}^R (R_{uj} - U_u^T V_j)^2 + \frac{\lambda_U}{2} \|\mathbf{U}\|_F^2 + \frac{\lambda_V}{2} \|\mathbf{V}\|_F^2 \right\} \quad (4.6)$$

Given the user-item rating matrix \mathbf{R} consisting of K users and N items, the MF represents the user-item rating matrix \mathbf{R} by two low-rank matrices, \mathbf{U} and

\mathbf{V} . A d -dimensional set of latent features is deployed to represent both users (in \mathbf{U}) and items (in \mathbf{V}). Note that we use U_u to denote a column d -dimensional feature vector of user u , V_j is a column d -dimensional feature vector of movie j , and R_{uj} denotes the user u 's rating on movie j . I_{uj}^R denotes an indicator function that is equal to 1 when $R_{uj} > 0$, and 0 otherwise. $\|\mathbf{U}\|_F$ and $\|\mathbf{V}\|_F$ are the Frobenius norms of \mathbf{U} and \mathbf{V} , that contribute to alleviating overfitting. λ_U, λ_V are regularization parameters for which we set $\lambda_U = \lambda_V = \lambda$ to simplify the model in this chapter.

In view of the discussion in the previous section, we now require that the movies being similar to each other with respect to the mood-specific similarity criterion in Eq. (4.4) share similar latent movie features. For this purpose, we formulate a context-aware loss function $L_1(\mathbf{V})$ as shown in Eq. (4.7).

$$L_1(\mathbf{V}) = \frac{1}{2} \sum_{j=1}^N \sum_{n=1}^N I_{jn}^{MS} \left(S_{jn}^{(m-spec)} - V_j^T V_n \right)^2 \quad (4.7)$$

where I_{jn}^{MS} denotes an indicator function that is equal to 1 when $S_{jn}^{(m-spec)} > 0$, and 0 otherwise.

Furthermore, we also assume that the movies similar to each other with respect to the PK-based similarity as in Eq. (4.5) should also share similar latent movie features, implying that the similarity of the plots is informative for mood-specific movie recommendation. Therefore, we formulate another context-aware loss function $L_2(\mathbf{V})$ as shown in Eq. (4.8).

$$L_2(\mathbf{V}) = \frac{1}{2} \sum_{j=1}^N \sum_{n=1}^N I_{jn}^{PK} \left(S_{jn}^{(Mov-PK)} - V_j^T V_n \right)^2 \quad (4.8)$$

Here, I_{jn}^{PK} denotes an indicator function that is equal to 1 when $S_{jn}^{(Mov-PK)} > 0$, and 0 otherwise.

Taking into account the context-aware loss functions as regularization terms in the basic MF model, a joint matrix factorization (JMF) model can be formulated as:

$$\begin{aligned} L(\mathbf{U}, \mathbf{V}) &= \frac{1}{2} \sum_{u=1}^K \sum_{j=1}^N I_{uj}^R (R_{uj} - U_u^T V_j)^2 \\ &\quad + \frac{\alpha}{2} \sum_{j=1}^N \sum_{n=1}^N I_{jn}^{MS} \left(S_{jn}^{(m-spec)} - V_j^T V_n \right)^2 \\ &\quad + \frac{\beta}{2} \sum_{j=1}^N \sum_{n=1}^N I_{jn}^{PK} \left(S_{jn}^{(Mov-PK)} - V_j^T V_n \right)^2 + \frac{\lambda}{2} (\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2) \end{aligned} \quad (4.9)$$

In this model, α and β are the tradeoff parameters for weighting the contributions of regularization by the mood-specific movie similarity and by the

PK-based movie similarity, respectively. Note that when both $\alpha = 0$ and $\beta = 0$, the JMF model converges to the basic MF model. For notation convenience, we use JMF-MS-PK to indicate the case that both $\alpha > 0$ and $\beta > 0$ and JMF-MS to indicate the model with $\alpha > 0$ and $\beta = 0$. Furthermore, we use JMF-MB to indicate the model with $\alpha > 0$ and $\beta = 0$ that exploits general mood-based movie similarity instead of the mood-specific movie similarity. A more elaborate investigation of the variations of the proposed JMF model is given in Section 5.

In summary, the advantage of the JMF model is two-fold: First, the additional contextual information, i.e., mood-specific movie similarity and PK-based movie similarity, alleviate the usual deficiencies of the rating matrix \mathbf{R} , i.e., data sparseness, since latent features of movies could be learned from contextual movie similarity matrices as well as from the rating matrix. Second, the mood-specific movie similarity could contribute to biasing recommendations towards movies with the specific mood.

Minimization of the objective function in Eq. (4.9) can be solved by gradient descend with alternatively fixed \mathbf{U} and \mathbf{V} . This process results in a local minimum solution. The gradients of $L(\mathbf{U}, \mathbf{V})$ with respect to \mathbf{U} and \mathbf{V} can be computed as:

$$\frac{\partial L}{\partial U_u} = \sum_{j=1}^N I_{uj}^R (U_u^T V_j - R_{uj}) V_j + \lambda U_u \quad (4.10)$$

$$\begin{aligned} \frac{\partial L}{\partial V_j} = & \sum_{u=1}^K I_{uj}^R (U_u^T V_j - R_{uj}) U_u + 2\alpha \sum_{n=1}^N I_{jn}^{MS} (V_j^T V_n - S_{jn}^{(m-spec)}) V_n \\ & + 2\beta \sum_{n=1}^N I_{jn}^{PK} (V_j^T V_n - S_{jn}^{(Mov-PK)}) V_n + \lambda V_j \end{aligned} \quad (4.11)$$

Note that in Eq. (4.11) we exploit the symmetry of $\mathbf{S}^{(m-spec)}$ and $\mathbf{S}^{(Mov-PK)}$. The JMF-MS-PK algorithm is described in detail in Algorithm 3.

4.4.4 Complexity Analysis

The complexity of computing the contextual movie similarity matrices is normally quadratic to the number of movies, i.e., $O(N^2)$. In the case that a new movie appears, the complexity of updating each contextual movie similarity matrix is linear in the number of current movies, i.e., $O(N)$. However, this computation could be carried out completely offline, since it is independent of learning latent features in JMF. By exploiting the data sparseness, the computation of the objective function in Eq. (4.9) is of complexity $O(d|\mathbf{R}| + d|\mathbf{S}^{(m-spec)}| + d|\mathbf{S}^{(Mov-PK)}| + d(K + N))$, where $|\mathbf{R}|$ denotes the

ALGORITHM 3: JMF-MS-PK

Input: User-movie rating matrix \mathbf{R} , mood-specific movie-to-movie similarity $\mathbf{S}^{(\mathbf{m}\text{-spec})}$, PK-based movie-to-movie similarity $\mathbf{S}^{(\mathbf{Mov}\text{-PK})}$, tradeoff parameters α, β , regularization parameter λ , stop condition ϵ .

Output: Complete user-movie relevance matrix \hat{R} .

Initialize $\mathbf{U}^{(0)}, \mathbf{V}^{(0)}$ with random values;

$t = 0$;

$f = 0$;

Compute $L^{(t)}$ as in Eq. (4.9);

repeat

$\eta = 1$;

 Compute $\frac{\partial L}{\partial U^{(t)}}, \frac{\partial L}{\partial V^{(t)}}$ as in Eq. (4.10) and (4.11);

repeat

$\eta = \eta/2$; // maximize learning step size

until $L(U^{(t)} - \eta \frac{\partial L}{\partial U^{(t)}}, V^{(t)} - \eta \frac{\partial L}{\partial V^{(t)}}) < L^{(t)}$;

$U^{(t+1)} = U^{(t)} - \eta \frac{\partial L}{\partial U^{(t)}}, V^{(t+1)} = V^{(t)} - \eta \frac{\partial L}{\partial V^{(t)}}$;

 Compute $L^{(t+1)}$ as in Eq. (4.9);

if $1 - L^{(t+1)}/L^{(t)} \leq \epsilon$ **then**

$f = 1$; // indicator of convergence

end

$t = t + 1$;

until $f = 1$;

$\hat{R} = U^{(t)T}V^{(t)}$;

number of observed ratings in a given user-movie rating matrix, $|\mathbf{S}^{(\mathbf{m}\text{-spec})}|$ the number of non-zero similarities in the mood-specific similarity matrix, and $|\mathbf{S}^{(\mathbf{Mov}\text{-PK})}|$ the number of non-zero similarities in the PK-based similarity matrix. The complexity of the gradients in Eq. (4.10) and (4.11) is $O(d|\mathbf{R}| + dK)$ and $O(d|\mathbf{R}| + d|\mathbf{S}^{(\mathbf{m}\text{-spec})}| + d|\mathbf{S}^{(\mathbf{Mov}\text{-PK})}| + dN)$, respectively. Considering the fact that we often have $|\mathbf{R}| \gg K, N$, i.e., the number of observed ratings is much larger than both the number of users and the number of movies in a collection, the total complexity in one iteration is $O(d|\mathbf{R}| + d|\mathbf{S}^{(\mathbf{m}\text{-spec})}| + d|\mathbf{S}^{(\mathbf{Mov}\text{-PK})}|)$. In practice, there are many more users than movies in movie recommender systems, e.g., the Netflix data set involves around 480K users and around 18K movies (www.netflixprize.com), and the MovieLens data set involves around 72K users and around 11K movies (www.grouplens.org/node/73). Therefore, the number of contextual links between movies could be much lower than user-movie links (ratings), leading to $|\mathbf{S}^{(\mathbf{m}\text{-spec})}|, |\mathbf{S}^{(\mathbf{Mov}\text{-PK})}| \ll |\mathbf{R}|$. The total complexity of the proposed algorithm could approximate to $O(d|\mathbf{R}|)$, which is linear with the number of observed ratings in the user-movie matrix. This analysis indicates that the proposed algorithm is computationally efficient and can be applied to large-scale cases.

4.5 Experimental Evaluation

In this section, we present the experiments we conducted to evaluate the proposed algorithm. The research questions that need to be answered through the experiments can be formulated as follows:

1. Does minimizing the objective function in Eq. (4.9) contribute to improving recommendation performance?
2. Can the proposed algorithm JMF-MS-PK outperform other state-of-the-art approaches?
3. How does the mood-specific movie similarity contribute to the performance of mood-specific recommendation in the Moviepilot challenge?
4. What is the contribution of integrating the PK-based movie similarity in addition to mood-specific movie similarity to the recommendation performance?

4.5.1 Experimental Setup

Data set. Our experiments are conducted on the dataset of the “Moviepilot mood track”, which consists of around 4.5M ratings (scale 0-100) assigned by around 105K users to a collection of around 25K movies. The data sparseness of the user-movie rating matrix is around 99.83%. Apart from the user-movie rating matrix, various contextual information is provided, e.g., gender and age of users, production year of movies, intended audience of movies, etc. The detail of statistics of the dataset is presented at [131]. As mentioned in the introduction, we only exploit the mood tags of movies and the plot keywords of movies in this work. The movie-mood tag (binary) matrix consists of around 25K movies and 16 mood tags, which in total involves 6712 mood tag assignments on movies. The movie-PK (binary) matrix consists of around 25K movies and 5683 PKs, which in total involves 92124 PK assignments to movies.

Evaluation Metrics. We use the precision of top-N recommendation list (P@N) and the Mean Average Precision (MAP) as the evaluation metrics for measuring the quality of the recommendation list [50, 57]. The P@N reflects the average ratio of the number of relevant movies over the top-N recommended movies for all test users. The definition of MAP is given as:

$$MAP = \frac{1}{K_{ts}} \sum_{u=1}^{K_{ts}} \frac{\sum_{j=1}^{N_u} (rel_u(j) \times P_u@j)}{\sum_{j=1}^{N_u} rel_u(j)} \quad (4.12)$$

where K_{ts} is the number of users for testing, and N_u denotes the number of recommended movies for the user u . $rel_u(j)$ is a binary indicator, which

is equal to 1 if the movie of rank j is relevant to user u , and is equal to 0 otherwise. $P_u@j$ is the precision of the top j recommended movies for the user u , i.e., the ratio of movies in the top j recommendation that are relevant to the user u . Since it is required that relevant movies are recommended as early as possible, usually a small value of N is chosen for $P@N$. In our experiments, we evaluated in the cases of $N = 1, 5, 10$. In addition, MAP reflects the quality of the entire recommendation list by considering the positions of all the relevant movies. Higher values for $P@N$ and MAP indicate a better recommendation performance.

Experimental Protocol. The Moviepilot challenge data set contains three pre-defined subsets: a training set, a validation set and a test set. The training set involves all users and all movies. The training set is used to generate recommendations. The validation and test sets involve a small number of users, i.e., 160 and 80 users, respectively, whose ratings are disjoint with their ratings in the training set. The validation set is used to tune the parameters in the proposed algorithm. The parameters in the baseline approaches, as discussed in Section 4.5.4, are also tuned to the validation set. Performance is reported based on recommendations for all the users in the test set, as demonstrated in Section 4.5.4. Moreover, according to the requirement in the Moviepilot challenge (cf. Section 4.2), the evaluation only concerns the movies with a specific mood tag (i.e., the mood with identifier 16) to be potentially relevant for the users. Recommended movies that are in the validation/test set, but do not have the specified mood tag, are counted as irrelevant for the target users, i.e., they do not contribute to improvement in recommendation as reflected by the evaluation metrics.

Note that in the proposed JMF algorithm, we set the dimensionality of latent features to be 10. Although the variation of dimensionality of latent features could influence the performance, we notice that, just like in a common MF technique [183], a further increase in the number of latent features would not introduce a large improvement, while requiring more computational cost (cf. Section 4.4.4). The regularization parameter λ is set to 1 based on the observation of the performance of the basic matrix factorization, which is also discussed in Section 4.5.4. The stopping condition ϵ in the learning process is set to 0.0001 in our experiments.

4.5.2 Impact of Tradeoff Parameters

The tradeoff parameters α and β in the proposed algorithm influence the relative contributions from the contextual movie similarities. By using the validation set, we investigate the impact of the tradeoff parameters by varying their values and measuring the recommendation performance in terms of $P@5$ and MAP.

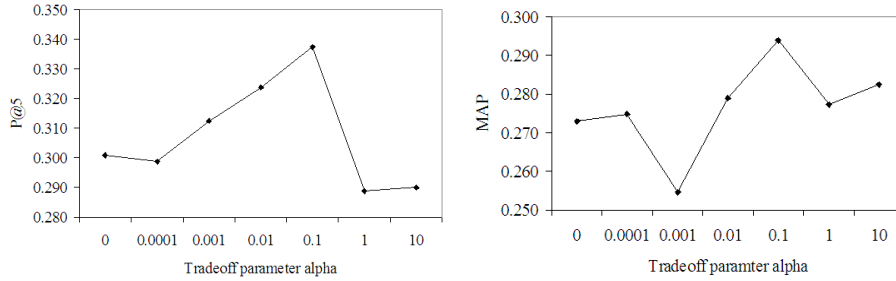


Figure 4.4: The impact of tradeoff parameter α on the recommendation performance of the proposed algorithm.

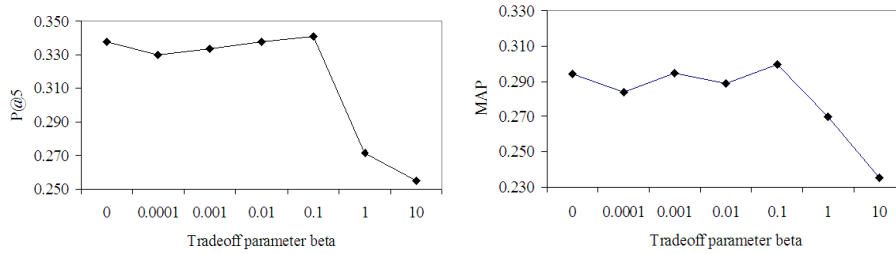


Figure 4.5: The impact of tradeoff parameter β on the recommendation performance of the proposed algorithm, when $\alpha = 0.1$.

We first set $\beta = 0$ and investigate the impact of α , as shown in Fig. 4.4. It can be seen that for both P@5 and MAP, the optimal value of α lies around 0.1. It also indicates that by only introducing the mood-specific movie similarity, additional improvement can be achieved over the basic MF model, i.e., the case when $\alpha = 0$. Then, we further investigate the impact of β , and keep the value of α fixed as $\alpha = 0.1$, as shown in Fig. 4.5. It can also be seen that for both P@5 and MAP, the optimal value of β is nearly 0.1. This indicates that in addition to the mood-specific movie similarity, there is still potential to further improve the recommendation performance by incorporating the PK-based movie similarity. Moreover, we can observe that the additional improvement stemming from the PK-based movie similarity is only slight compared to the case when only the mood-specific movie similarity is used, i.e., $\beta = 0$ in Fig. 4.5. This implies that the mood-specific movie similarity makes the major contribution among the used contextual information.

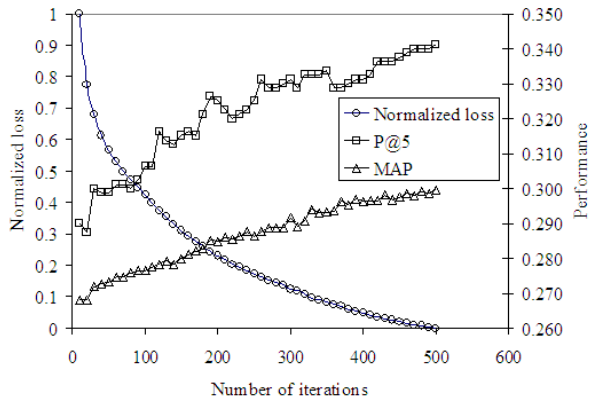


Figure 4.6: The effectiveness of the proposed algorithm in improving the recommendation performance.

4.5.3 Effectiveness

We further investigate the research question (1), namely to what extent the minimization of the objective function in Eq. (4.9) contributes to the improvement of the recommendation. To this end, we demonstrate the variation of the output of the objective function (normalized for demonstration purposes) and evaluation metrics, i.e., P@5 and MAP on the validation set, simultaneously during the iterations of the optimization process, as shown in Fig. 4.6. Note that in this experiment the tradeoff parameters used are the optimal values from the previous section, i.e., $\alpha = 0.1$ and $\beta = 0.1$. The results show that the recommendation performance generally increases monotonically with the minimization of the objective function, allowing us to give a positive answer to the first research question.

4.5.4 Performance Comparison

In this subsection, we compare the performance of the proposed algorithm JMF-MS-PK with a set of alternative recommendation approaches listed below. The performance is reported based on the test set. The tradeoff parameters are the optimal ones determined using the validation set as stated in Section 4.5.2.

- **PopRec:** Movies are recommended to users based on their popularity, which is defined in terms of the number of users who rated them in the training set. This approach constitutes a non-personalized and naive baseline since every test user will receive the same recommendation, i.e., a list of movies ranked in a descending order according to the number of their

Table 4.1: Comparison of recommendation performance between the proposed algorithm and other baseline approaches. “^” denotes a significant improvement of JMF-MS over JMF-MB, and “*” denotes a significant improvement of JMF-MS-PK over JMF-MS, according to Wilcoxon signed rank significance test with $p < 0.05$.

	P@1	P@5	P@10	MAP
PopRec	0.213	0.248	0.251	0.264
RWR	0.238	0.253	0.274	0.281
MF	0.325	0.305	0.241	0.252
JMF-MB	0.338	0.328	0.286	0.273
JMF-MS	0.350	0.335^	0.295^	0.289^
JMF-MS-PK	0.363	0.335	0.306*	0.290

ratings in the training set.

- **RWR**: This algorithm is a state-of-the-art recommendation approach that uses random walk with restarts (RWR) [72] on a graph encoding the relations between the users and items. Here, we set the restart probability to 0.8 based on the optimal performance achieved in the validation set.
- **MF**: This algorithm represents a basic state-of-the-art matrix factorization approach as in [75], which is also equivalent to Eq. (4.6). The dimensionality of the latent user and movie features is also set to 10. The regularization parameter λ is set to 1, which is tuned to achieve the optimal performance on the validation set. Note that the same corresponding parameters are used in the proposed algorithm as well.
- **JMF-MB**: This algorithm is a joint matrix factorization approach that incorporates the general mood-based movie similarity rather than mood-specific movie similarity. It also shares the same corresponding parameters as used in MF. In addition, we set the tradeoff parameter to 0.01 to again give the best performance on the validation set. The JMF-MB is used to compare with the proposed algorithm especially for validating the usefulness of the mood-specific movie similarity.
- **JMF-MS**: This algorithm is a joint matrix factorization approach that incorporates the mood-specific movie similarity as in Eq. (4.9). Note that in both JMF-MB and JMF-MS we do not use the PK-based movie similarity, i.e., $\beta = 0$.

The results of the comparative analysis are shown in Table 4.1, from which we can see the relative improvement achieved by the proposed algorithm JMF-MS-PK in terms of P@1, P@5, P@10 and MAP.

First, we can see that the JMF-MB approach outperforms non-context-aware approaches, i.e., PopRec, RWR, and MF, by over 5% in terms of P@1, P@5 and P@10. This improvement is statistically significant (based on Wilcoxon signed rank significance test, $p < 0.05$) in all cases and indicates that the contextual information indeed has the potential to help improve the recommendation performance. By incorporating both the mood-specific movie similarity and the PK-based movie similarity, the proposed algorithm JMF-MS-PK outperforms the non-context-aware approaches by over 10% across all the evaluation metrics, also with constant statistical significance. This gives an affirmative answer to the research question (2). In addition, the improvement of JMF over the basic MF approach also empirically indicates that exploiting contextual information via the JMF model could indeed contribute to alleviating data sparseness problem in the user-movie rating matrix.

Second, we observe that the JMF-MS achieves around 3%-5% improvement over the JMF-MB in most of the evaluation metrics, indicating that the mood-specific movie similarity is more beneficial for the mood-specific recommendation purpose compared to the general mood-based movie similarity. This observation provides an affirmative answer to the third research question.

Third, by further incorporating the PK-based movie similarity in addition to the mood-specific movie similarity, the JMF-MS-PK achieves around 3% significant improvement over the JMF-MS with respect to P@10, which results in an affirmative answer to the last research question. It can also be seen that the mood-specific movie similarity makes the major contribution under the mood-specific recommendation purpose, while other contextual movie similarities, e.g., the PK-based movie similarity, can be exploited via the JMF approach to make further enhancements of the recommendation performance.

Finally, we investigate the performance of the proposed algorithm with respect to users with varying numbers of rated movies, such as the results reported for P@10 in Table 4.2. Since the non-context-aware approaches perform generally worse than the proposed JMF algorithm as indicated before, we only select MF to represent the non-context-aware approaches. As can be seen from Table 4.2, the JMF algorithms outperform the basic MF across all users with various numbers of rated movies. We also notice that the users with relatively fewer rated movies (i.e., with no more than 100 rated movies) benefit most from the mood-specific movie similarity (i.e., from JMF-MS and JMF-MS-PK) compared to the general mood-based movie similarity (JMF-MB). We conjecture that the difference arises due to the following effect. Under mood-specific similarity less orthogonal pairs of movies exist in the set than under mood-based similarity. Effectively, the structure of the movie similarity space as defined by pair-wise relationships between movies is smoothed by the use of mood-specific similarity. Users who have rated large numbers of items do not benefit from this smoothing

Table 4.2: Comparison of P@10 performance between the proposed algorithm and other baseline approaches with respect to users with various numbers of rated movies.

Num. rated movies (Num. users)	MF	JMF-MB	JMF-MS	JMF-MS-PK
1~50 (19)	0.305	0.326	0.374	0.379
51~100 (16)	0.250	0.269	0.313	0.313
101~150 (13)	0.208	0.223	0.238	0.238
151~200 (12)	0.242	0.317	0.250	0.267
>200 (20)	0.195	0.285	0.270	0.300

since their profiles already contain enough information for reliable estimation of predictions. Users who have rated fewer items, however, effectively are able to cast a wider net if more pairs of movies are similar.

In real-world systems, users with limited numbers of rated movies are usually the majority in the community as reflected in the typically high data sparseness encountered by recommender systems. Our results indicate that exploiting the mood-specific movie similarity has the potential to benefit these users in particular. Compared to the JMF-MS, the JMF-MS-PK could be more beneficial for the users who rated relatively more movies. This observation may imply that the latent movie features that are learned from PK-based similarity can be better leveraged if there are more available ratings on those movies. If the user rates more movies, she could raise the chance that some of her rated movies are similar to other movies with respect to PK-based similarity, which could be recommended even if they are rated by few users overall within the collection. This observation also implies an affirmative answer to our last research question, i.e., the additional information introduced using PKs serves to improve recommendations in the case of users rating many movies.

4.6 Conclusion and Future Work

In this chapter, we present a novel context-aware recommendation algorithm that integrates contextual movie similarity, i.e., mood-specific movie similarity and PK-based movie similarity, together with the user-movie rating matrix into joint matrix factorization for the purpose of mood-specific movie recommendation, as defined in the Moviepilot challenge. The proposed algorithm is analyzed to be scalable for large-scale use cases. Our experiments at the Moviepilot challenge dataset show that the proposed algorithm outperforms several other state-of-the-art recommendation approaches. Substantial improvement can be achieved by exploiting contextual movie similarities, among which the mood-specific movie similarity is shown to make the major contribution to the recommendation performance and the PK-based movie similarity could fur-

ther enhance contribution. In addition, we specifically validate the usefulness of the mood-specific movie similarity compared to general mood-based movie similarity, which indeed leads to a substantial performance improvement. We also show the JMF with both mood-specific movie similarity and movie similarity in terms of plot keywords could be the most beneficial option for users across profiles containing different numbers of rated movies, compared to other variants.

We note that the algorithm proposed in this chapter could be generally applicable to other recommendation purposes, e.g., recommending movies with a specific actor, recommending music with a specific style. Exploiting contextual item similarity that is related to the specific recommendation purpose could be a way to make recommender systems context-aware.

Our future work will involve further exploration of the use of context-based information for recommendation. In particular, note that the Moviepilot challenge dataset was issued in an encoded form: we do not have direct knowledge of the identities of moods or plot keywords, rather these are represented within the data set as codes. In the future, we would like to experiment on a data set where we do have access to this information in order to compare our approach to approaches informed by external knowledge sources that could add explicit information on the relationship between the sources. We are also interested in moving beyond moods and plot keywords and understanding the suitability of our approaches for exploiting other sources of knowledge. In particular, we will address the question of what characteristics of a knowledge source must hold in order to be successfully exploited by our approach for the purpose of context-aware recommendation.

Chapter 5

Non-trivial Landmark Recommendation

Online photo-sharing sites provide a wealth of information about user behavior and their potential is increasing as it becomes ever-more common for images to be associated with location information in the form of geotags. In this paper, we propose a novel approach that exploits geotagged images from an online community for the purpose of personalized landmark recommendation. Under our formulation of the task, recommended landmarks should be relevant to user interests and additionally they should constitute non-trivial recommendations. In other words, recommendations of landmarks that are highly popular and frequently visited and can be easily discovered through other information sources such as travel guides should be avoided in favor of recommendations that relate to users' personal interests. We propose a collaborative filtering approach to the personalized landmark recommendation task within a matrix factorization framework. Our approach, WMF-CR, combines weighted matrix factorization and category-based regularization. The integrated weights emphasize the contribution of non-trivial landmarks in order to focus the recommendation model specifically on the generation of non-trivial recommendations. They support the judicious elimination of trivial landmarks from consideration without also discarding information valuable for recommendation. Category-based regularization addresses the sparse data problem, which is arguably even greater in the case of our landmark recommendation task than in other recommendation scenarios due to the limited amount of travel experience recorded in the on-

This work was first published as “Personalized landmark recommendation based on geotags from photo sharing sites” by Y. Shi, P. Serdyukov, A. Hanjalic, and M. Larson, in Proc. of ICWSM '11 [149]. This chapter is an extended version that has been accepted as “Non-trivial landmark recommendation using geotagged photos” for publication in ACM Transactions on Intelligent Systems and Technology, 4(3), 2013 [150].

line image set of any given user. We use category information extracted from Wikipedia in order to provide the system with a method to generalize the semantics of landmarks and allow the model to relate them not only on the basis of identity, but also on the basis of topical commonality. The proposed approach is computationally scalable, i.e., its complexity is linear with the number of observed preferences in the user-landmark preference matrix and the number of non-zero similarities in the category-based landmark similarity matrix. We evaluate the approach on a large collection of geotagged photos gathered from Flickr. Our experimental results demonstrate that WMF-CR outperforms several state-of-the-art baseline approaches in recommending non-trivial landmarks. Additionally, they demonstrate that the approach is well suited for addressing data sparseness and provides particular performance improvement in the case of users who have limited travel experience, i.e., have visited only few cities or few landmarks.

5.1 Introduction

Online photo-sharing sites such as Flickr¹ are a rich source of information on user photo-taking behavior, both at the individual and at the collective levels, representing a typical example of the social and community intelligence [195]. As GPS positioning becomes a standard functionality of mobile digital capture devices (i.e., cell phones or digital cameras), the amount of location information also available in online photo-sharing collections has increased. Location information takes the form of geotags, which encode where individual photos were taken.

Recent work has proposed various services that exploit the location information in photo-sharing sites, including geo-coordinate prediction [138], tag recommendation, content classification and clustering [158] and location recommendation [31]. The potential of location information in online photo-sharing collections is arguably not yet fully exploited. The richness of this potential is made clearer by closer consideration of the exact nature of an individual picture-taking act. Users deploy their mobile capture devices willfully and voluntarily. In other words, a click of the shutter is an explicit act of capture carried out on the part of a user. Insofar as it is possible to assume that users are triggered to take a picture by the feeling that a particular moment is special, then the capture of an image is effectively an act of tagging a particular moment as somehow important. If the image is associated with a geotag, then a user taking a picture is effectively tagging a place with an importance-related tag.

How exactly this importance should be understood or interpreted can be expected to vary widely from image to image and from user to user. However, when many acts of image capture are taken together, larger patterns emerge. These patterns can be exploited to implement intelligent, socially aware systems [195]. Seen from this high-level perspective, mobile devices are sensors capturing information on the importance of places for human visitors and with the appropriate algorithms the information can be processed in a way that makes it possible to provide back to users new information on places that they would probably find important and interesting.

In this paper, we give this high-level goal tangible form by proposing an algorithm that uses information from a photo-sharing website in order to provide users with personalized landmark recommendations. Our formulation of the landmark recommendation problem extends beyond the conventional location recommendation problem in two important respects. First, landmarks are considered to be places with a significance for history, culture or contemporary

¹<http://www.flickr.com/>

society. In short, they are places that have meaning for people and have a high potential for being of interest to travelers or cultural tourists. In contrast, locations are simply points on a map, which may or may not have a larger social significance or be associated with particular meaning. Second, under our formulation, landmark recommendation avoids popular landmarks, which duplicate information already available, e.g., in travel guides. Instead, we conceptualize personalization in the area of landmark recommendation to involve recommendation of ‘non-trivial’ landmarks – landmarks that are not typical destinations for mainstream travel and that a user would be unlikely to have easily found by other means. Note that there are multiple possible indicators that could be used to determine whether or not a particular landmark should be considered trivial. For example, one could consider a landmark is trivial if it is listed in a popular travel guide book, or if it is listed as highly popular in a travel website. In this paper, we define the triviality of a landmark based on the overall volume of attention that it receives from the users in an online community. We assume that users are motivated to visit non-trivial landmarks due to underlying topical interests in specific areas within domains such as architecture or history. In contrast, users are motivated to visit mainstream landmarks by a general desire to travel and see the world that is not topic specific. In sum, we consider personalization to involve a topical match between users’ own interests and the recommendations generated by the system. User satisfaction with the recommendations can be anticipated to rise if we are able to improve the quality of this match.

Recommending non-trivial landmarks is challenging because of the relative lack of information on past behavior of individual users that is available to create recommendations. This lack constitutes a formidable data sparseness challenge. Data sparseness is a problem because gaping holes are left in the information that would be desirable to create a fully fleshed-out picture of user preferences. Note that although Flickr contains an enormous number of photos, in order to use these photos to make recommendations, individual users must have photos taken at specific landmarks. Seen from this perspective, it is clear that Flickr is actually quite impoverished in terms of the information that it can offer to support landmark recommendation. Conventional recommender systems depend on past user consumption, such as profiles consisting of items either purchased or rated. Most users who travel, however, are relatively limited in the overall number of places that they visit. Travel is naturally constrained by a variety of factors that do not necessarily apply in other domains such as movie or book recommendation. Travel requires the availability of relatively large amounts of money and time, but also of travel documents (e.g., valid visa). The physical or cognitive stamina of the traveler also serves to keep the number of landmarks that a single person directly experiences relatively limited. Because of these limiting factors, a single individual visits only an extremely small frac-

tion of the total number of landmarks in existence. Consequently, personalized landmark recommendation must overcome a data sparseness problem that is arguably even larger than that presented by other domains in which there are fewer restrictions on the number of items with which users interact.

One conceivable approach would be to gather explicit information from users about their landmark preferences. Such a recommender system requires users to invest quite a bit of time and effort in explicitly informing the system of their interests. Further, if users have interests of which they are not consciously aware, the system will automatically fail to provide recommendations suiting these interests. Our approach to personalized landmark recommendation avoids these issues by exploiting implicit user preferences. For information on user landmark preference, we turn to the large and rich collections of user images available in online photo-sharing websites.

Using photo-sharing websites to approach the personalized landmark recommendation problem makes it possible to exploit user patterns that are implicit in photo-taking behavior, as described above. Collaborative filtering [129] (CF) allows us to make landmark recommendations in a new city for a traveler based on both the traveler's own preference on previously visited landmarks in other cities and also on other travelers' preference for landmarks in the new city. The underlying assumption is that a user in a new city may like landmarks that are already favored by other users who have had similar landmark visiting experiences in other cities in the past. Recommendations are made on the basis of images that the user has shared on a social media site and the user only needs to specify a destination city.

We specifically address data sparseness when designing our personalized landmark recommendation approach by incorporating category-based regularization, which exploits information concerning the general categories of landmarks. The categories are classes such as '17th-century architecture', 'Japanese gardens' or 'World War II sites'. We assume that user preferences are a reflection of underlying user topical interests. The use of categories allows us to counteract data sparseness by introducing information on similarity between landmarks on a more abstract, topical level. We obtain category information about landmarks from Wikipedia². Although another encyclopedic knowledge resource could have been used for the same purpose, we use Wikipedia due to its scope, availability and the fact that it is itself an online community resource.

The technique we propose is designed to be deployed in an application that uses the geotagged photos that a traveler (i.e., the target user) has uploaded to an online community to recommend landmarks in a new city for that user to visit. For example, if a traveler has used a smart phone to take a few photos in a new

²<http://www.wikipedia.org/>

city, then the proposed system can recommend some non-trivial landmarks in the city that could also fit her interest. We make use of the collaborative filtering paradigm [2, 129] in order to tackle the personalized landmark recommendation problem. We propose an approach, designated WMF-CR, that makes use of both *weighted matrix factorization* and *category-based regularization* in order to improve the performance of personalized landmark recommendation. This approach represents a substantive extension on its progenitor, CRMF (Category Regularized Matrix Factorization), which we proposed in the short paper [149] with which we initially introduced the landmark recommendation problem. The key innovation is our use of a weighting mechanism to balance the benefits of retaining as much data as possible on which to base the recommendation against the dangers of including data that will lead to highly popular, trivial recommendations. This characteristic sets our model apart from conventional CF models which do not have the capacity to prefer globally less-popular items over popular ones. We also notice that in this work we rely on the geotags of images rather than the image content for the recommendation process.

As mentioned before, recommendation using images from photo-sharing sites differs from recommendation in more conventional scenarios due to the lack of explicit user ratings, such as those often used for movie recommendation. Instead, our model incorporates user preference as expressed by the number of photos that users take at various locations. The experimental evidence presented in this paper will support the conclusion that WMF-CR effectively exploits user photo-taking behaviors to deal with data sparseness and that it is able to make non-trivial recommendations.

This paper presents the first fully mature approach to the personalized landmark recommendation problem and makes the following major contributions:

- We propose a novel approach, WMF-CR, that specifically addresses the issue of making non-trivial recommendations for the personalized landmark recommendation scenario.
- We demonstrate that WMF-CR outperforms other state-of-the-art approaches in recommending non-trivial landmarks.
- We show that WMF-CR is adequately scalable to deploy for very large collections.
- We provide evidence that WMF-CR accomplishes its design goal of addressing data sparseness by verifying its performance for users who only have limited travel experience, i.e., those most likely to suffer due to lack of information in their online photo-sharing profiles.

We would also like to note that in order to help support the new research topic,

namely, personalized landmark recommendation using users' geotagged photos, we make the data collection used for this research publicly available³.

The remainder of the paper is structured as follows. In the next section, we present a summary of the relevant literature and indicate how it is related to our own work. Then, in Section 5.3, the proposed WMF-CR model for non-trivial landmark recommendation is described in detail. Next, in Section 5.4, we introduce a data collection for the study of personalized landmark recommendation. The experimental evaluation of the proposed approach is presented in Section 5.5. The last section sums up the key aspects of our study and gives a brief outlook on future work.

5.2 Related Work

This section provides the necessary background information for our work. We discuss the emergence of non-trivial recommendation in the area of recommender systems. Then we present an overview of related work on recommendation that exploits location information, especially for social media sites. Finally, we present the necessary background on collaborative filtering (CF) techniques for recommendation.

5.2.1 Non-trivial Recommendations

There is emerging research interest in recommending non-trivial items, which we argue is particularly relevant for the area of travel destination recommendation. In the area of general travel destination recommendation, it has been observed that recent trends in tourism [19] have had an impact on the types of travel that people engage in. In particular, this work discusses the trend of *cultural tourism*, which involves an increasing demand for independent holidays during which people seek authentic and personal experiences that go beyond mainstream tourism. It emphasizes that in order for travel experiences to be personal, they should match the topical interests of individuals. Our algorithm for personalized landmark recommendation also adopts the assumption that personalization involves a topical interest aspect that is specific to a particular user.

Non-trivial recommendation shares the similar idea to the long tail investment, which has been analyzed to bring commercial benefits for internet companies [40]. Research effort has also been devoted to improving recommendation performance for items in the tail [118]. For the general recommendation sce-

³<http://dmirlab.tudelft.nl/users/yue-shi>

nario, a recent empirical study investigated several recommender algorithms for top-N recommendation tasks [33], revealing the importance of recommending non-trivial items by removing the most popular items in the evaluation. In this paper, we adopt a similar evaluation strategy as suggested in the work of [33].

Recently, evaluation issues have already attracted much effort in the recommender system research community. Various criteria have been proposed for evaluation of recommender systems beyond rating prediction accuracy [32, 46, 48, 57, 103, 110], including diversity, coverage, robustness, novelty and serendipity. However, standard evaluation metrics have not yet been established for measuring recommendation performance in terms of these different criteria. In particular, there is no evaluation metric in widespread use that measures the ability of recommendation approaches in recommending non-trivial items. In our work, we focus on evaluating the performance of recommending non-trivial landmarks by assuming that a specific number of the most popular landmarks in each city are irrelevant recommendations for users.

5.2.2 Exploiting Location Information for Recommendation

GPS-based Recommendation. Recent work has exploited explicit user-location data (e.g., GPS data) for location recommendation. Based on user location data collected directly from GPS devices, [86] proposed to mine user-to-user similarity from location histories in order to infer the correlation between different locations. The extension of this work [199] has shown that user similarity based on location history can be effectively exploited for personalized friend recommendation as well as location recommendation. Similarly, systems that make use of user similarity or location similarity mined from user location history have been built for shop recommendation [167] and restaurant recommendation [59]. In addition, a HITS (Hypertext Induced Topic Search)-based approach was proposed to recommend interesting locations and travel sequences within a region [200]. A further study based on this work has shown personalized recommendation can be achieved by mining correlations between locations based on user location history [198]. Another recent study demonstrated that user GPS history data can be exploited for location and activity recommendation via a joint matrix factorization model [197]. Compared to this previous work, a key difference in our work lies in that the user location data are obtained implicitly, i.e., from users' geotagged photos that are uploaded in social media sites, which requires much less effort from the user and the system. In other words, users do not need to spend large amounts of time uploading their location history, and systems do not need to preprocess or transform huge amount of user-uploaded data before being able to generate recommendations.

Geotag-informed Recommendation. Location data from the geotagged

photos of users has been exploited to approach various tasks. Based on the tags and geotags of Flickr photos together with information extracted from Yahoo Travel Guide⁴, a travel guidance system [44] has been designed to recognize and rank landmarks. Geotags have also been exploited to help travelers with trip planning [94], such as, to suggest places of interest, to find a proper path to view a landmark, and to find a proper route to travel from one landmark to another. Mining frequent trip patterns through geotagged photos including frequently visited city sequences and typical visit duration, has been proposed [8] to improve travel recommendation. Exploiting the similar trip patterns as studied in [8, 36] has been used to automatically construct travel itineraries. The differences between this previous work in the area of geotag-informed recommendation and our own work lies in the fact that we specifically address the personalized landmark recommendation task, i.e., making non-trivial recommendations of landmarks for individual users.

Personalized Geotag-informed Recommendation. To the best of our knowledge, there are only two recent studies that are closely related to our work, since they also target personalized location-based recommendation based on geotags from photo sharing sites. The first proposed to personalize location prediction by first generating recommendations based on location popularity and then re-rank recommendations based on similar interest from other users [31]. This work observed, but did not specifically address the issue that popular locations can dominate less frequently-visited locations in location recommendation. The authors suggest that recommendations for less-frequently visited/non-trivial locations would be more meaningful to travelers. Our work differs from [31] in three main respects. First, we target a new application, i.e., landmark recommendation rather than location recommendation. Second, we specifically address the data sparseness problem in the recommendation context. Third, the prediction model proposed by [31] relies on user similarities, which for large data sets grow to be computationally quite expensive, an issue which we return to below.

The second instance of closely related work, [77], proposed to approach personalized travel route recommendation by capturing both location dependence and user interest dependence. Location dependence was represented by the time-stamps of geotagged photos and user interest dependence was mined from users' travel routes that are extracted from geotags. Compared to this work, our approach has substantial differences in along two lines: First, we focus on landmark recommendation rather than route recommendation. Second, we design a recommendation model that specifically addresses the data sparseness problem and the challenge of recommending non-trivial landmarks, both of which were not investigated in the work of [77].

⁴<http://travel.yahoo.com>

5.2.3 Collaborative Filtering

Collaborative filtering is known as one of the most popular techniques for personalized recommendation. CF usually follows one of two basic approaches, memory-based or model-based [2]. In general, memory-based approaches make recommendations on the basis of similarities between users (user-based) [55], or on the basis of similarities between items (item-based) [38, 88, 136]. Among the aforementioned work on location-based recommendation, the work of [31, 59, 167, 199] involves memory-based CF. However, memory-based CF approaches usually suffer from computational cost in computing user-to-user or item-to-item similarities from a large number of users and items. Since there could be millions of users in social media sites, we do not choose the direction of memory-based CF for personalized landmark recommendation in this paper, while focusing on the direction of model-based CF.

Compared to memory-based CF, model-based approaches first fit prediction models based on training data and then use these models to predict users' preference on items. In particular, matrix factorization (MF) techniques have attracted much research attention in recommender systems because of their scalability and accuracy in rating/preference prediction, as witnessed by the Netflix contest [75]. Generally, MF techniques learn latent features of users and items from the observed preference in the user-item matrix and these features are then used to predict unobserved preferences. The rationale of MF is also illustrated from probabilistic point of view [134]. Rather than solely focusing on predicting users' preference scores, researchers have also formulated ranking-oriented CF approaches that specifically model users' pair-wise or list-wise preference based on their rating patterns, e.g., CofiRank [182], EigenRank [90], ListRank [144]. We also notice the existence of early work on weighted low-rank approximation [160], which describes a method that is referred to "weighted" but is close to standard MF. Another work [60] proposed to enhance the contribution of positive feedback by weighting each factorization based on a confidence estimate, which is proportionate to the strength of user-item preference. Note that we propose to design the weights in MF based on landmark popularity to specifically target recommending non-trivial landmarks. In contrast, in [160] the weights in MF are based on whether or not ratings are observed for particular items, and in [60] the weights are still based on user-item preference. Note that both of the two approaches have no particular consideration on trivial items, or on the exploitation of additional item information, the two issues studied in this paper.

Recently, the joint matrix factorization framework has been widely proposed to extend the basic MF model by taking into account different regularizations in order to make it suited for different purposes. For example, user social relationships have been exploited to regularize the factorization of user-item rating

matrix for improved rating prediction [96]. User activity correlation and location correlation in terms of location features have been exploited to regularize the factorization of a location-activity matrix for improved location and activity recommendation [197]. Contextual movie features have been exploited to regularize the factorization of the user-movie rating matrix for improved mood-specific movie recommendation [145]. Collective matrix factorization [155] has been proposed to factorize multiple matrices of related entities in order to leverage knowledge between different entities. In addition, since tags have been ubiquitous in recommender systems, researchers proposed not only to exploit tags for improving item recommendation [170], but also to exploit tensor decomposition techniques for improving tag recommendation [165]. Our work in this paper is closely related to aforementioned work in the sense that we also use the joint matrix factorization framework to exploit landmark categories from Wikipedia to regularize the factorization of user-landmark preference. However, we not only address a different application, our work is also substantially different from previous work in that our approach integrates a weighting scheme into the matrix factorization model, which allows the model to specifically target recommendation of non-trivial landmarks.

5.3 Non-trivial Landmark Recommendation

5.3.1 Overview

An overview of our personalized landmark recommender system is provided in Fig. 5.1. At the top (i.e., step 1), the photo-sharing collection is depicted from which the geotagged photos of the users are drawn and then transformed into a user-landmark matrix that encodes users' preference on landmarks. Note that we also use the photo-sharing site to extract an inventory of landmarks. The process of extracting landmarks from geotagged photos and extracting landmark categories from Wikipedia will be described in greater detail in Section 5.4. As previously mentioned, we use the number of photos that a user has taken around a landmark to indicate the user's preference for the landmark, i.e., a larger number of photos taken around a given landmark reflects a larger degree of preference.

In step 2, the landmark category information is extracted from Wikipedia and used to calculate the similarity between landmarks that are topically related, but not identical. Details of the category-based landmark similarity will be explained in Section 5.3.3. Once the user-landmark preference matrix and category-based landmark similarity matrix have been obtained, the proposed approach, WMF-CR, is applied (i.e., step 3) to learn the latent features of users and landmarks during the matrix decomposition process. The resulting model

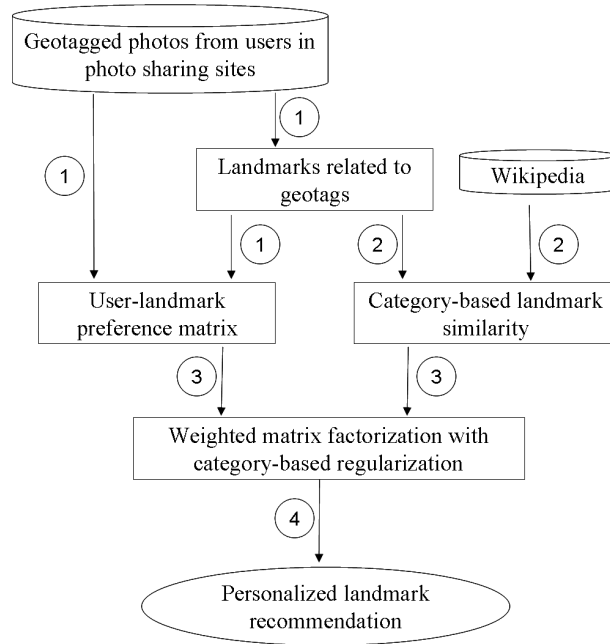


Figure 5.1: The block diagram of the proposed personalized landmark recommender system

is then used to generate non-trivial landmark recommendations to users, as illustrated in step 4.

In the following of this section, we present our Weighted Matrix Factorization with Category-based Regularization (WMF-CR) approach for the personalized landmark recommendation in detail. We first introduce the new weighting scheme incorporated by our approach and our method for using landmark categories available from Wikipedia to quantify the similarity between landmarks. Then, we put the individual parts together into the WMF-CR and conclude the section with a discussion that summarizes the characteristics of our algorithm that make it particularly suited to address the challenge of personalized landmark recommendation.

5.3.2 Weighted Matrix Factorization

Rationale. Recall from the introduction that our assumption is that users have different motivations for visiting different landmarks. The motivation for visiting frequently-visited landmarks is a general desire to travel and see the world. For this reason, such highly popular landmarks are considered trivial, since they are standard and do not have to be recommended on a person-by-

person basis. The motivation for visiting non-trivial landmarks is the interest of travelers in the topical aspects of the landmark. Our weighting scheme is motivated by the assumption that reducing the influence of highly popular landmarks will effectively reduce noise within the user-landmark matrix that is masking the person-dependent topical information implicit in user visiting patterns for non-trivial landmarks. A simplistic approach to emphasizing non-trivial landmarks is to eliminate all other landmarks from the user-landmark matrix. There are two reasons for which this approach is not to be preferred.

First, we can never be entirely sure about user motivations. In the case of the Eiffel Tower, we can probably safely assume that most visitors go there motivated by a general desire to have seen the world rather than motivated by a specific interest in the type of architecture it represents, namely a puddle iron lattice tower. However, in the larger majority of the cases it will be dangerous to apply this assumption. Rather than guessing at the topic-specific attraction of specific landmarks, we prefer to let the model learn which landmarks are less useful for personalized travel recommendation.

Second, we need to be very conservative about eliminating data from the user-landmark matrix in order to reduce the danger that we discard information that could prove useful for recommendation. As previously mentioned, the extreme sparseness of the user-landmark matrix presents us with a significant challenge. The sparse data problem could potentially be exacerbated if we are too aggressive in removing user preferences on landmarks from the user-landmark matrix. Instead, we would like to leave open the possibility that even relatively popular landmarks have topical interests inherent in the associated user visiting patterns that can be exploited to make personal landmark recommendations. Again, for this reason, we prefer to let the model learn which are useful.

Conventional Matrix Factorization. Our Weighted Matrix Factorization (WMF) approach extends a conventional matrix factorization approach with weights that control the relative contribution of non-trivial landmarks from the user-landmark matrix. The basic user-landmark matrix encodes the preferences of users for the individual landmarks. In order to represent user i 's preference on landmark j , we first calculate the number of user i 's photos that have geotags matching the landmark j . Note that the matching process between the photo geotags and the landmarks will be described in Section 5.4. Then, for each user i , we further normalize the number of her geotagged photos on landmark j to be within $[0, 1]$ (by dividing over the total number of the user's geo-tagged photos). The result is the user i 's preference score on landmark j , which is denoted by R_{ij} . The user-landmark matrix is designated \mathbf{R} , and it represents users in rows and their preferences on landmarks in columns.

The basic matrix factorization model [75] is expressed as:

$$U, V = \arg \min_{U, V} \left\{ \frac{1}{2} \sum_{i=1}^M \sum_{j=1}^N I_{ij} (R_{ij} - U_i^T V_j)^2 + \frac{\lambda_U}{2} \|U\|_F^2 + \frac{\lambda_V}{2} \|V\|_F^2 \right\}, \quad (5.1)$$

where \mathbf{R} consists of M users and N items. The vector U_i is the d -dimensional latent feature vector (a column vector) of user i . The vector V_j is the d -dimensional latent feature vector (a column vector) of landmark j . Note that the elements in the feature vectors are parameters (latent features) that need to be estimated from a set of training data. The idea of MF is to estimate \mathbf{U} and \mathbf{V} in terms of the known preference data \mathbf{R} and to use the learned \mathbf{U} and \mathbf{V} to predict the unknown users' preferences on landmarks. I_{ij} is an indicator function that is equal to 1 if $R_{ij} > 0$, and 0 otherwise. $\|\mathbf{U}\|_F$ and $\|\mathbf{V}\|_F$ are the Frobenius norms of \mathbf{U} and \mathbf{V} , respectively, which serve to alleviate overfitting. λ_U, λ_V are the norm regularization parameters, on which we impose the simplifying assumption $\lambda_U = \lambda_V = \lambda$.

Weighted Matrix Factorization. Under the basic MF formulation, all the observed preference scores contribute equally to learning the latent features of users and landmarks. However, for the purpose of personalized landmark recommendation, we wish to direct the recommendation process towards non-trivial landmarks, i.e., landmarks that are not widely visited, but that represent users personal, topical interest. We introduce a bias towards non-trivial recommendations into the model by reducing the influence of frequently-visited landmarks in the user-landmark matrix and increasing the influence of non-trivial landmarks.

We build the capacity to carefully adjust the balance between different kinds of landmarks into the model by introducing a factor W , which is integrated into Eq. (5.1) to yield the WMF formula:

$$U, V = \arg \min_{U, V} \left\{ \frac{1}{2} \sum_{i=1}^M \sum_{j=1}^N W_{ij} (R_{ij} - U_i^T V_j)^2 + \frac{\lambda}{2} (\|U\|_F^2 + \|V\|_F^2) \right\}. \quad (5.2)$$

The weight that is used for a given landmark j is determined by the global popularity of that landmark measured on the entire photo-sharing collection. Note that we define the popularity of a landmark by the number of users who have visited the landmark. We rank all landmarks in the collection by their overall popularity scores and retain influence only from the landmarks that are 'low enough' in the ranking. These landmarks are the ones that make the most effective contribution to biasing the model towards non-trivial recommendations. Note that in order to determine what should be considered as 'low enough', a threshold of ranking position needs to be determined empirically given a concrete use case, while the general idea is to eliminate the influence of the top popular landmarks on learning the recommendation model. Specifically, the

weighting coefficient W_{ij} in WMF is defined as:

$$W_{ij} = \begin{cases} 0, R_{ij} > 0 \wedge \text{poprank}(j) \leq K \\ 1, R_{ij} > 0 \wedge \text{poprank}(j) > K. \end{cases} \quad (5.3)$$

where $\text{poprank}(j)$ defines the rank of the popularity of landmark j . For instance, if landmark j is the most popular landmark in the collection, then $\text{poprank}(j) = 1$. For preferences that are unobserved, i.e., in cases where $R_{ij} = 0$, W_{ij} is trivially also set to 0. K takes the role of a threshold parameter, which controls how many of the most popular landmarks have their influence eliminated from the model. A larger K means that more highly popular landmarks are not used for training the recommendation model, resulting in a recommendation model that could be more biased to recommend non-trivial landmarks. However, the larger K also means that the more user-landmark preference data are discarded for training the recommendation model, resulting in a recommendation model that may suffer more from data sparseness. For this reason, in practice the optimal value of K needs to be tuned for a given data collection to attain a tradeoff between the aforementioned two aspects. We will demonstrate and discuss the impact of K in our experiments in Section 5.5. In sum, WMF explicitly reduces the influence of popular landmarks, effectively extending the model with the capacity to make non-trivial recommendations. Note that in the case of $K = 0$, WMF returns to the basic MF model.

5.3.3 Category-based Landmark Similarity

Our Category-based Regularization extends matrix factorization with regularization that integrates information on higher-level semantic similarities between landmarks. This approach explicitly addresses the problem of data sparseness in the user-landmark preference matrix, by making it possible to relate landmarks not only on the basis of identity, but also on the basis of topical similarity. For example, both landmark ‘London Bridge’ and landmark ‘Monument to the Great Fire of London’ belong to the category ‘History of the City of London’, indicating that they may be interesting to users who would like to know about London history. This topic-level similarity could allow the system increase the likelihood to recommend ‘Monument to the Great Fire of London’ to a user if he was already in favor of ‘London Bridge’, even in the case that there is limited preference data about ‘Monument to the Great Fire of London’. Incorporating auxiliary information to represent users and landmarks helps, in this way, compensate for the missing information in \mathbf{R} . For each landmark we collect category information from Wikipedia and create a binary landmark-category matrix, \mathbf{C} , which captures the links between landmarks and categories. There, $C_{jt} = 1$ if landmark j belongs to a category t , and 0 otherwise. Note that each landmark potentially belongs to multiple categories, e.g., ‘London Eye’ belongs to ‘Merlin Entertainments’, ‘Thames Path’, ‘Visitor attractions in London’, etc. For this reason, we can define the category-based landmark similarity between

landmark j and landmark n , by using the vector space similarity [135]:

$$S_{jn} = \frac{\sum_{t=1}^T C_{jt}C_{nt}}{\sqrt{\sum_{t=1}^T C_{jt}^2}\sqrt{\sum_{t=1}^T C_{nt}^2}}, \quad (5.4)$$

where T denotes the number of categories, and \mathbf{S} denotes the category-based landmark similarity matrix, which is symmetric and contains values between 0 and 1.

We build on the assumption that landmarks that are similar with respect to their categories might share similar characteristics pertinent for the purpose of landmark recommendation and that these characteristics are expected to be preserved under matrix decomposition. Exploiting this insight, a new loss function can be formulated as:

$$L(V) = \sum_{j=1}^N \sum_{n=1}^N J_{jn} (S_{jn} - V_j^T V_n)^2, \quad (5.5)$$

where J_{jn} is an indicator function that is equal to 1 if $S_{jn} > 0$, and 0 otherwise. It is important to note that the new loss function (5.5) has the potential to alleviate the data sparseness problem in \mathbf{R} . An extreme example illustrates this potential: if there is no user preference on landmark j , the latent features V_j cannot be learned from \mathbf{R} at all. However, V_j can still be learned from other landmarks that are similar in terms of categories, and are marked as being preferred by the users. Consequently, latent features of landmarks could be substantially better represented when exploiting not only user-landmark preference but also category-based landmark similarity.

5.3.4 Weighted Matrix Factorization with Category-based Regularization

We now combine WMF and category-based regularization, and present our integrated WMF-CR algorithm for personalized landmark recommendation. The objective function of WMF-CR is formulated as:

$$\begin{aligned} F(U, V) = & \frac{1}{2} \sum_{i=1}^M \sum_{j=1}^N W_{ij} (R_{ij} - U_i^T V_j)^2 \\ & + \frac{\beta}{2} \sum_{j=1}^N \sum_{n=1}^N J_{jn} (S_{jn} - V_j^T V_n)^2 + \frac{\lambda}{2} (\|U\|_F^2 + \|V\|_F^2) \end{aligned} \quad (5.6)$$

where β serves as a tradeoff parameter that controls the relative contribution from the category-based landmark similarity. Note that WMF-CR becomes equivalent to the WMF model when β is equal to 0. Minimizing the objective function results in the latent features of users and landmarks being learned

from the user-landmark preference in \mathbf{R} and from the category-based landmark similarity in \mathbf{S} .

Since the objective function in Eq. (5.6) is not jointly convex over \mathbf{U} and \mathbf{V} , we choose to use alternating gradient descent to solve the minimization problem. The gradients of $F(U, V)$ with respect to \mathbf{U} and \mathbf{V} can be computed as in Eq. (5.7) and (5.8).

$$\frac{\partial F}{\partial U_i} = \sum_{j=1}^N W_{ij} (U_i^T V_j - R_{ij}) V_j + \lambda U_i \quad (5.7)$$

$$\frac{\partial F}{\partial V_j} = \sum_{i=1}^M W_{ij} (U_i^T V_j - R_{ij}) U_i + 2\beta \sum_{n=1}^N J_{jn} (V_j^T V_n - S_{jn}) V_n + \lambda V_j \quad (5.8)$$

Note that in Eq. (5.8) we make use of the property that \mathbf{S} is symmetric. A local minimum solution of \mathbf{U} and \mathbf{V} is achieved by iteratively performing descent on \mathbf{U} and \mathbf{V} . With the learned latent features of \mathbf{U} and \mathbf{V} , we can predict user i 's preference on landmark j as $U_i^T V_j$. On the basis of this score, we generate the ranked list of landmarks that constitutes the personalized recommendation for user i .

The complexity of the objective function of WMF-CR in Eq. (5.6) is $O(d|\mathbf{R}|+d|\mathbf{S}|+d(M+N))$, where $|\mathbf{R}|$ denotes the number of known preference scores in the given user-landmark preference matrix \mathbf{R} , and $|\mathbf{S}|$ denotes the number of non-zero similarities in the category-based landmark similarity matrix \mathbf{S} . The complexity of the gradients in Eq.(5.7) and (5.8) is $O(d|\mathbf{R}|+dM)$ and $O(d|\mathbf{R}|+d|\mathbf{S}|+dN)$, respectively. Considering that we usually have $|\mathbf{R}|\gg M, N$, and $|\mathbf{S}|\gg N$, the total complexity of WMF-CR is $O(d(|\mathbf{R}|+|\mathbf{S}|))$, which is linear with the total number of known preference scores in \mathbf{R} and non-zero similarities in \mathbf{S} . Note that both \mathbf{R} and \mathbf{S} are very sparse in practice, e.g., in our data collection, as described in Section 5.4, we have $|\mathbf{R}|=260362$ and $|\mathbf{S}|=222778$. For this dataset, our MATLAB implementation of WMF-CR takes ca. 6.5 seconds for one iteration of the learning algorithm, running on a PC with 1.59 GHz CPU and 2.93 GB memory. This analysis indicates that WMF-CR is appropriate for application where it is necessary to scale up to very large use cases.

5.3.5 Discussion

We conclude our presentation of the proposed WMF-CR approach with a brief summary and discussion of the ways in which the algorithm addresses the specific challenges faced in personalized landmark recommendation.

1. **Non-trivial recommendation.** WMF-CR utilizes a weighting scheme in order to enhance the impact of non-trivial landmarks on the recom-

mentation process. We assume that non-trivial landmarks differ from trivial landmarks in that they contain more information on the underlying topical interests of users and, as such, are best suited to provide recommendations that are very specific for individuals. Effectively, within the personalized landmark recommendation scenario, the effect of landmarks that are highly popular and are visited independently of travelers specific topical interests amounts to noise. We reduce the influence of highly popular landmarks by implementing a careful balance between removing harmful data and retaining as much information as possible so as not to exacerbate the sparse data problem.

2. **Scalability.** The computational complexity of WMF-CR has been demonstrated to be linear in the total number of observed user preferences and non-zero category-based landmark similarities. Although the complexity of computing category-based landmark similarity is quadratic in the number of landmarks involved, this computation could be completely done offline. In addition, since the growth of the number of landmarks in the real world could be much slower than movies or music that are typical in recommender systems, the update of the category-based landmark similarity could have little cost. In sum, WMF-CR has great potential to be applied in very large scale social media sites.

3. **Data sparseness alleviation.** WMF-CR exploits landmark categories as an external resource to learn generalizations over the relationship between landmarks. Effectively, latent features of users and landmarks are learned not only from the user-landmark preference matrix, but also from the external knowledge. The external knowledge is cheap because it is extracted from Wikipedia. The ability of WMF-CR to address data sparseness is important for the personalized landmark recommendation scenario, since most of users usually have limited travel experience, i.e., a significant number of users could suffer from data sparseness problem. This issue will also be noted in our experimental evaluation in section 5.5.1.

In sum, WMF-CR is designed to generate effective personalized landmark recommendations while addressing the challenges of applying CF to large online photo-sharing collections. In the remainder of the paper, we present our data collection and an experimental analysis that demonstrates the strength of WMF-CR in practice.

Table 5.1: Statistics of the user-landmark preference dataset.

Max Num. landmarks visited by a user	524
Ave. Num. landmarks visited per user	6.5
Max Num. users visiting a landmark	4497
Ave. Num. users visiting per landmark	27.24
Sparseness	99.93%

5.4 Data Description

We collected the data for our experiments using the public API of Flickr. First, we downloaded metadata for 42.9 million geotagged photos and then we filtered this data according to the following considerations. For the purpose of personalized travel recommendation, we wish to focus our investigation on the images of users who are travelers. Since overwhelmingly large majority of Flickr users are based in the United States, we introduced a clearer focus on images most likely to belong to travelers by focusing only on users with geotagged photos taken outside the US. We limited our investigation to city landmarks and include in the dataset only the photos taken in the top 40 most visited cities. The above steps resulted in a dataset containing 126,123 geotagged photos from 40,084 users. We generated the landmarks associated with the images by making use of Wikipedia. We considered each geotagged Wikipedia article to constitute a landmark. Each photo was associated with all landmarks within one kilometer radius. This set was obtained by computing the geographical distance based on the geotags of the photos with the geotags of the Wikipedia articles, according to Haversine formula [157]. The set of landmarks was then filtered in a process that involved eliminating landmarks for which there was negligible overlap between the words in the title of the Wikipedia article and the tags assigned to the photo by the user. As a final step, already mentioned above, we normalized the number of a user’s photos related to a landmark in order to generate a score representing that user’s preference for a landmark. The resulting user-landmark preference matrix consists of 260,362 scores from 40,084 users and 9,557 landmarks. The main statistics of this data are given in Table 5.1.

In Fig. 5.2 we illustrate properties of the data with two plots showing how users are distributed over landmarks and landmarks over users. Note that Fig. 5.2(a) is based on the histogram of the number of landmarks visited by each user, while Fig. 5.2(b) is based on the histogram of the number of users who visited each landmark. Both distributions can be seen to closely follow a power-law, which illustrates the two underlying characteristics of the data, already mentioned above: 1) the data set is very sparse: most of the users in our dataset visited only a limited number of landmarks; 2) popular-landmarks threaten to dominate:

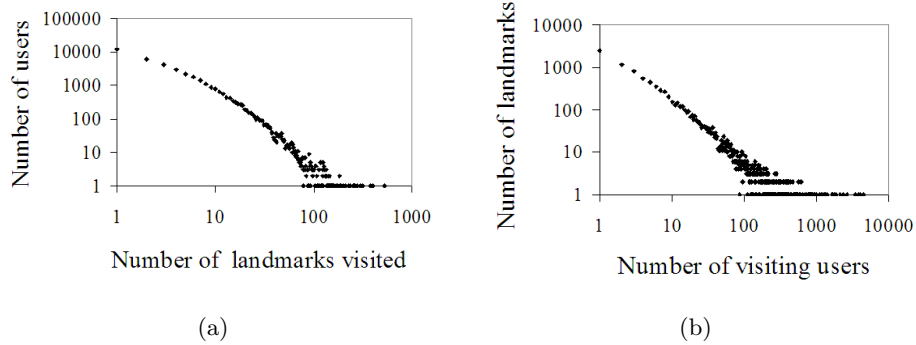


Figure 5.2: Log-log plot on (a) the distribution of the number of landmarks visited by the users, and (b) the distribution of the number of users visiting the landmarks

there are a few landmarks visited by lots of users, implying a phenomenon that the overlap of non-trivial (less-popular) landmarks across users may be very small, which makes the challenge and importance of recommending non-trivial landmarks quite substantial.

We extracted categories for each landmark from Wikipedia, resulting in a total of 7,379 categories. In our collection, the total number of landmark-category assignments is 20796, and in average there are 2.2 categories per landmark. The maximal number of categories that a landmark has is 28. This information is encoded to generate a landmark-category binary matrix with 7,379 categories.

5.5 Experimental Framework and Results

In this section, we report on a series of experiments conducted to evaluate the proposed WMF-CR model for personalized landmark recommendation. First, we introduce the experimental protocol, followed by an investigation of the impact of the parameters in the proposed WMF-CR model. Then, we compare the recommendation performance of WMF-CR with several state-of-the-art baseline approaches. We focus our evaluation on the ability of these approaches to make non-trivial landmark recommendations. Finally, we investigate the influence of user travel experience on the performance of personalized landmark recommendation.

Our experiments are designed in order to address the following research questions:

1. Is the weighting scheme effective? In other words, does it show evidence of achieving its aim of enhancing the influence of non-trivial landmarks

on recommendation?

2. Is the category regularization effective? In other words, do the landmark categories show evidence of achieving their aim of eliminating the negative impact of data sparseness?
3. Is WMF-CR beneficial for users who only have limited travel experience?

5.5.1 Evaluation Framework

Since we evaluate personalized landmark recommendation based on the ranked recommendation list for each user, which is comparable to the search result list for a query in document retrieval, we adopt two of the most widely used evaluation metrics, i.e., mean average precision (MAP) [57] and mean reciprocal rank (MRR) [175] to evaluate the recommendation quality. Specifically, AP and RR are defined for a given user m as:

$$AP_m = \frac{\sum_{j=1}^{N_m} (rel_m(j) \times Prec_m@j)}{\sum_{j=1}^{N_m} rel_m(j)} \quad (5.9)$$

$$RR_m = \frac{1}{\min \{j | rel_m(j) = 1, 1 \leq j \leq N_m\}} \quad (5.10)$$

where N_m denotes the number of recommended landmarks for the user m . $rel_m(j)$ is a binary indicator, which is equal to 1 if the j th landmark in the list is relevant to user m , and otherwise 0. $Prec_m@j$ is the precision of the top j recommended landmarks for the user m , i.e., the number of landmarks in the top j recommendation that are relevant to user m . MAP and MRR are average value of AP and RR, respectively, across all the users for evaluation. MAP reflects the quality of the entire recommendation list, while MRR emphasizes the ability of the system to recommend a relevant landmark as early as possible. The higher MAP and MRR scores, the better is the recommendation performance.

The most straightforward application from this study is to provide a user in a social photo sharing site with landmark recommendations, when she is visiting a new city. For this reason, we evaluate the proposed WMF-CR approach for personalized landmark recommendation under a simulated setting that approximates this application. In our experiments we only use the data of those users who have visited at least one landmark in each of at least two cities. By this means, we can define at least one “already visited” city and at least one “target” city for each user. The user-landmark matrix used for the experiments contains 14,031 users and all the landmarks, and has a sparseness of 99.89%, showing a challenging recommendation scenario in which the recommendation model would suffer from limited user preferences, while the proposed approach

is expected to benefit from external knowledge of landmarks. We also emphasize that the sparseness of this data collection is higher than conventional CF benchmark datasets, such as MovieLens 10M-rating dataset⁵ (sparseness ca. 97.7%) and Netflix 100M-rating dataset⁶ (sparseness ca. 98.8%).

The dataset is randomly split into three sets, i.e., a training set, a validation set and a test set. The training set contains 60% randomly selected users and their landmark preferences. Each of the validation set and the test set contains 20% randomly selected users and their landmark preferences. For each user in the validation and the test set, we randomly select one city that she has visited as that user's target city. Then, we remove the user's landmark preferences for this selected target city and try to recommend them. Note that the validation set is used to investigate the impact of parameters of the WMF-CR model, and the test set is used to evaluate the performance of WMF-CR and compare it to that of the baseline algorithms. We compare landmark recommendation algorithms on the basis of their ability to predict the withheld landmark preferences. Note that this method of evaluation provides a very conservative, lower-bound estimate of recommendation quality, which probably rather severely underestimates the usefulness of landmark recommendation algorithms in real-world applications. Recall that a user's landmark preferences for a city are taken to be all the landmarks associated with photos that the user has taken in that city. Users probably fall far short of photographing at all landmarks within a given city that would potentially be of interest to them, especially if the city is a large one. For this reason, landmarks that are not photographed might still be interesting recommendations for a user, even though these landmarks are treated as false alarms by our evaluation method. Under this evaluation framework, the resulting numbers of evaluation metrics are usually very low [33]. However, our purpose in evaluation is to determine the relative difference between recommendation approaches, and for this purpose our evaluation method is straightforward and well suited. Note that the regularization parameter λ in WMF-CR in Eq.(5.6) is set to 1 and the latent dimensionality d is set to 10, the optimal values as determined on the validation set for all the related MF baseline approaches.

Finally, as discussed before, popular and well-known landmarks are not the most useful or desirable recommendations for the task of personalized landmark recommendation. For this reason, in our evaluation we investigate the influence of the popular landmarks on the final result and focus on non-trivial, i.e., less popular, landmarks when discussing the recommendation performance.

⁵<http://www.grouplens.org/node/73>

⁶<http://www.netflixprize.com/>

Table 5.2: The top-ranked popular landmarks in the training set.

<i>poprank</i>	landmark
1	eiffel tower
2	louvre
3	london eye
4	big ben
5	tower bridge
6	colosseum
7	westminster
8	sagrada familia
9	trafalgar square
10	notre dame de paris
11	arc de triomphe
12	british museum
13	tate modern
14	reichstag
15	park guell

5.5.2 Impact of Parameters

In this section, we use the validation set to investigate the impact of the parameters of the WMF-CR, discuss their role in recommendation and determine the parameters settings that we use for the experiments.

First, we investigate the impact of parameter K in the WMF-CR model (cf. Eq. 5.6). K controls the balance of influence between non-trivial landmarks and mainstream landmarks on recommendation performance. In Table 5.2, we list the landmarks in the training set according to a descending order of their popularity, i.e., the number of users who have visited each of them.

It is evident that the top-ranked popular landmarks are well-known, frequently-visited, mainstream landmarks. By varying K , we eliminate the influence of those landmarks on the process of learning latent features that represent users and other landmarks. We examine the performance of WMF-CR on the validation set both in terms of MAP and MRR for the case in which the top five landmarks in each city have been removed.

Fig. 5.3 illustrates how the performance changes with K for non-trivial landmark recommendation and reveals that optimal recommendation performance is achieved when $K = 11$. In this case, non-trivial landmarks are taken to be all landmarks within a city except for the top-five most popular. Note that these top-five most popular landmarks in a given city are different from the ones eliminated during the learning of the recommendation model by the threshold K . This point is important since it demonstrates the ability of our approach to capture non-triviality as a phenomenon. In other words, it is clear that the approach goes beyond merely de-emphasizing specific globally popular

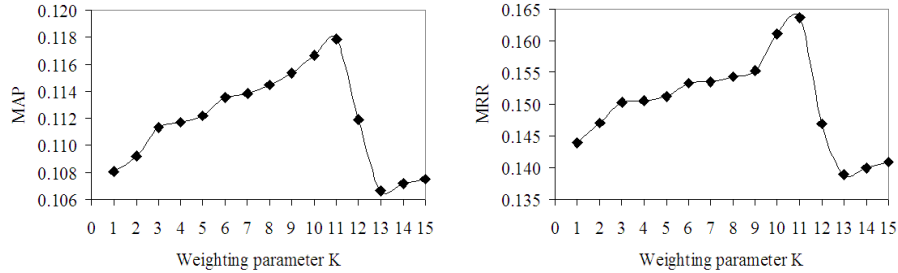


Figure 5.3: Impact of K on recommendation performance in the case that the five most popular landmarks in each city are taken as irrelevant. (Standard deviation of MAP: 0.004; Standard deviation of MRR: 0.007)

landmarks. This result also indicates that for the purpose of recommending non-trivial landmarks, the proposed model could benefit from eliminating the influence of a certain number of most popular landmarks. This observation provides initial evidence that our first research question concerning the effectiveness of the weighting scheme can be answered positively. As expected, there are a certain number of landmarks that are unhelpful for non-trivial landmark recommendation because, according to our initial conjecture, people’s motivations to visit them are general and not topical in nature. These landmarks thus constitute noise within our recommendation scenario and performance improves when they can be eliminated. More evidence on this point will be provided by the experimental results in the next section. The set of these landmarks is, however, relatively small. As can be also seen in Fig. 5.3, when more than a certain number K of top-visited landmarks are eliminated, performance drops off sharply. This observation indicates that excessively eliminating the influence of popular landmarks could also degrade the performance of recommending non-trivial landmarks, since a lot of user preference data associated with popular landmarks may be discarded. In the case of excessively increasing K , the performance remains low and stable with a small fluctuation. For this reason, it is important to attain a tradeoff of the weighting scheme that not only eliminates the influence of top-popular landmarks, but also maintains sufficient user preference data for model learning. Those retained landmarks carry information about helpful patterns, which we assume arise because of topical user motivation to visit landmarks. Note that for the purposes of personalized landmarks, it is not necessary to formulate an explicit understanding of what it means for a user to be motivated by interest in a particular topic to visit a landmark. Rather, user topical interests remain implicit in the data patterns, which can be effectively exploited by the CF approach without the need for explicit user profiles.

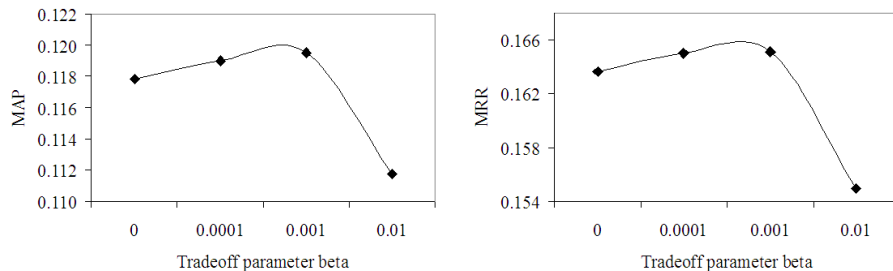


Figure 5.4: Impact of β on recommendation performance in the case that the five most popular landmarks in each city are taken as irrelevant.

Our next experiment in this section investigates the impact of the tradeoff parameter β in the WMF-CR model (as above, cf. Eq. 5.6). β controls the influence of the category-based landmark similarity in the proposed WMF-CR model. We fix K to the optimal value determined during the first experiment ($K = 11$) and vary the value of β in order to observe its influence on the ability of the algorithm to deal with data sparseness. As can be seen in Fig. 5.4, the optimal performance is achieved when $\beta = 0.001$. In the case of $\beta = 0$, WMF-CR does not exploit the category-based landmark similarity. The result indicates that the integration of landmark categories introduces performance improvement to WMF-CR. Again, additional evidence will be provided by the experimental results in the next section. Note that the optimal value β is collection-dependent, i.e., it needs to be tuned for different datasets, and it depends on the scale and the sparseness of both the user-landmark preference matrix and the category-based landmark similarity matrix. For this reason, the absolute value of β cannot reflect the proportions of contribution from each of the two matrices. We also notice Fig. 5.4 that when further increasing the tradeoff parameter, i.e., biasing the learning of latent landmark features towards category-based landmark similarity and away from user-landmark preference, the recommendation performance degrades. This effect illustrates the contribution that is made by the user behavior patterns in the user-landmark matrix and the importance of balancing the contribution of this matrix with the contribution of category information.

As a final point in this section, we examine one of our central design choices for WMF-CR, namely, the choice of using a hard cutoff to eliminate the influence of widely-visited, top-ranked landmarks, cf. Eq. 5.3. Top-ranked landmarks are completely eliminated from the learning process by having their weights set to zero. We performed a final exploratory experiment on the validation set to determine if this hard cutoff is indeed the right choice, or whether we should have weakened rather than eliminated the influence of these landmarks. Although the experiment was exploratory in nature, we report it here since it

provides additional evidence that WMF-CR achieves an optimal balance between eliminating and retaining information in the user-landmark matrix. In the exploratory experiment we set $K = 1$ and replaced the decision in Eq. 5.3 with the decision in Eq. 5.11, introducing a parameter α instead of 0 in order to slowly reduce the influence of the top-ranked landmark.

$$W_{ij} = \begin{cases} \alpha, & R_{ij} > 0 \wedge \text{poprank}(j) \leq K \\ 1, & R_{ij} > 0 \wedge \text{poprank}(j) > K \end{cases} \quad (5.11)$$

We found a slight trend towards dropping performance as α was increased from 0 to 1 and on this basis concluded that our decision to set $\alpha = 0$ was the correct one.

5.5.3 Evaluation

In this subsection, we analyze the performance of the proposed WMF-CR approach for the users in the test set, targeting personalized landmark recommendation. This analysis also includes an investigation of the influence of highly popular, frequently-visited landmarks and the influence of the number of cities/landmarks a user has visited on the recommendation performance. For comparison purposes, we also present the performance of several alternative approaches that we adopt as baselines:

- **PopRec:** Landmarks are recommended to users based on their popularity, which is defined in terms of the number of users who visited them in the training set. It is a non-personalized recommendation approach: for a given city, the same recommendations are always generated independently of the target user.
- **PureSVD:** The pure Singular Value Decomposition (SVD) approach is used to decompose the user-landmark preference matrix in order to obtain the latent user and item features, which are further used to generate the recommendations for each user [33].
- **MF:** The basic matrix factorization approach in Eq. (5.1) is included to represent a state-of-the-art CF approach [75]. Note that both PureSVD and MF only learn the latent user and item features from the user preference data.
- **CRMF:** This is the category-regularized matrix factorization approach that learns latent features from both the user preference data and the category-based landmark similarity [149]. Note that it is equivalent to WMF-CR (cf. Eq. (5.6)) in the case of parameters $\beta = 0.001$ and $K = 0$,

i.e., equally weighting the influence of landmarks with different popularity in the recommendation process.

- **WMF**: This is the weighted matrix factorization approach that learns latent features from the user preference data and weights the influence of different landmarks in terms of their popularity, as shown in Eq. (5.2) and (5.3). Note that it is equivalent to WMF-CR (cf. Eq. (5.6)) in the case that the parameter $\beta = 0$, i.e., it is a special case of WMF-CR in which the external landmark category information is not exploited in the recommendation process.

For WMF-CR and for the approaches related to it, we adopted the optimal values of the parameters that we had determined using the validation set as described in Section 5.5.2.

During our exploratory experimentation we also tried some traditional memory-based CF approaches (e.g., item-based CF by [38]) and wanted to deploy them as additional reference methods. However, we found the performance of these approaches was much worse than the other baseline approaches listed above. A possible reason might be that the memory-based approaches severely suffer from the data sparseness in the scenario of landmark recommendation. For this reason, we limited the comparative study to the above baselines.

Recommending non-trivial landmarks. By assuming that the numbers of most popular (and therefore irrelevant) landmarks may be different from case to case and by letting this number vary between 0 and 10, we can observe the performance of the proposed WMF-CR and its relative improvement over the baseline approaches, as shown in Table 5.3. Note that we use 0 (in the first column) to denote the case where no assumption of the relevance of most popular landmarks is made, and we measure the performance according to the ground truth in the test set. The results of our experiments on recommending non-trivial landmarks are treated in the remainder of this section.

First, as can be seen from Table 5.3, PopRec outperforms all the other approaches under the condition that no popular landmarks are assumed irrelevant. Recall, however, we are mainly concerned about the recommendation performance for landmarks less frequently visited by the general population, i.e., those a traveler may not know about beforehand. If just a few most popular landmarks in each city are ignored, we can see WMF-CR outperforms PopRec by a generous margin. For instance, WMF-CR outperforms PopRec over 40% in terms of MAP and over 100% in terms MRR in the case in which the top-4 most popular landmarks in each city are considered irrelevant. Note that all the improvements reported in our experiments are statistically significant according to the Wilcoxon signed rank significance test with $p < 0.05$ measured across all the users in the test set.

Second, we can observe that WMF-CR significantly outperforms CRMF in recommending non-trivial landmarks, i.e., up to ca. 10% in MAP and up to ca. 15% in MRR. This observation indicates that WMF-CR indeed contributes to improving the performance of recommending non-trivial landmarks by means of reducing the influence of highly popular landmarks. We can confirm a positive answer to our first research question about the effectiveness of the use of the weighting scheme. Note that we can also observe that similar improvements are achieved by WMF over PureSVD and MF, indicating that significant influence on performance can be still introduced by the weighting scheme in the case that the landmark categories are not exploited.

Third, we can observe that WMF-CR significantly improves over PureSVD, MF and WMF in recommending non-trivial landmarks, i.e., up to ca. 25% in MAP and up to ca. 35% in MRR over PureSVD; up to ca. 20% in MAP and up to ca. 35% in MRR over MF; and up to ca. 6% in MAP and up to ca. 7% in MRR over WMF. These improvements indicate that the category-based landmark similarity exploited in WMF-CR indeed contributes to alleviating data sparseness. We can confirm a positive answer to our second research question on the ability of WMF-CR to address the problem of sparse data. Note that we can also observe that improvement is achieved by CRMF over MF and PureSVD, indicating that substantive influence on performance can be still introduced by exploiting landmark categories even in case the weighting scheme is not involved.

Recommendations for users with limited travel experience. We divide the users from the test set into two groups according to the number of cities they have visited. For the purpose of counting the cities that a user has visited, we ignore that user’s target city. Out of 2807 users in the test set, over half of them (1589 users) have photos from only one city, and the rest (1218 users) have photos from visits to more than one city. Note that the distribution of the number of cities that users visit is skewed. For example, only 10 users visited more than 10 cities.

We show the recommendation performance of different approaches for the two groups of users in Table 5.4, given the condition that the top-5 most popular landmarks in each city are considered irrelevant. As can be observed, users with relatively more travel experience can consistently benefit more from recommendations by different approaches, indicating the user travel experience has significant impact on recommendation performance. The proposed WMF-CR approach achieves the best performance of all the approaches for the users that have visited more than one city, leading to, for example, improvement of ca. 40% in MAP and ca. 90% in MRR over PopRec, ca. 14% in MAP and ca. 17% in MRR over MF, and ca. 4% in MAP and ca. 5% in MRR over WMF. In comparison, we can also observe that the relative improvement of WMF-CR

Table 5.3: MAP comparison between WMF-CR and baseline approaches under removal of the x (first column) most popular landmarks in each city. Improvements achieved by WMF-CR over other baseline approaches are statistical significant, according to Wilcoxon signed rank significance test with $p < 0.05$ measured across all the users in the test set.

(a) Results of MAP.						
x	PopRec	PureSVD	MF	CRMF	WMF	WMF-CR
0	0.402	0.327	0.377	0.384	0.266	0.274
1	0.224	0.233	0.238	0.240	0.234	0.218
2	0.153	0.166	0.168	0.170	0.185	0.171
3	0.121	0.132	0.138	0.140	0.161	0.150
4	0.100	0.111	0.123	0.126	0.140	0.144
5	0.085	0.092	0.102	0.106	0.115	0.117
6	0.073	0.080	0.083	0.085	0.093	0.095
7	0.064	0.073	0.075	0.080	0.085	0.088
8	0.058	0.067	0.066	0.070	0.076	0.079
9	0.052	0.061	0.060	0.064	0.069	0.071
10	0.048	0.056	0.056	0.060	0.064	0.068

(b) Results of MRR.						
x	PopRec	PureSVD	MF	CRMF	WMF	WMF-CR
0	0.550	0.472	0.516	0.535	0.380	0.392
1	0.272	0.330	0.316	0.321	0.333	0.310
2	0.174	0.226	0.222	0.225	0.268	0.249
3	0.135	0.176	0.185	0.185	0.236	0.222
4	0.108	0.145	0.167	0.170	0.204	0.215
5	0.090	0.118	0.135	0.139	0.156	0.161
6	0.078	0.097	0.093	0.098	0.112	0.116
7	0.069	0.087	0.084	0.091	0.103	0.107
8	0.062	0.079	0.073	0.079	0.090	0.096
9	0.055	0.071	0.065	0.070	0.082	0.087
10	0.051	0.064	0.062	0.068	0.079	0.084

over baseline approaches is in nearly the same magnitude for users that only visited one city, e.g., ca. 37% in MAP and 73% in MRR over PopRec, ca. 13% in MAP and ca. 17% in MRR over MF, and ca. 4% in MAP and ca. 4% in MRR over WMF. Note that the magnitude of relative improvement introduced by the proposed WMF-CR approach over most of the baselines is not severely degraded for users who only visited one city. As a whole, these results imply that the proposed WMF-CR approach could be particularly beneficial for users that only have limited travel experience, indicating a positive answer to our third research question on the benefits of WMF-CR for users with limited travel experience.

We further divide the 1589 users who only visited one city into groups according to the number of landmarks they visited. We thereby also assume that users with fewer visited landmarks have less travel experience. We find that most of these users visited a very limited number of landmarks, e.g., over one-third only visited one landmark in a city, and users who visited a lot of landmarks are in

Table 5.4: Performance comparison for the users who visited only one city (1) and more than one city (>1) in the past. Results are achieved under the condition that the top-5 most popular landmarks in each city are taken as irrelevant.

(a) Results of MAP.						
Users with num. cities visited	PopRec	PureSVD	MF	CRMF	WMF	WMF-CR
1 (1589 users)	0.079	0.085	0.096	0.099	0.104	0.108
>1 (1218 users)	0.092	0.101	0.114	0.116	0.125	0.130
(b) Results of MRR.						
Users with num. cities visited	PopRec	PureSVD	MF	CRMF	WMF	WMF-CR
1 (1589 users)	0.085	0.109	0.126	0.129	0.141	0.147
>1 (1218 users)	0.095	0.130	0.153	0.154	0.170	0.179

Table 5.5: MAP comparison for the users who visited only one city, in terms of the number of their visited landmarks. Results are achieved under the condition that the top-5 most popular landmarks in each city are taken as irrelevant.

(a) Results of MAP.						
Users with num. visited landmarks	PopRec	PureSVD	MF	CRMF	WMF	WMF-CR
1 (676 users)	0.073	0.084	0.093	0.097	0.103	0.108
2~5 (635 users)	0.084	0.083	0.101	0.104	0.111	0.114
6~10 (153 users)	0.074	0.084	0.081	0.084	0.088	0.088
>10 (125 users)	0.094	0.095	0.093	0.096	0.101	0.102
(b) Results of MRR.						
Users with num. visited landmarks	PopRec	PureSVD	MF	CRMF	WMF	WMF-CR
1 (676 users)	0.081	0.107	0.116	0.120	0.134	0.142
2~5 (635 users)	0.089	0.104	0.137	0.139	0.150	0.155
6~10 (153 users)	0.081	0.117	0.122	0.124	0.132	0.134
>10 (125 users)	0.096	0.118	0.125	0.131	0.148	0.150

minority, e.g., only 125 users visited more than 10 landmarks in a particular target city. The performance for these different groups of users is shown in Table 5.5.

As can be seen, the highest relative improvements achieved by WMF-CR is for the group of users who only visited one landmark, e.g., ca. 48% over PopRec, ca. 11% over CRMF and ca. 5% over WMF in MAP; ca. 75% over PopRec; ca. 18% over CRMF; and ca. 6% over WMF in MRR. These results again indicate that WMF-CR could be particularly helpful for users with very limited travel experience, again supporting a positive answer to our third and final research question.

We also notice in the tables WMF-CR outperforms other approaches for other groups of users who have visited more than one landmark, although the magnitude of relative improvement is lower than for users who have visited only one landmark. These improvements support our conclusion that WMF-CR benefits

from alleviating data sparseness and biasing recommendation process in terms of landmark popularity and, as such, constitutes a highly effective technique for personalized landmark recommendation.

As a final note, we examine one particular user in more detail, who has only one city in her travel history (Barcelona) and has only visited two landmarks there ('Park Guell' and 'Sagrada Familia'). This case illustrates how WMF-CR works for users with very little travel experience. We generate top-10 recommendations for this user for her target city (London) using the different recommendation approaches. The results lists are given in Table 5.6. As can be seen, the approaches that do not specifically target recommending non-trivial landmarks, such as PureSVD, MF and CRMF, are heavily influenced by the most popular landmarks, e.g., at least two of the top-5 recommended landmarks are among the top-5 most popular landmarks in London. CRMF promoted the recommendation of 'London Zoo' probably because the underlying categories allow the algorithm to semantically relate 'London Zoo' to 'Park Guell'. This case suggests that landmark categories can indeed improve recommendation quality. Both WMF and WMF-CR managed to move less popular landmarks towards the top of the list. We can observe that the top-5 most popular landmarks in London are not recommended in either of the top-10 list. In addition, WMF-CR succeeded in recommending the relevant landmarks on the top, benefiting from not only the bias introduced towards non-trivial landmarks, but also the category information. It is interesting to note that the users two visited landmarks ('Park Guell' and 'Sagrada Familia') are among the most popular landmarks in Barcelona, but that the WMF-CR is able to successfully generate recommendations on the basis of this evidence of user interest. This example supports our conjecture that there is not a hard difference between landmarks that users visit motivated by their own topical interest and landmarks that users visit motivated by a general desire to travel. Here, relatively popular landmarks, that could also have been visited due to a general desire to travel, prove helpful in generating the recommendation.

5.6 Conclusion and Future Work

In this paper, we put forward a comprehensive approach to the new research challenge of personalized landmark recommendation based on geotagged photos from photo-sharing sites. Our formulation of the personalized landmark recommendation task includes the criterion that recommended landmarks should be 'non-trivial'. In other words, the system should not recommend landmarks that users could discover via conventional means. Our approach, called WMF-CR, incorporates *weighted matrix factorization* and *category-based regularization*. Weights are incorporated into the matrix factorization approach in order to re-

Table 5.6: Recommendations in London for a user who had only visited “park guell” and “sagrada familia” in Barcelona in the past. The ground truth withheld for this user are “british museum” and “london zoo”.

Rank	PopRec	PureSVD	MF	CRMF	WMF	WMF-CR
1	london eye	big ben	tate modern	tate modern	tate modern	british museum
2	big ben	tower bridge	big ben	tower bridge	hyde park london	london zoo
3	tower bridge	buckingham palace	buckingham palace	big ben	covent garden	tate modern
4	westminster	tate modern	hyde park london	trafalgar square	london zoo	canary wharf
5	trafalgar square	trafalgar square	tower bridge	london eye	kensington gardens	greenwich
6	british museum	westminster	tower of london	hyde park london	lloyd's of london	hyde park london
7	tate modern	canary wharf	trafalgar square	buckingham palace	british museum	natural history museum
8	buckingham palace	london bridge	westminster	london zoo	serpentine gallery	buckingham palace
9	covent garden	piccadilly circus	london eye	battersea power station	buckingham palace	tower of london
10	tower of london	st paul's cathedral	natural history museum	canary wharf	battersea power station	covent garden

inforce the importance of non-trivial landmarks for making non-trivial recommendations. Category-based regularization integrates information on topical-level similarities between landmarks by exploiting landmark category information available from Wikipedia. The computational complexity of WMF-CR is linear with the number of observed preferences in the user-landmark preference matrix and the number of non-zero similarities in the category-based landmark similarity matrix, meaning that it easily scales to large datasets.

The experimental results demonstrate that WMF-CR is capable of producing non-trivial personalized landmark recommendations. WMF-CR shows improved performance over a baseline based exclusively on popularity, over a conventional singular value decomposition approach (PureSVD) and over a standard matrix factorization approach (MF). Further, the addition of weights (WMF) or the use of category regularization (CRMF) both provide good performance when used separately. However, the best overall performance is achieved by the full WMF-CR approach. Additional evaluation shows that WMF-CR is particularly helpful for users with limited travel experience, i.e., in terms of either the number of cities or landmarks visited before. With these experiments we demonstrate that WMF-CR is indeed able to address the issue of sparseness of landmark data within Flickr.

Our future work will address the following challenges. First, although WMF-CR has clearly demonstrated the power of the assumption that the number of photos taken in the general location of a landmark reflects underlying interest that is useful for prediction, we would like to further refine the way in which our method captures user interest. Other factors that could be taken to reflect to user interest are: the distance of the photos from the landmark, the time that the user spent at the landmark as represented by the temporal spread of the photos, the detail with which the user has annotated the photos. In addition, it would be also interesting to exploit other resources to define the triviality of landmarks and extend our comparative study. Second, Wikipedia contains a large amount of semantic information concerning landmarks that goes above and beyond their topical categories. We would like to exploit these sources of information in order to enhance our ability to model topical relatedness between landmarks. Examples include people associated with places and times in history where places were particularly important. Such approaches could be implemented efficiently using DBpedia (<http://dbpedia.org>), a structured information resource extracted from Wikipedia. Semantic Web technologies could be used for semantically richer encoding of categorical information, introducing into the model the means to capture a finer-grained representation of user interests. This type of user interest representation may also contribute to landmark recommendation, in a similar way of landmark similarity. Third, the categories of landmarks we exploit are extracted from an external resource, i.e., Wikipedia. We could also exploit resources from within the photo-sharing

site in order to determine higher-level similarity between landmarks. Examples of promising internal resources are user annotations for individual photos taken at a landmark and the content features of photos. Fourth, another interesting direction is to take the use-context into account for real-time landmark recommendation. Since the application discussed in this paper is closely associated with GPS-enabled mobile devices, it would be promising to investigate approaches that instantly refine landmark recommendations based on the traveler's current location and landmarks visited on the same day. Finally, we would like to evaluate our landmark recommender system with real users in future. One earlier algorithm has been evaluated with a small-scale user study on a prototype system [68], in which the concept of off-the-beaten-track recommendations was highly appreciated by the users. It is interesting to investigate how user satisfaction could be improved by the proposed system in this paper. In sum, the new task of personalized landmark recommendation based on images from photo-sharing sites can be effectively addressed with our proposed WMF-CR approach and at the same time, the approach opens vistas for extension that are promising for future exploration.

Chapter 6

Optimizing MAP for Context-aware Recommendation

In this chapter, we tackle the problem of top-N context-aware recommendation for implicit feedback scenarios. We frame this challenge as a ranking problem in collaborative filtering (CF). Much of the past work on CF has not focused on evaluation metrics that lead to good top-N recommendation lists in designing recommendation models. In addition, previous work on context-aware recommendation has mainly focused on explicit feedback data, *i.e.*, ratings. We propose TFMAP, a model that directly maximizes Mean Average Precision with the aim of creating an optimally ranked list of items for individual users under a given context. TFMAP uses tensor factorization to model *implicit feedback* data (*e.g.*, purchases, clicks) with contextual information.

The optimization of MAP in a large data collection is computationally too complex to be tractable in practice. To address this computational bottleneck, we present a fast learning algorithm that exploits several intrinsic properties of average precision to improve the learning efficiency of TFMAP, and to ensure its scalability. We experimentally verify the effectiveness of the proposed fast learning algorithm, and demonstrate that TFMAP significantly outperforms state-of-the-art recommendation approaches.

This work has been published as “TFMAP: Optimizing MAP for top-N context-aware recommendation”, by Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, A. Hanjalic, and N. Oliver, in Proc. of ACM SIGIR '12 [140].

6.1 Introduction

Collaborative Filtering (CF) methods are at the core of most recommendation engines. Most of the data traces left by online users come in the form of implicit feedback, *i.e.*, we know which items a user interacted, *e.g.*, purchased, used, or clicked, etc., and possibly also the count of each interaction; however, we do not have an explicit rating, *i.e.*, a relevance score, that represents the strength of the user's interest in that item [60]. Learning the suggestion function from implicit feedback data, such as purchase or usage logs, can either be considered a classification problem, where items are classified to relevant or irrelevant, or a ranking problem where an optimal list of items is to be computed.

Top-N recommendation has recently attracted increased research interest because it generates a ranked list of results, which is directly connected to the end-user satisfaction [33]. Conventionally, recommender systems have been optimized to produced scores. A predicted score reflects the system's hypothesis of the strength of a particular user's preference for a particular item. In an overwhelmingly large number of recommender system use scenarios, users do not want preference strength information on all the items in the collection, but rather a compact list of top recommended items.

Although ranking-oriented CF approaches have been proposed for explicit feedback domains, *e.g.*, EigenRank [90] and CoFiRank [182], those approaches are difficult to apply to implicit feedback domains, since they require training examples that are derived from the users ratings on various items. In particular, implicit feedback is often binary in nature. We notice that in top-N recommendation, the quality of a recommendation list that contains items of binary relevance can be quantified using Mean Average Precision (MAP), a well known evaluation measure in the information retrieval (IR) community. MAP provides a single-figure measure of quality across recall levels, has especially good discrimination and stability properties, and roughly corresponds to the average area under the precision-recall curve [101]. It is thus a good measure of performance when a short list of the most relevant items is shown to users [139]. A state-of-the-art approach, Bayesian Personalized Ranking (BPR) [126], has been recently proposed to train recommendation models by optimizing the measure of the Area Under the ROC Curve (AUC), which is based on pairwise comparisons between items. Note that in the AUC measure mistakes at the top of the list carry equal weight to mistakes in the bottom of the recommendation list. In contrast to AUC, MAP is a list-wise measure, for which mistakes in the recommended items at the top of the list carry a higher penalty than mistakes at the bottom of the list [35, 193]. Users typically consider only few (5 -10) top-ranked items in the recommendation list, it is thus particularly important to get the recommendations at the top of the list right. The top-heavy bias of

MAP is thus particularly important in the recommendation problem. For this reason, we propose a recommendation model for implicit feedback domains by directly optimizing MAP.

Typically, recommender systems have access to additional information about the user-item interactions, such as the *context* that is associated with the user-item interaction [1]. The context could be the location where the user listened to a song on his/her mobile phone or the time of the user-item interaction. Context-aware recommendations (CARs) are a new paradigm that can significantly improve the recommendation relevance and quality, compared to conventional recommendations solely based on user-item interactions [1, 11, 65, 127]. In this chapter we present a generic CF model that is based on a generalization of matrix factorization to address context-aware recommendations. We extend the concept of matrix factorization to *tensor factorization*. A tensor is a generalization of the matrix concept to multiple dimensions. In the example above the user-item two-dimensional matrix is converted into a three-dimensional tensor of user-item-location interactions (see Figure 6.1).

Two key issues need to be considered in CARs: (1) *Context Integration*. The contextual information needs to be integrated in the recommendation model to be able to benefit the quality of the recommendation; and (2) *Optimization Function*. The recommendation model needs to be optimized under an objective function that corresponds to the recommendation quality for each user under each given context. Previous work in CARs has extensively studied the *context integration* issue, such as using tensor factorization [65] (TF) and factorization machines [127]. However, the second issue (*optimization function*) has only been addressed in a simplistic way. In the work of [65] and [127], the objective function in the recommendation model consists of minimizing the rating prediction error. This is an effective strategy where explicit feedback data is available from users, however, optimizing this objective is infeasible for scenarios with only implicit feedback data. In these scenarios, the quality of a recommendation list for a user is solely dependent on the positions of the relevant items in the list under the given context.

Here we propose a new context-aware recommendation approach based on tensor factorization for MAP maximization (TFMAP) that is designed to work with implicit feedback datasets. Taking insights from the area of *learning to rank*, TFMAP directly optimizes MAP for learning the model parameters, *i.e.*, latent factors/features of users, items and context types, which are then used to generate item recommendations for users under different types of context.

Directly optimizing MAP across all the users in a data collection is an expensive and non-trivial task. Therefore, we also propose a fast learning algorithm that exploits several properties of the average precision (AP) measure. We show that the computational complexity of the fast learning algorithm for TFMAP

is linear in the number of observed items in a given data collection. Our contributions in this chapter can be summarized as: 1) We propose a new generalized CF approach, TFMAP, that directly optimizes for MAP and leverages contextual information when available. We demonstrate that TFMAP outperforms state-of-the-art context-aware and context-free approaches. We observe significant improvements not only in MAP but also in precision at the top-N ranked recommendations. 2) To the best of our knowledge, TFMAP is also the first approach that can exploit datasets with implicit user feedback and contextual information. 3) We propose a fast learning algorithm that ensures the scalability of TFMAP and that exploits several properties of the AP measure.

The paper is organized as follows: in Section 6.2 we discuss the most relevant previous work and position our paper with respect to it. The research problem and the terminology used throughout the paper are presented in Section 6.3. In Section 6.4, we present the detail of TFMAP and the fast learning algorithm. Our experimental evaluation is reported in Section 6.5. Finally, Section 6.6 summarizes our main contributions and highlights a few areas of future work.

6.2 Related work

The work in this chapter closely relates to three research areas: CF with implicit feedback, context-aware recommendation, and learning to rank. In the following, we present the most relevant related work in each of them.

CF with Implicit Feedback. Most CF approaches in the literature deal with the rating prediction problem, as defined in the Netflix prize competition¹. A common approach to CF is to fit a latent factor model to the data, *e.g.*, latent semantic models [58, 152], and matrix factorization models, which learns a latent feature/factor vector for each user and item in the dataset such that the inner product of these features minimizes an explicit or implicit loss function [12]. Factor models have been shown to perform well in terms of predictive accuracy and scalability [3, 75, 134].

One of the first studies that used latent factor models for large implicit feedback datasets was introduced in [60]. It uses a least squares loss function and exploits the structure of the data (dominated by zero entries that correspond to negative preference), such that observed user-item interactions are weighted proportionately to the count of the interactions. Some extensions following this approach are introduced in [113] and [154]. In [126] a factorization approach based on the optimization of a smoothed pairwise ranking objective function was proposed. Optimizing the proposed objective function corresponds to maximizing

¹<http://www.netflixprize.com/>

the AUC. In this chapter, we propose to learn a recommendation model by optimizing MAP, whose top-bias property is a significant advantage over AUC for recommender systems, as discussed in Section 6.1. In addition, our work is substantially different from the aforementioned work, since various types of contextual information are exploited for the recommendation.

Context-aware recommendation (CAR). Early work in CAR utilized contextual information for pre-processing, where context drives data selection, or post-processing, where context is used to filter recommendations [1, 11]. Recent work has focused on building models that integrate contextual information with the user-item relations and model the user, item and context interactions directly. Two state-of-the-art approaches have been proposed to date, one based on tensor factorization [65, 185] and the other on factorization machines [127] (FM). However, both approaches have been designed for the *explicit* rating prediction problem.

In this chapter we utilize a tensor factorization approach, *i.e.*, the CANDECOMP/PARAFAC (CP) model [69], to represent the interactions among the user, the item and the context type. Our approach includes two substantial innovations, compared to the state of the art in CARs: (1) It targets recommendation scenarios with implicit feedback; and (2) it takes the evaluation metric (MAP) into account for learning the recommendation model.

Note that recommendation approaches have been proposed to take into account additional information (also referred as metadata, side information, or attributes) about users or items, *e.g.*, collective matrix factorization [155], localized factor models [4] and graph-based approaches [72]. However, this type of information would go beyond our definition of “context”, since we refer to context as information that is associated with both the user and the item at the same time. Finally, note that a recommended item set from a recommender is regarded as the “context” of user choice in the work of [189]. However, this type of context is still extracted from the user-item relations, and thus, does not fall in the scope of the context studied in this chapter.

Learning to Rank. Learning to rank has been an attractive research topic in both the machine learning and the information retrieval communities [93]. Our work in this chapter is closely related to recent research where proxies for common IR evaluation measures, such as NDCG and MAP, are used as the objective functions. The main difficulty of directly optimizing evaluation measures lies in their non-smoothness [17], *i.e.*, they are dependent on the rank values of ranked documents/items but not directly on the predicted relevance scores.

Ranking approaches can be broadly classified into two categories, those that implicitly optimize the IR measure and those that formulate an explicit ap-

proximation of the measure. LambdaRank [17] is a popular implicit optimization method, which was proposed to apply gradient descent on an implicit loss function, that is related to IR measures. Methods that explicitly optimize IR measures include structured estimation techniques [171] that minimize convex upper bounds of loss functions based on evaluation measures [187], *e.g.*, SVM-MAP [193] and AdaRank [186]. In the case of CF, CoFiRank [182] introduced a matrix factorization method where structured estimation was used to minimize over a convex upper bound of NDCG. SoftRank [169] was the first approach that proposed an explicit smoothed version of an evaluation measure, in which a rank distribution was employed, resulting in the expected values of document ranks that are smooth to the predicted relevance scores. In addition, a more general extension of SoftRank was presented by Chapelle et al. [26].

In this chapter, we also employ an explicit approximation of MAP, which is a smooth function of model parameters. Our work is different from aforementioned research, since we target context-aware recommendation rather than query-document search, and we propose a fast learning algorithm, which is critical for large-scale recommender systems.

6.3 Problem and Terminology

The research problem studied in this chapter is stated as follows: *Given implicit feedback and contextual information on user-item interactions, recommend to each user and under a given context, an optimal (from a MAP perspective) item list.*

We denote the implicit feedback data from M users to N items under K types of context as a binary tensor Y , *i.e.*, a 3-dimensional tensor, with $M \times N \times K$ entries which are denoted with y_{mik} : (1) $y_{mik} = 1$ indicates that user m has interacted (*i.e.* purchased, used) with item i under context type k . We can thus assume that the user has a preference for this item; and (2) $y_{mik} = 0$ indicates the absence of an interaction and thus the preference of user m to item i under context type k is unknown. $|Y|$ denotes the number of nonzero entries in Y . Y_{mk} denotes a binary vector that indicates the user m 's preference on all the items under context type k .

As mentioned in Section 6.2, the main idea behind factor models is to fit the original user-item interaction matrix with a low rank approximation. In this work we use tensor factorization (TF) as a generalization of the classical matrix factorization methods that accommodates for the contextual information. The latent features are stored in three matrices $U \in \mathcal{R}^{M \times D}$, $V \in \mathcal{R}^{N \times D}$ and $C \in \mathcal{R}^{K \times D}$ that correspond to users, items, and context types, respectively. We use U_m to denote a D -dimensional row vector, which represents the latent

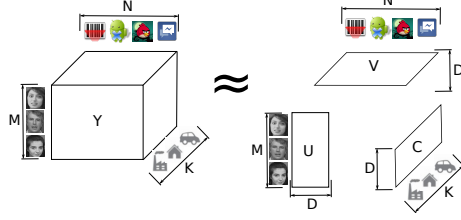


Figure 6.1: CP tensor factorization model.

features for user m . Similarly, V_i represents the latent features of item i , and C_k represents the latent features of context type k .

We use the CP model [69], as illustrated in Fig. 6.1, for tensor factorization, in which user m 's preference to item i under context type k is factorized as the inner product of the latent feature vectors, as shown below:

$$f_{mik} = \langle U_m, V_i, C_k \rangle = \sum_{d=1}^D U_{md} V_{id} C_{kd} \quad (6.1)$$

Based on user's m preference over all the items under context type k , we can then generate a recommendation list by ranking all the items in a descending order of the computed scores. Then, the AP of this list is defined as:

$$AP_{mk} = \frac{1}{\sum_{i=1}^N y_{mik}} \sum_{i=1}^N \frac{y_{mik}}{r_{mik}} \sum_{j=1}^N y_{mjk} \mathbb{I}(r_{mjk} \leq r_{mik}) \quad (6.2)$$

where r_{mik} denotes the rank of item i in the list of user m under context type k and $\mathbb{I}(\cdot)$ is an indicator function, which is equal to 1 if the condition is satisfied, and otherwise 0.

The MAP is defined as the average of AP across all the users and all the context types, as shown below:

$$MAP = \frac{1}{MK} \sum_{m=1}^M \sum_{k=1}^K \frac{\sum_{i=1}^N \frac{y_{mik}}{r_{mik}} \sum_{j=1}^N y_{mjk} \mathbb{I}(r_{mjk} \leq r_{mik})}{\sum_{i=1}^N y_{mik}} \quad (6.3)$$

6.4 TFMAP

In this section, we present the main technical contributions of this chapter: (1) our proposed *smooth approximation of MAP*, its *optimization* and associated complexity analysis; and (2) a novel *fast learning algorithm* for optimizing over the smooth MAP measure in a context-aware setting.

6.4.1 Smoothed Mean Average Precision

It is apparent from Eq. (6.2) and (6.3), that AP (or MAP) depends on the rankings of the items in the recommendation lists. The rankings of the items change in a non-smooth way with respect to the predicted user preference scores and thus, the AP measure ends up being a non-smooth function with respect to the latent features of users, items and context types. We thus cannot use any of the standard optimization methods that require smoothness in the objective function.

As previously mentioned, significant progress has been made in the area of learning to rank regarding the explicit optimization of evaluation metrics, such as MAP. The key issue is to approximate r_{mik} and $\mathbb{I}(r_{mjk} \leq r_{mik})$ in Eq. (6.2) and (6.3) by smoothed functions with respect to the model parameters, *i.e.*, U , V , and C .

Based on insights in [26], we approximate $\mathbb{I}(r_{mjk} \leq r_{mik})$ by the following logistic function:

$$\mathbb{I}(r_{mjk} \leq r_{mik}) \approx g(f_{mjk} - f_{mik}) = g(\langle U_m, V_j - V_i, C_k \rangle) \quad (6.4)$$

where $g(x) = 1/(1 + e^{-x})$. The basic assumption is that the condition of item j being ranked higher than item i is more likely to be satisfied, if item j has relatively higher relevance score than item i . The authors in [26] also proposed a sophisticated approximation for r_{mik} , which, to the best of our knowledge, has not been deployed in practice. In the case of MAP, we argue it is not necessary to approximate r_{mik} , since only $1/r_{mik}$ is in use. For this reason, we propose to directly approximate $1/r_{mik}$ with another logistic function:

$$\frac{1}{r_{mik}} \approx g(f_{mik}) = g(\langle U_m, V_i, C_k \rangle) \quad (6.5)$$

Note that the larger the predicted relevance score f_{mik} the closer $g(f_{mik})$ gets to 1, resulting in a low value of r_{mik} . Reversely, the lower f_{mik} , the larger is r_{mik} . Substituting Eq. (6.4) and (6.5) into Eq. (6.3), we obtain a smoothed approximation of MAP:

$$MAP = \frac{1}{MK} \sum_{m=1}^M \sum_{k=1}^K \frac{1}{\sum_{i=1}^N y_{mik}} \sum_{i=1}^N y_{mik} g(\langle U_m, V_i, C_k \rangle) \sum_{j=1}^N y_{mjk} g(\langle U_m, V_j - V_i, C_k \rangle) \quad (6.6)$$

6.4.2 Optimization

Since Eq. (6.6) is smooth with respect to U_m , V_i , and C_k , we can now optimize it using standard methods, such as gradient ascent. In order to avoid overfitting

, we add the Frobenius norms of the latent factors for regularization. Hence, the resulting TFMAP objective function is given by:

$$L(U, V, C) = \sum_{m=1}^M \sum_{k=1}^K \frac{1}{\sum_{i=1}^N y_{mik}} \sum_{i=1}^N y_{mik} g(\langle U_m, V_i, C_k \rangle) \sum_{j=1}^N y_{mjk} g(\langle U_m, V_j - V_i, C_k \rangle) - \frac{\lambda}{2} (\|U\|^2 + \|V\|^2 + \|C\|^2) \quad (6.7)$$

Note that we neglect the constant coefficient in MAP, since it has no influence on the optimization. Given a set of training data Y , a local maxima of Eq. (6.7) can be obtained by alternatively performing gradient ascent on one of the latent feature vectors at each step, while keeping the other latent vectors fixed. The gradients with respect to U , C , and V are given by Eq. (6.8~6.10).

$$\frac{\partial L}{\partial U_m} = \sum_{k=1}^K \frac{1}{\sum_{i=1}^N y_{mik}} \sum_{i=1}^N y_{mik} [\delta(V_i \odot C_k) + g(f_{mik}) \sum_{j=1}^N y_{mjk} g'(f_{m(j-i)k})(V_j \odot C_k)] - \lambda U_m \quad (6.8)$$

$$\frac{\partial L}{\partial C_k} = \sum_{m=1}^M \frac{1}{\sum_{i=1}^N y_{mik}} \sum_{i=1}^N y_{mik} [\delta(U_m \odot V_i) + g(f_{mik}) \sum_{j=1}^N y_{mjk} g'(f_{m(j-i)k})(U_m \odot V_j)] - \lambda C_k \quad (6.9)$$

$$\begin{aligned} \frac{\partial L}{\partial V_i} &= \sum_{m=1}^M \sum_{k=1}^K \frac{y_{mik}(U_m \odot C_k)}{\sum_{i=1}^N y_{mik}} \sum_{j=1}^N y_{mjk} [g'(f_{mik})g(f_{m(j-i)k}) \\ &\quad + (g(f_{mjk}) - g(f_{mik}))g'(f_{m(j-i)k})] - \lambda V_i \end{aligned} \quad (6.10)$$

where:

$$\begin{aligned} f_{mik} &= \langle U_m, V_i, C_k \rangle, \quad f_{m(j-i)k} = \langle U_m, V_j - V_i, C_k \rangle \\ \delta &= g'(f_{mik}) \sum_{j=1}^N y_{mjk} g(f_{m(j-i)k}) - g(f_{mik}) \sum_{j=1}^N y_{mjk} g'(f_{m(j-i)k}) \end{aligned}$$

The sign \odot denotes element-wise product, and $g'(x)$ denotes the derivative of $g(x)$. Note that since neither U_m or C_k is coupled with other latent feature vectors as in Eq. (6.6), the derivation of Eq. (6.8) and (6.9) is straightforward. However, V_i is coupled with other latent feature vectors in Eq. (6.6), resulting in a more complicate derivation of Eq. (6.10). We leave the detailed derivation of Eq. (6.10) in the Appendix.

In order to understand the practical utility of TFMAP, we analyze the complexity of the learning process for one iteration. Given the data sparseness in the

tensor Y and the fact that we usually have $|Y| \gg M, K$, the computational complexity of calculating the gradients in Eq. (6.8) and (6.9) is $O(D|Y|)$, which is linear to the number of observed user-item interactions in the given tensor. Hence, the computation of the gradients with respect to the latent user features and latent context features is tractable, and able to scale up for large-scale use cases. However, the complexity of Eq. (6.10) is $O(DN|Y|)$. Considering that we usually have $|Y| \gg N$, this complexity is even larger than quadratic in the number of items in the given collection. Thus, the computation of gradients regarding latent item features could be intractable in practice.

In the next section, we propose a novel *fast learning algorithm* to address the computational bottleneck in Eq. (6.10), reducing its complexity to $O(D|Y|)$.

6.4.3 Fast Learning

The proposed fast learning algorithm is outlined in Algorithm 4. Note that according to the definition of AP in Eq. (6.2), it is not necessary to optimize the latent features of all the items in order to maximize AP (as explained below).

The key idea of speeding up the learning process is to optimize, for each fixed pair of user m and context type k , the latent features of only a *set of representative* items, denoted as a buffer B_{mk} .

The gradient of the objective in Eq. (6.7) with respect to the latent features of item i in B_{mk} can be computed as:

$$\begin{aligned} \frac{\partial L}{\partial V_i} = & \sum_{m=1}^M \sum_{k=1}^K \frac{y_{mik}(U_m \odot C_k)}{\sum_{i \in B_{mk}} y_{mik}} \sum_{j \in B_{mk}} y_{mjk} [g'(f_{mik})g(f_{m(j-i)k}) \\ & + (g(f_{mjk}) - g(f_{mik}))g'(f_{m(j-i)k})] - \lambda V_i \end{aligned} \quad (6.11)$$

The computational complexity then depends on the size of the buffer, *i.e.*, the number of items selected for each pair of user-context type. When all items are included in the buffer, Eq. (6.11) is equal to Eq. (6.10), while selecting fewer items in the buffer results in lower complexity.

The key issue with this approach is finding the *right* items to include in the buffer, as the quality of the learning process and hence the resulting model directly depends on the items included in the buffer. The buffer needs to be constructed in such a way that it both reduces the computational complexity of the learning algorithm and conserves the necessary information to yield a high quality model.

Representative Item Selection. *Relevant Items.* For each user in a given context, we first include in the buffer all the items that have been observed by

the user in that context, *i.e.*, for which we have the user’s implicit feedback. These items are the basis for the computation of AP. Note that AP is defined based on the ranks of relevant items. Updating the latent features of relevant items should improve (*i.e.*, reduce) their rankings, thus, resulting in improved AP.

Irrelevant Items. Note that the ranking of irrelevant items influences AP indirectly, since their rankings are relative to the rankings of relevant items. Updating the latent features of irrelevant items will also improve (*i.e.*, raise) their rankings, thus, resulting in overall improved AP.

However, in practice, there are many more irrelevant items than relevant items for a user under a given context. The quantity of irrelevant items thus becomes the computational bottleneck in the learning algorithm of TFMAP.

For this reason, we choose to select only a relatively small number of irrelevant items in the buffer, n_{mk} , for user m and context type k . AP is a top-heavy list-wise ranking measure such that the lower the ranking of an item (the closer it is to the top of the list), the higher its influence in the final score. Top-ranked irrelevant items are the most influential items for AP optimization, yielding the following lemma:

Lemma 6.4.1. *If we try to improve the AP of a ranking list by optimizing (i.e., raising) the ranks of n irrelevant items, then raising the ranks of the top n irrelevant items should yield the largest improvement in AP.*

The proof in the case of $n = 1$ is provided in the Appendix. The proof for the case of $n > 1$ can be obtained in a similar way. Note that we could first sort all the irrelevant items for user m under context k in a descending order, according to the preference scores computed by the current model, *i.e.*, U_m , V and C_k in current iteration, and then select the top-ranked n_{mk} irrelevant items into the buffer.

In this work, we choose the set of irrelevant items in the buffer, n_{mk} , to be equal to the number of observed/relevant items for user m under context k , resulting in a total of $2n_{mk}$ items in the buffer.

We now optimize Eq. (6.7) for the latent features of the items within the buffer only. The complexity of Eq. (6.11) over each iteration is $O(2\tilde{n}^2MKD)$, where \tilde{n} denotes the average number of observed items per user and context type. Note that we have $\tilde{n}MK = |Y|$ and $|Y| \gg \tilde{n}$. Therefore, the complexity of Eq. (6.11) is $O(D|Y|)$, which is linear to the number of observed items in the given collection.

Efficient Buffer Construction. In order to select the top-ranked irrelevant items in each iteration, we need to make a prediction for each item and sort

ALGORITHM 4: Fast Learning TFMAP

Input: Training set Y , regularization parameter λ , sampling size n , learning rate γ , and the maximal number of iterations $itermax$.

Output: The learned latent features U , V , and C .

Initialize $U^{(0)}$, $V^{(0)}$, and $C^{(0)}$ with random values, and $t = 0$;

$p_0 = MAP$ based on Y and $U^{(0)}$, $V^{(0)}$, $C^{(0)}$;

repeat

for $m = 1, 2, \dots, M$ **do**

$U_m^{(t+1)} = U_m^{(t)} + \gamma \frac{\partial L}{\partial U_m^{(t)}}$ based on Eq. (6.8);

for $k = 1, 2, \dots, K$ **do**

$C_k^{(t+1)} = C_k^{(t)} + \gamma \frac{\partial L}{\partial C_k^{(t)}}$ based on Eq. (6.9);

for $m = 1, 2, \dots, M$ **do**

for $k = 1, 2, \dots, K$ **do**

$B_{mk} = \text{BufferConstruct}(Y_{mk}, U_m^{(t)}, V, C_k^{(t)}, n)$;

for $i \in B_{mk}$ **do**

$V_i^{(t+1)} = V_i^{(t)} + \gamma \frac{\partial L}{\partial V_i^{(t)}}$ based on Eq. (6.11);

$t = t + 1$;

$p = MAP$ based on Y and $U^{(t)}$, $V^{(t)}$, $C^{(t)}$;

if $p - p_0 \leq 0$ **then**

break ;

$p_0 = p$;

until $t \geq itermax$;

$U = U^{(t)}$, $V = V^{(t)}$, $C = C^{(t)}$

ALGORITHM 5: BufferConstruct

Input: User m 's preference on all the items under context type k , i.e., Y_{mk} , and U_m , V , C_k , and sampling size n .

Output: B_{mk} .

$B_{mk} = \emptyset$;

$B_{mk} = B_{mk} \cup \{i | y_{mik} = 1\}$;

$n_{mk} = \text{cardinality}(B_{mk})$;

$p = \min_{i, y_{mik}=1} \langle U_m, V_i, C_k \rangle$;

$S = \{i | y_{mik} = 0\} \cap \{i | \langle U_m, V_i, C_k \rangle > p\}$;

Randomly sample n items from S as Q ;

Descendingly sort items in Q , according to $\langle U_m, V_i, C_k \rangle, i \in Q$;

Set top-ranked n_{mk} items in Q as B^- ;

$B_{mk} = B_{mk} \cup B^-$;

them according to the current predicted scores. Considering that most recommender systems contain large numbers of items, the computational cost for the prediction and sorting process would be very high. For this reason, we propose

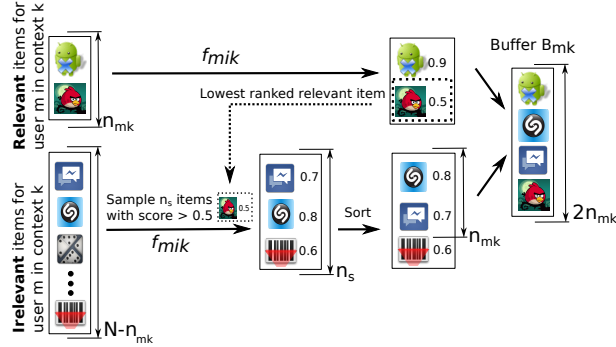


Figure 6.2: Illustration of buffer construction.

to sample a small set of irrelevant items and to select the top-ranked irrelevant items within the sampled set into the buffer.

We can maintain the representativeness of the top-ranked irrelevant items from the sampled set by using a key property of AP: *The items below the last relevant item in a ranked list have no contribution to AP*. This property can be easily understood from the definition of AP (see Equation 6.2).

Therefore, for each user under a given context type, we first find the relevant item with the lowest score. This operation is computationally cheap since the number of relevant items is usually very small. We then sample n_s irrelevant items from those irrelevant items (assuming that most unobserved items are irrelevant) that have higher predicted relevance scores than the minimum predicted relevance score of the relevant items. This sampled set has higher probability to contain the globally top-ranked irrelevant items than a randomly sampled set. Note that the relevance scores are calculated by the model in each iteration. We illustrate the buffer construction for user m under context type k in a single iteration in Fig. 6.2 and Algorithm 5.

In addition, since the model will become more accurate with each iteration, the minimum predicted score of the relevant items will also increase gradually. In other words, the position of the last relevant item in the ranked list will gradually move to the top of the list. As another by-product, this effect also helps to reduce the buffer construction time with each iteration. An experimental analysis confirming this property will be presented in Section 6.5.

Note that the sampling size for the irrelevant items does not only influence the buffer construction time, but also the quality of the learned latent item features. We investigate this tradeoff between buffer size (i.e., computational cost) and performance in Section 6.5.

Termination Criterion. Since Eq. (6.6) is an approximation of MAP for the

training data Y , we can use MAP (given by Eq. 6.3) as another termination criterion apart from conventional criteria, such as the number of iterations or the convergence rate. We stop the optimization process when we observe deteriorating values of MAP. Degrading values of MAP on the training data indicates that further optimizing the approximation of MAP as in Eq. (6.6) may not contribute to raising the true MAP.

6.5 Experimental Evaluation

In this section we present a collection of experiments that evaluate the proposed TFMAP. We first give a detailed description of the datasets and setup that are used in the experiments. Then, we investigate the impact of several parameters in the proposed fast learning algorithm that are critical for TFMAP, as mentioned in Section 6.4.3. Finally, we evaluate the recommendation performance of TFMAP, compared to several baselines, and analyze its scalability.

The experiments were designed to address the following research questions: 1) Does the proposed fast learning algorithm benefit TFMAP in achieving MAP maximization? 2) Does TFMAP outperform state-of-the-art context-aware and context-free approaches? 3) Is TFMAP scalable for large-scale context-aware recommendation?

6.5.1 Experimental Setup

Dataset. The main dataset we use in this chapter is from the *Appazaar* project² [15]. *Appazaar* recommends mobile applications to users from the Android Market. The application usage data is recorded in the form of implicit feedback since *Appazaar* logs which apps are run by each user. In addition, *Appazaar* also tracks available contextual information from the phone sensors, such as motion sensor and GPS. We use two contextual factors in the experiments, i.e., motion (unavailable, slow, fast) and location (workplace, home, elsewhere). Note that both of the contextual factors were inferred from a GPS trace. Hence, the context variable has 9 possible types that take into account all the combinations of the two contextual factors, i.e., $K = 9$ for C in Eq. (6.1). For example, context type “1” denotes that implicit feedback about a user running an application was observed when the user was at work and his/her motion status was unavailable. Finally, we represent one observation in the dataset as a triplet $(UserID, ItemID, ContextTypeID)$. The dataset contains 300469 triplets, 1767 users, 7701 items, 9 combinations of contextual features. On average, there are 18.9 app usage events per user and context type. A more

²<http://appazaar.net/>

detailed description of the dataset and its collection procedure can be found in [15].

Note that conventional CF benchmark datasets, *e.g.*, Netflix dataset, are not enriched with contextual information. Although the *Appazaar* dataset is not as large as these benchmark datasets, it is still much larger than datasets that have been used in previous context-aware recommendation work [65, 127]. Moreover, the datasets previously used in the CAR literature are all based on explicit ratings rather than implicit feedback, thus, not ideal for our study.

Experimental Protocol.

We separate the dataset into a training set and a test set, according to the timestamps. The training set consists of the first 80% implicit feedback data, while the test set contains the remaining 20% data. The target is to use the training set to learn a recommendation model, *i.e.*, U , V and C , which is then used to generate recommendation lists for each user under each type of context.

We use the MAP measure as in Eq. (6.3) to evaluate on the testset \tilde{Y} . Note that in order to have fair comparison with context-free approaches, we only preserve one context type for each user in the test set, *i.e.*, we randomly select one context type for each user in the test set and preserve the user's feedback within the selected context type, while excluding all the user's feedback data under other context types. To further clarify this design choice, we give a negative example in which a user in the test set has implicit feedback on the items under two different types of context. In this case, the MAP of context-aware approaches, such as TFMAP, should be measured according to AP under the two different types of context, while context-free approaches would only calculate AP based on the items and ignore the context. For this reason, our choice is necessary in order to attain fair comparative results to other context-free approaches.

In addition, note that since we only have implicit feedback from users, we cannot treat all the items that have no feedback in the test set as irrelevant/negative ones, in which case the recommendation performance could be severely underestimated. For this reason, we adopt a conventional widely-used evaluation strategy [33, 73], in which we randomly select 1000 items that have no feedback as irrelevant ones for each user in the test set. The performance is measured according to the recommendation list that only contains these 1000 items together with relevant items, *i.e.*, the items for which there is implicit feedback for that user.

In order to carry out our validation experiments, we randomly select 10% of all the implicit feedback data available in the training set. In our validation experiments we investigate the impact of the parameters and the fast learning algorithm in TFMAP.

Finally, note that we empirically tune the following conventional parameters so they yield the best performance in the validation test: regularization parameter $\lambda=0.001$, latent dimensionality $D=10$, and learning rate $\gamma=0.001$.

Setup for Comparison to FM. As mentioned in Section 6.2, the state-of-the-art context-aware approaches, such as FMs [127], are designed to tackle the rating prediction problem (explicit feedback), and hence they are difficult, if not impossible, to apply to implicit feedback data. For this reason, we use another dataset, *Food* dataset [111], which has also been used in the work on FMs [127]. This dataset contains ca. 6K 5-scale ratings from 212 users on 20 menus/items, and each rating is associated with 2 contextual factors, *i.e.* one factor about whether the user’s feeling about hunger is real or virtual (2 values: real, virtual) when she rated a menu, and the other factor about the user’s hunger degree (3 values: normal, hungry and full). By taking into account all the combinations of the two contextual factors, we obtain 6 types of context in the Food dataset.

In our experiments, we randomly select 80% of the ratings as the training set and the remaining ratings as the test set. Items with a rating higher than 3 in the test set are considered to be relevant. Note that a different rating threshold could be set to define the relevant items. Under this setting, we use FM approach to first predict the ratings of the users on the items under each context type, and then generate the recommendation list according to the predicted ratings. For TFMAP, we train the model by converting the training set to an *implicit feedback* dataset, in which each rated item is regarded as an indicator of implicit feedback (*i.e.*, the user tried the food item).

6.5.2 Validation: Impact of Fast Learning

We investigate the properties of the fast learning algorithm in TFMAP, presented in Section 6.4.3. The experimental results reported in this subsection are measured on the validation set previously described.

Impact of Sampling Size. By varying the sampling size in the fast learning algorithm of TFMAP, we investigate the buffer construction time and the performance variation in terms of MAP in the validation set, *i.e.*, an issue discussed in Section 6.4.3. We measure the buffer construction time cumulatively across all the users under all the context types in the training set over one iteration. The result is shown in Fig. 6.3.

Note that the buffer construction time increases almost linearly as the sampling size increases. Hence, with a relatively small sampling size, we could significantly reduce the buffer construction time compared to the case where all the irrelevant items for each user under a given context need to be ranked. For

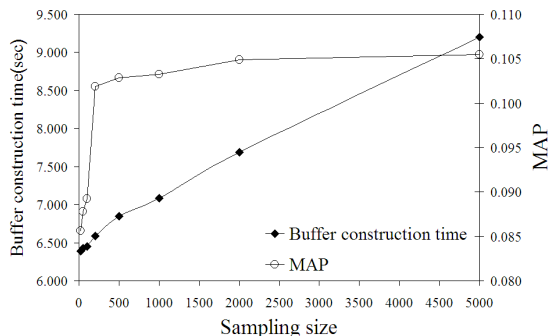


Figure 6.3: The impact of sampling size on buffer construction time and MAP of TFMAP.

example, in the *Appazaar* dataset we have over 7000 items, which means that a sampling size of 200 could save over 50% of the buffer construction time, as illustrated in Fig. 6.3. Also note that the recommendation performance in terms of MAP increases sharply as the sampling size increases up to 200, and then saturates. Therefore, even with a relatively small size of irrelevant items, *e.g.*, 200, (compared to all the irrelevant items), the top-ranked irrelevant items within the sampled set are sufficiently representative to be used for MAP optimization.

In sum, these results empirically verify the selection of a small set of irrelevant items to create the buffer in the fast learning algorithm of TFMAP and justify our algorithm design choices. For the remaining experiments we will keep a sampling size of 200.

Impact of Representative Irrelevant Items. Here we aim to understand the effectiveness of choosing the representative irrelevant items in the buffer. Rather than selecting representative irrelevant items to construct the buffer, an alternative is to use randomly selected irrelevant items. To test the random procedure we abandoned the ordering step of the algorithm and instead we randomly selected n_{mk} irrelevant items from the sampled set of size 200. In this case the accuracy yields a MAP of 0.083, dropping by 18.6% compared to the case where top-ranked irrelevant items are selected, *i.e.*, MAP of 0.102 as shown in Fig. 6.3. Increasing the sampling size further emphasizes the benefit of carefully selecting the representative items. When we choose to sample 5000 irrelevant items, the benefit over the random strategy is 21.7%. This experiment validates the benefit of using representative irrelevant items in the buffer, as discussed in Section 6.4.3.

Effect of the Lowest-ranked Relevant Item. As discussed in Section 6.4.3, it is not necessary to sample from all the irrelevant items in order to construct the buffer for a user in a given context, since the items ranked below the lowest-ranked relevant item have no influence on AP. Thus, the sampling process

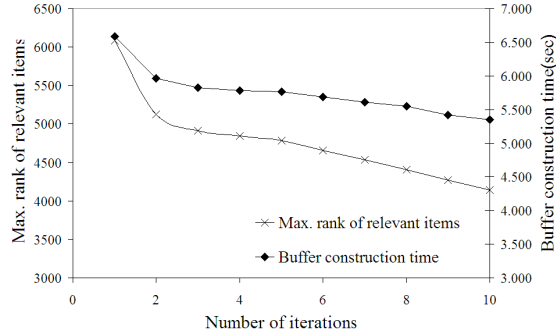


Figure 6.4: The average maximal rank of relevant items and the buffer construction time along iterations.

could be more efficient by neglecting the items ranked below the lowest-ranked relevant item.

Here, we present an experimental study that examines the change of the position of the lowest-ranked relevant item, *i.e.*, the maximal rank of relevant items in a recommendation list, during iterations, and also the change in the corresponding buffer construction time, as shown in Fig. 6.4. Note that this experiment is conducted on the validation set, with sampling size of 200 in TFMAP, and the results shown in Fig. 6.4 are the average values across all the users under all context types in each iteration.

We observe that the maximal rank of relevant items decreases with each iteration as the model is gradually optimized, *i.e.*, the model is more likely to rank relevant items higher in the list along iterations. This observation provides empirical evidence that exploiting the lowest-ranked relevant item in the sampling process does contribute to improving the quality of the representative irrelevant items, and also the efficiency of the buffer construction with each iteration. For example, the buffer construction time reduces by over 10% in the second iteration, compared to the first iteration.

Effectiveness of the Termination Criterion. Our final validation experiment investigates the effectiveness of the proposed termination criterion for the fast learning algorithm, as discussed in Section 6.4.3. We show the MAP measured in both the training (excluding the validation set) and the validation sets across the iterations, as in Fig. 6.5. Both MAP measures gradually improve towards an optimal value with only a few iterations (less than 20), indicating that TFMAP effectively learns latent features for users, items and context types for MAP optimization. Also note that both MAP measures start dropping consistently after a few iterations, indicating that it is effective to use the MAP measured in the training set as a termination criterion for the learning process to avoid model overfitting.

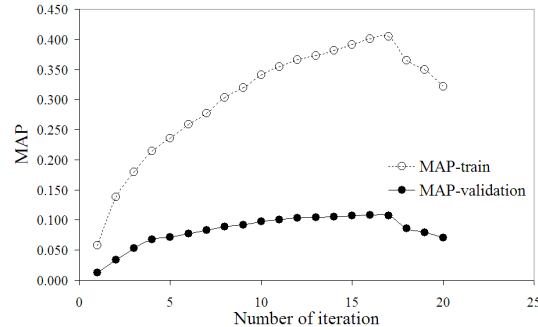


Figure 6.5: The MAP of the training set and the validation set in the learning process

From all the findings described in this section, we can give a positive answer to our first research question.

6.5.3 Performance Comparison

We now compare the performance of TFMAP with that of 5 baseline algorithms, according to the recommendation performance measured on the test set. The baseline approaches involved in this comparative experiment are listed below:

- **Pop.** A naive baseline that recommends items in terms of their popularity (*i.e.*, the number of observations from all the users) under the given context.
- **iMF.** A state-of-the-art CF approach proposed by Hu et al [60] for implicit feedback data.
- **BPR-MF.** Bayesian personalized ranking (BPR) represents another state-of-the-art optimization framework of CF for implicit feedback data [126]. BPR-MF represents the choice of using matrix factorization (MF) as the learning model with BPR optimization criterion. Note that the implementation of this baseline is done with the publicly available software MyMediaLite [42]. Although various learning models are available to be used with BPR, we find BPR-MF gives the best performance.
- **TFMAP-noC.** A variant of the proposed TFMAP, in which contextual information, *i.e.*, C , is not involved in the learning algorithm. Note that iMF, BPR-MF and TFMAP-noC are context-free methods, *i.e.*, the contextual information has no influence on the recommendations to individual users.
- **FM.** Factorization machine (FM) is a state-of-the-art context-aware approach [127]. As mentioned in Section 6.5.1, the comparison between FM and TFMAP is conducted on the *Food* dataset, due to the applicability

of FM. Note that the implementation of FM is done with the publicly available software libFM³.

Based on the Appazaar dataset, the recommendation performance of TFMAP and all the baselines except FM is shown in Table 6.1, from which we obtain three observations.

First, the context-free version of the proposed TFMAP, *i.e.*, TFMAP-noC significantly outperforms the other baselines in terms of MAP. Note that in our experiments, statistical significance is measured based on AP and precision values of all the users in the test set, according to Wilcoxon signed rank significance test with $p < 0.01$. This result indicates that in the case that contextual information is unavailable, directly optimizing MAP as proposed in TFMAP could still lead to substantial improvement over state-of-the-art context-free approaches, such as iMF and BPR-MF.

Second, we can see that both BPR-MF and TFMAP-noC attain dramatic improvement in MAP over the other two baselines, Pop and iMF. As mentioned in Section 6.2, BPR is designed to optimize the evaluation metric AUC. The superior performance of BPR-MF and TFMAP-noC suggests that directly optimizing an evaluation metric that measures the recommendation performance in implicit feedback systems would yield significant improvements in the recommendation performance. In addition, note that TFMAP-noC achieves a significant improvement in MAP of 3% over BPR-MF, and 4% improvement of P@1. This result indicates that optimizing MAP is a better choice for recommender systems than optimizing AUC, since the top-heavy bias in MAP is a critical factor that provides substantial benefit for the recommendation performance.

Third, as can be seen, TFMAP achieves an additional significant improvement over TFMAP-noC, *e.g.*, 5% in MAP and P@1. This result indicates that TFMAP succeeds in utilizing contextual information together with user-item implicit feedback for maximizing MAP. In addition, the exploitation of context could greatly improve implicit feedback recommenders; a similar conclusion was reached by previous work on CAR with explicit feedback [127, 65].

As mentioned before, we compare TFMAP with FM using the Food dataset, according to the protocol described in Section 6.5.1. The results are shown in Table 6.2. As can be observed, TFMAP significantly improves over FM to a large extent, *i.e.*, by more than 40% in MAP, 100% in P@1 and 50% in P@5 and 8% in P@10, showing a great competitiveness for top-N context-aware recommendation. From all the experimental results presented in this section, we confirm a positive answer to our second research question.

³<http://www.libfm.org/>

Table 6.1: Performance comparison of TFMAP and context-free baselines on Ap-pazaar dataset

	MAP	P@1	P@5	P@10
Pop	0.090	0.312	0.292	0.227
iMF	0.577	0.698	0.642	0.583
BPR-MF	0.612	0.800	0.712	0.602
TFMAP-noC	0.629	0.834	0.720	0.602
TFMAP	0.659	0.879	0.732	0.611

Table 6.2: Performance comparison of TFMAP and FM on Food dataset

	MAP	P@1	P@5	P@10
FM	0.152	0.036	0.050	0.055
TFMAP	0.219	0.089	0.075	0.059

6.5.4 Scalability

The last experiment investigates the scalability of TFMAP by measuring the model training time against the amount of data used for training the model. We use from 10% to 100% of the training data (the observed implicit feedback data in the training set) for learning the latent features, and the corresponding training times are shown in Fig. 6.6. Note that we have normalized the training time by the time that is required for training the model with all the data in the training set. It can be observed that the training time increases almost linearly with the amount of the training data, empirically verifying the property of linear computational complexity. This finding also allows us to answer our last research question positively.

6.6 Conclusions and future work

We have presented TFMAP, a novel top-N context-aware recommendation approach for implicit feedback domains. This approach utilizes tensor factorization to model each user’s preference for each item under each type of context, and the factorization model is learned by directly optimizing MAP. We also propose a fast learning algorithm that exploits several properties of AP to keep the complexity of TFMAP linear to the number of implicit feedback data in a given collection, thus, making TFMAP scalable. Our experimental results verify the effectiveness of the proposed fast learning algorithm for TFMAP, and demonstrate that TFMAP could outperform several state-of-the-art context-aware and context-free recommendation approaches.

Taking insights from recent statistical analysis on evaluation measures [179],

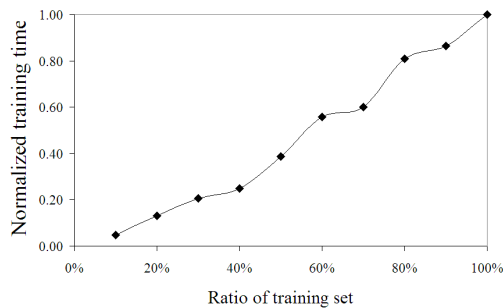


Figure 6.6: Scalability analysis of TFMAP

one line of our future work is to investigate the potential of optimizing other measures for context-aware recommendation, since different measures may represent different aspects of the recommendation quality. Another interesting topic of future work is to integrate contextual information together with metadata of users and items, as discussed in Section 6.2, to further advance the state-of-the-art in recommender systems.

6.A Derivation of Eq. (10)

Note that in the following derivation, we leave out the derivative of the regularization term, i.e., $-\lambda V_i$, due to the space limit.

$$\begin{aligned} \frac{\partial L}{\partial V_i} = & \sum_{m=1}^M \sum_{k=1}^K \frac{y_{mik}(U_m \odot C_k)}{\sum_{i=1}^N y_{mik}} \left(g'(f_{mik}) \sum_{j=1}^N y_{mjk} g(f_{m(j-i)k}) \right. \\ & \left. - g(f_{mik}) \sum_{\substack{j=1 \\ j \neq i}}^N y_{mjk} g'(f_{m(j-i)k}) + \sum_{\substack{p=1 \\ p \neq i}}^N y_{mpk} g(f_{mpk}) g'(f_{m(i-p)k}) \right) \end{aligned}$$

Since we have $g'(-x) = g'(x)$, we obtain:

$$\begin{aligned} \frac{\partial L}{\partial V_i} = & \sum_{m=1}^M \sum_{k=1}^K \frac{y_{mik}(U_m \odot C_k)}{\sum_{i=1}^N y_{mik}} \left(g'(f_{mik}) \sum_{j=1}^N y_{mjk} g(f_{m(j-i)k}) \right. \\ & \left. + \sum_{\substack{p=1 \\ p \neq i}}^N y_{mpk} g(f_{mpk}) g'(f_{m(p-i)k}) - g(f_{mik}) \sum_{\substack{j=1 \\ j \neq i}}^N y_{mjk} g'(f_{m(j-i)k}) \right) \end{aligned}$$

By replacing index “ p ” by “ j ”, we obtain:

$$\begin{aligned} \frac{\partial L}{\partial V_i} = & \sum_{m=1}^M \sum_{k=1}^K \frac{y_{mik}(U_m \odot C_k)}{\sum_{i=1}^N y_{mik}} \left(g'(f_{mik}) \sum_{j=1}^N y_{mjk} g(f_{m(j-i)k}) \right. \\ & \left. + \sum_{\substack{j=1 \\ j \neq i}}^N y_{mjk} (g(f_{mjk}) g'(f_{m(j-i)k}) - g(f_{mik}) g'(f_{m(j-i)k})) \right) \end{aligned}$$

Since the term in the last summation is 0 when $j = i$, we obtain:

$$\begin{aligned} \frac{\partial L}{\partial V_i} = & \sum_{m=1}^M \sum_{k=1}^K \frac{y_{mik}(U_m \odot C_k)}{\sum_{i=1}^N y_{mik}} \left(g'(f_{mik}) \sum_{j=1}^N y_{mjk} g(f_{m(j-i)k}) \right. \\ & \left. + \sum_{j=1}^N y_{mjk} (g(f_{mjk}) g'(f_{m(j-i)k}) - g(f_{mik}) g'(f_{m(j-i)k})) \right) \\ = & \sum_{m=1}^M \sum_{k=1}^K \frac{y_{mik}(U_m \odot C_k)}{\sum_{i=1}^N y_{mik}} \sum_{j=1}^N y_{mjk} \left(g'(f_{mik}) g(f_{m(j-i)k}) \right. \\ & \left. + (g(f_{mjk}) - g(f_{mik})) g'(f_{m(j-i)k}) \right) \end{aligned}$$

6.B Proof of Lemma 1

For the case of $n = 1$, consider that $r = [r_1, r_2, \dots, r_N]$ denotes the current ranks of all N items in a ranked list and r' denotes their ranks after some optimization. We can optimize the latent features of a single irrelevant item a using a very small step size until its rank would increase from $r_a = q$ to $r'_a = q + 1$. Consequently, the rank of item b that was ranked $q + 1$ would now be ranked q , i.e., $r_b = q + 1$ and $r'_b = q$. Now the Lemma can be proved by proving that $\Delta AP = AP_{r'} - AP_r$ is a

non-increasing function of q . It follows that the largest improvement ΔAP can be achieved by raising the rank of the irrelevant item with the lowest rank. Note that this proof is not related to a user or a context, so we simplify the notations. The ΔAP can be expressed as:

$$\Delta AP = \frac{1}{N_R} \sum_{i=1}^N \frac{y_i}{r'_i} \sum_{j=1}^N y_j \mathbb{I}(r'_j \leq r'_i) - \frac{1}{N_R} \sum_{i=1}^N \frac{y_i}{r_i} \sum_{j=1}^N y_j \mathbb{I}(r_j \leq r_i)$$

where N_R denotes the number of relevant items. Since we have the condition: $r_i = r'_i$, if $i \neq a, b$, we can obtain

$$\begin{aligned} \Delta AP &= \frac{1}{N_R} \left(\frac{y_a}{r'_a} \sum_{j=1}^N y_j \mathbb{I}(r'_j \leq r'_a) + \frac{y_b}{r'_b} \sum_{j=1}^N y_j \mathbb{I}(r'_j \leq r'_b) \right. \\ &\quad \left. - \frac{y_a}{r_a} \sum_{j=1}^N y_j \mathbb{I}(r_j \leq r_a) - \frac{y_b}{r_b} \sum_{j=1}^N y_j \mathbb{I}(r_j \leq r_b) \right) \end{aligned}$$

Since we also have $y_a = 0$ as known, we further obtain:

$$\Delta AP = \frac{1}{N_R} \left(\frac{y_b}{r'_b} \sum_{j=1}^N y_j \mathbb{I}(r'_j \leq r'_b) - \frac{y_b}{r_b} \sum_{j=1}^N y_j \mathbb{I}(r_j \leq r_b) \right)$$

Substituting $r_b = q + 1$ and $r'_b = q$, we have:

$$\Delta AP = \frac{1}{N_R} y_b \sum_{j=1}^N y_j \left(\frac{1}{q} \mathbb{I}(r'_j \leq q) - \frac{1}{q+1} \mathbb{I}(r_j \leq q+1) \right)$$

Again, when $j \neq a, b$, we have $r_j = r'_j$, and:

$$\begin{aligned} \mathbb{I}(r'_j \leq q) &= \mathbb{I}(r_j \leq q+1) = 1, \quad \text{if } r'_j \leq q \\ \mathbb{I}(r'_j \leq q) &= \mathbb{I}(r_j \leq q+1) = 0, \quad \text{if } r'_j > q+1 \end{aligned}$$

Accordingly, we obtain:

$$\sum_{j=1, j \neq a, b}^N y_j \left(\frac{1}{q} \mathbb{I}(r'_j \leq q) - \frac{1}{q+1} \mathbb{I}(r_j \leq q+1) \right) = N_q \left(\frac{1}{q} - \frac{1}{q+1} \right)$$

where N_q denotes the number of relevant items within top- q items. Finally, we obtain:

$$\begin{aligned} \Delta AP &= \frac{1}{N_R} y_b \left(\sum_{j=1, j \neq a, b}^N y_j \left(\frac{1}{q} \mathbb{I}(r'_j \leq q) - \frac{1}{q+1} \mathbb{I}(r_j \leq q+1) \right) \right. \\ &\quad \left. + y_a \left(\frac{1}{q} \mathbb{I}(r'_a \leq q) - \frac{1}{q+1} \mathbb{I}(r_a \leq q+1) \right) \right. \\ &\quad \left. + y_b \left(\frac{1}{q} \mathbb{I}(r'_b \leq q) - \frac{1}{q+1} \mathbb{I}(r_b \leq q+1) \right) \right) \\ &= \frac{1}{N_R} y_b \left(N_q \left(\frac{1}{q} - \frac{1}{q+1} \right) + y_b \left(\frac{1}{q} - \frac{1}{q+1} \right) \right) \\ &= \frac{1}{N_R} \frac{N_q y_b + y_b^2}{q(q+1)} \end{aligned}$$

Note that $N_q \leq q$ and $y_b \in \{0, 1\}$. We complete the proof with ΔAP is a non-increasing function of q . A similar proof can be deduced for the case of $n > 1$.

Chapter 7

Future Challenges

In previous chapters of this thesis, we introduced our contribution to the performance improvement of the CF-based recommender systems. The contribution addresses two critical aspects of recommendation, namely the ranking of the recommended items and the context-awareness of the recommendation process.

Regarding the ranking models for recommender systems, we proposed a Unified Recommendation Model (URM) in Chapter 2 and Collaborative Less-is-More Filtering (CLiMF) in Chapter 3, which were developed to improve the recommendation in the scenarios with explicit user feedback data and implicit feedback data, respectively. The proposed methods both achieved significant improvements over the state-of-the-art in their respective scenarios. In our attempt to make the CF-based recommendation more context-aware, we proposed methods for incorporating two types of contextual information into the recommendation process. The first type is the side information collected about the users or items (Chapter 4 and 5) and second type is the information associated with user-item interactions (Chapter 6). Just like in the case of ranking optimization, the proposed methods for context-awareness were shown to outperform the state-of-the-art.

In this chapter we look beyond ranking optimization and context-awareness and identify and investigate other challenges for which we believe to shape the research on recommender systems in the future. We group the future challenges into two main categories, as shown in Fig. 7.1. We refer to the first category as the ‘Challenges of New Conditions and Tasks’. These are the consequence of many new conditions under which recommendation needs to be performed and many new use scenarios that require recommendation to maximally exploit the available information resources. We focus here on the challenges of *social*

This chapter is based on a submission to ACM Computing Surveys, by Y. Shi, M. Larson, and A. Hanjalic, in Mar. 2013.

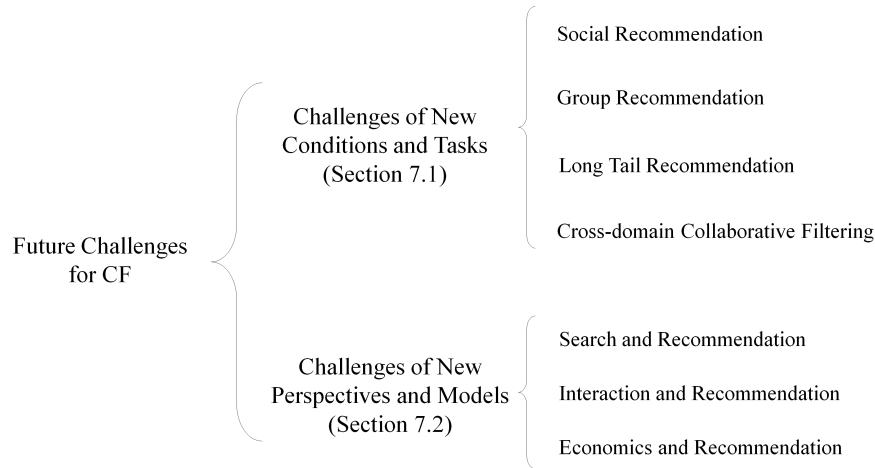


Figure 7.1: Overview of challenges for collaborative filtering

recommendation, group recommendation, long-tail recommendation and cross-domain recommendation. They are covered in detail in Section 7.1. The second category of challenges is what we refer to as the ‘Challenges of New Perspectives and Models’. These challenges, covered in Section 7.2, have their origins outside the domain of recommender systems and emerge from the convergence between recommender systems and other areas, such as search, human-computer interaction and economics.

7.1 Challenges of New Conditions and Tasks

Rapid growth and the emerging new concepts of systems and platforms accommodating recommendation mechanisms have resulted in increasing demands posed on these mechanisms to adjust to new conditions and tasks. Innovative recommendation concepts are required that can operate in the increasingly social web context where much more can be derived about relations between users or between a user and an item than from the traditional user-item matrix. Furthermore, increasing accessibility of the web to new social groups, like elderly people, opens new perspectives for designing recommendation algorithms, like those that can satisfy not a single user, but groups of users, e.g. at elderly homes. In the same way, increasing demands for improved and personalized (mobile) services, like travel location recommendation, force the recommender systems to get the most from the available information resources, for example by focusing on the long tail of the popularity-based item list. Finally, service providers exploiting different commercial domains have discovered a high potential of learning user preferences across these domains, which gives rise to cross-domain recommendation. For each of these challenges we describe in the following its significance, open issues to be addressed by new research and the state-of-the-art approaches that, in one way or the other, have attempted to address these challenges.

7.1.1 Social Recommendation

Significance Social networks can be seen as a valuable source of information about users and items that can benefit CF. Relations between users or between users and items (e.g. derived from a social network graph [72]) can be exploited together with CF approaches for recommending videos, photos, music and news [99]. By adopting the terminology that has already become established in literature regarding the recommender systems that incorporate information derived from social networks, we refer to such systems as *social recommendation* systems [153]. Social recommendation naturally plays a central role in social networks and social media sites. Good examples are contact recommendations *People you may know* that are offered to the users of the LinkedIn professional network (<http://www.linkedin.com/>) and the *Recommendations Bar* offered by the FaceBook social network (<http://www.facebook.com/>). We anticipate that increasing socialization via online platforms will make social recommendation one of the main recommendation mechanisms in the future.

Open Issues Here, we would like to discuss three issues that social recommendation faces and sketch the possibilities for future research directions.

- First, *how do inherent properties of social networks interact with social recommendation?* Existing work in this direction generally neglects the intrinsic nature of social networks and its influence on recommendation. For example, connectivity in online social networks is known to be characterized by *Power Law* distributions [30, 107]. In other words, it is a defining characteristic of social networks that few users have many connections, while many users only have few connections. If social recommendations are influenced by users' connectivity degrees, some users stand to benefit much more than others from the integration of social network into recommender system algorithms. The impact of varying levels of connectivity on recommender system performance is, of yet, only poorly understood and deserves further investigation. Similar questions can be raised concerning the impact of other social network properties on recommender systems, such as their "small world" property [180] or the their "shrinking diameters" property [81]. In short, researchers have yet to fully explore issues related to benefits that social relationship can bring to recommendation. As reported in most related literature, in general, social relationships make it possible to improve recommender system performance over what can be achieved using the U-I matrix alone. The absolute gain, however, is not tremendous. It has also been observed that naive prediction models based on the U-I matrix can attain recommendation performance comparable to that achieved using social relationships [102, 146]. One possible method that can be used to investigate the issue has been introduced in preliminary work by the authors of [143]. This work proposes to exploit social network modeling techniques, such as those of [79], to simulate the social relationship between users in the recommender system. The simulation makes it possible to investigate the upper bounds of the benefit that the exploitation of social networks can be expected to provide for recommender system algorithms.
- Second, *how can mutual benefits between recommender systems and social networks be promoted?* In other words, how can we improve both the content item recommendation and users' social engagement via social recommendation? Recommender system researchers have devoted significant effort to exploiting user social relationship for improving recommendation performance, while research on how to exploit recommendations for improving social relationships has remained relatively limited. For this reason, an open issue is to investigate the potential of using the U-I matrix simultaneously for social relationship prediction and recommendation. A pioneering contribution in this area was made by the authors of [188], who demonstrated the mutual benefit between recommenders and social networks in terms of product recommendation and social friendship prediction. However, a fundamental question that researchers need to ad-

dress in this direction is the exact nature of the correlation between the similarity of users' interest and the social relationship. This question is important since the core assumption that must hold in order to guarantee mutual benefit is that socially related users share similar interests. Previous work has shown that there is a correlation between user online communication behavior and social relationships [156], which may also indicate the potential correlation between recommender system users and their social relationships. An initial study, performed in [133], investigated the impact of user social relationships on their tastes in movies and showed a positive correlation between the two. However, more convincing research in this direction is still needed, so as to provide evidence for understanding the predictability between each other of the users' interest in recommender systems and their social relationships.

- Third, *how to exploit negative social relationships for social recommendation?* Examples of negative relationships in social networks involve distrust and blacklisting. One example of the relatively limited work in this area is [97], where it was proposed to regularize the factorization of the U-I matrix by imposing a constraint that users with distrust relationships should have dissimilar latent factors. This work suggests that exploiting distrust relationships could have a positive effect for improving recommendation performance. However, another recent study in [174] has compared several trust-based and distrust-based recommendation approaches, and observed that distrust relationships make only a marginal contribution. In future, we consider that one of the challenging issues that researchers need to tackle in order to effectively use negative relationships for recommendation is the ways in which negative relationships propagate in social networks. We point out that the propagation of negative relationships can be expected to demonstrate a fundamentally different dynamic than the propagation of positive relationships. The simple assumption that a friend's friend is a friend, captures the natural propagation of positive relationships relatively well. However, the assumption that "an enemy's enemy is a friend" is less reliable and suggests that the propagation of negative relationship is much more complicated. The research on social network analysis has started investigating the propagation of the negative relationships and its predictability [49, 80]. It can be expected that findings from research in this related area could inform the exploitation of the negative relationship for social recommendation.

State of the Art In the literature, in addition to using social networks for item recommendation, a few contributions have been made to social connection recommendation. One of the first contributions to use recommender system techniques for people recommendation in social networks was conducted in the

domain of an enterprise social network in [52]. This work empirically investigated user profile representations by structured information sources, e.g., the user's co-authorship, and the effectiveness of content-based recommendation approaches. The results demonstrated the feasibility and the usefulness of exploiting rich side information sources for online connection recommendation. Inspired by this work, [53] have carried out comprehensive research on followee recommendation in the Twitter social network, by means of both content-based approaches and CF approaches and the combination of the two. Their work also demonstrated the usefulness of the content features from tweets (which, in contrast to the information sources used in [52], is unstructured data) for improving recommendation over that solely based on the social graph. We also notice that other similar work investigated different user profile representations with structured data in Flickr social network for a variety of recommendation purposes, including friend recommendation [153]. In the area of social tagging networks, [166, 201] exploited user tagging data for user connection recommendation, but did not examine the usefulness of the social graph. As mentioned previously, more recent work related to the topic of people recommendation has been carried out in [188]. This work proposed to jointly exploit both the user-service/item relations and the user-user social graph for both service/item and friend recommendation. In addition to people recommendation, community recommendation in social networks has also been attempted. *Combinational CF* [29] was one of the first attempts proposed for community recommendation. [173] investigated both MF and graph-based approaches for community recommendation, exploiting both the user-user friendship network and the user-community network.

Summarizing, research work in social recommendation has mainly focused on the exploitation of social networks for item recommendation, while an effective model/framework of social recommendation that can introduce mutual benefits between social networks and recommender systems, is still missing. Since social recommendation, due to its myriad of applications, is expected to remain a productive research topic in recommender systems, the effort towards addressing the three open issues covered here holds the potential for high payoff in terms of impact on the recommender system community.

7.1.2 Group Recommendation

Significance Although most recommender systems target providing quality recommendations for individual users, in some scenarios, recommendations are required that satisfy the needs of a group, e.g., as movie recommendation for a family, restaurant recommendation for dating partners, and event recommendation for online communities. In such applications, the success of a given

recommendation is not dependent on the opinion of one user, but rather on the user group as a whole. Because the recommendation needs of groups are complex and go beyond the sum of the needs of the individual group members, group recommendation has been identified as a research challenge in recommender systems [63].

Open Issues We cover four issues that distinguish group recommendation from recommendation for individual users. These issues constitute the key aspects that need to be addressed for this research challenge. We note that a recent overview work in [119] has also discussed the new perspectives on group recommendation. In this chapter, we restrict ourselves to highlighting only those group recommendation issues that are most critical in our viewpoint, with the goal of complementing the information in [119].

- First, *how to model group-level preference?* Intuitively, a good recommendation for a group should be something that fits the group-level preference. However, modeling the group-level preference is difficult, since in most scenarios we only have the preferences of individual users, and the side information of individual users and items. A simplistic model would take group-level preference to be the intersection of all the members' individual preferences. However, such an aggregation approach potentially suffers two drawbacks. First, it might result in more severe data sparseness for CF, since the common interests among all the members could be very limited. Second, it might overlook the relationship between the members and the group, since members can possibly adjust their personal preference to accommodate those of other group members, who they know to enjoy or consume different sorts of items. To overcome the drawbacks of this simplistic model, significant research effort towards effective recommendation algorithms that can model the group-level preference in a reasonable and interpretable manner is needed.
- Second, *what is the impact of group structure, and how to exploit it for group recommendation?* Members in a social system/organization usually have different roles, such as leaders and followers. In this sense, members in a group should have different types of influence on items recommended to the group as a whole. Here, again, we note that a good group recommendation is not necessarily the “common” interest of all the group members. For example, if plenty of research themes are available/relevant to be recommended to a research group, the group leader, who holds a ten-year strategic view on this group, but also understands the expertise within all the group members, should have much stronger opinion to the relative importance of different themes, than the group members with less

experience, who may only consider the relevance of those themes based on their own expertise and a significantly narrower understanding of the field. In this case, a good recommendation should be more biased to the group leader's preference. Another example is if a parent takes a young child to a movie, then the recommendation should be more heavily based on what will interest the child rather than the parent. Because of such asymmetries, group structure needs to be investigated and exploited for steering group recommendation. Although the explicit group structures may not be available for individual groups, there could exist possibilities for mining group structures from the side information about group members, such as the interaction information and the social relationship. Then, one could further study how the inferred group structures benefit group recommendation. To the best of our knowledge, the issue of group structures has not been raised or studied in the community, although it is relatively clear that research addressing this issue stands to make a significant contribution to group recommendation.

- Third, *how to take into account the dynamics of a group for group recommendation?* This issue has been raised in the work of [119]. We also highlight this issue, since we consider dynamics to be a key characteristic that makes group recommendation different from other recommendation tasks. For example, it is natural that online groups can be growing (i.e., more and more new members joining in) or dying (i.e., more and more members leaving) [64]. Little is known about the impact of such trends on group recommendation. Further, research investigation has yet to explore if changes in the group structure can be used actually to inform recommendations. For example, the information that a particular member left a group, could potentially shed light on the ways how recommendation could be improved for the remaining group members. The challenge is inferring the implications of the changes, e.g., if the member dropped out due to interests that diverged with those of the group. Conversely, if a member is observed to join the group, this information could be useful for improving recommendations as long as the reason for joining can be inferred.
- Finally, *how to evaluate group recommendation?* Although evaluation by itself has been recognized as a big challenge in the entire scope of recommender systems [39, 50, 71, 124], we particularly focus on the open issues regarding the evaluation for group recommendation in this chapter. Note that the general issues regarding the evaluation of recommender systems are not covered in the scope of this thesis. In the specific case of group recommendation, the main difficulty lies in the ambiguity of the ground truth. In practice, the relevance of an item recommended to a group is usually derived from its relevance to individual group members. For ex-

ample, using the *worst-case* strategy, one may consider an item relevant to a group only if it is relevant to all the members. However, using the *average satisfaction* strategy, one may consider whether an item is relevant to a group based on its average relevance scores across all the members. As a result, if different strategies are used for interpreting the data as ground truth, contradictory evaluation results could arise, e.g., some approaches may perform well under one particular strategy, but not others. Inconsistencies in evaluation may result in the loss of opportunities for valuable insight and conclusions made on the basis of experimental evaluation. Ideally, the ground truth data for group recommendation should be a relevance score for each item on behalf of all the group members, such as the case that a group representative rated a few movies after the group watched them together in a movie night event. Obviously, collecting such data for group recommendation is already a very difficult task. However, it is one that is worth tackling since successful research towards this challenge would serve to drive group recommendation research forward significantly.

State of the Art A few research contributions have been made to address the challenge of group recommendation. Most of them have focused on the first open issue mentioned above, namely preference modeling. Two strategies have been attempted for modeling group-level preferences, i.e., one is to first generate a group profile by aggregating the user profiles in the group and then make recommendations for the group profile, and the other is to first generate recommendations for all the users in the group and then aggregate the results as the final output for the group [5, 20]. The effectiveness of the two strategies for group recommendation has been investigated in recent studies, by using either simulated data of user groups [10], or real data about families of users [13]. In addition, some recent work has exploited the social relationship of group members for group recommendation [45]. Other recent work has exploited item content features or metadata for modifying group recommendation that is solely based on the joint preferences of group members [137]. However, we can see that those approaches fall into simplistic methods for modeling group-level preferences, discussed above. The two drawbacks of these models, namely, the even-worse data sparseness and the conditional relationship between the members and their group, have not been addressed, nor have they even been widely recognized. A recent competition focusing on group recommendation task [132] has, again, highlighted the difficulty of addressing this challenge¹. Finally, we point that the other three open issues also present challenges in need of attention from the research community.

¹<http://2011.camrachallenge.com/news/>

7.1.3 Long Tail Recommendation

Significance According to the terminology introduced in [7], the *Long Tail* within the context of recommender systems refers to the items that have low popularity or have just been added to the recommender system domain. Long tail recommendation can be understood as distinguishing highly-personalized recommendation from less-personalized and non-personalized recommendation (e.g., popularity-based recommendation). For this reason, the ability of a recommender system to recommend the long tail items is a critical indicator of the usefulness of recommender systems. For example, a user may like a popular movie (she may already know it) that suits her interest, but may be more in favor of a movie recommendation that is less obvious and surprises her. The ability to recommend items that users would not have otherwise found or thought of raises their appreciation of the system. Similarly in the travel domain, a traveler would appreciate a recommendation for a place that fits her particular interest, rather than a popular location that is described in every tour guide. In short, long tail recommendation plays an important role in most recommendation applications, since it helps, to a large extent, to improve users' satisfaction, and, by stimulating curiosity, also their engagement.

Open Issues We summarize our perspective on three issues that make long tail recommendation extremely challenging.

- First, *how to promote the recommendation of tail items?* As mentioned above, the difficulty of recommending long tail items lies in the fact that such items have very limited user preferences in their history. Such a problem is also known as the *item cold start* problem. One possibility is to address this problem by taking into account either content information derived from the items, or rich side information associated with them, when applying CF approaches. Another possibility is to first explicitly identify the tail items in a given collection, and then generate recommendations intentionally biased to those tail items. However, both of these options involve heuristics. Consequently, on the whole, the field continues to suffer from the lack of a solid theoretical ground for addressing the long tail recommendation challenge. In other words, we are still missing a unified and well-established recommendation model or framework that makes it possible to explicitly target tail items.
- Second, *what is the added value of tail items, and how to exploit it in recommendation algorithms?* We also notice that there has been little investigation on how recommendation of tail items can influence user satisfaction, or how additional revenue can be generated by the recommender

systems from the tail items. In general, principled answers are lacking to the question of why, and in which cases, recommending a tail item is more important than a head item. Extensive experimental research may be necessary in order to understand and explain the potential added value to be derived from long tail recommendation. Pioneering work on this issue was carried out by the authors of [108], who studied the revenue influenced by recommendation of tail items and head items in Amazon.com. They found that the recommender system helps to improve the revenue from the tail items, but at the same time, reduces the revenue from the head items. Although this chapter does not directly address the challenge of improving tail recommendations, it serves as an example of work that will inform future research that addresses the underlying question, “Why recommend tail items?”.

- Finally, *how to match long tail recommendations and users’ topical needs?* Recommendation in the long tail means not only recommending items receiving less overall user attention, it also means satisfying user recommendation needs that are relatively speaking more exotic. Adapting recommender systems not only to niche items, but also to niche preferences is a formidable research challenge. Consider a travel recommender system that recommends that a user visit a relatively popular destination “London Eye”. The user can be satisfied with this recommendation for a relatively popular reason, namely because of a general desire to visit famous attractions. However, the user can also be satisfied with this recommendation because of a technical interest in large ‘observation wheels’, which is shared by relatively fewer people. Because this interest applies to a very small group of users, it will not be well represented in the traditional U-I matrix. Facing the challenge of recommending items for long-tail reasons requires methods capable of adapting recommendations to user topical interests. Approaches to this challenge could derive benefit from analyzing user topical interests from various information sources, in order to determine the specific nature of the long-tail adaptation that would best suit a user. At the current time, awareness of the importance of this open issue is not widespread, and significant efforts are necessary both in order to understand the nature of highly specialized user interests, and also how to adapt recommendation to address them.

State of the Art In literature, the long tail problem in recommender systems was first formulated by the authors of [118], who specifically focused on improving recommendations of items in the long tail, i.e., the items with only few ratings. The authors proposed to first split the item set into head items and tail items, and then only use the ratings in the clusters of tail items to generate the recommendations for the tail items. Their research demonstrated

the importance of tail items, since the effectiveness of their approach relies on achieving the proper split between head and tail items. More recently, [161] has proposed to specifically exploit item popularity (i.e., the number of ratings for an item) for refining the evaluation metric used to measure recommendation accuracy so that it places more emphasis on successful recommendations of tail items. Our work of non-trivial landmark recommendation [150], as presented in Chapter 5, addresses the tail recommendation by proposing a weighting scheme that biases recommendations towards non-trivial items. We also note that long tail recommendation is closely related to the issue of *novelty/serendipity* in recommender systems [48]. Researchers focusing on this issue have argued that it is important to recommend items that are not only relevant but also can provide users with a positive sense of surprise [61, 106, 109, 112]. In short, the contributions that have been made thus far in this area, have mainly focused on mechanisms that promote the recommendation of tail items, i.e., they address the first technical problem as discussed above. There has been a marked shortage of contributions that treat the theoretical aspects of long tail recommendation models and the second and the third issues discussed above remain open research challenges.

7.1.4 Cross-domain Collaborative Filtering

Significance Cross-domain collaborative filtering (CDCF) has recently started to draw research attention [82]. The core concept of CDCF is to exploit information from multiple U-I matrices (i.e., domains) in order to allow the recommendation performance of one domain to benefit from information from one or more other domains. In other words, we can regard CDCF as CF on one U-I matrix/domain, while taking other U-I matrices as contextual information sources. The CDCF techniques hold particular importance for recommender systems for two reasons. First, they can be exploited by megadata owners (e.g., internet companies with a variety of online services) for further optimizing recommendations for their users under different sites. Second, they can introduce mutual benefit for different data owners (e.g., two companies running businesses that offer different online products) for further improving their service quality. Recently, a new online application, *Tipflare*², has been developed at MIT as a pioneering application of cross-domain recommendation. In all, CDCF has become one of the major challenges for the research of recommender systems.

Open Issues As a new research topic in recommender systems, CDCF is in search of answers to two fundamental questions: first, what could be the

²<https://www.tipflare.com/>

common knowledge/data that can be transferred/shared between different domains, or simply, “What to share?”, and, second, what could be the optimal way to transfer/share knowledge between different domains, or simply, “How to share?”. In the following, we elaborate on our understanding of these two issues.

- First, *what to share?* This problem focuses on the usefulness and the reliability of information patterns that could be exploited for CDCF. Users (or items) in different domains could be mutually exclusive, thus, making it difficult to establish links between users (or items) from different domains. An interesting direction is to explore the knowledge about characteristics that are shared between domains and is represented in user-contributed information, such as tags [147]. In addition, since social networks can interconnect users across different domains, it might be also promising to derive knowledge that is common between two domains by analyzing information, such as votes/likes on different domain products, contributed by socially connected users. It is important to pay careful attention to the reliability of information that is common between two domains. In other words, in cases in which it is possible to automatically identify information about characteristics shared between two domains, it is still questionable whether, or which of, those characteristics are reliable enough to improve CDCF. For this reason, it is important that researchers also gain an understanding of cases in which CDCF could degrade the recommendation quality.
- Second, *how to share?* Addressing this issue requires the development of new algorithms for optimally exploiting mutual benefit from multiple domains. On one hand, the link (or the correlation) between user preferences in different domains may be hidden. Methods that focus on discovering cross-domain correlations hold promise to improve the performance of CDCF. On the other hand, there might be multiple links between different domains that could be used for knowledge transfer. In this case, algorithms are needed that are not only capable of exploiting multiple links simultaneously, but that are also able to automatically discover the relative importance of different links. In addition, as mentioned before, there are many contextual information sources available in each of the individual domains. Individually the domains may already be large scale, and taken together they may pose an even more serious scale challenge for CDCF. Massive amounts of information from multiple domains needs to be processed with a reasonable computational cost.

State of the Art Some of the earliest work on CDCF was carried out by Berkovsky et al. [14], who deployed several mediation approaches for importing

and aggregating user rating vectors from different domains. Recently, research on CDCF has been influenced by, and benefitted from, progress in the area of transfer learning [115], a machine learning paradigm for sharing knowledge among different domains. For example, approach called Coordinate System Transfer [117] first learns latent features of users and items from an auxiliary domain (which has relatively more user preference data), and then adapts them to a target domain (which has relatively less user preference data). Further, an extension of this approach has been proposed that exploits implicit user feedback, rather than explicit user ratings, to constitute the auxiliary domain [116]. However, these approaches require that either users or items are shared between the domains, which is a condition not commonly encountered in practical applications. Codebook transfer (CBT) [83] and rating-matrix generative model (RMGM) [84] are two approaches that transfer knowledge from an auxiliary domain by learning an implicit cluster-level rating pattern that can be shared with a target domain. Similarly, multi-domain CF is an approach that extends PMF in multiple domains involving explicit user preference [196] or implicit user feedback [168] by learning an implicit correlation matrix, which links different domains for knowledge transfer. One of the latest contributions has adopted the CDCF framework of RMGM [84] to address the problem of dynamic CF [85]. However, those approaches rely on implicit domain correlations that are mined solely from user preference data, and no explicit links are exploited. [147] have proposed tag-induced cross-domain collaborative filtering (TagCDCF) to use common tags as bridges to link different domains for improving CDCF. On the whole, very limited work has been devoted to exploiting explicit links between different domains for CDCF. For this reason, the first technical problem, i.e., what to share, still a significant open issue. In addition, as discussed with respect to the second CDCF issue mentioned above, the exploitation of various contextual information sources and the consideration of multiple cross-domain links have not been fully explored by the research community. Thus, many opportunities remain open for addressing the challenge of CDCF.

7.2 Challenges of New Perspectives and Models

Outside of the core research area of recommender systems, there are a number of other research areas that are rapidly developing and which have the potential to inform and stimulate new developments in recommender systems. In this section, we cover three of these areas that we consider to be particularly promising sources of the new perspectives and new techniques necessary to stimulate innovation and progress in recommender system research.

7.2.1 Search and Recommendation

Significance Search and recommendation are both technologies that have come in to their own with the rise of the Internet. From the application perspective, the difference between the two lies in whether or not users are required to express their information need by explicitly, via queries (as in search) or whether the information needs are implicit, e.g., encoded in rating and consumption behavior (as in recommendation). Because the function and benefit of the two technologies are complementary, it can be expected that many online applications will have the need for both, with various levels of integration. A recent example of the convergence of search and recommendation is the +1-button offered by *Google+*³ that allows users to vote on search results. The quality of search results stands to benefit significantly from integrating explicit feedback from human users with similar search needs. In addition, another recent Google application *Google Now*⁴ aims to achieve personalized search by integrating context-aware recommendation. Because of the wide reach and enormous importance of search engines, the integration of search and recommendation technologies has become an attractive research topic, and presents a substantial challenge for researchers from both communities.

Open Issues We would like to discuss two challenges that we expect to be of central importance for future research on the integration of search and recommendation.

- First, *how can recommendation techniques help improve the quality of search results in the long tail?* In the case of web search, which involves billions of webpages, millions of which can be relevant to a single query, there could be a tremendous number of webpages that are only visited by users an extremely limited number of times. If webpages are infrequently viewed, they will be infrequently voted upon by users, even when voting is effortless, as with *Google+* as described before. There is a real danger that a given relevant webpage does not accumulate any votes at all. We note that this challenge can be regarded as a special case of the general “Long Tail Recommendation” challenge (see Section 7.1.3) in the search scenario. However, we emphasize that one major/particular issue here is that the result webpages (which correspond to items) are conditioned on particular queries. It is important to keep in mind that in search scenarios, the long tail involves the interaction between the frequency of user votes and the frequencies of queries. Note that there are many queries which are issued only infrequently, and for this reason, low voting volume might fail to

³<http://www.google.com/+1/button/>

⁴<http://www.google.com/landing/now/>

reinforce not only the importance of the webpage, but also the relevance relationship between the webpage and the query. Another danger is that user votes will create a snowball feedback effect. In other words, a few user votes will lead to certain webpages being ranked higher, where they will be more easily seen and accumulate more votes. Webpages that are relevant, but happen not to establish their popularity early, risk falling to the bottom of the ranking and never being discovered. For this reason, we would suggest two directions for addressing this problem. First, one could develop methods to predict votes for webpages, and then use the inferred votes for improving search results. These methods would create a minimum vote volume for tail webpages and could also prevent webpages to be lost in the snowball voting patterns. A major open issue for vote prediction is how vote prediction algorithms differ from the algorithms that carry out the main calculation of the relevance match between query and webpages. Second, instead of the voting system for the search results, one could consider developing a voting system for queries so as to avoid the constraint from tail webpages. In this way, the search results are supposed to be improved not by the collaboratively recommended results, but by means of using collaboratively recommended queries.

- Second, *how to allow search results to benefit from user votes, but also maintain attack resistance?* Conventionally, *attacks* in recommender systems refer to cases in which malicious users (attackers) assign high ratings deliberately to particular items in order to promote (or denigrate) those items [78, 104]. In the case of a voting system for search results, malicious users could also shill the system by giving deliberate votes to particular results (e.g., webpages). We emphasize that this issue could be more severe than that in recommender systems, since the queries are used to express the users' information needs. For example, the query "New York" is often used by users who are planning a travel to New York. This information need can be easily used by malicious users who may deliberately promote some results, e.g., a particular hotel, by assigning a lot of votes to them. This issue also opens plenty of opportunities for future research towards attack resistant mechanisms for collaboratively recommending search results.

State of the Art The relationship between search and recommendation was formally raised in a panel discussion in 2010 [51]. A few recent research contributions have demonstrated the possibilities of exploiting information from search engines for item recommendation [87, 159], especially in social settings. However, to our knowledge, there have been no specific attempts that address the open issues for the integration of search and recommendation that we have identified here. We anticipate that a sizable number of studies on this challenge

will be carried out in near future, leading to significant new developments for online applications.

7.2.2 Interaction and Recommendation

Significance Today, the interaction between users and recommender systems is no longer focused on ratings, and most recommender systems have become more interactive than before. Note that the term “interaction” in this subsection refers to a process in which the system elicits particular information/reactions from the user and integrates this information to refine the recommendation results. Conversation has been recognized as one of the most important types of interaction for recommender systems [172]. Typically, a conversation is used to guide the users to express their information needs more explicitly, providing a basis for fine-grained adaption of recommendation to user needs [100]. For instance, a movie recommender may first ask the user some situational questions, e.g., “Are you alone or with friends?”, before generating recommendations. The answers to this kind of questions could help the system to increase the relevance of recommendations. Another example is that the system can ask the user for feedback on recommendations. One possibility would be that if a user did not choose any of the top (e.g., top-10) recommended items, the system may ask the user why she was disappointed. The answer to this question could also be used in improving recommendation algorithms, by allowing them to adjust the recommendations for this user [28]. In addition, explanation of recommendations has also been considered as a critical function for recommender systems. Explanations provide the users with rationale that motivates why the items/products have been recommended [41, 56, 67, 164]. Two main effects can be attained by explaining recommendation results. First, it helps users to better understand the mechanism of the system. Understanding, in turn, potentially improves user satisfaction, since users could learn to adjust their behavior and expectations to the system [190]. In addition, it also builds trust because the users can tell if they agree with the factors influencing recommendations produced by the system. Second, it may allow recommender systems to serve users better with serendipitous results, since the users may discover new interest inspired by the explanations [192]. For example, in the case that a user wants to enjoy some movies and consults a movie recommender, the user may like some recommended movies which are certainly popular at that moment, and which fit his interest well. However, if the user has never heard of the recommended movie, but the recommender system provides a convincing explanation of why the user might like the movie, the user might prefer to watch this movie instead of other “predictable” recommended movies. In view of such usefulness of various interactions for recommender systems, the integration of interaction and recommendation is expected to be a trend for most of online

services. At the same time, it will remain a challenging research topic that requires effort from different research areas.

Open Issues Although various types of interaction exist between recommenders and users, the key issue we would like to highlight is *how the information from the interactions, such as conversations and explanations, can be exploited effectively for improving recommendation quality?* In other words, we need to address the question, “Which algorithms/paradigms would be the most effective or promising for interactive recommender systems?”. One possibility to address this problem is to consult results from the field of decision-making theory [130], which has been identified as a viable basis for developing new recommendation algorithms [62]. Another aspect of this problem is whether researchers should seek a generalized mechanism that can handle all kinds of interactions for recommendation, or whether different mechanisms that are specialized for different types of interaction are needed. This challenge also provides valuable opportunities for future research on interactive recommendation with different types of interaction data. Looking back on past research progress in CF, we see that major innovations have been made in the face of the rise of new types of data, such as the contextual information reviewed in this paper. We anticipate that, along with the growing availability of various types of interaction data, a wave of new contributions to CF beyond the user-item matrix will be proposed for integrating interaction and recommendation.

State of the Art In the most basic case, interactive recommendation is studied as a case of the problem of online CF [91, 162]. Under such a view, the key issue is how to constantly take into account new user preference data for improving recommendation results. One recent contribution has exploited the information of user choice in recommendation sessions for model training [189]. Here, a key constraint imposed is that the chosen items in a session should have higher relevance for the user than the unchosen items. In general, however, current research remains in a stage that focuses on user preference data. Research has yet to turn its attention to use cases in which information about a variety of interactions between users and recommenders is available.

7.2.3 Economics and Recommendation

Significance The study of economics provides a valuable source of models and insights that can be used to improve recommender systems. In economic systems, for example, in commodities markets, actors pursue specific objectives under the limitations of specific constraints. Recommender systems are also characterized by interactions between actors with objectives and constraints.

As recommender system scenarios grow more complicated, multiple objectives of multiple actors and a growing number of constraints must be taken into consideration. For example, recommender systems play a key role in e-commerce since they mediate the interaction between buyers and sellers. Recommendations must be optimized in order to satisfy both sellers business metrics, such as sales and customer retention, and also to generate recommendations that buyers find interesting and useful. Economic models are ideally suited for capturing these complex interactions. The ability of economic models to reflect and explain the dynamics underlying recommendation scenarios makes them uniquely suited for understanding and improving recommender systems.

Open Issues The main challenge that must be faced in order to allow recommender systems to productively exploit economic models is *selecting and integrating economic models that optimize recommender system output*. Obviously, different economic theories may relate to different aspects of recommender systems. The discussion in this thesis has focused on recommender systems that exploit context. We would like to explicitly point out that the availability of context information, such item categories or item prices, increases the complexity of the recommendation problem and thereby also of the ways in which economic models can be exploited by recommender systems.

State of the Art Here, we mention a few examples of the work in the area of recommender systems that has drawn on economic theories. These examples are chosen to illustrate the diversity of economic models that are relevant for recommendation scenarios.

Early work connecting economic models and recommender systems highlighted the correspondence between CF and Social Choice theory [121]. Social Choice theory is a framework for analyzing how the preferences of individuals can be combined to obtain decisions at the level of the social collective. The authors of [121] suggest that voting mechanisms provide a valuable source of possibilities for refined CF. Economic models have been used by the authors of [54] in order to explain user rating behavior in recommender systems. These authors adopt a straightforward economic model which models raters as a set of agents that work to maximize their objectives given constraints. The model integrates factors that influence users' willingness to rate movies, including their desire for high-quality movie recommendations and the limited time and effort that they are willing to spend rating movies. The model is able to explain a significant portion of users rating behavior. However, the authors caution that a thorough understanding of the user population under investigation is required in order to create economic models that explain user behavior. A market-place model has been used in [181]. This work is based on the insight that the strengths of

multiple recommender systems can be combined, if these systems are allowed to compete in a marketplace for positions within the recommendation list that is presented to the user.

A consumer behavior model has been used in [177]. This work makes use of the economic concept of ‘utility’, which is defined as the satisfaction or pleasure that an individual derives as a result of purchasing a product. In [177] an existing recommender system is extended to take into account the dependence of a product’s utility on user past purchasing behavior. For example, a user who has just purchased a consumable such as diapers, will soon derive high additional utility from another similar purchase. For durable items, for example consumer electronics such as cameras, more time must elapse before similar purchases provide high additional utility. Portfolio theory has been exploited in [151] to improve lists of recommended items. This work observes that the usefulness of an item recommendation is dependent not only on that particular recommendation, but rather on the entire list of recommended items. The approach takes this list to be an investment portfolio, and applies optimization techniques used in the financial world. The optimization handles uncertainty and also maximizes the diversity of the list in a way that respects the user’s preference for topical breadth.

7.3 Conclusions

In this chapter we discussed the challenges that we anticipate to be most influential regarding the future research on CF, and also the opportunities for addressing each of these challenges. The treatment that we have given these challenges in this thesis is based on our understanding of application demands, fundamental problems and outreach connections in the area of recommender systems. We expect that these challenges will attract significant research effort and lead to productive research outcomes in the following 5-10 years.

Without doubt, new challenges in the scope of CF with contextual information, above and beyond those discussed in this thesis, will continue to arise in the coming years. The emergence of new challenges is influenced by a variety of factors. These factors include: the availability of new types of contextual information in recommender systems, the development of new applications involving recommendation technology, the reform of evaluation methodologies for recommendation performance, the exploration of new crossovers between recommender systems and other areas, and the recognition of new fundamentals and theories in recommender systems. As a result of these new developments, we believe that recommender systems will continue to be a productive and interesting research field, and that the opportunities for research work to achieve high impact in this area will remain ample and attractive.

Bibliography

- [1] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Trans. Inf. Syst.*, 23:103–145, January 2005.
- [2] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.
- [3] D. Agarwal and B.-C. Chen. Regression-based latent factor models. KDD '09, pages 19–28. ACM, 2009.
- [4] D. Agarwal, B.-C. Chen, and B. Long. Localized factor models for multi-context recommendation. KDD '11, pages 609–617. ACM, 2011.
- [5] S. Amer-Yahia, S. B. Roy, A. Chawlat, G. Das, and C. Yu. Group recommendation: semantics and efficiency. *Proc. VLDB Endow.*, 2(1):754–765, Aug. 2009.
- [6] S. S. Anand and B. Mobasher. From web to social web: Discovering and deploying user and content profiles. chapter Contextual Recommendation, pages 142–160. Springer-Verlag, Berlin, Heidelberg, 2007.
- [7] C. Anderson. *The Long Tail: Why The Future Of Business Is Selling Less Of More*. Hyperion press, 2006.
- [8] Y. Arase, X. Xie, T. Hara, and S. Nishio. Mining people’s trips from large scale geo-tagged photos. MM '10, pages 133–142. ACM, 2010.
- [9] S. Balakrishnan and S. Chopra. Collaborative ranking. WSDM '12, pages 143–152. ACM, 2012.
- [10] L. Baltrunas, T. Makcinskas, and F. Ricci. Group recommendations with rank aggregation and collaborative filtering. RecSys '10, pages 119–126. ACM, 2010.
- [11] L. Baltrunas and F. Ricci. Context-based splitting of item ratings in collaborative filtering. RecSys '09, pages 245–248. ACM, 2009.
- [12] J. Basilico and T. Hofmann. Unifying collaborative and content-based filtering. In *Proceedings of the twenty-first international conference on Machine learning*, ICML '04, pages 9–, New York, NY, USA, 2004. ACM.

-
- [13] S. Berkovsky and J. Freyne. Group-based recipe recommendations: analysis of data aggregation strategies. *RecSys '10*, pages 111–118. ACM, 2010.
- [14] S. Berkovsky, T. Kuflik, and F. Ricci. Cross-domain mediation in collaborative filtering. *UM '07*, pages 355–359. Springer-Verlag, 2007.
- [15] M. Böhmer, B. Hecht, J. Schöning, A. Krüger, and G. Bauer. Falling asleep with angry birds, facebook and kindle - a large scale study on mobile application usage. In *Proc. of Mobile HCI '11*, 2011.
- [16] J. S. Breese, D. Heckerman, and C. M. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. *UAI '98*, pages 43–52, 1998.
- [17] C. J. C. Burges, R. Ragno, and Q. V. Le. Learning to rank with nonsmooth cost functions. *NIPS '06*, pages 193–200, 2006.
- [18] R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12:331–370, November 2002.
- [19] M. L. Caballero. Near2me: Design and evaluation of a personalized recommender and explorer for off-the-beaten-track travel destinations. *Technical Report, Eindhoven University of Technology*, 2010.
- [20] L. M. Campos, J. M. Fernández-Luna, J. F. Huete, and M. A. Rueda-Morales. Managing uncertainty in group recommending processes. *User Modeling and User-Adapted Interaction*, 19(3):207–242, Aug. 2009.
- [21] I. Cantador and P. Castells. Semantic contextualisation in a news recommender system. *Proceedings of the 2009 Workshop on Context-Aware Recommender Systems*, 2009.
- [22] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, , and H. Li. Learning to rank: From pairwise approach to listwise approach. *Technical Report MSR-TR-2007-40*, Microsoft Research, 2007.
- [23] B. D. Carolis, I. Mazzotta, N. Novielli, and V. Silvestri. Using common sense in providing personalized recommendations in the tourism domain. In *Proceedings of the 2009 Workshop on Context-Aware Recommender Systems*, 2009.
- [24] S. Chakrabarti, R. Khanna, U. Sawant, and C. Bhattacharyya. Structured learning for non-smooth ranking losses. *KDD '08*, pages 88–96. ACM, 2008.
- [25] O. Chapelle, D. Metlzer, Y. Zhang, and P. Grinspan. Expected reciprocal rank for graded relevance. *CIKM '09*, pages 621–630. ACM, 2009.
- [26] O. Chapelle and M. Wu. Gradient descent optimization of smoothed information retrieval metrics. *Inf. Retr.*, 13:216–235, June 2010.
- [27] H. Chen and D. R. Karger. Less is more: probabilistic models for retrieving fewer relevant documents. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 429–436, New York, NY, USA, 2006. ACM.

-
- [28] L. Chen and P. Pu. Critiquing-based recommenders: survey and emerging trends. *User Modeling and User-Adapted Interaction*, 22(1-2):125–150, Apr. 2012.
- [29] W.-Y. Chen, D. Zhang, and E. Y. Chang. Combinational collaborative filtering for personalized community recommendation. KDD '08, pages 115–123. ACM, 2008.
- [30] A. Clauset, C. R. Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *SIAM Rev.*, 51(4):661–703, Nov. 2009.
- [31] M. Clements, P. Serdyukov, A. P. de Vries, and M. J. Reinders. Using flickr geotags to predict user travel behaviour. SIGIR '10, pages 851–852. ACM, 2010.
- [32] P. Cremonesi, F. Garzotto, S. Negro, A. Papadopoulos, and R. Turrin. Comparative evaluation of recommender system quality. In *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems*, CHI EA '11, pages 1927–1932. ACM, 2011.
- [33] P. Cremonesi, Y. Koren, and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. RecSys '10, pages 39–46. ACM, 2010.
- [34] J. Davidson, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston, and D. Sampath. The youtube video recommendation system. RecSys '10, pages 293–296. ACM, 2010.
- [35] J. Davis and M. Goadrich. The relationship between precision-recall and roc curves. ICML '06, pages 233–240, 2006.
- [36] M. De Choudhury, M. Feldman, S. Amer-Yahia, N. Golbandi, R. Lempel, and C. Yu. Automatic construction of travel itineraries using social breadcrumbs. HT '10, pages 35–44. ACM, 2010.
- [37] H. Deng, M. R. Lyu, and I. King. Effective latent space graph-based re-ranking model with global consistency. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, WSDM '09, pages 212–221, New York, NY, USA, 2009. ACM.
- [38] M. Deshpande and G. Karypis. Item-based top-n recommendation algorithms. *ACM Trans. Inf. Syst.*, 22:143–177, January 2004.
- [39] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan. Collaborative filtering recommender systems. *Foundations and Trends in Human-Computer Interaction*, 4(2):81–173, 2011.
- [40] A. Elberse. Should you invest in the long tail? *Harvard Business Review*, July-August 2008.
- [41] G. Friedrich and M. Zanker. A taxonomy for generating explanations in recommender systems. *AI Magazine*, 32(3):90–98, 2011.
- [42] Z. Gantner, S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. Mymedialite: a free recommender system library. RecSys '11, pages 305–308. ACM, 2011.

-
- [43] Z. Gantner, S. Rendle, and S.-T. Lars. Factorization models for context-/time-aware movie recommendations. *CAMRa '10*, pages 14–19. ACM, 2010.
- [44] Y. Gao, J. Tang, R. Hong, Q. Dai, T.-S. Chua, and R. Jain. W2go: a travel guidance system by automatic landmark ranking. *MM '10*, pages 123–132. ACM, 2010.
- [45] M. Gartrell, X. Xing, Q. Lv, A. Beach, R. Han, S. Mishra, and K. Seada. Enhancing group recommendation by incorporating social relationship interactions. *GROUP '10*, pages 97–106. ACM, 2010.
- [46] M. Ge, C. Delgado-Battenfeld, and D. Jannach. Beyond accuracy: evaluating recommender systems by coverage and serendipity. *RecSys '10*, pages 257–260. ACM, 2010.
- [47] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70, Dec. 1992.
- [48] L. Grossman. How computers know what we want - before we do. *TIME.*, 175(20), May 2010.
- [49] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. *WWW '04*, pages 403–412. ACM, 2004.
- [50] A. Gunawardana and G. Shani. A survey of accuracy evaluation metrics of recommendation tasks. *J. Mach. Learn. Res.*, 10:2935–2962, December 2009.
- [51] I. Guy, A. Jaimes, P. Agulló, P. Moore, P. Nandy, C. Nastar, and H. Schinzel. Will recommenders kill search?: recommender systems - an industry perspective. *RecSys '10*, pages 7–12. ACM, 2010.
- [52] I. Guy, I. Ronen, and E. Wilcox. Do you know?: recommending people to invite into your social network. In *Proceedings of the 14th international conference on Intelligent user interfaces*, IUI '09, pages 77–86, New York, NY, USA, 2009. ACM.
- [53] J. Hannon, M. Bennett, and B. Smyth. Recommending twitter users to follow using content and collaborative filtering approaches. *RecSys '10*, pages 199–206. ACM, 2010.
- [54] F. M. Harper, X. Li, Y. Chen, and J. A. Konstan. An economic model of user rating in an online recommender system. In *Proceedings of the 10th international conference on User Modeling*, UM'05, pages 307–316, Berlin, Heidelberg, 2005. Springer-Verlag.
- [55] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. *SIGIR '99*, pages 230–237. ACM, 1999.
- [56] J. L. Herlocker, J. A. Konstan, and J. Riedl. Explaining collaborative filtering recommendations. *CSCW '00*, pages 241–250. ACM, 2000.

-
- [57] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22:5–53, January 2004.
- [58] T. Hofmann. Latent semantic models for collaborative filtering. *ACM Trans. Inf. Syst.*, 22:89–115, January 2004.
- [59] T. Horozov, N. Narasimhan, and V. Vasudevan. Using location for personalized poi recommendations in mobile environments. In *Proceedings of the International Symposium on Applications on Internet*, pages 124–129, Washington, DC, USA, 2006. IEEE Computer Society.
- [60] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. ICDM '08, pages 263–272. IEEE Computer Society, 2008.
- [61] N. Hurley and M. Zhang. Novelty and diversity in top-n recommendation – analysis and evaluation. *ACM Trans. Internet Technol.*, 10(4):14:1–14:30, Mar. 2011.
- [62] A. Jameson. What should recommender systems people know about the psychology of choice and decision making? Keynote at Workshop on Decision Making and Recommendation Acceptance Issues in Recommender Systems, 2011.
- [63] A. Jameson and B. Smyth. The adaptive web. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *Recommendation to groups*, pages 596–627. Springer-Verlag, Berlin, Heidelberg, 2007.
- [64] S. R. Kairam, D. J. Wang, and J. Leskovec. The life and death of online groups: predicting group growth and longevity. WSDM '12, pages 673–682. ACM, 2012.
- [65] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. RecSys '10, pages 79–86. ACM, 2010.
- [66] J. Kleinberg and M. Sandler. Using mixture models for collaborative filtering. *J. Comput. Syst. Sci.*, 74:49–69, February 2008.
- [67] B. Knijnenburg, M. Willemsen, Z. Gantner, H. Soncu, and C. Newell. Explaining the user experience of recommender systems. *User Modeling and User-Adapted Interaction*, pages 1–64, 2012.
- [68] C. Kofler, L. Caballero, M. Menendez, V. Occhialini, and M. Larson. Near2me: an authentic and personalized social media-based recommender for travel destinations. In *Proceedings of the 3rd ACM SIGMM international workshop on Social media*, WSM '11, pages 47–52. ACM, 2011.
- [69] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Rev.*, 51(3):455–500, Aug. 2009.
- [70] T. G. Kolda and J. Sun. Scalable tensor decompositions for multi-aspect data mining. ICDM '08, pages 363–372. IEEE Computer Society, 2008.

-
- [71] J. Konstan and J. Riedl. Recommender systems: from algorithms to user experience. *User Modeling and User-Adapted Interaction*, 22:101–123, 2012.
- [72] I. Konstas, V. Stathopoulos, and J. M. Jose. On social networks and collaborative recommendation. SIGIR '09, pages 195–202. ACM, 2009.
- [73] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. KDD '08, pages 426–434. ACM, 2008.
- [74] Y. Koren. Collaborative filtering with temporal dynamics. KDD '09, pages 447–456. ACM, 2009.
- [75] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42:30–37, August 2009.
- [76] Y. Koren and J. Sill. Ordrec: an ordinal model for predicting personalized item rating distributions. RecSys '11, pages 117–124. ACM, 2011.
- [77] T. Kurashima, T. Iwata, G. Irie, and K. Fujimura. Travel route recommendation using geotags in photo sharing sites. CIKM '10, pages 579–588. ACM, 2010.
- [78] S. K. Lam and J. Riedl. Shilling recommender systems for fun and profit. WWW '04, pages 393–402. ACM, 2004.
- [79] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani. Kronecker graphs: An approach to modeling networks. *J. Mach. Learn. Res.*, 11:985–1042, Mar. 2010.
- [80] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Predicting positive and negative links in online social networks. WWW '10, pages 641–650. ACM, 2010.
- [81] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. KDD '05, pages 177–187. ACM, 2005.
- [82] B. Li. Cross-domain collaborative filtering: A brief survey. ICTAI '11, pages 1085–1086. IEEE Computer Society, 2011.
- [83] B. Li, Q. Yang, and X. Xue. Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction. IJCAI '09, pages 2052–2057, 2009.
- [84] B. Li, Q. Yang, and X. Xue. Transfer learning for collaborative filtering via a rating-matrix generative model. ICML '09, pages 617–624. ACM, 2009.
- [85] B. Li, X. Zhu, R. Li, C. Zhang, X. Xue, and X. Wu. Cross-domain collaborative filtering over time. AAAI '11, pages 2293–2298. AAAI, 2011.
- [86] Q. Li, Y. Zheng, X. Xie, Y. Chen, W. Liu, and W.-Y. Ma. Mining user similarity based on location history. GIS '08, pages 34:1–34:10. ACM, 2008.
- [87] Y. Li, J. Hu, C. Zhai, and Y. Chen. Improving one-class collaborative filtering by incorporating rich user information. CIKM '10, pages 959–968. ACM, 2010.

-
- [88] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7:76–80, 2003.
- [89] N. N. Liu, B. Cao, M. Zhao, and Q. Yang. Adapting neighborhood and matrix factorization models for context aware recommendation. CAMRa '10, pages 7–13. ACM, 2010.
- [90] N. N. Liu and Q. Yang. Eigenrank: a ranking-oriented approach to collaborative filtering. SIGIR '08, pages 83–90. ACM, 2008.
- [91] N. N. Liu, M. Zhao, E. Xiang, and Q. Yang. Online evolutionary collaborative filtering. RecSys '10, pages 95–102. ACM, 2010.
- [92] N. N. Liu, M. Zhao, and Q. Yang. Probabilistic latent preference analysis for collaborative filtering. CIKM '09, pages 759–766. ACM, 2009.
- [93] T.-Y. Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, 2009.
- [94] X. Lu, C. Wang, J.-M. Yang, Y. Pang, and L. Zhang. Photo2trip: generating travel routes from geo-tagged photos for trip planning. MM '10, pages 143–152. ACM, 2010.
- [95] H. Ma, I. King, and M. R. Lyu. Effective missing data prediction for collaborative filtering. SIGIR '07, pages 39–46. ACM, 2007.
- [96] H. Ma, I. King, and M. R. Lyu. Learning to recommend with explicit and implicit social relations. *ACM Trans. Intell. Syst. Technol.*, 2:29:1–29:19, May 2011.
- [97] H. Ma, M. R. Lyu, and I. King. Learning to recommend with trust and distrust relationships. RecSys '09, pages 189–196. ACM, 2009.
- [98] H. Ma, H. Yang, M. R. Lyu, and I. King. Sorec: social recommendation using probabilistic matrix factorization. CIKM '08, pages 931–940. ACM, 2008.
- [99] H. Ma, T. C. Zhou, M. R. Lyu, and I. King. Improving recommender systems by incorporating social contextual information. *ACM Trans. Inf. Syst.*, 29:9:1–9:23, April 2011.
- [100] T. Mahmood and F. Ricci. Improving recommender systems with adaptive conversational strategies. HT '09, pages 73–82. ACM, 2009.
- [101] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*. Cambridge Univ. Press, Cambridge [u.a.], 1. publ. edition, 2008.
- [102] P. Massa and P. Avesani. Trust-aware recommender systems. RecSys '07, pages 17–24. ACM, 2007.
- [103] S. M. McNee, J. Riedl, and J. A. Konstan. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI '06 extended abstracts on Human factors in computing systems*, CHI EA '06, pages 1097–1101. ACM, 2006.

-
- [104] B. Mobasher, R. Burke, R. Bhaumik, and C. Williams. Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness. *ACM Trans. Internet Technol.*, 7(4), Oct. 2007.
- [105] J. M. Morales-Del-Castillo, E. Peis, and E. Herrera-Viedma. A filtering and recommender system prototype for scholarly users of digital libraries. In *Proceedings of the 2nd World Summit on the Knowledge Society: Visioning and Engineering the Knowledge Society. A Web Science Perspective*, WSKS '09, pages 108–117. Springer-Verlag, 2009.
- [106] M. Nakatsuji, Y. Fujiwara, A. Tanaka, T. Uchiyama, K. Fujimura, and T. Ishida. Classical music for rock fans?: novel recommendations for expanding user interests. *CIKM '10*, pages 949–958. ACM, 2010.
- [107] M. E. J. Newman. Power laws, Pareto distributions and Zipf's law. *Contemporary Physics*, 46(5):323–351, May 2005.
- [108] G. Oestreicher-Singer and A. Sundararajan. Recommendation networks and the long tail of electronic commerce. *MIS Quarterly*, 36:65–83, 2012.
- [109] J. Oh, S. Park, H. Yu, M. Song, and S.-T. Park. Novel recommendation based on personal popularity tendency. *ICDM '11*, pages 507–516, 2011.
- [110] M. O'Mahony, N. Hurley, N. Kushmerick, and G. Silvestre. Collaborative recommendation: A robustness analysis. *ACM Trans. Internet Technol.*, 4:344–377, November 2004.
- [111] C. Ono, Y. Takishima, Y. Motomura, and H. Asoh. Context-aware preference model based on a study of difference between real and supposed situation data. *UMAP '09*, pages 102–113. Springer-Verlag, 2009.
- [112] K. Onuma, H. Tong, and C. Faloutsos. Tangent: a novel, 'surprise me', recommendation algorithm. *KDD '09*, pages 657–666. ACM, 2009.
- [113] R. Pan and M. Scholz. Mind the gaps: weighting the unknown in large-scale one-class collaborative filtering. *KDD '09*, pages 667–676. ACM, 2009.
- [114] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. M. Lukose, M. Scholz, and Q. Yang. One-class collaborative filtering. *ICDM '08*, pages 502–511, 2008.
- [115] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Trans. on Knowl. and Data Eng.*, 22:1345–1359, October 2010.
- [116] W. Pan, N. Liu, E. Xiang, and Q. Yang. Transfer learning to predict missing ratings via heterogeneous user feedbacks. *IJCAI'11*. Morgan Kaufmann Publishers Inc., 2011.
- [117] W. Pan, E. Xiang, N. Liu, and Q. Yang. Transfer learning in collaborative filtering for sparsity reduction. *AAAI '10*, 2010.
- [118] Y.-J. Park and A. Tuzhilin. The long tail of recommender systems and how to leverage it. *RecSys '08*, pages 11–18. ACM, 2008.

-
- [119] J. J. Pazos Arias, A. Fernández Vilas, R. P. Daz Redondo, I. Cantador, and P. Castells. *Group Recommender Systems: New Perspectives in the Social Web*, volume 32 of *Intelligent Systems Reference Library*, pages 139–157. Springer Berlin Heidelberg, 2012.
- [120] M. J. Pazzani and D. Billsus. Content-based recommendation systems. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *The adaptive web*, pages 325–341. Springer-Verlag, Berlin, Heidelberg, 2007.
- [121] D. M. Pennock, E. Horvitz, and C. L. Giles. Social choice theory and recommender systems: Analysis of the axiomatic foundations of collaborative filtering. In *Proc. 17th AAAI*, 2000.
- [122] L. Pizzato, T. Rej, T. Chung, I. Koprinska, and J. Kay. Recon: a reciprocal recommender for online dating. *RecSys '10*, pages 207–214. ACM, 2010.
- [123] R. L. Plackett. The analysis of permutations. *Applied Statistics*, 24(2):193–202, 1975.
- [124] P. Pu, L. Chen, and R. Hu. Evaluating recommender systems from the users perspective: survey of the state of the art. *User Modeling and User-Adapted Interaction*, pages 1–39, 2012.
- [125] S. Rendle, L. Balby Marinho, A. Nanopoulos, and L. Schmidt-Thieme. Learning optimal ranking with tensor factorization for tag recommendation. *KDD '09*, pages 727–736. ACM, 2009.
- [126] S. Rendle, C. Freudenthaler, Z. Gantner, and S.-T. Lars. Bpr: Bayesian personalized ranking from implicit feedback. *UAI '09*, pages 452–461. AUAI Press, 2009.
- [127] S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme. Fast context-aware recommendations with factorization machines. *SIGIR '11*, pages 635–644. ACM, 2011.
- [128] S. Rendle and L. Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. *WSDM '10*, pages 81–90. ACM, 2010.
- [129] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: an open architecture for collaborative filtering of netnews. *CSCW '94*, pages 175–186. ACM, 1994.
- [130] R. M. Roe, J. R. Busemeyer, and J. T. Townsend. Multialternative decision field theory: a dynamic connectionist model of decision making. *Psychological Review*, 108(2):370–392, 2001.
- [131] A. Said, S. Berkovsky, and E. W. De Luca. Putting things in context: Challenge on context-aware movie recommendation. *CAMRa '10*, pages 2–6. ACM, 2010.
- [132] A. Said, S. Berkovsky, and E. W. De Luca. Group recommendation in context. *CAMRa '11*, pages 2–4. ACM, 2011.

-
- [133] A. Said, E. W. De Luca, and S. Albayrak. How social relationships affect user similarities. In *Proceedings of the ACM IUI'10 Workshop on Social Recommender Systems*, 2010.
- [134] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. volume 20 of *NIPS '08*, 2008.
- [135] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18:613–620, November 1975.
- [136] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. *WWW '01*, pages 285–295. ACM, 2001.
- [137] S. Seko, T. Yagi, M. Motegi, and S. Muto. Group recommendation using feature space representing behavioral tendency and power balance among members. *RecSys '11*, pages 101–108. ACM, 2011.
- [138] P. Serdyukov, V. Murdock, and R. van Zwol. Placing flickr photos on a map. *SIGIR '09*, pages 484–491. ACM, 2009.
- [139] G. Shani and A. Gunawardana. Evaluating recommendation systems. In F. Ricci, L. Rokach, and B. Shapira, editors, *Recommender Systems Handbook*, pages 257–298. 2010.
- [140] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, A. Hanjalic, and N. Oliver. Tfmap: optimizing map for top-n context-aware recommendation. *SIGIR '12*, pages 155–164. ACM, 2012.
- [141] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, N. Oliver, and A. Hanjalic. Climf: learning to maximize reciprocal rank with collaborative less-is-more filtering. *RecSys '12*, pages 139–146. ACM, 2012.
- [142] Y. Shi, M. Larson, and A. Hanjalic. Exploiting user similarity based on rated-item pools for improved user-based collaborative filtering. *RecSys '09*, pages 125–132. ACM, 2009.
- [143] Y. Shi, M. Larson, and A. Hanjalic. Connecting with the collective: self-contained reranking for collaborative recommendation. *CMM '10*, pages 9–14. ACM, 2010.
- [144] Y. Shi, M. Larson, and A. Hanjalic. List-wise learning to rank with matrix factorization for collaborative filtering. *RecSys '10*, pages 269–272. ACM, 2010.
- [145] Y. Shi, M. Larson, and A. Hanjalic. Mining mood-specific movie similarity with matrix factorization for context-aware recommendation. *CAMRa '10*, pages 34–40. ACM, 2010.
- [146] Y. Shi, M. Larson, and A. Hanjalic. How far are we in trust-aware recommendation? *ECIR'11*, pages 704–707. Springer-Verlag, 2011.
- [147] Y. Shi, M. Larson, and A. Hanjalic. Tags as bridges between domains: improving recommendation with tag-induced cross-domain collaborative filtering. *UMAP'11*, pages 305–316. Springer-Verlag, 2011.

-
- [148] Y. Shi, M. Larson, and A. Hanjalic. Mining contextual movie similarity with matrix factorization for context-aware recommendation. *ACM Trans. Intell. Syst. Technol.*, 4(1), 2013.
- [149] Y. Shi, P. Serdyukov, A. Hanjalic, and M. Larson. Personalized landmark recommendation based on geotags from photo sharing sites. ICWSM '11, pages 622–625. AAAI, 2011.
- [150] Y. Shi, P. Serdyukov, A. Hanjalic, and M. Larson. Non-trivial landmark recommendation using geotagged photos. *ACM Trans. Intell. Syst. Technol.*, 2013.
- [151] Y. Shi, X. Zhao, J. Wang, M. Larson, and A. Hanjalic. Adaptive diversification of recommendation results via latent factor portfolio. SIGIR '12, pages 175–184. ACM, 2012.
- [152] L. Si and R. Jin. Flexible mixture model for collaborative filtering. ICML '03, pages 704–711, 2003.
- [153] S. Siersdorfer and S. Sizov. Social recommender systems for web 2.0 folksonomies. HT '09, pages 261–270. ACM, 2009.
- [154] V. Sindhwani, S. S. Bucak, J. Hu, and A. Mojsilovic. One-class matrix completion with low-density factorizations. ICDM '10, pages 1055–1060. IEEE Computer Society, 2010.
- [155] A. P. Singh and G. J. Gordon. Relational learning via collective matrix factorization. KDD '08, pages 650–658. ACM, 2008.
- [156] P. Singla and M. Richardson. Yes, there is a correlation: - from social networks to personal behavior on the web. WWW '08, pages 655–664. ACM, 2008.
- [157] R. W. Sinnott. Virtues of the haversine. *Sky and Telescope*, 68(2):158, 1984.
- [158] S. Sizov. Geofolk: latent spatial semantics in web 2.0 social media. WSDM '10, pages 281–290. ACM, 2010.
- [159] B. Smyth, J. Freyne, M. Coyle, and P. Briggs. Recommendation as collaboration in web search. *AI Magazine*, 32(3):35–45, 2011.
- [160] N. Srebro and T. Jaakkola. Weighted low-rank approximations. ICML '03, pages 720–727. AAAI Press, 2003.
- [161] H. Steck. Item popularity and recommendation accuracy. RecSys '11, pages 125–132. ACM, 2011.
- [162] D. H. Stern, R. Herbrich, and T. Graepel. Matchbox: large scale online bayesian recommendations. WWW '09, pages 111–120. ACM, 2009.
- [163] J.-H. Su, H.-H. Yeh, P. S. Yu, and V. S. Tseng. Music recommendation using content and context information mining. *IEEE Intelligent Systems*, 25:16–26, January 2010.

-
- [164] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos. Providing justifications in recommender systems. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 38(6):1262–1272, nov. 2008.
- [165] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos. Tag recommendations based on tensor dimensionality reduction. RecSys '08, pages 43–50. ACM, 2008.
- [166] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos. A unified framework for providing recommendations in social tagging systems based on ternary semantic analysis. *IEEE Trans. on Knowl. and Data Eng.*, 22:179–192, February 2010.
- [167] Y. Takeuchi and M. Sugimoto. Cityvoyager: An outdoor recommendation system based on user location history. In *Ubiquitous Intelligence and Computing*, volume 4159 of *Lecture Notes in Computer Science*, pages 625–636. Springer Berlin / Heidelberg, 2006.
- [168] J. Tang, J. Yan, L. Ji, M. Zhang, S. Guo, N. Liu, X. Wang, and Z. Chen. Collaborative users' brand preference mining across multiple domains from implicit feedbacks. AAAI '11. AAAI Press, 2011.
- [169] M. Taylor, J. Guiver, S. Robertson, and T. Minka. Sofrank: optimizing non-smooth rank metrics. WSDM '08, pages 77–86. ACM, 2008.
- [170] K. H. L. Tso-Sutter, L. B. Marinho, and L. Schmidt-Thieme. Tag-aware recommender systems by fusion of collaborative filtering algorithms. SAC '08, pages 1995–1999. ACM, 2008.
- [171] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *J. Mach. Learn. Res.*, 6:1453–1484, 2005.
- [172] D. Tunkelang. Recommendations as a conversation with the user. RecSys '11, pages 11–12. ACM, 2011.
- [173] V. Vasuki, N. Natarajan, Z. Lu, B. Savas, and I. Dhillon. Scalable affiliation recommendation using auxiliary networks. *ACM Trans. Intell. Syst. Technol.*, 3(1):3:1–3:20, Oct. 2011.
- [174] P. Victor, C. Cornelis, M. D. Cock, and A. M. Teredesai. Trust- and distrust-based recommendations for controversial reviews. *IEEE Intelligent Systems*, 26:48–55, 2011.
- [175] E. M. Voorhees. The trec-8 question answering track report. In *TREC-8*, 1999.
- [176] J. Wang, A. P. de Vries, and M. J. T. Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. SIGIR '06, pages 501–508. ACM, 2006.
- [177] J. Wang and Y. Zhang. Utilizing marginal net utility for recommendation in e-commerce. SIGIR '11, pages 1003–1012. ACM, 2011.
- [178] X. Wang, J.-T. Sun, Z. Chen, and C. Zhai. Latent semantic analysis for multiple-type interrelated data objects. SIGIR '06, pages 236–243. ACM, 2006.

- [179] Y. Wang, G. Cong, G. Song, and K. Xie. Community-based greedy algorithm for mining top-k influential nodes in mobile social networks. *KDD '10*, pages 1039–1048. ACM, 2010.
- [180] D. J. Watts and S. H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393(6684):440–442, June 1998.
- [181] Y. Z. Wei, L. Moreau, and N. R. Jennings. A market-based approach to recommender systems. *ACM Trans. Inf. Syst.*, 23(3):227–266, July 2005.
- [182] M. Weimer, A. Karatzoglou, Q. Le, and A. Smola. Cofrank - maximum margin matrix factorization for collaborative ranking. *NIPS '07*, pages 1593–1600, 2007.
- [183] M. Weimer, A. Karatzoglou, and A. Smola. Improving maximum margin matrix factorization. *Mach. Learn.*, 72:263–276, September 2008.
- [184] R. Wetzker, W. Umbrath, and A. Said. A hybrid approach to item recommendation in folksonomies. In *Proceedings of the WSDM '09 Workshop on Exploiting Semantic Annotations in Information Retrieval*, ESAIR '09, pages 25–29. ACM, 2009.
- [185] L. Xiong, X. Chen, T.-K. Huang, J. Schneider, and J. G. Carbonell. Temporal collaborative filtering with bayesian probabilistic tensor factorization. *SDM '10*, pages 211–222, 2010.
- [186] J. Xu and H. Li. Adarank: a boosting algorithm for information retrieval. *SIGIR '07*, pages 391–398. ACM, 2007.
- [187] J. Xu, T.-Y. Liu, M. Lu, H. Li, and W.-Y. Ma. Directly optimizing evaluation measures in learning to rank. *SIGIR '08*, pages 107–114. ACM, 2008.
- [188] S.-H. Yang, B. Long, A. Smola, N. Sadagopan, Z. Zheng, and H. Zha. Like like alike: joint friendship and interest propagation in social networks. *WWW '11*, pages 537–546. ACM, 2011.
- [189] S.-H. Yang, B. Long, A. J. Smola, H. Zha, and Z. Zheng. Collaborative competitive filtering: learning recommender using context of user choice. *SIGIR '11*, pages 295–304. ACM, 2011.
- [190] L. R. Ye and P. E. Johnson. The impact of explanation facilities on user acceptance of expert systems advice. *MIS Q.*, 19(2):157–172, June 1995.
- [191] H. Yildirim and M. S. Krishnamoorthy. A random walk method for alleviating the sparsity problem in collaborative filtering. *RecSys '08*, pages 131–138. ACM, 2008.
- [192] K.-H. Yoo and U. Gretzel. Creating more credible and persuasive recommender systems: The influence of source characteristics on recommender system evaluations. In *Recommender Systems Handbook*, pages 455–477. Springer US, 2011.
- [193] Y. Yue, T. Finley, F. Radlinski, and T. Joachims. A support vector method for optimizing average precision. *SIGIR '07*, pages 271–278. ACM, 2007.

-
- [194] J. Zhang and P. Pu. A recursive prediction algorithm for collaborative filtering recommender systems. *RecSys '07*, pages 57–64. ACM, 2007.
- [195] L. Zhang, Y. Zhang, and Q. Xing. Filtering semi-structured documents based on faceted feedback. *SIGIR '11*, pages 645–654. ACM, 2011.
- [196] Y. Zhang, B. Cao, and D. yan Yeung. Multi-domain collaborative filtering. *UAI '10*, 2010.
- [197] V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang. Collaborative location and activity recommendations with gps history data. *WWW '10*, pages 1029–1038. ACM, 2010.
- [198] Y. Zheng and X. Xie. Learning travel recommendations from user-generated gps traces. *ACM Trans. Intell. Syst. Technol.*, 2:2:1–2:29, January 2011.
- [199] Y. Zheng, L. Zhang, Z. Ma, X. Xie, and W.-Y. Ma. Recommending friends and locations based on individual location history. *ACM Trans. Web*, 5:5:1–5:44, February 2011.
- [200] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma. Mining interesting locations and travel sequences from gps trajectories. *WWW '09*, pages 791–800. ACM, 2009.
- [201] T. Zhou, H. Ma, M. Lyu, and I. King. Userrec: A user recommendation framework in social tagging systems. *AAAI '10*, AAAI Press, 2010.
- [202] S. Zhu, K. Yu, Y. Chi, and Y. Gong. Combining content and link for classification using matrix factorization. *SIGIR '07*, pages 487–494. ACM, 2007.

Acknowledgements

I would like to express my highest gratitude to my supervisors Alan, Inald and Martha, without whom this thesis would not have been possible. It was truly enjoyable to work with all of them, and their guidance and assistance have provided critical support to my research progress.

I am also grateful to have excellent colleagues and support staff in the DMIR Lab. Over the four years, I have appreciated the help and friendship of Stevan, Maarten, Ronald, Christoph, Carsten, Peng, Wen, Xinchao, Babak, Alessio, Linjun, Arjen, Saskia, and Robbert. In particular, I would like to thank Cynthia and Raynor, who were also always kind in helping me to improve my understanding of the Dutch culture.

Conducting research within the PetaMedia project was an important experience during my PhD journey. Many thanks to the PetaMedia partners from TU Berlin, EPFL, and Queen Mary University of London.

In addition, I would like to thank Alexandros, Linas, and Nuria for the productive and inspiring internship in Barcelona, and for our continued collaboration. I am also grateful to Jun for inspiring discussion and constructive suggestions during the course of my thesis work.

Many Chinese friends brought a lot of joy to my life in the four years. It is impossible to list all their names and what I can say is only, "Thank you all."

Finally, I would like to express my deepest gratitude to my wife Hui Tang for too many things to write down here.

Curriculum Vitae

Yue Shi was born in Jiangsu, China on Oct. 24, 1983. He obtained his B.E degree (in 2006) and M.E. degree (in 2008) both in Electronic Science and Engineering from Southeast University, Nanjing, China.

From Dec. 2008 to Dec. 2012, he was a Ph.D. student at the Multimedia Information Retrieval Lab, Department of Intelligent Systems, Delft University of Technology, the Netherlands, supervised by Prof. Alan Hanjalic, Prof. Inald Lagendijk and Dr. Martha Larson. His Ph.D. research was mainly on the topic of recommender systems, and involved in the EU FP7 project PetaMedia. In 2011, he spent three months working as an intern at Telefonica Research, Barcelona, Spain, mentored by Dr. Alexandros Karatzoglou, Dr. Linas Baltrunas, and Dr. Nuria Oliver. Yue has published over 20 research papers in journals and conference proceedings including ACM SIGIR, ACM RecSys, ICWSM, ECIR. He received the Best Paper award at ACM RecSys 2012 for his work on “Collaborative Less-is-More Filtering”, and Overall Winner Award at ACM RecSys Challenge on Context-aware Movie Recommendation 2010 for his work on “Mood-specific movie recommendation”. He has also served as a program committee member and reviewer for several conferences and journals, such as AAAI, RecSys, CIKM, ACM TOIS, TIST, IEEE TKDE, TMM, TCSVT, Elsevier IPM.