Personalized and adaptive recommendation

Liu, Yong

2016

Liu, Y. (2016). Personalized and adaptive recommendation. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/66909>

<https://doi.org/10.32657/10356/66909>

Personalized and Adaptive Recommendation

LIU YONG

School of Computer Science and Engineering

A thesis submitted to the Nanyang Technological University
in fulfilment of the requirement for the degree of
Doctor of Philosophy

2016

Acknowledgments

I would like to express my gratitude to my supervisor, **Professor Chunyan Miao**, for her constant encouragement and guidance in my academic studies. She allows me an ideal level of freedom in pursuing ideas and research topics. The discussions with her also prompt me to think my research from different perspectives.

I am also indebted to **Professor Aixin Sun** for his influential discussions and suggestions for my research, and his useful comments for the drafts of the papers and thesis. I would like to express my sincere thanks to **Professor Bingsheng He**, who brought me into the research field of location-based social networks.

I would also like to thank my colleagues, Ailiya Borjigin, Chaohai Chen, Han Yu, Peng Cheng, Qiong Wu, Shaohua Li, Siyuan Liu, Xiaogang Han, Yundong Cai, and other great guys in Joint NTU-UBC Research Centre of Excellence in Active Living for Elderly (LILY) for their direct and indirect help in my research work.

I would also want to acknowledge my good friends, Chenliang Li, Haifeng Lu, Jinfeng Zhuang, Juyong Zhang, Linhong Zhu, Lixin Duan, Miao Lin, Min Wu, Ming Yang, Mingkui Tan, Peilin Zhao, Qi Duan, Shenghua Gao, Shengbo Yang, Xiangang Cheng, Xiao Teng, Xin Liu, Xing Jiang, Yu Zhang, Zhiqiang Xu, Zongyang Ma, who made my Ph.D. life much more colorful.

Last, but not least, I would like to thank my wife Sicong Chen, for her support, encouragement, understanding, and love during these years. A special thanks to my cat, Dai Dai, for the happiness he brings me. My parents should receive my deepest gratitude and love for their loving considerations and great confidences in me.

Contents

Acknowledgments	i
Summary	v
List of Figures	vii
List of Tables	ix
List of Abbreviations	x
1 Introduction	1
1.1 Background	1
1.2 Thesis Contributions	3
1.2.1 Personalized Location Recommendation	4
1.2.2 Top- N Recommendation with Users' Implicit Feedback	5
1.2.3 Drug-Target Interaction Recommendation	6
1.3 Thesis Organization	7
2 Background and Related Work	8
2.1 Collaborative Filtering	8
2.1.1 Neighborhood-based Methods	10
2.1.2 Model-based Methods	11
2.2 Personalized Location Recommendation	15
2.2.1 Methods Using GPS Data	15
2.2.2 Methods Using Check-in Data	16
2.3 Structured Sparsity Learning	18
2.4 Boosting	19

2.5	Drug-Target Interaction Recommendation	20
3	Personalized Location Recommendation	23
3.1	Introduction	23
3.2	Empirical Data Analysis	25
3.3	Personalized Location Recommendation by Matrix Factorization . . .	28
3.4	Exploitation of Geographical Characteristics	29
3.4.1	Instance-Level Exploitation	30
3.4.2	Region-Level Exploitation	31
3.4.3	Instance-Region Neighborhood Matrix Factorization	33
3.5	Experiments	36
3.5.1	Experimental Settings	36
3.5.2	Performance Comparison	40
3.5.3	Parameter Sensitivity Analysis	43
3.5.4	Impact of Different Clustering Strategies	45
3.6	Summary	46
4	Top-N Recommendation with Users' Implicit Feedback	47
4.1	Introduction	47
4.2	The Boosting Recommendation Framework	48
4.2.1	Adaptive Boosting Personalized Ranking	49
4.2.2	Constructing Component Recommender	52
4.3	Experiments	54
4.3.1	Experimental Settings	54
4.3.2	Performance Comparison	56
4.3.3	Parameter Sensitivity Analysis	58
4.4	Summary	59
5	Drug-Target Interaction Recommendation	60
5.1	Introduction	60

5.2	Problem Formalization	62
5.3	Drug-Target Interaction Recommendation Model	63
5.3.1	Logistic Matrix Factorization	63
5.3.2	Regularized by Neighborhood	65
5.3.3	Neighborhood Regularized Logistic Matrix Factorization . . .	67
5.4	Experiments	70
5.4.1	Experimental Settings	70
5.4.2	Performance Comparison	72
5.4.3	Neighborhood Benefits	75
5.4.4	Parameter Sensitivity Analysis	77
5.4.5	Recommending Novel Interactions	78
5.5	Summary	82
6	Conclusions and Future Work	84
6.1	Summary of Contributions	84
6.2	Future Work	86
6.2.1	Personalized Location Recommendation	86
6.2.2	Top- N Recommendation with Users' Implicit Feedback	89
6.2.3	Drug-Target Interaction Recommendation	90
	Appendix A Proof for Theorems	92
A.1	The Proof for Theorem 1	92
A.2	The Proof for Theorem 2	93
	Appendix B List of Publications	96
	References	98

Summary

Recommender systems have become an important research area. Plenty of algorithms have been proposed to solve the recommendation problems in various scenarios. This thesis presents the personalized and adaptive recommendation algorithms developed for three emerging application scenarios: personalized location recommendation, top- N recommendation with users' implicit feedback, and drug-target interaction recommendation.

Personalized location recommendation is an important application in the rapid growing location-based social networks. Its objective is to recommend a user a list of locations that she may prefer but has not visited before. This thesis proposes a novel personalized location recommendation method, namely **Instance-Region Neighborhood Matrix Factorization (IRenMF)**. To improve recommendation accuracy, IRenMF exploits two levels of geographical neighborhood characteristics of users' check-in data generated in location-based social networks: (a) instance-level characteristics, *i.e.*, nearest neighboring locations which tend to share more similar user preferences; and (b) region-level characteristics, *i.e.*, locations in the same geographical region which may share similar user preferences.

More generally, many newly emerging recommendation tasks are usually formulated as top- N recommendation problems based on users' implicit feedback instead of explicit feedback. Here, explicit feedback refers to users' ratings to items, while implicit feedback is derived from users' interactions with items, *e.g.*, number of times a user plays a song. This thesis proposes a boosting recommendation algorithm, named **Addaptive Boosting Personalized Ranking (AdaBPR)**, for top- N rec-

ommendation using users’ implicit feedback. In the AdaBPR framework, multiple homogeneous component recommenders are linearly combined to create an ensemble model, for better recommendation accuracy. The component recommenders are constructed based on a fixed matrix factorization algorithm by using a re-weighting strategy, which assigns a dynamic weight distribution on the observed user-item interactions.

In addition, this thesis also presents a study that adopts personalized recommendation technique for the drug-target interaction (DTI) recommendation task in the drug discovery process of pharmaceutical sciences. A novel recommendation approach, called **Neighborhood Regularized Logistic Matrix Factorization (NRLMF)**, has been proposed to suggest a list of potential DTIs. More specifically, NRLMF focuses on modelling the interaction probability of a drug-target pair using logistic matrix factorization. Because the positive observations (*i.e.*, the observed interacting drug-target pairs) are already experimentally verified, they are more trustworthy than the negative observations (*i.e.*, the unknown pairs). NRLMF assigns higher importance levels to positive observations. To further improve the accuracy, NRLMF exploits the local structure of DTI data by neighborhood regularization.

Extensive experiments on real datasets have been performed to demonstrate the effectiveness of the proposed methods, compared with state-of-the-art methods.

List of Figures

1.1	An example of product recommendations in Amazon.	2
1.2	An example of rating prediction and top- N recommendation tasks. .	3
3.1	The instance-level neighborhood characteristics of check-in data. . . .	26
3.2	The region-level neighborhood characteristics of check-in data.	27
3.3	An example of the structured sparsity assumption of the location latent factors.	32
3.4	Performances of methods without geographical characteristics.	40
3.5	Performances of methods with geographical characteristics.	42
3.6	Impact of geographical neighborhood characteristics in IRenMF. . . .	44
3.7	The location recommendation accuracy with respect to different clustering strategies.	45
4.1	Performance trend of AdaBPR measured by MAP, AUC, and NDCG@10.	58
5.1	Performance trend of NRLMF measured by AUC and AUPR with different settings of K_1 under cross-validation setting CVS1.	76
5.2	Performance trend of NRLMF measured by AUC and AUPR with different settings of K_2 under cross-validation setting CVS1.	77
5.3	Performance trend of NRLMF measured by AUC and AUPR with different settings of c under cross-validation setting CVS1.	78
5.4	Performance trend of NRLMF measured by AUC and AUPR with different settings of r under cross-validation setting CVS1.	79

6.1	An example of the category tree associated with locations.	87
6.2	An example of the tips given by foursquare users.	88

List of Tables

3.1	The statistics of users' check-in data in four popular cities.	37
4.1	The statistics of the experimental datasets.	54
4.2	The accuracies of different recommendation algorithms measured by MAP, AUC, and NDCG@10.	57
5.1	The statistics of the drug-target interaction datasets.	70
5.2	The AUC and AUPR obtained under cross-validation setting CVS1. .	72
5.3	The AUC and AUPR obtained under cross-validation setting CVS2. .	73
5.4	The AUC and AUPR obtained under cross-validation setting CVS3. .	74
5.5	The top 30 novel interactions predicted by NRLMF on GPCR dataset.	80
5.6	The fractions of true DTIs among the predicted top N ($N=10, 30, 50$) interactions under cross-validation setting CVS1.	81

List of Abbreviations

AdaBPR Adaptive Boosting Personalized Ranking

AUC Area Under the ROC Curve

AUPR Area Under the Precision-Recall Curve

CF Collaborative Filtering

CV Cross-Validation

CVS Cross-Validation Setting

DTI Drug-Target Interaction

IRenMF Instance-Region Neighborhood Matrix Factorization

LBSNs Location-based Social Networks

MAP Mean Average Precision

MF Matrix Factorization

NDCG Normalized Discounted Cumulative Gain

NRLMF Neighborhood Regularized Logistic Matrix Factorization

PLRs Personalized Location Recommender Systems

Chapter 1

Introduction

1.1 Background

The exponential expansion of data generated in our lives makes the information filtering technologies, such as recommender systems, which can help users filter out useful information from a massive amount of data, become increasingly important. A recommender system is usually designed to recommend a user a list of information items (*e.g.*, books, movies, and products) that may be preferred by her. In recent years, recommender systems have been widely used for E-commerce, for example the movie recommendation in Netflix¹ and product recommendation in Amazon². Figure 1.1 shows an example of the product recommendations in Amazon. There are also other recommender systems developed for the newly emerging application scenarios, such as friend recommendation in online social networks [KLZ12] and location recommendation in location-based social networks [BZWM15].

Generally speaking, there are two main categories of recommendation tasks: **rating prediction** and **top- N recommendation**. The objective of rating prediction is to recover the missing entries of the underlying rating matrix based on a given subset of its entries. Examples include movie rating prediction in Netflix [BK07,

¹<https://www.netflix.com/>

²<http://www.amazon.com/>

Figure 1.1: An example of product recommendations in Amazon.

ZWSP08, Kor08, KBV09, Kor09] and business rating prediction in Yelp³ [BEF13, HSL14, TBK⁺14, LZZW15]. For top- N recommendation, the recommender systems focus on ranking the missing entries instead of predicting the exact values of the missing entries. Product recommendation in Amazon [LSY03], friend recommendation in online social networks [CGD⁺09], and location recommendation in location-based social networks [LLAM13, LWSM14, BZWM15] are examples of top- N recommendation tasks. Figure 1.2 shows an example of different recommendation tasks.

From the data perspective, two types of data, *i.e.*, **explicit feedback** and **implicit feedback**, are usually used to build the recommendation models. Considering the user-item interaction data as an example, explicit feedback refers to users' ratings to items, and implicit feedback is derived from users' interactions with items, *e.g.*, the number of times a user visits a location and the number of times a user plays a song. The explicit feedback data are usually represented using numeric values, for example the 1-to-5 scale ratings in Netflix. The implicit feedback data are usually described using binary ratings [DK04]. If a user has interacted with an item, the rating of the user-item pair is set to 1; and 0 otherwise. Note that 0 does not explicitly indicate that the user is not interested in the item. It may be the result that the user does not even know the existence of the item.

³<http://www.yelp.com/>

Figure 1.2: An example of rating prediction and top- N recommendation tasks.

Due to the great research challenges and the potential huge commercial value, recommender systems have gotten a lot of attentions from both the academia and the industry. Lots of algorithms and real systems have been developed during the past few years [AT05, SK09, SLH14]. In the literature, the collaborative filtering based methods have been widely used to solve the rating prediction and top- N recommendation problems, and matrix factorization based models are the most successful collaborative filtering methods. The objective of matrix factorization is to project the high dimensional data to low dimensional latent spaces, for better understanding of the data and thus achieving more accurate recommendations.

1.2 Thesis Contributions

In particular, this thesis focuses on developing novel recommendation algorithms for three emerging application scenarios, including personalized location recommendation, top- N recommendation with users' implicit feedback, and drug-target interaction recommendation.

1.2.1 Personalized Location Recommendation

The rapid development of mobile technologies motivates the emergence of location-based social networks (LBSNs), which constitute a framework for previous independently launched location-based services, *e.g.*, location-based search, friend finder, and people tracking, to complement and reinforce each other. LBSNs mainly focus on geographical social interactions. The users are encouraged to explore their surroundings and share experiences with online friends. Usually, LBSNs are grounded on the concept of “check-in”: users willingly share their locations with online friends by broadcasting their current places through a location-sensing mobile device [SM11, Zhe12].

The available check-in data generated in LBSNs contain rich knowledge about users’ interests and thus are beneficial to a wide range of applications, *e.g.*, location recommendation [ZCZ⁺10], friend recommendation [SNM11], and activity recommendation [ZZXY10]. As an important application in LBSNs, the personalized location recommender system can help users explore new locations to enrich their experiences, via recommending a user a set of locations that she may prefer but has not visited before. On the other hand, the personalized location recommender system can also facilitate third-party developers (*e.g.*, advertisers) to provide more relevant services at the right locations.

In previous work, the geographical characteristics derived from users’ historical check-in data have been reported effective in improving location recommendation accuracy. However, previous studies mainly exploit geographical characteristics from a user’s perspective, via modeling the geographical distribution of each individual user’s check-ins. This thesis (Chapter 3) focuses on exploiting the geographical characteristics from the location perspective, by modeling the geographical neighborhood of a location. The neighborhood is modeled at two different levels: the instance-level neighborhood defined by a few nearest neighbors of the location and the region-level neighborhood for the geographical region where the location exists. This thesis introduces a novel location recommendation approach, namely Instance-

Region Neighborhood Matrix Factorization (IRenMF), which exploits two levels of geographical neighborhood characteristics: (a) instance-level characteristics, *i.e.*, nearest neighboring locations which tend to share more similar user preferences; and (b) region-level characteristics, *i.e.*, locations in the same geographical region which may share similar user preferences. In the proposed IRenMF method, the two levels of geographical characteristics are naturally incorporated into the learning of latent features of users and locations, so that IRenMF predicts users' preferences on locations more accurately.

1.2.2 Top- N Recommendation with Users' Implicit Feedback

In many application scenarios, the recommendation tasks are formulated as top- N recommendation problems based on users' implicit feedback instead of explicit feedback. Here explicit feedback refers to users' ratings to items while implicit feedback is derived from users' interactions with items, such as the number of times a user plays a song. For item recommendation with implicit feedback, users' preferences on items are unobservable. For example, in the music recommendation task, the number of times a user plays a song (*i.e.*, implicit feedback) is observable but not her preference rating to the song (*i.e.*, explicit feedback). Recommendation models are then built based on some pre-defined preference assumptions, *e.g.*, the more number of times a user plays a song, the more the user likes the song. Such assumptions may not accurately describe users preferences. In addition, the observed user-item interaction data is generally very sparse, which makes the preference modeling even more challenging. As a result, existing solutions often deliver unsatisfactory recommendation accuracies.

Instead of directly addressing these challenges in item recommendation with implicit feedback, this thesis (Chapter 4) exploits a boosting technique to improve the recommendation accuracy via combining the recommendation power of multiple "weak recommenders". A boosting recommendation method named Adaptive Boosting Personalized Ranking (AdaBPR) has been proposed for top- N item recom-

mendation with users' implicit feedback. In the proposed AdaBPR framework, multiple homogeneous component recommenders are linearly combined to create an ensemble model, for better recommendation accuracy. The component recommenders are constructed based on a fixed collaborative filtering algorithm by using a re-weighting strategy, which assigns a dynamic weight distribution on the observed user-item interactions.

1.2.3 Drug-Target Interaction Recommendation

The drug discovery is one of the primary objectives of pharmaceutical science, which is an interdisciplinary research field of fundamental sciences covering biology, chemistry, physics, statistics, *etc.* In the drug discovery process, the identification of drug-target interactions (DTIs) is a crucial step. The most common targets of currently marketed drugs include proteins (*e.g.*, G protein-coupled receptors, enzymes, and ion channels) and nucleic acids [OALH06, LG08]. As the experimental validation is laborious and costly, only a small portion of DTIs have been experimentally validated. To improve the drug discovery efficiency, there is a great need for the development of accurate computational approaches that can recommend potential DTIs to guide the experimental validation and reduce costs.

The DTI recommendation problem is a top- N recommendation task, which aims to recommend biologists or clinicians a list of potential interacting drug-target pairs. Thus, some recent work exploits matrix factorization for DTI recommendation [Gön12, CLH⁺13, ZDMZ13]. This thesis (Chapter 5) proposes a novel DTI recommendation approach, called Neighborhood Regularized Logistic Matrix Factorization (NRLMF). Specifically, the proposed NRLMF focuses on modelling the probability that a drug would interact with a target by logistic matrix factorization, where the properties of drugs and targets are represented using drug-specific and target-specific latent vectors, respectively. In addition, NRLMF assigns higher importance levels to positive observations (*i.e.*, the observed interacting drug-target pairs) than negative observations (*i.e.*, the unknown pairs). Because the positive observations are already experimen-

tally verified, they are usually more trustworthy. Furthermore, the local structure of the interaction data has also been exploited via neighborhood regularization to achieve better accuracy.

1.3 Thesis Organization

This thesis is organized as follows. Chapter 2 first reviews the existing research work that is most related to different problems presented in this thesis. From Chapter 3 to Chapter 5, the novel recommendation methods proposed in this thesis are presented in details.

Chapter 3 introduces the proposed IRenMF model for personalized location recommendation in LBSNs. Extensive experiments on the real data collected from Gowalla, a popular location-based social network, have been conducted to demonstrate the effectiveness and advantages of the proposed approach.

Chapter 4 presents the details of the AdaBPR model that exploits boosting to improve the accuracy of top- N recommendation based on users' implicit feedback. Empirical experiments demonstrate AdaBPR outperforms four strong baselines on three real datasets.

Chapter 5 presents a novel DTI recommendation approach, named NRLMF, which aims to predict the interaction probability of a given drug-target pair. The performances of NRLMF are empirically evaluated on four benchmark datasets, compared with five state-of-the-art methods.

Finally, Chapter 6 draws the conclusion of this thesis and outlines potential directions for future work.

Chapter 2

Background and Related Work

This chapter reviews the state-of-the-art related to the research problems presented in this thesis, particularly focusing on the following research areas: collaborative filtering, personalized location recommendation, structured sparsity learning, boosting, and drug-target interaction recommendation.

2.1 Collaborative Filtering

In the last decades, lots of algorithms have been proposed to solve the rating prediction and top- N recommendation problems [AT05, SK09, SLH14]. Typically, existing methods can be categorized into three main groups¹:

- *Content-based filtering*: The user will be recommended items that are similar to the ones she preferred in the past.
- *Collaborative filtering (CF)*: The user will be recommended items that have been liked by other users with similar preferences.
- *Hybrid approaches*: These approaches combine both content-based and collaborative filtering approaches to recommend items.

¹For better understanding, this section only focuses on the recommender systems developed based on the user-item interaction data. The recommender systems built with other data, *e.g.*, user-user interaction data and drug-target interaction data, can also be similarly categorized.

This thesis only focus on the CF based approaches, as they are most widely adopted for rating predicting and top- N recommendation tasks.

Most existing recommendation approaches are proposed to solve the rating prediction problems, *e.g.*, Bayesian models [CG99, MP00, MP02, SK06], clustering models [UF98, Mer99, CHW01, SJ03a, BKG⁺05, GM05, XLY⁺05], latent semantic models [HP99, Hof01, Hof03, Hof04, Mar04], and matrix factorization models [SJ03b, SRJ04, ZWFM06, SM08, MS08]. Among these approaches, the matrix factorization based models are the most successful recommendation approaches and have been applied to various rating prediction tasks [KBV09, WHZL13, LHWZ13, HSL14]. The primary objective of rating prediction is to predict the exact rating a user may assign to an item. Therefore, the rating prediction algorithms are developed based on users' explicit ratings, and the performances of the algorithms are usually evaluated by error metrics, *e.g.*, Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).

In many practical scenarios, the objective of recommender systems is to suggest a list of interesting items that may be most preferred by a user, which is usually referred as a top- N recommendation task and formulated as a ranking problem. All the recommendation problems studied in this thesis are top- N recommendation problems. An intuitive solution to the top- N recommendation problems is: (a) recovering missing entries of the partially observed rating matrix using a rating prediction algorithm, and (b) ranking the missing entries according to the predicted rating values and choosing top-ranked entries as recommendations. However, for top- N recommendation tasks, the performances of recommendation models are evaluated by accuracy metrics, *e.g.*, precision, recall, the Area Under the ROC Curve (AUC), and the Mean Average Precision (MAP). These metrics focus on whether the user will interact with the item rather than the exact rating value the user will assign to the item. There is no trivial relationship between the error metrics and the accuracy metrics. Thus, the rating prediction algorithms usually do not perform well in top- N recommendation tasks [CKT10]. In the literature, many alternative algorithms have been proposed to specifically solve the top- N recommendation problems.

Existing CF approaches developed for the top- N recommendation tasks can be categorized into two main groups: neighborhood-based (or memory-based) CF methods and model-based CF methods.

2.1.1 Neighborhood-based Methods

The most widely used neighborhood-based CF approaches are user-based CF approaches [BHK98, HKBR99, JCS04] and item-based CF approaches [SKKR01, LSY03, DK04]. The user-based approaches predict a user's preference on a given item, considering the preferences of several most similar users on the target item. In contrast, the item-based approaches first find several most similar items to the target item and then predict a user's preference on the target item based on her preferences on the most similar items.

Pearson correlation coefficient [RIS⁺94] and Cosine similarity [BHK98] are usually adopted to compute the user similarities and item similarities. Let $\mathbf{R} \in \mathbb{R}^{m \times n}$ denote the rating matrix used to describe the preferences of m users $\{u_i\}_{i=1}^m$ over n items $\{v_j\}_{j=1}^n$. In $\mathbf{R}$, the (i, j) element R_{ij} denotes u_i 's preference on v_j . The Pearson correlation coefficient based similarity $\Phi(u_i, u_{i'})$ and Cosine similarity $\Psi(u_i, u_{i'})$ between two users u_i and $u_{i'}$ are defined as follows:

$$\begin{aligned}\Phi(u_i, u_{i'}) &= \frac{\sum_{v_j \in I(u_i) \cap I(u_{i'})} (R_{ij} - \bar{R}_{u_i}) \cdot (R_{i'j} - \bar{R}_{u_{i'}})}{\sqrt{\sum_{v_j \in I(u_i) \cap I(u_{i'})} (R_{ij} - \bar{R}_{u_i})^2} \cdot \sqrt{\sum_{v_j \in I(u_i) \cap I(u_{i'})} (R_{i'j} - \bar{R}_{u_{i'}})^2}} \\ \Psi(u_i, u_{i'}) &= \frac{\sum_{v_j \in I(u_i) \cap I(u_{i'})} R_{ij} \cdot R_{i'j}}{\sqrt{\sum_{v_j \in I(u_i) \cap I(u_{i'})} R_{ij}^2} \cdot \sqrt{\sum_{v_j \in I(u_i) \cap I(u_{i'})} R_{i'j}^2}},\end{aligned}\tag{2.1}$$

where $\bar{R}_{u_i}$ represents the average ratings given by u_i , and $I(u_i)$ denotes the set of items u_i has rated. The item-based approaches are similar to the user-based approaches. The difference is that the item-based approaches compute the item similarities instead of the user similarities. In a similar manner, the Pearson correlation coefficient based similarity $\Phi(v_j, v_{j'})$ and Cosine similarity $\Psi(v_j, v_{j'})$ between two items v_j and $v_{j'}$ are

defined as follows:

$$\begin{aligned}
 \Phi(v_j, v_{j'}) &= \frac{\sum_{u_i \in U(v_j) \cap U(v_{j'})} (R_{ij} - \bar{R}_{v_j}) \cdot (R_{ij'} - \bar{R}_{v_{j'}})}{\sqrt{\sum_{u_i \in U(v_j) \cap U(v_{j'})} (R_{ij} - \bar{R}_{v_j})^2} \cdot \sqrt{\sum_{u_i \in U(v_j) \cap U(v_{j'})} (R_{ij'} - \bar{R}_{v_{j'}})^2}}, \\
 \Psi(v_j, v_{j'}) &= \frac{\sum_{u_i \in U(v_j) \cap U(v_{j'})} R_{ij} \cdot R_{ij'}}{\sqrt{\sum_{u_i \in U(v_j) \cap U(v_{j'})} R_{ij}^2} \cdot \sqrt{\sum_{u_i \in U(v_j) \cap U(v_{j'})} R_{ij'}^2}}, \tag{2.2}
 \end{aligned}$$

where $U(v_j)$ denotes the set of users that have rated v_j , and $\bar{R}_{v_j}$ represents the average rating value given to v_j .

Once completing the calculation of user similarities or item similarities, the prediction $\hat{R}_{ij}$ of the missing rating to v_j , given by u_i , can be determined via aggregating the ratings of the most similar users or items. For example, the following aggregating functions are usually used in the user-based approaches [AT05]:

$$\begin{aligned}
 (1) \hat{R}_{ij} &= \frac{1}{\sum_{u_{i'} \in N(u_i)} \Psi(u_i, u_{i'})} \sum_{u_{i'} \in N(u_i)} \Psi(u_i, u_{i'}) \cdot R_{i'j}, \\
 (2) \hat{R}_{ij} &= \bar{R}_{u_i} + \frac{1}{\sum_{u_{i'} \in N(u_i)} \Psi(u_i, u_{i'})} \sum_{u_{i'} \in N(u_i)} \Psi(u_i, u_{i'}) \cdot (R_{i'j} - \bar{R}_{u_{i'}}), \tag{2.3}
 \end{aligned}$$

where $N(u_i)$ denotes the set of users that are most similar with u_i .

2.1.2 Model-based Methods

For top- N recommendation, the primary objective of model-based approaches is to learn different ranking functions to address the recommendation problems. These approaches can be categorized into three main groups: pointwise, pairwise, and listwise approaches, according to the loss functions used to build the models [KBS13].

Pointwise Methods. The most representative pointwise approach is Weighted Regularized Matrix Factorization (WRMF) [PZC⁺08, HKV08, PS09, Ste10], which treats all observed entries in the user-item interaction matrix as positive examples and all missing entries as negative examples. Different confidence levels are assigned

to the positive and negative examples. A least square loss function is then used to learn the latent features of users and items by approximating the pre-defined ratings. Sindhwani *et al.* [SBHM10] treated the missing entries in the user-item interaction matrix as optimization variables. They proposed a weighted, low-rank non-negative matrix factorization model to obtain the optimal labels for the missing entries. Yao *et al.* [YTY⁺14] proposed a dual-regularized model, which integrated the weighting-based method [PZC⁺08] and imputation-based method [SBHM10] for top- N recommendation. In [CKT10], Cremonesi *et al.* presented the PureSVD method that modeled each user as a combination of the item features. It did not need to learn user-specific parameters thus it can offer convenient optimization. Koren and Sill [KS11] treated user feedback on items as ordinal and proposed a pointwise ordinal model, namely OrdRec, for top- N recommendation. The OrdRec method predicted a probability distribution, rather than a single score, for the target user-item rating. In [NK11], Ning and Karypis proposed a Sparse Linear Method (SLIM), which employed a sparse linear model for top- N recommendation. In SLIM, the recommendation score for a new item was calculated as an aggregation of other items. This approach was extended to incorporate the side information of items to improve the recommendation accuracy in [NK12]. Recently, Kabbur *et al.* [KNK13] proposed a item-based top- N recommendation approach that learned the item similarity matrix as the product of two low dimensional latent factor matrices. Johnson [Joh14] proposed a logistic matrix factorization model which focused on modeling the probability that a user would prefer a specific item.

Pairwise Methods. A common assumption of the pairwise approaches is that a user prefers the items that she has interacted with than those items that she has not interacted with before. The Bayesian Personalized Ranking (BPR) [RFGST09] is the most representative pairwise approach. The objective of BPR is to rank the interacted items higher than the un-interacted items in the ranking list of items for each user. In [RFGST09], the optimization criterion proposed for BPR was demonstrated to be equivalent to optimizing the ranking metric AUC of the model on the task. The BPR

framework was also extended to incorporate other auxiliary data for top- N recommendation, *e.g.*, temporal information [RFST10], user social relationship [DLS11], and the taxonomy structure associated with items [KAP⁺12]. In [PC13a], Pan and Chen proposed the CoFiSet model, which generalized BPR via proposing a new assumption that an individual user is likely to prefer a set of observed items to a set of unobserved items. Moreover, Pan and Chen [PC13b] proposed the Group Bayesian Personalized Ranking (GBPR) model that combined the pairwise preference and group preference for top- N recommendation. Zhang *et al.* [ZCWX13] proposed to optimize the recommendation accuracy by dynamically choosing negative training samples from the current prediction model and iteratively updating the proposed model. In [ZPX⁺14], Zhong *et al.* introduced an adaptive pairwise preference learning algorithm to improve the efficiency and accuracy of pairwise preference learning algorithms. Rendle and Freudenthaler [RF14] developed a non-uniform item sampler to improve the convergence of widely used stochastic gradient descent solution for BPR-based approaches. In the recent work [WWY13, HDD13], the Weighted Approximately Ranked Pairwise (WARP) loss, which was first proposed in [UBG09], was demonstrated to be effective for top- N recommendation tasks. Minimizing the WARP loss defined based on different parameters can achieve the optimization of different accuracy metrics, *e.g.*, precision, AUC, and MAP. In addition, Lee *et al.* [LBK⁺14] proposed a Local Collaborative Ranking (LCR) model, which assumed the user-item interaction matrix was locally low-rank. The LCR model combined a set of local low-rank matrices, each of which was trained by minimizing the pairwise ranking loss. Kapicioglu *et al.* [KRSJ14] developed the Collaborative Local Ranking (CLR) framework that explored users' preferences on subsets of items. The CLR model was learned by minimizing a local ranking loss, which can be viewed as a local version of AUC.

Listwise Methods. The objective of listwise approaches was to optimize the loss function defined based on more complex ranking metrics. For example, Weimer *et al.* [WKLS07] proposed the CofiRank method that employed Maximum Margin Matrix Factorization (MMMF) [SRJ04] to optimize the ranking measure, *i.e.*, the

Normalized Discounted Cumulative Gain (NDCG). Shi *et al.* [SLH10] presented a ranking approach, called ListRank-MF, which combined a listwise learn-to-rank algorithm with matrix factorization for top- N recommendation. In [SKB⁺12a], Shi *et al.* developed the Tensor Factorization for MAP maximization (TFMAP), which directly optimized the ranking metric MAP and created an optimally ranked item list for an individual user under a given context. Moreover, the Collaborative Less-is-More Filtering (CLiMF) model proposed in [SKB⁺12b] explored the optimization of Mean Reciprocal Rank (MRR), a metric for measuring top- N item recommendation. Liu *et al.* [LZSM15] introduced a recommendation algorithm, called Adaptive Boosting Personalized Ranking (AdaBPR), which exploited boosting to improve the recommendation accuracy. AdaBPR adopted the recommendation power of multiple component models that were created by optimizing the weighted AUC using a pairwise approach.

Other Methods. To improve the recommendation accuracy, Li *et al.* [LHZC10] proposed two methods to exploit the rich information associated with users, *e.g.*, searching history and browsing history. One was to linearly combine the scores from different sources, and the other one was to embed the user information into matrix factorization based models. Yang *et al.* [YSGL12] proposed to incorporate the social relationship between users into the collaborative filtering models for more accurate top- N recommendations. Moreover, in [YLS⁺11], Yang *et al.* presented the Collaborative Competitive Filtering (CCF) framework to learn the user preferences by modeling users' behaviors in their interactions with the recommender systems. In [BC12], Balakrishnan and Chopra utilized Probabilistic Matrix Factorization (PMF) [MS07] to learn latent features for users and items. Then, they employed the pointwise and pairwise learning-to-rank models to optimize the ranking metric NDCG. Recently, Paquet and Koenigstein [PK13] proposed a Bayesian graphical model, in which the users and items were connected using a random graph. The proposed model was learned by combining stochastic gradient descent and mean field variational inference over the random graph samples.

2.2 Personalized Location Recommendation

The location recommendation is an important feature provided by the location-based services. The objective of location recommendation is to suggest users a set of interesting locations that they may prefer but have not visited before, considering their visited locations.

2.2.1 Methods Using GPS Data

Traditional location recommender systems were mainly based on users' location history extracted from their GPS trajectories. Takeuchi and Sugimoto [TS06] employed an item-based CF algorithm to suggest shops to users according to their movement history and the geographical conditions of the city. Simialarly, Horozov *et al.* [HNV06] proposed a user-based collaborative location recommender system, called Geowhiz, to recommend an individual user relevant restaurants that other similar users usually visit. Zheng *et al.* [ZZXM09] reported a research about mining interesting places and travel sequences from the movement history of a group of users over a period of one year in the real world. A Tree-Based Hierarchical Graph (TBHG) was proposed to model users' movement history. Moreover, an inference model based on Hyperlink-Induced Topic Search (HITS) was used to compute the interestingness scores of locations and authority scores of users, considering the relationship between the interestingness of a place and the travel experience of a user. Recently, the location-activity collaborative recommendation tasks were studied in [ZZXY10,ZCZ⁺10,LLL11]. The objective was to recommend users interesting locations where they can perform a specific activity or recommend them interesting activities that they can perform at the specified location. In [ZZXY10,ZCZ⁺10], the input user-location-activity relations were modeled using a tensor representation. A collective matrix factorization based approach was utilized to take these input information to train a location and activity recommender. Leung *et al.* [LLL11] tried to use activity as an additional information to improve the accuracy of location recommendation. The users, loca-

tions, and activities were incorporated into a Community Location Model (CLM). A dynamic co-clustering algorithm, *i.e.*, Community-based Agglomerative-Divisive Clustering (CADC), was proposed to cluster GPS trajectories into groups of similar users, similar activities, and similar locations. The groups would be incrementally updated when a new user or movement trajectory coming.

2.2.2 Methods Using Check-in Data

In recent years, the popularity of location-based social networks (LBSNs) motivated the development of location recommender systems based on users' check-in data generated in LBSNs [YYL10, BS11, AS11, BZWM15]. Ye *et al.* [YYL10] reported the first research work about providing location recommendation services for large scale LBSNs. They proposed two friend-based collaborative filtering approaches to predict a user's preferences on locations, considering the preferences of her social friends. To achieve more accurate recommendations, Ye *et al.* [YYLL11] proposed a unified recommendation framework to incorporate the user preference, geographical influence and social influence into one recommendation process. They designed a probabilistic model to capture the geographical influence from a specific user's visited locations. Cheng *et al.* [CYKL12] studied users' multi-center check-in behavior and employed a multi-center gaussian model to capture the geographical influence. Then, a matrix factorization model was fused with geographical and social influence for location recommendation in LBSNs. Bao *et al.* [BZM12] generated location recommendations taking advantages of both the user preferences and the opinions from local experts in a new city. Levandoski *et al.* [LSEM12] proposed a location-aware recommender system, named LARS, which employed the location-based ratings to provide recommendations. In [LFYX13], Liu *et al.* introduced a probabilistic recommendation framework that employed the geographical influence on a user's check-in behavior, the user mobility pattern, and the user check-in count data for location recommendation. Zhang *et al.* [ZC13] developed a unified geo-social recommendation framework, called iGSLR, in which a kernel density estimation approach was used to personalize the

geographical influence on users' check-in behaviors as individual distributions. Wang *et al.* [WTM13] proposed algorithms that generated recommendations based on the past user behavior (visited locations), the longitude and latitude of each location, the social relationships among users, and the similarity between users. Moreover, Yin *et al.* [YSC⁺13] introduced a location-content-aware recommender system that offered a user a set of venues or events, considering both personal interest and the local preference of each individual city. In [LZX⁺14], Lian *et al.* utilized matrix factorization to jointly model users' activity areas and the influence areas of locations for location recommendation.

Additionally, there also existed some location recommender systems that exploited the temporal characteristics of users' check-in data for location recommendation. For example, Yuan *et al.* [YCM⁺13] developed a time-aware recommendation model to recommend locations for a given user at a specified time in a day. To further enhance the recommendation model, they also considered the geographical information using a probabilistic approach similar to [YYLL11]. In [LLAM13], Liu *et al.* studied the transition patterns of users' preference over location categories for location recommendation. The matrix factorization technique was employed to predict a user's preference transitions over categories and then her preference on locations in the corresponding categories. Similarly, Ye *et al.* [YZC13] employed the location category information to model the user movement pattern. They utilized a mixed hidden Markov model to predict the category of user activity at the next step and then predicted the most likely location given the estimated category distribution. Cheng *et al.* [CYLK13] proposed a matrix factorization model to solve the successive check-in location recommendation task. Two properties of users' check-in sequences (*i.e.*, personalized Markov chain and region localization) were exploited. Gao *et al.* [GTHL13] proposed a location recommendation framework that leveraged two temporal properties (*i.e.*, non-uniformness and consecutiveness) of a user's daily check-in preferences. They also introduced four temporal aggregation strategies to integrate a user's check-in preferences of different temporal states. Yuan *et al.* [YCS14] introduced the

Geographical-Temporal influences Aware Graph (GTAG) to model users' check-in records, geographical influence, and temporal influence for the time-aware location recommendation problem.

2.3 Structured Sparsity Learning

For feature learning problems in high-dimensional space, sparsity is a desirable property of feature vectors. The L_1 -norm penalty has been widely used to enforce sparsity to the learned feature vectors. Numerous methods have been proposed to solve the L_1 -regularization regression problems in statistics, machine learning, and computer vision. For more discussions, please refer to [HTFF09] (Chapter 18). However, the standard L_1 -norm penalty does not consider any dependency (*e.g.*, the group structure) among inputs. This limits its applicability in many complex application scenarios.

In recent years, the idea of using structure in addition to sparsity has been explored. A lot of structured sparsity learning approaches have been proposed to induce different joint sparsity patterns to related variables, by employing more structured constraints on inputs. Among existing methods, the group structure has received much attention recently [Hua11, Che13]. For example, Yuan and Lin [YL06] proposed the group lasso model that assumes disjoint group structures of inputs and enforces sparsity on the pre-defined groups of features. It employed a L_1/L_p mixed norm penalty, with $p > 1$, to achieve group level variable selection. In practice, L_1/L_2 and L_1/L_∞ were the most popular mixed norms. Theoretically, it has been demonstrated that the L_1/L_2 penalty has the potential to improve the accuracy of estimator [HZ10]. Moreover, the group lasso has been extended to allow more complex structures such as hierarchical groups and overlapping groups [JOV09, ZRY09].

The structured sparsity learning approaches have been applied to different real scenarios. For example, Kim and Xing [KX10] studied the gene selection problems. They proposed a tree-guided group lasso regression method, which applied group lasso

to the groups of output variables defined by a hierarchical clustering tree. In [JOB10], Jenatton *et al.* presented an extension of sparse PCA based on a structured regularization that encoded high order information of the input data. Their experiments on face recognition task demonstrated the advantages of structured approach over unstructured approaches. Moreover, the group lasso models have also been applied in time series data analysis problems [KSX09].

In this thesis, we adopt the structured sparsity learning technique to incorporate the geographical group structure associated with users' check-in data into the learning of user-specific and location-specific latent features. Please refer to Chapter 3 for more details.

2.4 Boosting

Boosting is a general technique that can improve the accuracy of a given weak learning algorithm [FS95]. The basic idea is to repeatedly construct a set of weak models using the weak learning algorithm on re-weighted training data. The weak models are then linearly combined to form a strong model, for better accuracy. The boosting technique was originally developed for the binary classification setting, where AdaBoost [FS95] was the most well-known algorithm. It has also been extended to deal with other problems, including multi-class classification [SS99], regression [DH02], and ranking [XL07].

Recently, the boosting technique has also been introduced to solve recommendation problems. In [JNG⁺13], two boosting frameworks based on AdaBoost have been proposed to enhance the performances of collaborative filtering approaches in rating prediction tasks. Each framework consisted of a set of homogeneous recommenders created by the same collaborative filtering algorithm with different sample weights. The recent work [CXZ⁺14] exploited the gradient boosting for context-aware recommendation. Specifically, a Gradient Boosting Factorization Machine (GBFM) model was proposed to select effective interaction features for rating pre-

diction. In [WSZ14], the Adaptive Boosting Matrix Factorization (AdaMF) model extended AdaRank [XL07] for item recommendation, where the sample weights were assigned to individual users, and the component recommenders were constructed using the collaborative filtering algorithm for rating prediction.

Among the exiting recommendation frameworks, the proposed AdaBPR in Chapter 4 is most related to AdaMF, but with two significant differences. First, in AdaBPR, a sample weight is assigned to each observed *user-item interaction*, but not an *individual user* as in AdaMF. Second, the component recommenders of AdaBPR are created by directly optimizing the weighted ranking metric (*i.e.*, weighted AUC), not approximating users' ratings in AdaMF. Empirical evaluations show that these two differences make AdaBPR much more effective for item recommendation tasks.

2.5 Drug-Target Interaction Recommendation

In the drug discovery process, drug-target interaction (DTI) recommendation is an important step that aims to suggest potential new drugs or new targets for existing drugs. Therefore, it can help to guide the experimental validation and reduce costs. The DTI recommendation has recently attracted vast research attentions, and numerous algorithms have been proposed [DZ15].

In general, traditional computational methods proposed for DTI recommendation can be categorized into two main groups: docking simulation approaches² and ligand-based approaches³ [KLB⁺09, LAJ11, WSZ⁺13]. The docking simulation approaches predict potential DTIs, considering the structural information of the target proteins. However, the docking simulation is extensively time-consuming, and the structural information may not be available for some protein families, for example the G-Protein

²In the filed of molecular modeling, docking is a method that predicts the preferred orientation of one molecule to a second when bound to each other to form a stable complex [LR96].

³In biochemistry and pharmacology, a ligand is a substance (e.g., drug and hormone) that forms a complex with a bio-molecule to serve a biological purpose [Ken09]. For example, in protein-ligand binding, the ligand is usually a signal-triggering molecule binding to a site on a target protein. The binding typically results in a change of conformation of the target protein.

Coupled Receptors (GPCRs). The ligand-based approaches compare a candidate ligand with the known ligands of the target proteins for DTI recommendation. The assumption is that two similar ligands are likely to have similar properties and will bind to the same group of proteins [KRA⁺07]. The chemical similarity of the ligands are calculated using the ligand topology structures. This kind of approaches may not perform well for the targets with a small number of ligands.

The recent quick development of machine learning techniques provides effective and efficient ways to suggest DTIs [LZC09, CLJ⁺12, CDW12]. An intuitive idea is to formulate the DTI recommendation as a binary classification problem, where the drug-target pairs are treated as instances, and the chemical structures of drugs and the amino acid subsequences of targets are treated as features. Then, classical classification methods, *e.g.*, Support Vector Machines (SVM) and Regularized Least Square (RLS), can be used. For example, in [NS07], a SVM model was utilized to classify a given drug-target pair into interaction (*i.e.*, positive class) and non-interaction (*i.e.*, negative class), considering the amino acid sequences of proteins, chemical structures, and the mass spectrometry data. Bleakley and Yamanishi proposed a supervised approach for DTI recommendation based on the bipartite local models (BLMs), where SVM was used to build the local models [BY09]. Xia *et al.* proposed a semi-supervised DTI recommendation approach, namely Laplacian regularized least square (LapRLS), and extended it to incorporate the kernel constructed from the known DTI network [XWZW10]. van Laarhoven *et al.* defined a Gaussian Interaction Profile (GIP) kernel to represent the interactions between drugs and targets, and they employed RLS with the GIP kernel for DTI recommendation problems [vLNM11, vLM13]. Mei *et al.* integrated the BLM method with a neighbor-based interaction-profile inferring (NII) procedure to form a DTI recommendation approach called BLM-NII, where the RLS classifier with GIP kernel was used as the local model [MKY⁺13]. Moreover, in [YKM⁺14], Yamanishi *et al.* introduced a web server called DINIES, which utilized supervised machine learning techniques, *e.g.*, pairwise kernel learning and distance metric learning, to suggest unknown DTIs from different sources of biological data.

Another line of research for DTI recommendation is the application of low rank matrix factorization technique, which maps both drugs and targets into a shared latent space with low dimensionality. The drug-target interactions are modeled using the linear combinations of the latent representations of drugs and targets. For example, Gönen proposed a kernelized Bayesian matrix factorization (KBMF) method, which combined the kernel-based dimensionality reduction, matrix factorization, and binary classification for DTI recommendation [Gön12]. Cobanoglu *et al.* utilized probabilistic matrix factorization (PMF) [MS07] for recommending unknown DTIs [CLH⁺13]. The accuracy of the PMF based approach was further improved by an active learning strategy. Moreover, Zheng *et al.* introduced the multiple similarities collaborative matrix factorization (MSCMF) model, which extended collaborative matrix factorization (CMF) to exploit multiple kinds of drug similarities and target similarities for DTI recommendation [ZDMZ13].

Chapter 3

Personalized Location Recommendation

3.1 Introduction

In recent years, we have witnessed the rapid growth and increasing popularity of location-based social networks (LBSNs), *e.g.*, BrightKite, Foursquare, and Gowalla, where users explore their surrounding places and share life experiences via “check-ins”. The available check-in data in LBSNs contain rich knowledge about users’ interests, and thus are beneficial to a wide range of applications such as location recommendation [ZCZ⁺10], friend recommendation [SNM11], and activity recommendation [ZZXY10]. As an important application of LBSNs, the personalized location recommender systems (PLRs) can help users explore new locations to enrich their experiences. On the other hand, PLRs can also facilitate third-party developers (*e.g.*, advertisers) to provide more relevant services at the right locations. Therefore, PLRs have drawn great research attention from both academia and industry in recent years [Zhe12,ZX11]. For example, Foursquare, a well-known LBSN that has over

55 million users and 7 billion check-ins (last updated June, 2015), has launched its location recommendation engine since March 2011.¹

The primary idea of PLRs is to predict a user's preferences on unvisited locations. Among existing PLRs developed for LBSNs, the approaches established on collaborative filtering technique are most widely used to model users' preferences, according to the historical interactions between users and locations [ZZXY10]. To improve location recommendation accuracy, auxiliary information such as social information and geographical information, has also been exploited. For instance, social opinions from users' online social friends [YYL10] or local experts in a new city [BZM12] have been considered for location recommendation. In contrast to social information, geographical information has been found to play a much more important role in the location recommendation task [CYKL12].

Prior work that employs geographical information for location recommendation mainly explores geographical characteristics of check-in data from user perspective. Based on the empirical observation that users tend to check-in at nearby locations, the geographical distances between users and locations are considered for location recommendation tasks, using an inverse proportional rule in [LFYX13]. To better describe users' preferences on locations, different methods have been proposed to model users' check-in behaviors in LBSNs. For example, in [YYLL11], a power-law probabilistic model is used to describe the power-law distribution of a user's check-ins. The approach in [CYKL12] utilizes a multi-center Gaussian model to study users' multi-center check-in behavior. To summarize, existing PLRs only exploit the geographical characteristics from user perspective, via modeling the geographical distribution (*e.g.*, the power-law distribution or multi-center distribution) specific to a particular user's check-ins. The geographical characteristics from location perspective have not been studied for location recommendation. This kind of characteristics are

¹<http://engineering.foursquare.com/2011/03/22/building-a-recommendation-engine-foursquare-style/>

independent from individual users and should have potential benefits to location recommendation.

This chapter introduces a novel location recommendation method, namely Instance-Region Neighborhood Matrix Factorization (IRenMF). This method exploits the geographical characteristics of check-in data from the location perspective, with an emphasis on the geographical neighborhood. Through the empirical analysis of user's check-in data, the nearest neighboring locations have been found to share more common visitors. This observation indicates that, in general, a user shares similar preferences on nearest neighboring locations. In IRenMF, this pattern is used as the **instance-level geographical neighborhood characteristics**. Specifically, by introducing a similarity measure, the relationship underlying a user's preferences on a few nearest neighboring locations is captured and utilized to characterize her preference on a target location. In addition, the empirical study on check-in data also shows that locations in a geographical region may share similar user preferences. Each geographical region is usually associated with a specific function (*e.g.*, business, entertainment, and education) [CSHS12, YZX12]. Such geographical regions contain additional prior knowledge about the interactions between users and locations. In IRenMF, the geographical region structure is used as the **region-level geographical neighborhood characteristics**. Specifically, a group lasso penalty is employed to integrate the geographical region structure within the check-in data into the learning of latent factors of users and locations.

3.2 Empirical Data Analysis

To gain a better understanding of users' check-in activities, an empirical analysis is conducted to study the check-in data collected from Gowalla, a popular LBSN (see Section 3.5.1 for details). More specifically, the geographical neighborhood characteristics are studied at two levels: instance-level and region-level. The following two observations are made from the data analysis: (a) nearest neighboring locations which

Figure 3.1: The instance-level neighborhood characteristics of check-in data.

tend to share more similar user preferences; and (b) locations in the same geographical region which may share similar user preferences. These patterns are independent of any particular user. The details of the data analysis are as follows.

Instance Level. At the instance level, the similarity of users' check-ins at two different locations ℓ_j and ℓ_k is measured using Jaccard similarity coefficient, denoted by $\beta(\ell_j, \ell_k)$. This measure is also known as the *common visitor ratio* of two locations in the discussion of this chapter, and its definition is as follows:

$$\beta(\ell_j, \ell_k) = \frac{|\mathcal{U}(\ell_j) \cap \mathcal{U}(\ell_k)|}{|\mathcal{U}(\ell_j) \cup \mathcal{U}(\ell_k)|}, \quad (3.1)$$

where $\mathcal{U}(\ell_j)$ and $\mathcal{U}(\ell_k)$ denote the sets of users who have visited ℓ_j and ℓ_k respectively.

Figure 3.1(a) plots the relationship between $\beta(\ell_j, \ell_k)$ and the geographical distance $d(\ell_j, \ell_k)$ between ℓ_j and ℓ_k . The common visitor ratio is averaged over all pairs of locations with same distances. In this chapter, the distance between two locations is computed by using the Haversine formula². As shown in Figure 3.1(a), β decreases with the increase of d , which indicates that geographically adjacent locations tend to share more common visitors. In other words, neighboring locations are more likely to be visited by the same set of users, or users have similar preferences on neighboring locations. As an example, Figure 3.1(b) shows the common visitor ratio

²https://en.wikipedia.org/wiki/Haversine_formula

Figure 3.2: The region-level neighborhood characteristics of check-in data.

between a randomly sampled location (id: 23092, latitude: 51.5144313985, longitude: -0.152746439) and the other locations. It reflects that location 23092 shares more common visitors with its geographical neighboring locations.

Region Level. To study the geographical characteristics of locations at region level, the locations are clustered into groups. The clustering can be conducted by using different similarity/distance measures between locations, *e.g.*, geographical distance, user check-in similarity (*i.e.*, $\beta(\ell_j, \ell_k)$), or the combination of the two.³ For illustration purposes, this chapter presents a clustering result by using user check-in similarity with geographical constraint, similar to that reported in [CSHS12]. A spectral clustering algorithm is utilized to group the locations [NJW02]. More specifically, an affinity matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$ is computed over all observed locations $\{\ell_j\}_{j=1}^n$. For a given location ℓ_j , let $\mathcal{N}(\ell_j)$ denote the set of K closest locations to ℓ_j by the geographical

³The three kinds of similarity/distance measures are evaluated in the experiments (see Section 3.5.4).

distance. In this analysis, K is set at 10. The element M_{jk} in $\mathbf{M}$ is determined as,

$$M_{jk} = \begin{cases} \beta(\ell_j, \ell_k) + \varepsilon, & \text{if } \ell_j \in \mathcal{N}(\ell_k) \text{ or } \ell_k \in \mathcal{N}(\ell_j) \\ 0, & \text{otherwise,} \end{cases} \quad (3.2)$$

where ε is a small constant used to keep each location connecting to arbitrary other locations either directly or indirectly. Then, the spectral algorithm is used to partition locations into groups, considering users' check-ins at each location and the geographical distances between locations.

Figure 3.2 shows an example of 10 clusters of the locations around Chicago area. The location clusters are denoted in different colors. Observe that the locations are distributed in geographical regions and the shapes of the regions are determined by user check-in behavior. In other words, locations in the same geographical region may share similar user preferences. Suggested in earlier studies, each of these geographical regions is usually associated with a specific function such as education, business, entertainment and so on, considering the category information of the locations in the region [CSHS12, YZX12]. This observation is consistent with the instance-level geographical characteristics. Nevertheless, a region refers to a much larger geographical area than the area containing a few nearest neighbors of a specific location.

Motivated by these two observations, the above geographical characteristics of check-in data are proposed to be incorporated for location recommendation, so as to improve location recommendation accuracy.

3.3 Personalized Location Recommendation by Matrix Factorization

The recommendation task studied in this chapter can be defined as: given the historical check-ins of m users $\{u_i\}_{i=1}^m$ over n locations $\{\ell_j\}_{j=1}^n$, for a target user, recommend her a set of locations that she might prefer but has not visited before. A binary rat-

ing matrix $\mathbf{R} \in \mathbb{R}^{m \times n}$ is used to describe users' preferences on locations, where each element $R_{ij} \in \{0, 1\}$ denotes u_i 's preference on ℓ_j . If u_i has checked-in at ℓ_j at least once, R_{ij} is set to 1; otherwise, R_{ij} is set to 0. Note that $R_{ij} = 0$ does not explicitly indicate u_i is not interested in ℓ_j . It can also be caused by that u_i does not know ℓ_j .

The matrix factorization based approaches are the most successful recommendation methods [KBV09]. The primary idea of matrix factorization is to map both users and locations into a shared space with dimension $r \ll \min(m, n)$, and represent u_i and ℓ_j using $\mathbf{U}_i \in \mathbb{R}^{1 \times r}$ and $\mathbf{L}_j \in \mathbb{R}^{1 \times r}$, which are known as the latent factors of u_i and ℓ_j respectively. The preference of u_i on ℓ_j can then be approximated using

$$\hat{R}_{ij} = \mathbf{U}_i \mathbf{L}_j^\top. \quad (3.3)$$

Biases for users and locations can also be incorporated in Eq. (3.3) to produce more accurate models [KBV09]. In general, the latent factors of all users and locations are denoted by $\mathbf{U} \in \mathbb{R}^{m \times r}$ and $\mathbf{L} \in \mathbb{R}^{n \times r}$ respectively, where $\mathbf{U}_i$ is the i^{th} row in $\mathbf{U}$ and $\mathbf{L}_j$ is the j^{th} row in $\mathbf{L}$. The model parameters, $\mathbf{U}$ and $\mathbf{L}$, can be learned by minimizing a weighted regularized square error loss [HKV08, PZC⁺08, KBV09],

$$\min_{\mathbf{U}, \mathbf{L}} \frac{1}{2} \sum_{i,j} W_{ij} (R_{ij} - \hat{R}_{ij})^2 + \frac{\lambda_1}{2} \|\mathbf{U}\|_F^2 + \frac{\lambda_2}{2} \|\mathbf{L}\|_F^2, \quad (3.4)$$

where W_{ij} gives a confidence level to the preference feedback R_{ij} , $\|\cdot\|_F$ is the Frobenius norm of a matrix, λ_1 and λ_2 are regularization parameters. Once the latent factors $\mathbf{U}$ and $\mathbf{L}$ have been learned, the preference value associated with any user-location pair (u_i, ℓ_j) can be predicted using Eq. (3.3).

3.4 Exploitation of Geographical Characteristics

This section presents the details of IrenMF, the objective of which is to exploit two levels (*i.e.*, instance-level and region-level) of neighborhood characteristics to provide

more accurate location recommendations. Firstly, at the instance level, a geographical weighting strategy is used to consider the strong relations among users' preferences on the target location and a few nearest neighbors of it. Secondly, at the region level, a group lasso penalty is introduced to take advantage of the geographical region structure derived from the check-in data.

3.4.1 Instance-Level Exploitation

Through mapping users and locations into a shared latent space, matrix factorization can effectively estimate the overall relations associated with most or all locations. However, the classical matrix factorization based approach introduced in Section 3.3 ignores the strong relations among geographical nearest neighboring locations.

The empirical analysis of check-in data shows that nearest neighboring locations tend to share more common visitors. Inspired by this observation, the preference of u_i on ℓ_j is proposed to be characterized by u_i 's preferences on a few nearest neighboring locations of ℓ_j . Thus, the prediction of R_{ij} is modified as,

$$\hat{R}_{ij}^{new} = \alpha \mathbf{U}_i \mathbf{L}_j^\top + (1 - \alpha) \frac{1}{Z(\ell_j)} \sum_{\ell_k \in \mathcal{N}(\ell_j)} Sim(\ell_j, \ell_k) \mathbf{U}_i \mathbf{L}_k^\top, \quad (3.5)$$

where $\alpha \in [0, 1]$ is the instance weighting parameter used to control the influence from neighboring locations, $\mathcal{N}(\ell_j)$ denotes the set of K nearest neighboring locations of ℓ_j . In the following experiments, K is empirically set at 10.⁴ In Eq. (3.5), $Sim(\ell_j, \ell_k)$ is the geographical weight denoting the geographical influence of ℓ_k on ℓ_j . $Z(\ell_j)$ is a normalizing factor formulated as $Z(\ell_j) = \sum_{\ell_k \in \mathcal{N}(\ell_j)} Sim(\ell_j, \ell_k)$. Here, $Sim(\ell_j, \ell_k)$ is defined using a Gaussian function as follows based on the observations in Figures 3.1(a) and 3.1(b).

$$Sim(\ell_j, \ell_k) = e^{-\frac{\|\mathbf{x}_j - \mathbf{x}_k\|^2}{\sigma^2}} \quad \forall \ell_k \in \mathcal{N}(\ell_j), \quad (3.6)$$

⁴The performances with respect to different setting of K have also been evaluated. The experimental results indicate that the location recommendation accuracy is not very sensitive to the size of nearest neighbors.

where $\mathbf{x}_j$ and $\mathbf{x}_k$ represent the geographical coordinates (latitude and longitude) of ℓ_j and ℓ_k respectively, σ is a constant which is empirically set at 0.1 in the experiments.

Note that this instance-level exploitation is essentially different from the approach proposed in [YYLL11]. The approach proposed in this chapter is from the location perspective and considers the relationship between a target location ℓ_j and a few nearest geographical neighbors $\mathcal{N}(\ell_j)$ of ℓ_j . The prediction of u_i 's preference on ℓ_j is determined by the characteristics of u_i , ℓ_j , and $\mathcal{N}(\ell_j)$. However, in [YYLL11], the probability that u_i would check-in at ℓ_j is determined by the distances between ℓ_j and u_i 's **visited locations** $\mathcal{L}(u_i)$ (*i.e.*, from user perspective). Usually, $\mathcal{N}(\ell_j)$ and $\mathcal{L}(u_i)$ are entirely different or with a very small overlap, as u_i has only visited a small proportion of locations. Therefore, **these two methods are modeling two different kinds of geographical characteristics**. In all experiments, the proposed instance-level exploitation outperforms the approach in [YYLL11] (see Section 3.5.2 for details).

3.4.2 Region-Level Exploitation

As aforementioned, locations in the same geographical region may share similar user preferences. This section focuses on incorporating this region-level characteristics into the learning of latent factors (or latent features) of users and locations.

In the feature learning problems, the structured sparsity learning approaches (*e.g.*, group lasso) have been widely used to consider the prior knowledge of the group structure of inputs. Inspired by its successful applications in real scenarios [KX10, HZM11, JOB10, KSX09], the proposed method utilizes group lasso to exploit the geographical region structure of locations, via assuming the latent factors of locations from the same region share the same sparsity pattern, namely the structured sparsity assumption of location latent factors. In this assumption, the locations are assumed to be clustered into G disjoint groups: $\{\mathcal{L}_{(g)}\}_{g=1}^G$. The location latent factors $\mathbf{L}$ are divided into G groups as $\mathbf{L} = \{\mathbf{L}_{(g)}\}_{g=1}^G$, where $\mathbf{L}_{(g)} \in \mathbb{R}^{n_g \times r}$ denotes the latent factors

Figure 3.3: An example of the structured sparsity assumption of the location latent factors.

of locations belonging to $\mathcal{L}_{(g)}$, n_g is the number of locations in $\mathcal{L}_{(g)}$. Accordingly, the rating matrix $\mathbf{R}$ is divided into G groups as $\mathbf{R} = \{\mathbf{R}_{(g)}\}_{g=1}^G$, where $\mathbf{R}_{(g)} \in \mathbb{R}^{m \times n_g}$.

Figure 3.3 gives an example of the structured sparsity assumption. The locations are clustered into 4 groups, and both users and locations are mapped into a shared latent space with dimension 4. In $\mathbf{L}^\top$, the red and white cells denote nonzero and zeros values respectively. As shown in Figure 3.3, the column vectors in $\mathbf{L}_{(2)}^\top$ share the same sparsity pattern since all their third rows are zeros. Therefore, the reconstruction of elements in $\mathbf{R}_{(2)}$ only relies on the first, second and fourth row components in $\mathbf{L}_{(2)}^\top$. With this sparsity structure of $\mathbf{L}$, the associations of latent factors of users and locations can be understood at the group level instead of instance level, as the geographical structure information of locations is embedded in the location latent factors. Taking into account the structured sparsity assumption, a group lasso penalty of $\mathbf{L}$ is used as,

$$\lambda_3 \sum_{g=1}^G \sum_{d=1}^r \omega_g \|\mathbf{L}_{(g)}^d\|_2, \quad (3.7)$$

to integrate the geographical region structure information into the learning of latent factors $\mathbf{U}$ and $\mathbf{L}$. In Eq. (3.7), λ_3 is the group sparsity regularization parameter. $\mathbf{L}_{(g)}^d \in \mathbb{R}^{n_g \times 1}$ is the d^{th} column vector in $\mathbf{L}_{(g)}$. ω_g is a weight assigned to $\mathbf{L}_{(g)}^d$. A simple setting of ω_g is $\omega_g = n_g^{-\frac{1}{2}}$, which enables the amount of penalization to be adjusted according to the size of each group [YL06].

3.4.3 Instance-Region Neighborhood Matrix Factorization

This section presents the final formulation of IRenMF. By plugging Eq. (3.5) and Eq. (3.7) into Eq. (3.4), the optimization problem of IRenMF is formulated as follows,

$$\begin{aligned} \min_{\mathbf{U}, \mathbf{L}} \mathcal{F}(\mathbf{U}, \mathbf{L}) = & \frac{1}{2} \sum_{i,j} W_{ij} (R_{ij} - \hat{R}_{ij}^{new})^2 + \frac{\lambda_1}{2} \|\mathbf{U}\|_F^2 + \frac{\lambda_2}{2} \|\mathbf{L}\|_F^2 \\ & + \lambda_3 \sum_{g=1}^G \sum_{d=1}^r \omega_g \|\mathbf{L}_{(g)}^d\|_2. \end{aligned} \quad (3.8)$$

For simplicity, the problem in Eq. (3.8) is rewritten as,

$$\begin{aligned} \min_{\mathbf{U}, \mathbf{L}} \mathcal{F}(\mathbf{U}, \mathbf{L}) = & \frac{1}{2} \|\widetilde{\mathbf{W}} \circ (\mathbf{R} - \mathbf{U}\mathbf{L}^\top \mathbf{S}_\alpha)\|_F^2 + \frac{\lambda_1}{2} \|\mathbf{U}\|_F^2 + \frac{\lambda_2}{2} \|\mathbf{L}\|_F^2 \\ & + \lambda_3 \sum_{g=1}^G \sum_{d=1}^r \omega_g \|\mathbf{L}_{(g)}^d\|_2, \end{aligned} \quad (3.9)$$

where $\widetilde{\mathbf{W}} \in \mathbb{R}^{m \times n}$ is a weighting matrix. $\circ$ denotes the Hadamard product of two matrices. $\mathbf{S}_\alpha = \alpha \mathbf{I} + (1 - \alpha) \mathbf{S}^\top$. $\mathbf{I} \in \mathbb{R}^{n \times n}$ is an identity matrix. $\mathbf{S} \in \mathbb{R}^{n \times n}$, in which each element $S_{jk} = \text{Sim}(\ell_j, \ell_k) / Z(\ell_j)$.

In general, as u_i 's check-in times C_{ij} at ℓ_j grows, we have a strong indication that u_i indeed likes ℓ_j . Therefore, a simple setting of the element $\widetilde{W}_{ij}$ in $\widetilde{\mathbf{W}}$ is as,

$$\widetilde{W}_{ij} = (1 + \delta C_{ij})^{\frac{1}{2}} \quad (3.10)$$

where the constant δ is used to control the rate of increase. Empirically, δ is set at 10 in the experiments.

The optimization problem stated in Eq. (3.9) is a non-convex problem, and can be further decomposed into two convex optimization problems by fixing one variable ($\mathbf{U}$ or $\mathbf{L}$) and optimizing the other one. Therefore, we use the following two-step alternating optimization strategy to solve the problem in Eq. (3.9).

Algorithm 1: APG Optimization Algorithm

Input : $\mathbf{L}^{(0)}, \lambda_2, \lambda_3, \tau^{(0)}, \varepsilon, \eta, t^{(0)}, \{\omega_g\}_{g=1}^G$
Output: $\mathbf{L}^{(k)}$

- 1 Initialize: $\mathbf{L}^{(1)} \leftarrow \mathbf{L}^{(0)}, k \leftarrow 1, \sigma \leftarrow 1.0$
- 2 Use 3.12 and $\mathbf{L}^{(0)}$ to calculate $\varphi(\mathbf{L}^{(0)})$
- 3 **while** $k \leq \text{maxIter}$ && $\sigma \geq \varepsilon$ **do**
 - // maxIter is the maximum number of iterations
 - 4 $\tau \leftarrow \eta\tau^{(k-1)}, t^{(k)} \leftarrow \frac{1+\sqrt{1+4(t^{(k-1)})^2}}{2}, \mathbf{X}^{(k)} \leftarrow \mathbf{L}^{(k)} + \frac{t^{(k-1)}-1}{t^{(k)}}(\mathbf{L}^{(k)} - \mathbf{L}^{(k-1)})$
 - 5 **while** 1 **do**
 - 6 $\mathbf{Z} \leftarrow \mathbf{X}^{(k)} - \frac{1}{\tau}\nabla p(\mathbf{X}^{(k)})$
 - 7 Use Eq. (3.16) to calculate $\mathbf{L}^*$
 - 8 Use Eq. (3.13) and Eq. (3.14) to compute $p(\mathbf{L}^*)$ and $p_\tau(\mathbf{L}^*, \mathbf{X}^{(k)})$
 - 9 **if** $p(\mathbf{L}^*) \leq p_\tau(\mathbf{L}^*, \mathbf{X}^{(k)})$ **then**
 - 10 $\tau^{(k)} \leftarrow \tau$, break
 - 11 **end**
 - 12 **else**
 - 13 $\tau \leftarrow \frac{\tau}{\eta}$
 - 14 **end**
 - 15 **end**
 - 16 $\mathbf{L}^{(k)} \leftarrow \mathbf{L}^*$
 - 17 Use Eq. (3.12) to update $\varphi(\mathbf{L}^{(k-1)})$ and $\varphi(\mathbf{L}^{(k)})$
 - 18 $\sigma \leftarrow \frac{|\varphi(\mathbf{L}^{(k)}) - \varphi(\mathbf{L}^{(k-1)})|}{|\varphi(\mathbf{L}^{(k-1)})|}$
 - 19 $k \leftarrow k + 1$
 - 20 **end**
 - 21 Return $\mathbf{L}^{(k)}$;

Optimize U: when $\mathbf{L}$ is fixed, the optimization problem with respect to $\mathbf{U}$ is as follows,

$$\min_{\mathbf{U}} \phi(\mathbf{U}) = \frac{1}{2} \|\widetilde{\mathbf{W}} \circ (\mathbf{R} - \mathbf{U}\mathbf{L}^\top \mathbf{S}_\alpha)\|_F^2 + \frac{\lambda_1}{2} \|\mathbf{U}\|_F^2. \quad (3.11)$$

The alternating least squares (ALS) algorithm proposed in [HKV08, PZC⁺08] is directly used to solve the problem in Eq. (3.11).

Optimize L: when $\mathbf{U}$ is fixed, the optimization problem becomes,

$$\min_{\mathbf{L}} \varphi(\mathbf{L}) = p(\mathbf{L}) + \frac{\lambda_2}{2} \|\mathbf{L}\|_F^2 + \lambda_3 \sum_{g=1}^G \sum_{d=1}^r \omega_g \|\mathbf{L}_{(g)}^d\|_2, \quad (3.12)$$

where

$$p(\mathbf{L}) = \frac{1}{2} \|\widetilde{\mathbf{W}} \circ (\mathbf{R} - \mathbf{U}\mathbf{L}^\top \mathbf{S}_\alpha)\|_F^2. \quad (3.13)$$

The problem in Eq. (3.12) can be solved using the accelerated proximal gradient (APG) method [TY10]. The key point of APG approaches is to construct the Taylor approximation of $p(\mathbf{L})$ at the point $\mathbf{X}$ as,

$$p_\tau(\mathbf{L}, \mathbf{X}) = \frac{\tau}{2} \|\mathbf{L} - \mathbf{Z}\|_F^2 - \frac{1}{2\tau} \|\nabla p(\mathbf{X})\|_F^2 + p(\mathbf{X}) \quad (3.14)$$

where $\tau > 0$ and $\mathbf{Z} = \mathbf{X} - \frac{1}{\tau} \nabla p(\mathbf{X})$. For a fixed point $\mathbf{X}$, we need to solve the following optimization problem,

$$\min_{\mathbf{L}} \psi(\mathbf{L}) = \frac{\tau}{2} \|\mathbf{L} - \mathbf{Z}\|_F^2 + \frac{\lambda_2}{2} \|\mathbf{L}\|_F^2 + \lambda_3 \sum_{g=1}^G \sum_{d=1}^r \omega_g \|\mathbf{L}_{(g)}^d\|_2. \quad (3.15)$$

The solution to the problem in Eq. (3.15) is given in Theorem 1. The proof for Theorem 1 can be found in the Appendix A.1. The details of the APG solution to Eq. (3.12) are presented in Algorithm 1.

Theorem 1. *Let $\mathbf{L}^*$ be the optimal solution to problem (3.15). $\mathbf{L}^*$ is unique and can be calculated via a soft-threshold operator defined as*

$$[\mathbf{L}^*]_{(g)}^d = \begin{cases} \frac{\mathbf{Z}_{(g)}^d (\|\mathbf{Z}_{(g)}^d\|_2 - \frac{\lambda_3 \omega_g}{\tau})}{(1 + \frac{\lambda_2}{\tau}) \|\mathbf{Z}_{(g)}^d\|_2}, & \text{if } \|\mathbf{Z}_{(g)}^d\|_2 > \frac{\lambda_3 \omega_g}{\tau} \\ \mathbf{0}, & \text{otherwise,} \end{cases} \quad (3.16)$$

where $[\mathbf{L}^*]_{(g)}^d \in \mathbb{R}^{n_g \times 1}$ and $\mathbf{Z}_{(g)}^d \in \mathbb{R}^{n_g \times 1}$ are the sub-vectors at the d^{th} dimension of the g^{th} group of $\mathbf{L}^*$ and $\mathbf{Z}$ respectively.

The entire optimization procedure for the problem in Eq. (3.9) is summarized in Algorithm 2. Although the objective function of IRenMF (like the objective function of other “standard” recommender systems) is certainly non-convex [KBV09] and subject to local minima, in the experiments it yields stable recommendation accuracy when restarting with different initial conditions.

Algorithm 2: IRenMF Optimization Algorithm

Input : $\mathbf{R}, \widetilde{\mathbf{W}}, r, \varepsilon, \{\lambda_i\}_{i=1}^3, \{\omega_g\}_{g=1}^G$
Output: $\mathbf{U}^{(k)}, \mathbf{L}^{(k)}$

- 1 Initialize $\mathbf{U}^{(0)}$ and $\mathbf{L}^{(0)}$ with random elements, $k \leftarrow 1, \sigma \leftarrow 1.0$
- 2 **while** $k \leq \text{maxIter}$ && $\sigma \geq \varepsilon$ **do**
 - 3 // maxIter is the maximum number of iterations
 - 3 Compute $\mathbf{U}^{(k)}$ by solving the problem in Eq. (3.11)
 - 4 Compute $\mathbf{L}^{(k)}$ by solving the problem in Eq. (3.12)
 - 5 $\sigma \leftarrow \frac{|\mathcal{F}(\mathbf{U}^{(k)}, \mathbf{L}^{(k)}) - \mathcal{F}(\mathbf{U}^{(k-1)}, \mathbf{L}^{(k-1)})|}{|\mathcal{F}(\mathbf{U}^{(k-1)}, \mathbf{L}^{(k-1)})|}$
 - 6 $k \leftarrow k + 1$
- 7 **end**
- 8 Return $\mathbf{U}^{(k)}$ and $\mathbf{L}^{(k)}$;

3.5 Experiments

This section first introduces the experimental settings and then presents the analysis of experimental results. The experiments focus on the following aspects: (a) the comparison of the performances of IRenMF and baseline recommendation algorithms, (b) the sensitivity of IRenMF to different parameter settings, and (c) the impact of different location clustering strategies.

3.5.1 Experimental Settings

Dataset Description. The experimental data used in this study was collected from Gowalla, a popular LBSN, which has more than 600,000 users since November 2010 and was acquired by Facebook in December 2011.⁵ In practice, the Gowalla APIs were used to collect user profiles and check-in data made before June 1, 2011. Finally, 36,001,959 check-ins made by 319,063 users over 2,844,076 locations have been obtained. Each check-in contains user id, location id, longitude, latitude, timestamp, etc. Four datasets were constructed to evaluate the performances of IRenMF, via extracting the check-in data generated in four popular cities: Berlin, London, Chicago,

⁵<http://techcrunch.com/2011/12/02/report-facebook-has-acquired-gowalla/>

Table 3.1: The statistics of users' check-in data in four popular cities.

	Berlin	Chicago	London	San Francisco
#users	5,510	13,852	17,112	21,591
#locations	15,528	38,505	63,466	66,142
#check-ins	238,972	490,998	942,877	1,545,407
average #users per location	23.98	20.28	32.11	41.35
average #locations per user	8.51	7.30	8.66	13.50

and San Francisco. The detailed statistics of the check-in data in the four datasets are summarized in Table 3.1.

Data Partition. In the experiments, each of the four datasets is partitioned into three non-overlapping parts. More specifically, within one dataset, if a user u_i has visited n_i different locations $\mathcal{L}(u_i) = \{\ell_i^1, \ell_i^2, \dots, \ell_i^{n_i}\}$, 60% of her visited locations $\mathcal{L}(u_i)$ are randomly chosen as training data, and another 10% of $\mathcal{L}(u_i)$ are chosen as development data to tune parameters. The remaining 30% of $\mathcal{L}(u_i)$ are used as testing data to evaluate the effectiveness of recommendation methods. Similar experimental settings have been used to validate the performances of location recommendation methods in previous work [CYKL12, YYLL11, YCM⁺13]. After splitting the check-in data, the densities of the training data of Berlin, Chicago, London and San Francisco datasets are 1.17×10^{-3} , 4.13×10^{-4} , 3.84×10^{-4} , and 4.68×10^{-4} respectively. The density of the training data is defined as follows:

$$density = \frac{N_i^{Tr}}{N_u^{Tr} \times N_\ell^{Tr}}, \quad (3.17)$$

where N_u^{Tr} denotes the number of users in the training data, N_ℓ^{Tr} denotes the number of locations in the training data, N_i^{Tr} denotes the number of observed user-location interaction pairs in the training data.

Evaluation Metrics. The quality of location recommendations is assessed using two metrics (precision and recall), as suggested by [CYKL12, YYLL11]. The precision and recall of the top- N location recommendations to a target user are denoted by $P@N$ and $R@N$ respectively. $P@N$ measures the ratio of recovered locations to the N

recommended locations, and $R@N$ measures the ratio of recovered locations to the set of locations in the testing data. Given an individual user u_i , $\mathcal{L}^T(u_i)$ denotes the set of corresponding visited locations in the testing data, and $\mathcal{L}^R(u_i)$ denotes the set of recommended locations by a method. The definitions of $P@N$ and $R@N$ are:

$$\begin{aligned} P@N &= \frac{1}{|\mathcal{T}|} \sum_{u_i \in \mathcal{T}} \frac{|\mathcal{L}^T(u_i) \cap \mathcal{L}^R(u_i)|}{K}, \\ R@N &= \frac{1}{|\mathcal{T}|} \sum_{u_i \in \mathcal{T}} \frac{|\mathcal{L}^T(u_i) \cap \mathcal{L}^R(u_i)|}{|\mathcal{L}^T(u_i)|}, \end{aligned} \quad (3.18)$$

where $\mathcal{T}$ denotes the set of users in the testing data. Specifically, $P@5$, $P@10$, $R@5$, and $R@10$ are used as evaluation metrics in the experiments. Note that all the four datasets in this study have low density, which usually leads to relatively low precision and recall. Thus, the low values of precision and recall shown in the experiments are reasonable and in similar range of values reported in other work [CYKL12, YYLL11, YCM⁺13]. This section presents the relative improvements the proposed approach achieved, instead of the absolute values.

Evaluated Recommendation Algorithms. The performances of the following personalized location recommendation methods are evaluated in this section:

UserCF: This is the user-based CF approach used in [YYLL11] that predicts a user's preferences, considering the preferences of other similar users. The user similarity is computed using cosine similarity and $k = 150$.

ItemCF: This is the item-based CF approach [SKKR01] that predicts a user's preference on a target location, considering her preferences on similar locations. The location similarity is computed using cosine similarity and $k = 150$.

WRMF: This is the weighted regularized matrix factorization model designed for processing large scale implicit feedback datasets [HKV08, PZC⁺08]. It is a special case of IRenMF without considering the geographical characteristics of check-in data (via setting $\alpha = 1$ and $\lambda_3 = 0$ in Eq. (3.9)).

BPRMF: The Bayesian personalized ranking (BPR) is another state-of-the-art CF framework for implicit feedback data [RFGST09]. BPRMF indicates the choice of using matrix factorization as the learning model in the BPR optimization criterion. In this method, the visited locations are assumed to be more interesting than unvisited locations to an individual user.

GeoCF: This is the user preference/geographical influence based recommendation model proposed in [YYLL11]. In this method, a power-law distribution model is used to capture the geographical influence from a user’s visited locations. A unified location recommendation framework is used to linearly combine the geographical influence and the user preference derived by a user-based CF approach.

MGMF: This approach [CYKL12] is based on the observation that a user tends to check-in around several centers. A Multi-center Gaussian Model (MGM) is employed to model the probability that a user will check-in at a given location. Moreover, a fusion framework is used to fuse the user preference (derived by a MF model) and the MGM check-in probability for location recommendation.

InMF: It is a special case of IRenMF that only exploits the instance-level geographical neighborhood characteristics for location recommendation (via setting $\lambda_3 = 0$ in Eq. (3.9)).

RenMF: It is a special case of IRenMF that only exploits the region-level geographical neighborhood characteristics for location recommendation (via setting $\alpha = 1$ in Eq. (3.9)).

IRenMF: This is the proposed approach that exploits both instance- and region-level geographical neighborhood characteristics for location recommendation.

Parameter Settings. In the matrix factorization based approaches, the dimension of latent space r is set at 200, considering both efficiency and effectiveness. The

Figure 3.4: Performances of methods without geographical characteristics.

regularization parameters λ_1 and λ_2 of IrenMF, are chosen by cross-validation and finally set at 0.015. The group sparsity regularization parameter λ_3 is set at 1 on all datasets. The instance weighting parameter α of IrenMF is set at 0.4, 0.4, 0.6, 0.4 on Berlin, Chicago, London, and San Francisco datasets respectively. The impact of different settings of α and λ_3 is discussed in Section 3.5.3. As a preprocessing step, the k -means algorithm is used to cluster locations into 50 groups according to their latitudes and longitudes. The impact of the region structures discovered by different clustering strategies is discussed in Section 3.5.4. In the experiments, the matrix factorization based approaches have been ran five times, and the average results on the top-5 and top-10 recommendations are reported.

3.5.2 Performance Comparison

This section reports two sets of results for methods with and without utilizing geographical characteristics separately.

Methods without Geographical Characteristics. The precision and recall of the four state-of-the-art top- N recommendation algorithms (*i.e.*, UserCF, ItemCF, WRMF, and BPRMF) are reported in Figure 3.4. On all datasets, UserCF outperforms WRMF and BPRMF in terms of precision metrics. Particular, UserCF attains 12.37% and 132.52% better performance than WRMF and BPRMF over all datasets, in terms of P@5. A possible reason is that the classical matrix factorization based CF approaches only assume a global structure of the check-in data, but ignore the local structure embedded in the check-in data, *e.g.*, the strong relations among nearest geographical neighboring locations. On the other hand, WRMF achieves highest recall values in terms of R@5 and R@10, on the Chicago, London and San Francisco datasets. This indicates that WRMF is an appropriate choice to incorporate geographical characteristics for location recommendation. In addition, as shown in Figure 3.4, UserCF significantly outperforms ItemCF. One possible explanation is that the user similarity is more accurate than location similarity, as many locations only have a few user check-ins. In this sense, the results shown in this section are consistent with the results reported in [YYLL11].

Methods with Geographical Characteristics. The precision and recall of the proposed methods utilizing geographical characteristics from the location perspective and other two state-of-the-art methods utilizing geographical characteristics from the user perspective [CYKL12, YYLL11] are reported in Figure 3.5.

Observing from Figure 3.5, InMF, RenMF, and IRenMF achieve better accuracy than WRMF in terms of both precision and recall, on all datasets. In terms of P@5, InMF outperforms WRMF by 43.18%, 37.96%, 22.90%, and 19.08% on the four datasets, respectively. RenMF outperforms WRMF by 21.76%, 19.39%, 17.71%, and 9.94% on the four datasets. This indicates that both instance-level and region-level neighborhood characteristics benefit the location recommendation for higher accuracy. In addition, the observation that InMF has better accuracy than RenMF shows that the instance-level characteristics are more effective than the region-level characteristics. This result suggests that nearest neighbouring locations tend to share

Figure 3.5: Performances of methods with geographical characteristics.

more user preference. By incorporating both kinds of neighborhood characteristics into the learning of latent factors, the recommendation accuracy of the classical matrix factorization model (*i.e.*, WRMF) can be significantly improved. For example, in terms of P@5, the improvements of IRenMF over WRMF are 56.48%, 43.36%, 29.35%, and 24.67% respectively, on all datasets.

To predict a user's preference on a target location, GeoCF considers the geographical influence from her visited locations (*i.e.*, the geographical characteristics from user's point of view). In contrast, InMF exploits geographical influence of the nearest neighbouring locations to the target location (*i.e.*, the instance-level geographical characteristics from location's point of view). Compared to GeoCF, InMF always achieves better results in terms of all measures. For example, in terms of P@10, InMF outperforms GeoCF by 11.78%, on average. This result indicates that the instance-level geographical neighborhood characteristics derived from location perspective have more impacts on the location recommendation accuracy than the geographical characteristics derived from user perspective. Taking advantage of both

levels of neighborhood characteristics, IRenMF outperforms GeoCF by 17.90%, in terms of P@10, on average.

Compared to MGMMF, the proposed InMF, RenMF and IRenMF perform significantly better on all datasets. The average improvements of InMF, RenMF and IRenMF over MGMMF, in terms of P@5, are 115.47%, 91.38%, and 127.38%, respectively. Moreover, MGMMF is also observed performing poorer than WRMF. One possible reason is that MGMMF is based on the probabilistic factor model, which only models users' check-in times at locations, instead of users' preferences.

In Figure 3.5, IRenMF always achieves the best results in terms of all the evaluation metrics, on all datasets. This demonstrates the advantage of combining both levels of geographical neighborhood characteristics for location recommendation. Most importantly, on average, the proposed IRenMF consistently outperforms the competitors WRMF, GeoCF, and MGMMF, in terms of P@10, by 38.63%, 17.90%, and 116.49% respectively. In all experiments, the standard deviations of the recommendation accuracy of IRenMF are below 0.005, showing the good stability of the proposed method.

3.5.3 Parameter Sensitivity Analysis

In IRenMF, the impact of geographical neighborhood characteristics on location recommendation are controlled by the instance weighting parameter α and group sparsity regularization parameter λ_3 . Figure 3.6(a) shows P@5 and R@5 of IRenMF (setting λ_3 at 0) with respect to different α ranging from 0 to 1 with an increment of 0.1. As shown in Figure 3.6(a), IRenMF achieves comparable results on Berlin, Chicago and San Francisco datasets with setting α at 0 and 1. This demonstrates that it is possible to characterize users' preferences on a target location, using their preferences on a few nearest neighboring locations of it. Specifically, in terms of P@5, it is observed that α set at 0.4 on Berlin dataset obtains the best results, as well as 0.4 on Chicago dataset, 0.6 on London dataset, and 0.4 on San Francisco dataset respectively. It suggests that α well tradeoff the importance between a user's preference on the target location and

Figure 3.6: Impact of geographical neighborhood characteristics in IrenMF.

her preferences on nearest neighboring locations. In particular, it is found that only considering one of them (*i.e.*, $\alpha = 0$ or $\alpha = 1$) will lead to degraded accuracy.

The performances of IrenMF have been studied in the grid $\lambda_3 \in \{0.001, 0.01, 0.1, 1, 10, 100\}$, while setting α at 1. The performances of IrenMF are shown in Figure 3.6(b), which indicates the recommendation accuracy can be gradually influenced by the setting of the group sparsity regularization parameter λ_3 , and $\lambda_3 = 1$ is the most suitable setting in all four datasets. As shown in Figure 3.6(b), the performance of IrenMF drops drastically by changing λ_3 from 10 to 100. This is because that the learned latent factors of locations are too sparse, and thus the reconstruction of users' preferences on locations are insufficient to provide accurate location recommendations. For example, in Berlin dataset, the sparsity of the location latent factors is 90.70% and 98.89% when setting λ_3 at 10 and 100 respectively.

Figure 3.7: The location recommendation accuracy with respect to different clustering strategies.

3.5.4 Impact of Different Clustering Strategies

This section explores the impact of different clustering strategies on the location recommendation accuracy. To single out the impact of region-level neighborhood characteristics, the instance weighting parameter α in IRenMF is set at 1, effectively only evaluating the performance of RenMF.

The performances of RenMF with respect to different region structures have been studied. Here, the region structures are obtained by clustering locations into 50 groups with the following clustering methods: (1) **KG**, the k -means clustering solely based on the geographical distance between locations; (2) **KH**, the k -means clustering solely based on users' check-ins generated at each location; (3) **SP**, the spectral clustering [NJW02], which considers both users' check-ins at locations and the geographical distance between locations. Figure 3.7 presents the recommendation accuracy of RenMF with respect to different region structures obtained by KG, KH and SP. It is observed that RenMF-KG, RenMF-KH, and RenMF-SP usually outperform WRMF in terms of P@5 and R@5. The average improvements of RenMF-KG, RenMF-KH, and RenMF-SP over WRMF are 17.2%, 4.73%, and 16.74% respectively, in terms of P@5. This demonstrates the importance of region-level neighborhood characteristics to location recommendation. Moreover, RenMF-KG and RenMF-SP achieve better performance than RenMF-KH on all datasets. This indicates that the region structure obtained from the geographical information (longitude and latitude) contributes

more to location recommendation accuracy than the interactions between users and locations.

This again highlights the importance of geographical neighborhood characteristics at region level.

3.6 Summary

This chapter empirically analyzes the geographical characteristics from the location perspective using historical check-in data collected from Gowalla. Two levels (*i.e.*, instance-level and region-level) of geographical neighborhood characteristics have been studied. Moreover, a novel location recommendation framework, *i.e.*, Instance-Region Neighborhood Matrix Factorization (IRenMF), is proposed to incorporate aforementioned geographical neighborhood characteristics for improving location recommendation accuracy. In particular, to solve the optimization problem in IRenMF, an alternating optimization strategy is proposed, in which an accelerated proximal gradient (APG) method is used to learn the latent factors of locations, considering the geographical region structure of the check-in data. Furthermore, extensive experiments on the data collected from Gowalla are conducted to evaluate IRenMF. Experimental results show that: (a) Compared to the baseline matrix factorization model (*i.e.*, WRMF) without utilizing geographical characteristics, the proposed methods that use either instance-level or region-level geographical neighborhood characteristics significantly improves recommendation accuracy; the improvement is on average by 30.78% and 17.2% respectively, in terms of precision metric P@5; (b) Utilizing both two levels of geographical neighborhood characteristics, IRenMF substantially outperforms two state-of-the-art location recommendation methods, which consider geographical characteristics derived from the user perspective.

Chapter 4

Top- N Recommendation with Users' Implicit Feedback

4.1 Introduction

For item recommendation with implicit feedback, users' preferences on items are unobservable. For example, in the music recommendation task¹, the number of times a user plays a song is observable (*i.e.*, implicit feedback) but not her preference rating to the song (*i.e.*, explicit feedback). Recommendation models are then built based on some pre-defined preference assumptions, *e.g.*, the more number of times a user plays a song, the more the user likes the song. Such assumptions may not accurately describe users' preferences. Moreover, the observed user-item interaction data is generally very sparse, which makes the preference modeling even more challenging. As a result, existing solutions often deliver unsatisfactory recommendation accuracies.

Instead of directly addressing these challenges in item recommendation with implicit feedback, this chapter considers an alternative approach, which exploits a boosting technique to improve the recommendation accuracy by combining the recommendation power of multiple “weak recommenders”. The boosting technique is originally proposed to improve the accuracy of classification models by combining multiple

¹<http://labrosa.ee.columbia.edu/millionsong/challenge>

weak classifiers [FS95]. It is then extended to solve ranking tasks. For example, AdaRank is a boosting framework that optimizes ranking metrics such as the Area Under the ROC Curve (AUC), the Mean Average Precision (MAP), and the Normalized Discounted Cumulative Gain (NDCG) [XL07]. Recent studies show that accuracies of previous recommendation methods can be improved by utilizing boosting techniques [JNG⁺13, CXZ⁺14, WSZ14]. However, all existing solutions are based on the recommendation models designed for rating prediction tasks, which are not optimal for item recommendation with implicit feedback.

This chapter introduces a novel boosting algorithm, namely AdaPtive Boosting Personalized Ranking (AdaBPR), for item recommendation with users' implicit feedback. In the proposed AdaBPR framework, multiple homogeneous component recommenders are linearly combined to achieve more accurate recommendation. The component recommenders are learned based on a re-weighting strategy that assigns a dynamic weight to each observed user-item interaction.

4.2 The Boosting Recommendation Framework

Given the historical interactions between m users and n items, the task of item recommendation is to recommend a target user u a list of items that she may prefer but has not interacted with before [DK04]. In many real-world scenarios, the item recommendation tasks are based on users' implicit preference feedback, *e.g.*, the number of times that a user has interacted with an item. Based on this observation, for simplicity, we introduce a set of binary variables $y_{ui} \in \{0, 1\}$ to represent users' implicit feedback². If a user u has interacted with an item i , y_{ui} is set to 1, and 0 otherwise. Note that $y_{ui} = 0$ does not explicitly indicate that u is not interested in i . It may be the result that user u does not even know the existence of i . In this chapter, the set

²This is the most widely used representation of users' implicit feedback data for item recommendation tasks. Indeed, we can also use y_{ui} to denote the number of times that a user u has interacted with an item i . The numeric variables can be handled with slight modifications to the proposed model.

of users and items are denoted by U and V , respectively. For a user u , the set of her historical items is denoted by $V_u^+ = \{i | y_{ui} = 1, i \in V\}$, and the set of items that has not interacted with u is denoted by $V_u^- = V \setminus V_u^+$. Moreover, the set of all observed user-item interactions is denoted by $\mathcal{D} = \{(u, i) | u \in U, i \in V_u^+\}$.

For item recommendation tasks, the accuracy of a recommendation model is usually evaluated using the ranking metrics, *e.g.*, AUC, MAP, NDCG, Precision, and Recall. The definitions of AUC and MAP are given below.

$$\text{AUC} = \frac{1}{|U|} \sum_{u \in U} \frac{1}{|V_u^+|} \sum_{i \in V_u^+} \frac{1}{|V_u^-|} \sum_{j \in V_u^-} \mathbb{I}(\pi_{ui} < \pi_{uj}) \quad (4.1)$$

$$\text{MAP} = \frac{1}{|U|} \sum_{u \in U} \frac{\sum_{i \in V_u^+} \frac{\sum_{j \in V} \mathbb{I}(y_{uj} > 0) \mathbb{I}(\pi_{uj} \leq \pi_{ui})}{\pi_{ui}}}{|V_u^+|} \quad (4.2)$$

where $|\mathcal{X}|$ denotes the cardinality of the set $\mathcal{X}$, π_{ui} is the rank position of i in the ranked item list that is created by sorting items according to u 's preferences on them in descending order, and $\mathbb{I}(\cdot) = 1$ if the condition is true, and 0 otherwise.

4.2.1 Adaptive Boosting Personalized Ranking

The proposed AdaBPR framework aims to optimize the loss function defined based on a personalized ranking metric.

Observe from Eq. (4.1) and Eq. (4.2) that the accuracy of a recommendation model is determined by the rank positions of the historical items V_u^+ of each user u . Thus, a general function $E[\pi(u, i, f)]$ is used to denote the ranking accuracy associated with each observed interaction pair (u, i) where $i \in V_u^+$. The argument of the general function $\pi(u, i, f)$ is the rank position of item i in the ranked item list of u , resulted by a learned ranking model f . Then, the ranking accuracy in terms of a ranking metric, *e.g.*, AUC or MAP, on the training data is rewritten as below, where $\beta_u = \frac{1}{|V_u^+|}$.

$$\frac{1}{|U|} \sum_{u \in U} \frac{1}{|V_u^+|} \sum_{i \in V_u^+} E[\pi(u, i, f)] \propto \sum_{(u, i) \in \mathcal{D}} \beta_u E[\pi(u, i, f)]. \quad (4.3)$$

To maximize the ranking accuracy, we can minimize the following loss function:

$$\arg \min_{f \in \mathcal{F}} \sum_{(u,i) \in \mathcal{D}} \beta_u \{1 - E[\pi(u, i, f)]\}, \quad (4.4)$$

where $\mathcal{F}$ is the set of possible ranking functions. Observation that this minimization is equivalent to maximizing the performance measures (*e.g.*, AUC or MAP). Because E is a non-continuous function, it is difficult to optimize the loss function defined in Eq. (4.4). To solve this issue, we can minimize its upper bound as follows:

$$\arg \min_{f \in \mathcal{F}} \sum_{(u,i) \in \mathcal{D}} \beta_u \exp\{-E[\pi(u, i, f)]\}. \quad (4.5)$$

The primary idea of applying boosting for item recommendation is to learn a set of homogeneous component recommenders and then create an ensemble of the component recommenders to predict users' preferences. Here, a linear combination of component recommenders is used as the final recommendation model:

$$f = \sum_{t=1}^T \alpha_t h^{(t)}, \quad (4.6)$$

where $h^{(t)}$ is the t^{th} component recommender and α_t is a positive weight assigned to $h^{(t)}$ to determine its contribution in the final recommendation model. In the training process, AdaBPR runs for T rounds, and one component recommender is created at each round. At the t^{th} round, given the former $t - 1$ component recommenders, the optimization problem in Eq. (4.5) is converted to

$$\arg \min_{\alpha_t, h^{(t)} \in \mathcal{H}} \sum_{(u,i) \in \mathcal{D}} \beta_u \exp\{-E[\pi(u, i, f^{(t-1)} + \alpha_t h^{(t)})]\}, \quad (4.7)$$

where $\mathcal{H}$ is the set of possible component recommenders and $f^{(t-1)} = \sum_{k=1}^{t-1} \alpha_k h^{(k)}$. To solve Eq. (4.7), AdaBPR first creates an optimal component recommender $h^{(t)}$ by using a re-weighting strategy, which assigns a dynamic weight $w_{ui}^{(t)}$ on each observed

Algorithm 3: The AdaBPR Algorithm

Input : The observed user-item interactions $\mathcal{D}$, and parameters E and T
Output: The ensemble recommender $f^{(T)}$.
 1 Initialize $w_{ui}^{(1)} = \beta_u / \sum_{(u,i) \in \mathcal{D}} \beta_u, \forall (u, i) \in \mathcal{D}$;
 2 **for** $t = 1, \dots, T$ **do**
 3 Create the component recommender $h^{(t)}$ using Alg. 4, based on $\mathcal{D}$ and the
 dynamic weights $w_{ui}^{(t)}, \forall (u, i) \in \mathcal{D}$;
 4 Compute the ranking accuracy measure $E[\pi(u, i, h^{(t)})], \forall (u, i) \in \mathcal{D}$;
 5 Compute the weight α_t assigned to $h^{(t)}$,
 6 $\alpha_t = \frac{1}{2} \ln \frac{\sum_{(u,i) \in \mathcal{D}} w_{ui}^{(t)} \{1 + E[\pi(u, i, h^{(t)})]\}}{\sum_{(u,i) \in \mathcal{D}} w_{ui}^{(t)} \{1 - E[\pi(u, i, h^{(t)})]\}};$
 7 Create the ensemble recommender $f^{(t)}$,
 8 $f^{(t)} = \sum_{k=1}^t \alpha_k h^{(k)}$;
 9 **foreach** $(u, i) \in \mathcal{D}$ **do**
 10 $w_{ui}^{(t+1)} = \frac{\beta_u \exp\{-E[\pi(u, i, f^{(t)})]\}}{\sum_{(u,i) \in \mathcal{D}} \beta_u \exp\{-E[\pi(u, i, f^{(t)})]\}};$
 11 **end**
 12 **end**

user-item interaction $(u, i) \in \mathcal{D}$. At each round, AdaBPR increases the weights of the observed user-item interactions that are not ranked well by the ensemble recommender created so far. The learning of the next component recommender will then pay more attention to those “hard” interactions. Suppose $h^{(t)}$ is given, its weight α_t can be solved. The details of the AdaBPR framework is presented in Alg. 3. A lower bound on the personalized ranking accuracy for AdaBPR is presented in Theorem 2. The proof for Theorem 2 can be found in the Appendix A.2.

Theorem 2. *The following bound holds on the personalized ranking accuracy of the AdaBPR algorithm on training data:*

$$\frac{1}{|U|} \sum_{u \in U} \frac{1}{|V_u^+|} \sum_{i \in V_u^+} E[\pi(u, i, f)] \geq \frac{\sum_{(u,i) \in \mathcal{D}} \beta_u}{|U|} \left[1 - \prod_{t=1}^T e^{-\delta_{\min}^t \sqrt{1 - \varphi(t)^2}} \right],$$

where $\varphi(t) = \sum_{(u,i) \in \mathcal{D}} w_{ui}^{(t)} E[\pi(u, i, h^{(t)})]$, $\delta_{\min}^t = \min_{(u,i) \in \mathcal{D}} \delta_{ui}^t$, and $\delta_{ui}^t = E[\pi(u, i, f^{(t-1)} + \alpha_t h^{(t)})] - E[\pi(u, i, f^{(t-1)})] - \alpha_t E[\pi(u, i, h^{(t)})]$, for all $(u, i) \in \mathcal{D}$ and $t = 1, 2, \dots, T$.

4.2.2 Constructing Component Recommender

To construct component recommenders, AdaBPR adopts the latent factor based models, which are the most successful recommendation models in literature [SK09]. The objective of latent factor model is to map both users and items into a shared latent space, with a low dimensionality $d \ll \min(|U|, |V|)$. Specifically, two latent vectors $\mathbf{p}_u^{(t)} \in \mathbb{R}^{1 \times d}$ and $\mathbf{q}_i^{(t)} \in \mathbb{R}^{1 \times d}$ are used to denote the properties of u and i respectively, in a component recommender $h^{(t)}$. The preference score of u to i is approximated by:

$$h_{ui}^{(t)} = \mathbf{p}_u^{(t)} \mathbf{q}_i^{(t)\top}. \quad (4.8)$$

Biases for users and items can also be incorporated in Eq. (4.8) to produce more accurate models. The latent factors of all users and all items are further denoted by $\mathbf{P}^{(t)} \in \mathbb{R}^{|U| \times d}$ and $\mathbf{Q}^{(t)} \in \mathbb{R}^{|V| \times d}$ respectively.

At each round, the accuracy of the component recommender $h^{(t)}$ can be evaluated by the ranking performance measure E weighted by $w_{ui}^{(t)}$. The optimal $h^{(t)}$ is then obtained by consistently optimizing the weighted ranking measure. In the implementation, AUC is chosen as the ranking metric to learn AdaBPR, considering both the effectiveness and efficiency factors. Note that AUC may be replaced by other ranking metrics whose range is within $[-1, +1]$, *e.g.*, MAP and NDCG [XL07].

Given the weight distribution $w_{ui}^{(t)}$, the accuracy of the component recommender $h^{(t)}$, measured by weighted AUC, is defined as follows:

$$\begin{aligned} \text{wAUC} &= \sum_{(u,i) \in \mathcal{D}} w_{ui}^{(t)} \frac{1}{|V_u^-|} \sum_{j \in V_u^-} \mathbb{I}(\pi_{ui}^{(t)} < \pi_{uj}^{(t)}) \\ &= \sum_{(u,i) \in \mathcal{D}} w_{ui}^{(t)} \frac{1}{|V_u^-|} \sum_{j \in V_u^-} \mathbb{I}(h_{ui}^{(t)} > h_{uj}^{(t)}), \end{aligned} \quad (4.9)$$

where $\pi_{ui}^{(t)}$ denotes the rank position of the item i in the list ranked by $h^{(t)}$ for u . Maximizing of the weighted AUC is equivalent to minimizing the following loss func-

Algorithm 4: Component Recommender Construction Algorithm

Input : The observed interactions $\mathcal{D}$, the weight distribution $w_{ui}^{(t)}$, the learning rate η , and the regularization parameter λ .
Output: The learned user latent factors $\mathbf{P}^{(t)}$ and item latent factors $\mathbf{Q}^{(t)}$.

```

1 Initialize  $\mathbf{P}^{(t)}$  and  $\mathbf{Q}^{(t)}$  randomly;
2 for  $e = 1, \dots, \text{max\_iter}$  do
3   Randomly permute the observed interactions  $\mathcal{D}$ ;
4   foreach  $(u, i) \in \mathcal{D}$  do
5     Randomly draw an item  $j$  from  $V_u^-$ ;
6     Update the model parameters  $\theta$  as follows,
       
$$\theta \leftarrow \theta - \eta \left[ \hat{w}_{ui}^{(t)} \cdot \frac{\partial \ell(\Delta_{uij}^{(t)})}{\partial \Delta_{uij}^{(t)}} \cdot \frac{\partial \Delta_{uij}^{(t)}}{\partial \theta} + \lambda \theta \right],$$

       where  $\theta = \mathbf{p}_u^{(t)}$ ,  $\mathbf{q}_i^{(t)}$ , and  $\mathbf{q}_j^{(t)}$ ;
7   end
8 end
    
```

tion:

$$\min_{\mathbf{P}^{(t)}, \mathbf{Q}^{(t)}} \sum_{(u,i) \in \mathcal{D}} w_{ui}^{(t)} \frac{1}{|V_u^-|} \sum_{j \in V_u^-} \mathbb{I}(h_{ui}^{(t)} \leq h_{uj}^{(t)}). \quad (4.10)$$

To solve this problem, the indicator function is replaced using a convex surrogate, *i.e.*, the hinge loss function, as follows:

$$\ell(\Delta_{uij}^{(t)}) = \max(0, 1 - \Delta_{uij}^{(t)}), \quad (4.11)$$

where $\Delta_{uij}^{(t)} = h_{ui}^{(t)} - h_{uj}^{(t)}$. The optimal component recommender $h^{(t)}$ is found by optimizing the following objective function:

$$\min_{\mathbf{P}^{(t)}, \mathbf{Q}^{(t)}} \sum_{(u,i) \in \mathcal{D}} \hat{w}_{ui}^{(t)} \sum_{j \in V_u^-} \frac{1}{|V_u^-|} \ell(\Delta_{uij}^{(t)}) + \frac{\lambda}{2} \left(\|\mathbf{P}^{(t)}\|_F^2 + \|\mathbf{Q}^{(t)}\|_F^2 \right), \quad (4.12)$$

where $\hat{w}_{ui}^{(t)} = w_{ui}^{(t)} \cdot |\mathcal{D}|$, λ is a regularization parameter, and $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. The problem in Eq. (4.12) can be solved by stochastic gradient descent, detailed in Alg. 4. Note that in Line 6, the component $\frac{1}{|V_u^-|}$ does not appear in the updating of model parameter θ . The reason is that for a given observed

Table 4.1: The statistics of the experimental datasets.

Datasets	#users	#items	#user-item pairs
Movielens-100K	943	1,574	82,520
Movielens-1M	6,039	3,628	836,478
TPMSD-790K	27,216	9,994	789,915

interaction $(u, i) \in \mathcal{D}$, an item j is sampled from V_u^- with uniform probability $\frac{1}{|V_u^-|}$ to form a triplet (u, i, j) used to train the component recommender.

4.3 Experiments

This section first empirically evaluates AdaBPR against state-of-the-art methods and then studies the impact of different parameter settings to the performances of AdaBPR.

4.3.1 Experimental Settings

Dataset Description. The experiments are conducted on three public datasets, including the MovieLens-100K³, MovieLens-1M, and the Taste Profile Subset of the Million Song Dataset (TPMSD) [BMEWL11]. MovieLens-100K contains 100,000 ratings given by 943 users to 1,682 movies. MovieLens-1M contains 1,000,209 ratings given by 6,040 users to 3,952 movies. For these two datasets, the ratings larger than 3 are kept as observed preference feedback, to simulate the implicit feedback scenario. Similar settings have been used in earlier studies [PC13b]. The TPMSD contains 48,373,586 listening records made by 1,019,318 users over 384,546 songs. In the evaluation, a subset of TPMSD is used with the following sampling strategy. Firstly, 10,000 songs that have been listened by at least 50 different users are randomly sampled. Secondly, the users who have listened 20 or more different sampled songs and their listening history over the sampled songs are extracted. As a result, about 790K interactions made by 27,216 users to 9,994 songs are collected, and this dataset

³<http://grouplens.org/datasets/movielens/>

is denoted by TPMSD-790K. Table 5.1 summarizes the three datasets where the movies and songs are “items”.

Each dataset is randomly partitioned into two non-overlapping sets for training and testing. For each user u , 70% of her historical items V_u^+ are randomly chosen as training data, and the remaining 30% of V_u^+ are used for testing. The densities of the user-item interaction matrix generated from the training data of the three datasets MovieLens-100K, MovieLens-1M, and TPMSD-790K are 4.05×10^{-2} , 2.70×10^{-2} , and 2.08×10^{-3} , respectively. The density of the training data is defined as in Eq. (3.17).

Evaluation Metrics. The accuracy of a recommendation model is measured by using three metrics, namely MAP, AUC, and NDCG@10. The former two metrics are widely adopted for evaluation of ranking accuracy. Because users are usually interested in a few top-ranked items, NDCG@ N is also used to compare the top- N recommendation performance. In the experiments, N is set at 10. For each metric, the accuracy for each user on the testing data is computed, and then the averaged accuracy over all users is reported.

Evaluated Recommendation Algorithms. This section evaluates the performances of AdaBPR and 4 baseline algorithms: PopRank, BPRMF, GBPR, and AdaMF.

- PopRank is a naïve baseline that recommends items to users purely based on the popularity of items.
- BPRMF is a state-of-the-art approach designed for item recommendation with users' implicit feedback [RFGST09]. It is a pairwise approach and has outperformed pointwise methods [PZC⁺08, HKV08], especially in terms of the ranking metric AUC.
- GBPR is an extension of BPRMF. It combines both the pairwise preference and group preference for item recommendation [PC13b].

- AdaMF is a boosting approach that extends AdaRank for item recommendation [WSZ14]. It generates item recommendations by approximating the explicit ratings (*e.g.*, 1-5) given by users. This original AdaMF algorithm is denoted by AdaMF^{exp} . This section also evaluates AdaMF^{imp} , which is AdaMF with implicit feedback. The rating of an observed user-item interaction pair is set to 1 and the rating of an un-observed interaction pair is 0. For a user u with item history V_u^+ , the same number of items are uniformly sampled from V_u^- to train the model.

Parameter Settings. The cross-validation strategy is adopted to choose the parameters for all the evaluated algorithms. Similar to that in [PC13b], the validation data are constructed by randomly taking 1 historical item of each user from the training data. For AdaBPR, the dimensionality of the latent space d is selected from $\{10, 30, 50\}$ (see Section 4.2.2), and for the other models, d is chosen from $\{10, 30, 50, 70, 90\}$. The regularization parameters are chosen from $\{10^{-6}, 10^{-5}, \dots, 1\}$ (see Eq. 4.12), and the optimal learning rates are selected from $\{2^{-10}, 2^{-9}, \dots, 1\}$ (see Alg. 4). For GBPR, the group size is chosen from $\{2, 3, 4, 5\}$ and the parameter ρ is chosen from $\{0.2, 0.4, 0.6, 0.8, 1.0\}$. The number of component models in AdaMF^{exp} , AdaMF^{imp} , and AdaBPR are all set to 30. All experiments are repeated for 5 times, each with a different random seed. The results reported are average of the 5 runs.

4.3.2 Performance Comparison

The recommendation accuracies of AdaBPR and other baseline methods are summarized in Table 4.2. The following observations are made from the experiments:

- On all three datasets, the proposed AdaBPR algorithm significantly outperforms all the other baseline algorithms, measured by the three metrics. The improvements over the baselines are statistically significant with $p < 0.001$ using Wilcoxon signed rank significance test [SG11].

Table 4.2: The accuracies of different recommendation algorithms measured by MAP, AUC, and NDCG@10.

Datasets	Methods	MAP	AUC	NDCG@10
Movielens-100k	PopRank	0.1485	0.8541	0.4962
	BPRMF	0.2780 ± 0.0029	0.9320 ± 0.0005	0.6705 ± 0.0005
	GBPR	<u>0.2907 ± 0.0017</u>	<u>0.9338 ± 0.0006</u>	<u>0.6942 ± 0.0031</u>
	AdaMF ^{exp}	0.1296 ± 0.0009	0.8288 ± 0.0014	0.4601 ± 0.0076
	AdaMF ^{imp}	0.2893 ± 0.0004	0.9322 ± 0.0002	0.6928 ± 0.0037
	AdaBPR	0.3132 ± 0.0007	0.9365 ± 0.0002	0.7310 ± 0.0040
Movielens-1M	PopRank	0.1206	0.8649	0.4875
	BPRMF	0.2172 ± 0.0005	0.9277 ± 0.0002	0.6352 ± 0.0005
	GBPR	<u>0.2295 ± 0.0010</u>	0.9315 ± 0.0002	<u>0.6617 ± 0.0022</u>
	AdaMF ^{exp}	0.1097 ± 0.0004	0.8458 ± 0.0007	0.4541 ± 0.0009
	AdaMF ^{imp}	0.2286 ± 0.0004	<u>0.9318 ± 0.0001</u>	0.6417 ± 0.0015
	AdaBPR	0.2629 ± 0.0004	0.9395 ± 0.0001	0.7124 ± 0.0015
TPMSD-790K	PopRank	0.0466	0.8345	0.1609
	BPRMF	0.0723 ± 0.0012	0.9048 ± 0.0007	0.2624 ± 0.0027
	GBPR	<u>0.0792 ± 0.0017</u>	0.9065 ± 0.0016	<u>0.2795 ± 0.0038</u>
	AdaMF ^{exp}	N.A.	N.A.	N.A.
	AdaMF ^{imp}	0.0735 ± 0.0006	<u>0.9079 ± 0.0004</u>	0.2501 ± 0.0021
	AdaBPR	0.1032 ± 0.0002	0.9190 ± 0.0001	0.3474 ± 0.0007

The best results are in **bold faces** and the second best results are underlined.

- The proposed AdaBPR attains significant improvement over other pairwise approaches without boosting, *i.e.*, BPRMF and GBPR. For instance, in terms of NDCG@10, AdaBPR outperforms BPRMF and GBPR by 32.39% and 24.29%, respectively, on TPMSD-790K dataset. This result shows that the accuracy of item recommendation can be largely improved by using boosting technique.
- Compared with AdaMF^{imp}, AdaBPR achieves much better results on all datasets for all the metrics. For example, in terms of NDCG@10, AdaBPR outperforms AdaMF^{imp} by 5.51%, 11.02%, and 38.90%, respectively, on the three datasets. This result clearly demonstrates the effectiveness of the proposed algorithm. The main reason is that the component recommenders in AdaBPR are learnt by using the algorithm designed for personalized ranking problems, while the component recommenders in AdaMF are built based on the algorithms for rating prediction tasks.

Figure 4.1: Performance trend of AdaBPR measured by MAP, AUC, and NDCG@10.

- AdaMF^{exp} performs much poorer than AdaMF^{imp} and other baseline methods. This suggests that AdaMF is more effective with implicit feedback than with explicit feedback, for item recommendation tasks.

4.3.3 Parameter Sensitivity Analysis

To evaluate the sensitivity to parameters for AdaBPR, an empirical analysis is conducted to study the performances of AdaBPR with respect to different parameter settings. The number of component recommenders is adjusted from 1 to 50 and the dimensionality of the latent space d of each component recommender is varied in $\{10, 30, 50\}$. The results on all the three datasets are summarized in Figure 4.1, in terms of MAP, AUC, and NDCG@10. Observed that the ranking performance gener-

ally improves with the increase of number of component recommenders, particularly when the number of component recommenders are fewer than 10. When there are more than 10 component recommenders, introducing more component recommenders leads to marginal improvements. Regarding the dimensionality of latent space factor d , larger d generally achieves better results. The only exception is the NDCG@10 measure on TPSMD-790K dataset where $d=10$ leads to better result than $d = 30$. Nevertheless, $d = 50$ achieves the best results or the second best results in all the experiment settings.

4.4 Summary

This chapter proposes a novel boosting algorithm, named AdaBPR (Adaptive Boosting Personalized Ranking), for the item recommendation tasks with users' implicit feedback. AdaBPR creates an ensemble of a set of homogeneous component recommenders built by the same collaborative filtering algorithm, to predict users' preferences on items. The performances of AdaBPR and four strong baseline methods are evaluated on three benchmark datasets, namely MovieLens-100K, MovieLens-1M, and the Taste Profile Subset of the Million Song Dataset. AdaBPR achieves the best recommendation accuracy on all the three datasets, evaluated on three widely adopted measures, MAP, AUC, and NDCG@10.

Chapter 5

Drug-Target Interaction Recommendation

5.1 Introduction

The drug discovery is one of the primary objectives of the pharmaceutical sciences, which is an interdisciplinary research field of fundamental sciences covering biology, chemistry, physics, statistics, etc. In the drug discovery process, the drug-target interaction (DTI) recommendation is an important step that aims to suggest a list of potential new drugs or new targets for existing drugs. Therefore, it can help to guide the experimental validation and reduce costs. Recently, the DTI recommendation has attracted vast research attentions and numerous algorithms have been proposed [DZ15]. Existing methods suggest potential DTIs based on a small number of experimentally validated interactions in existing databases, *e.g.*, ChEMBL [GBB⁺12], DrugBank [WKG⁺08], KEGG DRUG [KGS⁺12], and SuperTarget [GKD⁺08]. Previous studies have shown that a fraction of new interactions between drugs and targets can be predicted based on the experimentally validated DTIs, and the computational methods for identifying DTIs can significantly improve the drug discovery efficiency [DTMZ13, DZ15].

The recent rapid development of machine learning techniques provides effective and efficient ways to DTI recommendation. An intuitive idea is to formulate the DTI recommendation as a binary classification problem, where the drug-target pairs are treated as instances, and the chemical structures of drugs and the amino acid subsequences of targets are treated as features. Then, classical classification methods, *e.g.*, Support Vector Machines (SVM) and Regularized Least Square (RLS), can be used [NS07, BY09, XWZW10, vLNM11, vLM13, MKY⁺13, YKM⁺14]. Essentially, the DTI recommendation problem is a top- N recommendation task that aims to suggest a list of potential DTIs. Thus, another line of research for DTI recommendation is the application of low rank matrix factorization technique, which maps both drugs and targets into a shared latent space with low dimensionality [Gön12, CLH⁺13, ZDMZ13]. The drug-target interactions are modeled using the linear combinations of the latent representations of drugs and targets.

This chapter proposes a DTI recommendation approach, namely Neighborhood Regularized Logistic Matrix Factorization (NRLMF), which focuses on predicting the probability that a drug would interact with a target. Specifically, the properties of a drug and a target are represented by two vectors in the shared low dimensional latent space, respectively. For each drug-target pair, the interaction probability is modeled by a logistic function of the drug-specific and target-specific latent vectors. This is different from the KBMF method [Gön12] that predicts the interaction probability using a standard normal cumulative distribution function of the drug-specific and target-specific latent vectors [GK14]. In NRLMF, an observed interacting drug-target pair (*i.e.*, positive observation) is treated as c ($c \geq 1$) positive examples, while an unknown pair (*i.e.*, negative observation) is treated as a single negative example. As such, NRLMF assigns higher importance levels to positive observations than negatives. Because the positive observations are biologically validated and thus usually more trustworthy, however the negative observations could contain potential DTIs and are thus unreliable. This differs from previous matrix factorization methods that treat the interacting and unknown pairs equally [Gön12, CLH⁺13, ZDMZ13].

In addition, NRLMF also studies the local structure of the interaction data to further improve the DTI recommendation accuracy, by exploiting the neighborhood influences from most similar drugs and most similar targets. In particular, NRLMF imposes individual regularization constraints between the latent representations of a drug and its nearest neighbors, which are most similar with the given drug. Similar neighborhood regularization constraints have also been added on the latent representations of targets. Note that this neighborhood regularization method is different from previous approaches that exploit the drug similarities and target similarities using kernels [JV08, XWZW10, vLNM11, MKY⁺13] or factorizing the similarity matrices [ZDMZ13]. Moreover, the proposed approach only considers nearest neighbors instead of all similar neighbors as used in previous approaches, avoiding noisy information, thus achieves more accurate results.

5.2 Problem Formalization

In this chapter, the set of drugs is denoted by $D = \{d_i\}_{i=1}^m$, and the set of targets is denoted by $T = \{t_j\}_{j=1}^n$, where m and n are the number of drugs and number of targets, respectively. The interactions between drugs and targets are represented by a binary matrix $\mathbf{Y} \in \mathbb{R}^{m \times n}$, where each element $y_{ij} \in \{0, 1\}$. If a drug d_i has been experimentally verified to interact with a target t_j , y_{ij} is set to 1; otherwise, y_{ij} is set to 0. The non-zero elements in $\mathbf{Y}$ are called “interaction pairs” and regarded as positive observations. The zero elements in $\mathbf{Y}$ are called “unknown pairs” and regarded as negative observations. We define the set of positive drugs and targets as $D^+ = \{d_i | \sum_{j=1}^n y_{ij} > 0, \forall 1 \leq i \leq m\}$ and $T^+ = \{t_j | \sum_{i=1}^m y_{ij} > 0, \forall 1 \leq j \leq n\}$, respectively. Then, the set of negative drugs (*i.e.*, new drugs without any known interaction targets) and negative targets (*i.e.*, new targets without any known interaction drugs) are defined as $D^- = D \setminus D^+$ and $T^- = T \setminus T^+$, respectively. In addition, the drug similarities are represented by $\mathbf{S}^d \in \mathbb{R}^{m \times m}$, where the (i, μ) element $s_{i\mu}^d$ is the

similarity between d_i and d_μ . The target similarities are described using $\mathbf{S}^t \in \mathbb{R}^{n \times n}$, where the (j, ν) element $s_{j\nu}^t$ is the similarity between t_j and t_ν .

The objective of this study is to first predict the interaction probability of a drug-target pair and subsequently rank the candidate drug-target pairs according to the predicted probabilities in descending order, such that the top-ranked pairs are the most likely to interact.

5.3 Drug-Target Interaction Recommendation Model

5.3.1 Logistic Matrix Factorization

The matrix factorization technique has been successfully applied for DTI recommendation in previous studies. In this work, we develop the DTI recommendation model based on logistic matrix factorization (LMF) [Joh14], which has been demonstrated to be effective for personalized recommendations. The primary idea of applying LMF for DTI recommendation is to model the probability that a drug would interact with a target. In particular, both drugs and targets are mapped into a shared latent space, with a low dimensionality r , where $r \ll \min(m, n)$. The properties of a drug d_i and a target t_j are described by two latent vectors $\mathbf{u}_i \in \mathbb{R}^{1 \times r}$ and $\mathbf{v}_j \in \mathbb{R}^{1 \times r}$, respectively. Then, the interaction probability p_{ij} of a drug-target pair (d_i, t_j) is modeled by the following logistic function:

$$p_{ij} = \frac{\exp(\mathbf{u}_i \mathbf{v}_j^\top)}{1 + \exp(\mathbf{u}_i \mathbf{v}_j^\top)}. \quad (5.1)$$

For simplicity, we further denote the latent vectors of all drugs and all targets by $\mathbf{U} \in \mathbb{R}^{m \times r}$ and $\mathbf{V} \in \mathbb{R}^{n \times r}$ respectively.

In DTI recommendation tasks, the observed interacting drug-target pairs have been experimentally verified, thus they are more trustworthy and important than the unknown pairs. Towards a more accurate modeling for DTI recommendation, we propose to assign higher importance levels to the interaction pairs than unknown pairs. In particular, each interaction pair is treated as c ($c \geq 1$) positive training

examples, and each unknown pair is treated as a single negative training example. Here, c is a constant used to control the importance levels of observed interactions and is empirically set to 5 in the experiments. This importance weighting strategy has been demonstrated to be effective for personalized recommendations [HKV08, Joh14, LWSM14]. However, it has not been explored for DTI recommendation.

By assuming that all the training examples are independent, the probability of the observations is as follows:

$$p(\mathbf{Y}|\mathbf{U}, \mathbf{V}) = \left(\prod_{1 \leq i \leq m, 1 \leq j \leq n, y_{ij}=1} [p_{ij}^{y_{ij}} (1 - p_{ij})^{(1-y_{ij})}]^c \right) \times \left(\prod_{1 \leq i \leq m, 1 \leq j \leq n, y_{ij}=0} p_{ij}^{y_{ij}} (1 - p_{ij})^{(1-y_{ij})} \right). \quad (5.2)$$

Note that when $y_{ij} = 1$, $c(1 - y_{ij}) = 1 - y_{ij}$, and when $y_{ij} = 0$, $cy_{ij} = y_{ij}$. Hence, we can rewrite Eq. (5.2) as follows:

$$\begin{aligned} p(\mathbf{Y}|\mathbf{U}, \mathbf{V}) &= \left(\prod_{1 \leq i \leq m, 1 \leq j \leq n, y_{ij}=1} p_{ij}^{cy_{ij}} (1 - p_{ij})^{(1-y_{ij})} \right) \times \left(\prod_{1 \leq i \leq m, 1 \leq j \leq n, y_{ij}=0} p_{ij}^{cy_{ij}} (1 - p_{ij})^{(1-y_{ij})} \right) \\ &= \prod_{i=1}^m \prod_{j=1}^n p_{ij}^{cy_{ij}} (1 - p_{ij})^{(1-y_{ij})}. \end{aligned} \quad (5.3)$$

In addition, we also place zero-mean spherical Gaussian priors on the latent vectors of drugs and targets as:

$$p(\mathbf{U}|\sigma_d^2) = \prod_{i=1}^m \mathcal{N}(\mathbf{u}_i|0, \sigma_d^2 \mathbf{I}), \quad p(\mathbf{V}|\sigma_t^2) = \prod_{j=1}^n \mathcal{N}(\mathbf{v}_j|0, \sigma_t^2 \mathbf{I}), \quad (5.4)$$

where σ_d^2 and σ_t^2 are parameters controlling the variances of Gaussian distributions, and $\mathbf{I}$ denotes the identity matrix. Hence, through a Bayesian inference, we have

$$p(\mathbf{U}, \mathbf{V}|\mathbf{Y}, \sigma_d^2, \sigma_t^2) \propto p(\mathbf{Y}|\mathbf{U}, \mathbf{V})p(\mathbf{U}|\sigma_d^2)p(\mathbf{V}|\sigma_t^2). \quad (5.5)$$

The log of the posterior distribution is thus derived as follows:

$$\begin{aligned} \log p(\mathbf{U}, \mathbf{V} | \mathbf{Y}, \sigma_d^2, \sigma_t^2) &= \sum_{i=1}^m \sum_{j=1}^n cy_{ij} \mathbf{u}_i \mathbf{v}_j^\top - (1 + cy_{ij} - y_{ij}) \log [1 + \exp(\mathbf{u}_i \mathbf{v}_j^\top)] \\ &\quad - \frac{1}{2\sigma_d^2} \sum_{i=1}^m \|\mathbf{u}_i\|_2^2 - \frac{1}{2\sigma_t^2} \sum_{j=1}^n \|\mathbf{v}_j\|_2^2 + C, \end{aligned} \quad (5.6)$$

where C is a constant term independent with the model parameters (*i.e.*, $\mathbf{U}$ and $\mathbf{V}$). The model parameters can then be learned by maximizing the posterior distribution, which is equivalent with minimizing the following objective function:

$$\min_{\mathbf{U}, \mathbf{V}} \sum_{i=1}^m \sum_{j=1}^n (1 + cy_{ij} - y_{ij}) \log [1 + \exp(\mathbf{u}_i \mathbf{v}_j^\top)] - cy_{ij} \mathbf{u}_i \mathbf{v}_j^\top + \frac{\lambda_d}{2} \|\mathbf{U}\|_F^2 + \frac{\lambda_t}{2} \|\mathbf{V}\|_F^2, \quad (5.7)$$

where $\lambda_d = \frac{1}{\sigma_d^2}$, $\lambda_t = \frac{1}{\sigma_t^2}$, and $\|\cdot\|_F$ denotes the Frobenius norm of a matrix.

5.3.2 Regularized by Neighborhood

Through mapping both drugs and targets into a shared latent space, the LMF model can effectively estimate the global structure of the DTI data. However, LMF ignores the strong neighborhood associations among a small set of closely related drugs or targets. Thus, this chapter proposes to exploit the nearest neighborhood of a drug and that of a target to further improve the DTI recommendation accuracy. For a drug d_i , the set of its nearest neighbors is denoted by $N(d_i) \in D \setminus d_i$, where $N(d_i)$ is constructed by choosing K_1 most similar drugs with d_i . The set of the nearest neighbors of a target t_j is denoted by $N(t_j) \in T \setminus t_j$, which consists of the K_1 most similar targets with t_j . In the experiments, K_1 is empirically set to 5.

In this chapter, the drug neighborhood information is represented using an adjacency matrix $\mathbf{A}$, where the (i, μ) element $a_{i\mu}$ is defined as follows:

$$a_{i\mu} = \begin{cases} s_{i\mu}^d & \text{if } d_\mu \in N(d_i) \\ 0 & \text{otherwise.} \end{cases} \quad (5.8)$$

Similarly, the adjacency matrix used to describe the target neighborhood information is denoted by $\mathbf{B}$, where its (j, ν) element $b_{j\nu}$ is defined as follows:

$$b_{j\nu} = \begin{cases} s_{j\nu}^t & \text{if } t_\nu \in N(t_j) \\ 0 & \text{otherwise.} \end{cases} \quad (5.9)$$

Note that the adjacency matrices $\mathbf{A}$ and $\mathbf{B}$ are not symmetric.

The primary idea of exploiting the drug neighborhood information for DTI recommendation is to minimize the distances between d_i and its nearest neighbors $N(d_i)$ in the latent space. This objective can be achieved by minimizing the following objective function:

$$\begin{aligned} & \frac{\alpha}{2} \sum_{i=1}^m \sum_{\mu=1}^m a_{i\mu} \|\mathbf{u}_i - \mathbf{u}_\mu\|_F^2 \\ = & \frac{\alpha}{2} \left[\sum_{i=1}^m \left(\sum_{\mu=1}^m a_{i\mu} \right) \mathbf{u}_i \mathbf{u}_i^\top + \sum_{\mu=1}^m \left(\sum_{i=1}^m a_{i\mu} \right) \mathbf{u}_\mu \mathbf{u}_\mu^\top \right] - \frac{\alpha}{2} \text{tr}(\mathbf{U}^\top \mathbf{A} \mathbf{U}) - \frac{\alpha}{2} \text{tr}(\mathbf{U}^\top \mathbf{A}^\top \mathbf{U}) \\ = & \frac{\alpha}{2} \text{tr}(\mathbf{U}^\top \mathbf{L}^d \mathbf{U}), \end{aligned} \quad (5.10)$$

where $\text{tr}(\cdot)$ is the trace of a matrix, $\mathbf{L}^d = (\mathbf{D}^d + \tilde{\mathbf{D}}^d) - (\mathbf{A} + \mathbf{A}^\top)$. $\mathbf{D}^d$ and $\tilde{\mathbf{D}}^d$ are two diagonal matrices, in which the diagonal elements are $D_{ii}^d = \sum_{\mu=1}^m a_{i\mu}$ and $\tilde{D}_{\mu\mu}^d = \sum_{i=1}^m a_{i\mu}$ respectively. Moreover, the neighborhood information of targets is also exploited for DTI recommendation by minimizing the following objective function:

$$\frac{\beta}{2} \sum_{j=1}^n \sum_{\nu=1}^n b_{j\nu} \|\mathbf{v}_j - \mathbf{v}_\nu\|_F^2 = \frac{\beta}{2} \text{tr}(\mathbf{V}^\top \mathbf{L}^t \mathbf{V}), \quad (5.11)$$

where $\mathbf{L}^t = (\mathbf{D}^t + \tilde{\mathbf{D}}^t) - (\mathbf{B} + \mathbf{B}^\top)$, $\mathbf{D}^t$ and $\tilde{\mathbf{D}}^t$ are two diagonal matrices, in which the diagonal elements are $D_{jj}^t = \sum_{\nu=1}^n b_{j\nu}$ and $\tilde{D}_{\nu\nu}^t = \sum_{j=1}^n b_{j\nu}$. Note that the proposed neighborhood regularization only considers influences from the K_1 nearest neighbors of each drug and each target. It is different from the graph Laplacian constraints used in previous studies [HAX⁺12, GJP14], which consider influences from all similar drugs

and targets. Clearly, given a drug-target pair, the proposed approach would like to leverage their nearest neighbors, instead of all the neighbors that could potentially introduce noisy information, to enhance the recommendation accuracy.

5.3.3 Neighborhood Regularized Logistic Matrix Factorization

The final DTI recommendation model can be formulated by considering the drug-target interactions as well as the neighborhood of drugs and targets. By plugging Eq. (5.10) and Eq. (5.11) into Eq. (5.7), the proposed NRLMF model is formulated as follows:

$$\begin{aligned} \min_{\mathbf{U}, \mathbf{V}} \sum_{i=1}^m \sum_{j=1}^n (1 + cy_{ij} - y_{ij}) \ln [1 + \exp(\mathbf{u}_i \mathbf{v}_j^\top)] - cy_{ij} \mathbf{u}_i \mathbf{v}_j^\top \\ + \frac{1}{2} \text{tr} [\mathbf{U}^\top (\lambda_d \mathbf{I} + \alpha \mathbf{L}^d) \mathbf{U}] + \frac{1}{2} \text{tr} [\mathbf{V}^\top (\lambda_t \mathbf{I} + \beta \mathbf{L}^t) \mathbf{V}]. \end{aligned} \quad (5.12)$$

The optimization problem in Eq. (5.12) can be solved by an alternating gradient ascent procedure. Denoting the objective function in Eq. (5.12) by L , the partial gradients with respect to $\mathbf{U}$ and $\mathbf{V}$ are as follows:

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{U}} &= \mathbf{P} \mathbf{V} + (c - 1)(\mathbf{Y} \odot \mathbf{P}) \mathbf{V} - c \mathbf{Y} \mathbf{V} + (\lambda_d \mathbf{I} + \alpha \mathbf{L}^d) \mathbf{U} \\ \frac{\partial L}{\partial \mathbf{V}} &= \mathbf{P}^\top \mathbf{U} + (c - 1)(\mathbf{Y}^\top \odot \mathbf{P}^\top) \mathbf{U} - c \mathbf{Y}^\top \mathbf{U} + (\lambda_t \mathbf{I} + \beta \mathbf{L}^t) \mathbf{V}, \end{aligned} \quad (5.13)$$

where $\mathbf{P} \in \mathbb{R}^{m \times n}$, in which the (i, j) element is p_{ij} (see Eq. (5.1)), $\odot$ denotes the Hadamard product of two matrices. To accelerate the convergence of the gradient descent optimization methods, the AdaGrad algorithm [DHS11] is used to adaptively choose the gradient step size. The details of the optimization algorithm to the proposed NRLMF model are described in Alg. 5, where $\mathbf{U}$ and $\mathbf{V}$ are randomly initialized using a Gaussian distribution with mean 0, standard deviation $\frac{1}{\sqrt{r}}$.

Algorithm 5: NRLMF Optimization Algorithm

Input : $\mathbf{Y}, \mathbf{S}^d, \mathbf{S}^t, c, r, K_1, K_2, \lambda_d, \lambda_t, \alpha, \beta, \gamma$
Output: $\mathbf{U}, \mathbf{V}$

- 1 Initialize $\mathbf{U}$ and $\mathbf{V}$ randomly, and set $\varphi_{ik} = 0, \phi_{jk} = 0, \forall 1 \leq i \leq m, 1 \leq j \leq n,$
and $1 \leq k \leq r$;
- 2 Construct the adjacency matrices $\mathbf{A}$ and $\mathbf{B}$ according to Eq. (5.8) and
Eq. (5.9) respectively;
- 3 Compute the neighborhood regularization matrices $\mathbf{L}^d$ and $\mathbf{L}^t$ according to
Eq. (5.10) and Eq. (5.11) respectively;
- 4 **for** $t = 1, \dots, \text{max_iter}$ **do**
- 5 $\mathbf{G}^d \leftarrow \frac{\partial L}{\partial \mathbf{U}}$; // fix $\mathbf{V}$ and compute the gradient with respect to $\mathbf{U}$
- 6 **for** $i = 1, \dots, m$ **do**
- 7 **for** $k = 1, \dots, r$ **do**
- 8 // g_{ik}^d and u_{ik} are the (i, k) elements in $\mathbf{G}^d$ and $\mathbf{U}$.
- 9 $\varphi_{ik} \leftarrow \varphi_{ik} + g_{ik}^d \cdot g_{ik}^d$;
- 10 $u_{ik} \leftarrow u_{ik} - \gamma \frac{g_{ik}^d}{\sqrt{\varphi_{ik}}}$; // update each element of d_i 's latent vector
- 11 **end**
- 12 **end**
- 13 $\mathbf{G}^t \leftarrow \frac{\partial L}{\partial \mathbf{V}}$; // fix $\mathbf{U}$ and compute the gradient with respect to $\mathbf{V}$
- 14 **for** $j = 1, \dots, n$ **do**
- 15 **for** $k = 1, \dots, r$ **do**
- 16 // g_{jk}^t and v_{jk} are the (j, k) element in $\mathbf{G}^t$ and $\mathbf{V}$.
- 17 $\phi_{jk} \leftarrow \phi_{jk} + g_{jk}^t \cdot g_{jk}^t$;
- 18 $v_{jk} \leftarrow v_{jk} - \gamma \frac{g_{jk}^t}{\sqrt{\phi_{jk}}}$; // update each element of t_j 's latent vector
- 19 **end**
- 20 **end**
- 21 **end**

Once the latent vectors $\mathbf{U}$ and $\mathbf{V}$ have been learned, the probability associated with any unknown drug-target pair (d_i, t_j) can be predicted using Eq. (5.1). However, in the training procedure, the latent vectors of drugs belonging to the negative drug set D^- and those of the targets belonging to the negative target set T^- are learned solely based on negative observations (*i.e.*, unknown pairs). As we know, some negative observations may be potential positive DTIs. Due to such uncertainty over negative observations, the learned latent vectors of the negative drugs and targets may not be accurate enough to describe their properties. One remedy for this problem is to replace the latent vector of a negative drug/target using the linear combination of the latent vectors of its nearest neighbors in the positive set. For a drug $d_i \in D^-$, the set of its K_2 nearest neighbors in D^+ is denoted by $N^+(d_i)$. Similarly, for a target $t_j \in T^-$, the set of its K_2 nearest neighbors in T^+ is denoted by $N^+(t_j)$. Note that $N^+(d_i)$ and $N^+(t_j)$ are built using the same criteria as that used to construct the neighborhood in the training procedure. Then, the prediction of the interaction probability of a drug-target pair (u_i, v_j) is modified as,

$$\hat{p}_{ij} = \frac{\exp(\tilde{\mathbf{u}}_i \tilde{\mathbf{v}}_j^\top)}{1 + \exp(\tilde{\mathbf{u}}_i \tilde{\mathbf{v}}_j^\top)}, \quad (5.14)$$

where

$$\begin{aligned} \tilde{\mathbf{u}}_i &= \begin{cases} \mathbf{u}_i & \text{if } d_i \in D^+ \\ \frac{1}{\sum_{\mu \in N^+(d_i)} s_{i\mu}^d} \sum_{\mu \in N^+(d_i)} s_{i\mu}^d \mathbf{u}_\mu & \text{if } d_i \in D^-, \end{cases} \\ \tilde{\mathbf{v}}_j &= \begin{cases} \mathbf{v}_j & \text{if } t_j \in T^+ \\ \frac{1}{\sum_{\nu \in N^+(t_j)} s_{j\nu}^t} \sum_{\nu \in N^+(t_j)} s_{j\nu}^t \mathbf{v}_\nu & \text{if } t_j \in T^-. \end{cases} \end{aligned} \quad (5.15)$$

Note that Eq. (5.15) shows a general case for smoothing the learned drug-specific and target-specific latent vectors. In the experiments, K_2 is empirically set to 5 to simplify the model.

5.4 Experiments

Extensive experiments on the benchmark datasets have been performed to evaluate the performances of the proposed NRLMF method.

5.4.1 Experimental Settings

The performances of DTI recommendation algorithms are evaluated on four benchmark datasets, including Nuclear Receptors, G-Protein Coupled Receptors (GPCR), Ion Channels, and Enzymes. These datasets are originally provided by [YAG⁺08] and are publicly available at <http://web.kuicr.kyoto-u.ac.jp/supp/yoshi/drugtarget/>. Table 5.1 summarizes the statistics of all four datasets. Each dataset contains three types of information: 1) the observed DTIs, 2) the drug similarities, and 3) the target similarities. Particularly, the observed DTIs are retrieved from public databases KEGG BRITE [KGH⁺06], BRENDA [SCE⁺04], SuperTarget [GKD⁺08], and Drug-Bank [WKG⁺08]. The drug similarities are computed based on the chemical structures of the compounds derived from the DRUG and COMPOUND sections in the KEGG LIGAND database [KGH⁺06]. For a pair of compounds, the similarity between their chemical structures is measured by the SIMCOMP algorithm [HOGK03]. The target similarities, on the other hand, are calculated based on the amino acid sequences of target proteins retrieved from the KEGG GENES database [KGH⁺06]. The sequence similarity between two proteins is computed using the normalized Smith-Waterman score.

Table 5.1: The statistics of the drug-target interaction datasets.

Dataset	# Drugs	# Targets	# Interactions
Nuclear Receptor	54	26	90
GPCR	223	95	635
Ion Channel	210	204	1476
Enzyme	445	664	2926

The cross-validation is adopted to evaluate the performances of DTI recommendation methods. For NRLMF and other competing methods, 10-fold cross-validation

is performed for 5 times, each with a different random seed. Following previous studies [MKY⁺13, vLM13, ZDMZ13], the widely adopted measures, *i.e.*, the Area Under the ROC Curve (AUC) and the Area Under the Precision-Recall Curve (AUPR), are used as the evaluation metrics. In particular, an AUC score in each repetition of cross-validation is calculated, and a final AUC score that is the average over the 5 repetitions is reported. The AUPR is calculated using the same strategy.

The drug-target interaction matrix $\mathbf{Y} \in \mathbb{R}^{m \times n}$ had m rows for drugs and n columns for targets. We conducted CV under three different settings as follows [ZDMZ13, PAP⁺15].

- CVS1: CV on drug-target pairs — random entries in $\mathbf{Y}$ (*i.e.*, drug-target pairs) are selected for testing.
- CVS2: CV on drugs — random rows in $\mathbf{Y}$ (*i.e.*, drugs) are blinded for testing.
- CVS3: CV on targets — random columns in $\mathbf{Y}$ (*i.e.*, targets) are blinded for testing.

These 3 settings CVS1, CVS2, and CVS3 refer to the DTI recommendation for (a) new (unknown) interactions, (b) new drugs, and (c) new targets, respectively. In this chapter, the proposed NRLMF model is compared with the following DTI recommendation methods, namely NetLapRLS [XWZW10], KBMF2K [Gön12], BLM-NII [MKY⁺13], WNN-GIP [vLM13], and CMF [ZDMZ13], by testing their recommendation capabilities under the above three settings.

In the matrix factorization based methods (*i.e.*, KBMF2K, CMF, and NRLMF), the dimensionality of the latent space r is selected from $\{50, 100\}$ [ZDMZ13]. For NRLMF, λ_d is set to be the same with λ_t , and these two parameters are chosen from $\{2^{-5}, 2^{-4}, \dots, 2^1\}$ (see Eq. (5.7)). The neighborhood regularization parameters α and β of NRLMF are selected from $\{2^{-5}, 2^{-4}, \dots, 2^2\}$ and $\{2^{-5}, 2^{-4}, \dots, 2^0\}$, respectively (see Eq. (5.12)), and the optimal learning rate γ is selected from $\{2^{-3}, 2^{-2}, \dots, 2^0\}$ (see Alg. 5). The parameters of the remaining methods (*i.e.*, NetLapRLS, KBMF2K,

Table 5.2: The AUC and AUPR obtained under cross-validation setting CVS1.

AUC					
	Nuclear Receptor	GPCR	Ion Channel	Enzyme	Avg.
NetLapRLS	0.850±0.021*	0.915±0.006*	0.969±0.003*	0.972±0.002*	0.927
BLM-NII	<u>0.905±0.023*</u>	<u>0.950±0.006*</u>	<u>0.981±0.002*</u>	<u>0.978±0.002*</u>	<u>0.954</u>
WNN-GIP	0.901±0.017*	0.944±0.005*	0.959±0.003*	0.964±0.003*	0.942
KBMF2K	0.877±0.023*	0.926±0.006*	0.961±0.003*	0.905±0.003*	0.917
CMF	0.864±0.026*	0.940±0.007*	0.981±0.002*	0.969±0.002*	0.939
NRLMF	0.950±0.011	0.969±0.004	0.989±0.001	0.987±0.001	0.974
AUPR					
	Nuclear Receptor	GPCR	Ion Channel	Enzyme	Avg.
NetLapRLS	0.465±0.044*	0.616±0.015*	0.837±0.009*	0.789±0.005*	0.677
BLM-NII	<u>0.659±0.039*</u>	0.524±0.024*	0.821±0.012*	0.752±0.011*	0.689
WNN-GIP	0.589±0.034*	0.520±0.021*	0.717±0.020*	0.706±0.017*	0.633
KBMF2K	0.534±0.050*	0.578±0.018*	0.771±0.009*	0.654±0.008*	0.634
CMF	0.584±0.042*	<u>0.745±0.013</u>	0.923±0.006	<u>0.877±0.005*</u>	<u>0.782</u>
NRLMF	0.728±0.041	0.749±0.015	<u>0.906±0.008</u>	0.892±0.006	0.819

“Avg.” shows the average AUC/AUPR over four datasets. In each column, the best results are in **bold faces** and the second best results are underlined. * indicates NRLMF significantly outperforms the competitor with $p < 0.05$ using t-test.

BLM-NII, WNN-GIP, and CMF) are specified following the corresponding references. For each CV setting, the best AUC and the corresponding AUPR values of each method, obtained by different parameter combinations, are reported.

5.4.2 Performance Comparison

Table 5.2 shows the AUC and AUPR values obtained by various methods under the cross-validation setting CVS1. As shown in Table 5.2, NRLMF attains the best AUC values over all datasets. The final average AUC obtained by NRLMF is 0.974, which is 2.10% better than the second method BLM-NII. Moreover, NRLMF achieves the highest AUPR over 3 datasets (*i.e.*, Nuclear Receptor, GPCR, and Enzyme) and obtains the second best AUPR on the Ion Channel dataset, where CMF outperforms NRLMF (0.923 for CMF vs. 0.906 for NRLMF). The average AUPR obtained by NRLMF is 0.819, which is 4.73% higher than that obtained by the second best method CMF. In summary, under the cross-validation setting CVS1, NRLMF outperforms

Table 5.3: The AUC and AUPR obtained under cross-validation setting CVS2.

AUC					
	Nuclear Receptor	GPCR	Ion Channel	Enzyme	Avg.
NetLapRLS	0.789±0.039*	0.817±0.015*	0.757±0.025*	0.786±0.023*	0.787
BLM-NII	0.799±0.037*	0.838±0.016*	0.796±0.025	0.813±0.022*	0.812
WNN-GIP	<u>0.890</u> ±0.023	<u>0.891</u> ±0.010	0.797±0.028	0.882 ±0.015	<u>0.865</u>
KBMF2K	0.844±0.023*	0.839±0.020*	<u>0.799</u> ±0.019	0.713±0.029*	0.799
CMF	0.818±0.036*	0.857±0.014*	0.743±0.029*	0.829±0.019*	0.812
NRLMF	0.900 ±0.021	0.895 ±0.011	0.813 ±0.027	<u>0.871</u> ±0.017	0.870
AUPR					
	Nuclear Receptor	GPCR	Ion Channel	Enzyme	Avg.
NetLapRLS	0.417±0.048*	0.229±0.017*	0.200±0.026*	0.123±0.009*	0.242
BLM-NII	0.438±0.048*	0.315±0.022*	0.302±0.033	0.253±0.023*	0.327
WNN-GIP	<u>0.504</u> ±0.056	0.295±0.025*	0.258±0.032*	<u>0.278</u> ±0.037*	0.334
KBMF2K	0.477±0.049	0.366 ±0.024	0.308±0.038	0.263±0.033*	<u>0.354</u>
CMF	0.488±0.050	<u>0.365</u> ±0.022	0.286±0.030*	0.229±0.020*	0.342
NRLMF	0.545 ±0.054	0.364±0.023	0.344 ±0.033	0.358 ±0.040	0.403

“Avg.” shows the average AUC/AUPR over four datasets. In each column, the best results are in **bold faces** and the second best results are underlined. * indicates NRLMF significantly outperforms the competitor with $p < 0.05$ using t-test.

other competing methods, being statistically significant except two comparison cases with CMF at the significant level of 0.05 using t-test.

The results obtained under the setting CVS2 for new drugs are shown in Table 5.3. In particular, NRLMF outperforms the other methods over the Nuclear Receptor, GPCR, and Ion Channel datasets, in terms of AUC. On the Enzyme dataset, WNN-GIP achieves a little better AUC than NRLMF (0.882 for WNN-GIP vs. 0.871 for NRLMF). Over all datasets, NRLMF obtains the best average AUC value 0.870. For the AUPR metric, NRLMF achieves the best results on all datasets except the GPCR dataset, where KBMF2K and CMF are slightly better than NRLMF. Overall, NRLMF achieves the best average AUPR 0.403, which is 13.84% higher than the second-best method KBMF2K and 17.84% higher than the third-best method CMF.

In addition, Table 5.4 summarizes the results obtained under the setting CVS3 for new targets. It is observed that WNN-GIP outperforms other methods on the Nuclear Receptor dataset, in terms of AUC and AUPR. On the other 3 datastes, the proposed NRLMF achieves the best AUC and AUPR values. Over all datasets, WNN-GIP achieves the highest average AUC value 0.940, which is 1.29% better than the

Table 5.4: The AUC and AUPR obtained under cross-validation setting CVS3.

AUC					
	Nuclear Receptor	GPCR	Ion Channel	Enzyme	Avg.
NetLapRLS	0.655±0.046*	0.770±0.024*	0.914±0.012*	0.905±0.014*	0.811
BLM-NII	0.534±0.086*	0.778±0.025*	0.914±0.012*	0.909±0.014*	0.784
WNN-GIP	0.935 ±0.017	<u>0.926</u> ±0.013	<u>0.950</u> ±0.007*	<u>0.947</u> ±0.008*	0.940
KBMF2K	0.668±0.060*	0.882±0.016*	0.938±0.008*	0.876±0.012*	0.8288
CMF	0.680±0.066*	0.837±0.019*	0.905±0.012*	0.915±0.013*	0.834
NRLMF	<u>0.851</u> ±0.027	0.930 ±0.012	0.964 ±0.007	0.966 ±0.005	<u>0.928</u>
AUPR					
NetLapRLS	<u>0.449</u> ±0.074	0.334±0.025*	0.737±0.020*	0.669±0.021*	0.547
BLM-NII	0.402±0.083	0.341±0.034*	<u>0.762</u> ±0.020	<u>0.735</u> ±0.022*	0.560
WNN-GIP	0.531 ±0.073	<u>0.550</u> ±0.047	0.696±0.035*	0.566±0.038*	<u>0.586</u>
KBMF2K	0.324±0.071	0.516±0.045	0.677±0.021*	0.565±0.023*	0.521
CMF	0.400±0.077	0.433±0.028*	0.620±0.027*	0.698±0.021*	0.538
NRLMF	<u>0.449</u> ±0.079	0.556 ±0.038	0.785 ±0.028	0.812 ±0.018	0.651

“Avg.” shows the average AUC/AUPR over four datasets. In each column, the best results are in **bold faces** and the second best results are underlined. * indicates NRLMF significantly outperforms the competitor with $p < 0.05$ using t-test.

second-best method NRLMF. For AUPR, NRLMF achieves the best average AUPR 0.651, which is a 11.09% better than the second-best method WNN-GIP.

The task under the setting CVS1 focuses on predicting the unknown pair (d_i, t_j) , where at least one DTI is known for d_i and t_j respectively in the training data. However, the tasks under CVS2 and CVS3 focus on the recommendations for new drugs and new targets respectively, where no DTIs are observed for new drugs and new targets in the training data. Therefore, the task under CVS1 is easier than those under CVS2 and CVS3, and the AUC and AUPR values obtained by DTI recommendation methods under CVS1 are higher than those obtained under CVS2 and CVS3 as expected. For all CV settings, the proposed NRLMF method achieves the best AUC values in 10 out of 12 scenarios (*i.e.*, 3 CV settings on 4 datasets) via integrating LMF with neighborhood regularization. In the remaining 2 scenarios (*i.e.*, CVS2 on Enzyme dataset and CVS3 on Nuclear Receptor dataset), WNN-GIP attains better AUC values than NRLMF. The results in these 2 scenarios can be interpreted as follows. For instance, under CVS2, the interactions for 10% of the drugs (*i.e.*, the set of negative drugs D^-) have been blinded in the training phase.

The latent vectors of D^- are learned solely based on negative observations, and thus are not accurate. This may lead to the inaccuracies of the learned latent vectors of targets (see Eq. (5.13)). Especially, for the targets with only one interaction, the accuracies of the learned latent vectors may be drastically reduced. In NRLMF, the latent vectors of negative drugs and targets are smoothed using their nearest neighbors. However, there is no smoothing for the latent vectors of targets with only one interaction (see Eq. (5.15)). As such, the performances of NRLMF under CVS2 may be affected more on the dataset with a higher percentage of targets that have only one interaction. Interestingly, over 4 datasets, the percentage of targets that have only one interaction is 30.77%, 35.79%, 11.27%, and 43.37%, for Nuclear Receptor, GPCR, Ion Channel, and Enzyme, respectively. Enzyme dataset has the highest percentage of targets with only one interaction, and thus the performance of NRLMF on this dataset under CVS2 is likely to be affected most. Similarly, the percentage of drugs with only one interaction is 72.22% for Nuclear Receptor, 47.53% for GPCR, 38.57% for Ion Channel, and 39.78% for Enzyme. Thus, by blinding the interactions of 10% targets (*i.e.*, under CVS3), the performance of NRLMF on Nuclear Receptor dataset is the most likely to be affected. For the AUPR metric, NRLMF attains the best AUPR values in 9 out of 12 scenarios, which is to be expected, since the methods that optimize AUC are not guaranteed to optimize AUPR [?]. In addition, the target sequence similarity $\mathbf{S}^t$ is more reliable and informative than the drug chemical similarity $\mathbf{S}^d$ [vLNM11]. Hence, the information propagated from the neighbors to the new targets by the regularization term in Eq. (5.11) will be more accurate than those to new drugs by the term in Eq. (5.10). This explains the results well that various methods usually achieve higher AUC and AUPR under CVS3 than CVS2.

5.4.3 Neighborhood Benefits

The proposed NRLMF incorporates neighborhood information for DTI recommendation via the neighborhood regularization in training and the neighborhood smoothing

Figure 5.1: Performance trend of NRLMF measured by AUC and AUPR with different settings of K_1 under cross-validation setting CVS1.

in prediction. This section studies how the neighborhood information benefits DTI recommendation under cross-validation setting CVS1.

Figure 5.1(a) shows the AUC values obtained by NRLMF with respect to different settings of the neighborhood size K_1 used for the neighborhood regularization in the training procedure. As shown in Figure 5.1(a), the optimal values of K_1 are 3, 5, 5, and 5, for four datasets, respectively. Under the setting CVS1, the average AUC of NRLMF is 0.958 when K_1 is set as 0 (*i.e.*, without neighborhood regularization in training), while it is increased to 0.974 when K_1 is set as 5. Figure 5.1(b) illustrates the AUPR values with respect to different settings of K_1 . Observed that NRLMF achieves the best AUPR by setting K_1 as 7, 7, 9, and 3, respectively. When $K_1 = 0$, the average AUPR achieved by NRLMF without neighborhood regularization is 0.772, while it is increased to 0.818 by setting $K_1 = 5$. These results highlight that the neighborhood regularization is highly desirable for DTI recommendation.

In addition, this section also studies the impact of the neighborhood size K_2 used for neighborhood smoothing in the prediction procedure. Figures 5.2(a) and 5.2(b) plot the AUC and AUPR values obtained by NRLMF with respect to different settings of K_2 . As shown in Figure 5.2(a), NRLMF achieves best AUC via setting K_2 as 5, 3, 5, and 5, respectively. For AUPR measure, the best results are achieved by

Figure 5.2: Performance trend of NRLMF measured by AUC and AUPR with different settings of K_2 under cross-validation setting CVS1.

setting K_2 as 5, 3, 9, and 5, respectively. Over all datasets, when $K_2 = 0$ (*i.e.*, without neighborhood smoothing in prediction), the average AUC and AUPR values obtained by NRLMF are 0.950 and 0.772, respectively, while these values are 0.974 and 0.819 when $K_2 = 5$. These observations demonstrate the effectiveness of using nearest neighbors to predict the interaction probability for a given drug-target pair. In addition, when K_1 and K_2 are set at 5, NRLMF can get reasonably good results for both AUC and AUPR, respectively.

5.4.4 Parameter Sensitivity Analysis

This section focuses on the sensitivity analysis for other two parameters, *i.e.*, the importance levels of observed DTIs c and the dimensionality of the latent space r , under the setting CVS1.

As shown in Figure 5.3(a), when the importance level c is set as 1, NRLMF achieves the lowest AUC on all the datasets, and it is able to achieve the optimal AUC over all datasets with $c = 5$. Similarly, Figure 5.3(b) shows that NRLMF achieves better AUPR under the setting $c > 1$ than under the setting $c = 1$, on the GPCR, Ion Channel, and Enzyme datasets. On the Nuclear Receptor dataset, NRLMF attains slightly better AUPR under the setting $c = 1$ than under the other

Figure 5.3: Performance trend of NRLMF measured by AUC and AUPR with different settings of c under cross-validation setting CVS1.

settings. These observations demonstrate that assigning more importance on the observed interactions can boost the performance of NRLMF. However, when c is large enough, the performance of NRLMF tends to become saturated, where further increasing c has very limited improvement.

The impact of the dimensionality of the latent space r on the performance of NRLMF, in terms of AUC and AUPR, is shown in Figure 5.4(a) and 5.4(b), respectively. It is found that larger r generally achieves better results. The two exceptions are the AUPR measure on Nuclear Receptor and Ion Channel datasets, where $r = 30$ leads to slightly better results than $r = 50$. Nevertheless, $r = 100$ achieves the best results or the second best results measured by AUC and AUPR, on all datasets. Thus, the parameter r is recommended to be set in the range $[50, 100]$, which is consistent with previous studies [ZDMZ13].

5.4.5 Recommending Novel Interactions

This section evaluates the practical ability of NRLMF on recommending novel interactions, which refer to interactions with high probabilities that do not occur in the benchmark datasets. Following similar settings in previous studies [BY09, vLNM11, Gön12], four well-known biological databases, *i.e.*, ChEMBL [GBB⁺12],

Figure 5.4: Performance trend of NRLMF measured by AUC and AUPR with different settings of r under cross-validation setting CVS1.

DrugBank [KLJ⁺11], KEGG [KGS⁺12], and Matador [GKD⁺08], are used as reference to verify whether the suggested new DTIs are true or not.

To conduct this study, the online profiles associated with the drugs and targets in each benchmark dataset are crawled from the online reference databases, and the approved drug-target interactions are collected. Over all benchmark datasets, there are 791 drugs and 986 targets, and 1,999 novel interactions have been confirmed in one or more reference databases. The number of confirmed novel interactions in Nuclear Receptor, GPCR, Ion Channel, and Enzyme datasets are 21, 512, 1034, and 432, respectively. For each dataset, the entire dataset is used as training set. The unknown interactions will be ranked based on the interaction probabilities predicted using the optimal parameters learned under cross-validation setting CVS1 instead of those learned under other two settings (*i.e.*, CVS2 and CVS3). This is because that the objective is to suggest those novel likely drug-target interactions, instead of focusing a new drug or a new target. Then, the recommended novel interactions are the top ranked unknown drug-target interaction pairs.

Table 5.5 shows the top 30 novel interactions predicted by NRLMF on the GPCR dataset. In this table, the DTIs are bolded to indicate that they exist in one or more of the reference databases. The third column of Table 5.5 shows the predicted interaction probability of a drug-target pair. For each pair, the databases containing it are listed

Table 5.5: The top 30 novel interactions predicted by NRLMF on GPCR dataset.

Rank	Drug	Target	Probability	Databases		
1	D00283	hsa1814	0.9181	C	D	M
2	D02358	hsa154	0.8828		D	
3	D04625	hsa154	0.8550		D	K
4	D02614	hsa154	0.8373			
5	D00227	hsa136	0.8370	C		
6	D01712	hsa136	0.7736		D	
7	D01352	hsa5731	0.7718			
8	D02250	hsa6751	0.7611			K
9	D02884	hsa136	0.7605			
10	D02354	hsa1814	0.7581			
11	D02147	hsa153	0.7500		D	M
12	D01871	hsa3269	0.7266			
13	D00371	hsa134	0.7064	C	D	K
14	D00371	hsa135	0.7012	C	D	K
15	D02725	hsa5737	0.6904			
16	D00682	hsa5739	0.6852			
17	D04006	hsa135	0.6803		D	K
18	D00049	hsa8843	0.6659		D	
19	D04006	hsa134	0.6606		D	K
20	D00604	hsa147	0.6603		D	
21	D00715	hsa1129	0.6584		D	K
22	D00503	hsa3356	0.6538			
23	D01103	hsa1129	0.6417			K
24	D02359	hsa153	0.6367			
25	D00079	hsa5731	0.6117	C	D	
26	D00765	hsa1128	0.6089			
27	D02340	hsa1812	0.6063		D	
28	D00442	hsa6753	0.5940			K
29	D00397	hsa1131	0.5829	C	D	K
30	D00095	hsa155	0.5797	C	D	K

in the last column of the table, where **C** is short for ChEMBL, **D** for DrugBank, **K** for KEGG, and **M** for Matador. For example, the highest ranked DTI is (D00283, hsa1814) with predicted probability 0.9181, which exists in the databases ChEMBL, DrugBank, and Metador. The results indicate that 67% of the recommendation (20 out of 30) are currently confirmed in at least one of the reference databases. Since these databases are still being updated as new DTIs are found, the fraction of new

Table 5.6: The fractions of true DTIs among the predicted top N ($N=10, 30, 50$) interactions under cross-validation setting CVS1.

Dataset	Method	Top 10	Top 30	Top 50
Nuclear Receptor	NetLapRLS	10%	23%	26%
	BLM-NII	30%	27%	16%
	WNN-GIP	0%	20%	14%
	KBMF2K	40%	30%	22%
	CMF	10%	20%	24%
	NRLMF	50%	43%	28%
GPCR	NetLapRLS	40%	40%	46%
	BLM-NII	70%	60%	58%
	WNN-GIP	30%	43%	38%
	KBMF2K	90%	53%	52%
	CMF	50%	40%	36%
	NRLMF	60%	67%	60%
Ion Channel	NetLapRLS	60%	47%	38%
	BLM-NII	30%	30%	34%
	WNN-GIP	30%	43%	48%
	KBMF2K	100%	83%	84%
	CMF	0%	0%	6%
	NRLMF	50%	33%	34%
Enzyme	NetLapRLS	70%	50%	40%
	BLM-NII	70%	60%	46%
	WNN-GIP	70%	50%	40%
	KBMF2K	70%	43%	28%
	CMF	20%	7%	4%
	NRLMF	90%	60%	44%

DTIs correctly predicted by NRLMF may increase in the future. This encouraging result that NRLMF can successfully detect quite a few novel interactions that are not in the GPCR dataset, implies that NRLMF is very effective in suggesting new true DTIs from sparse matrices consisted of very few DTIs.

Finally, Table 5.6 summarizes the fractions of true DTIs among the top N ($N = 10, 30, 50$) recommendations generated by various DTI methods, using the optimal parameters learned under the cross-validation setting CVS1. Observed that NRLMF is able to achieve consistently accurate recommendation results across all the datasets. For example, the fractions of true DTIs among the top 10 predicted interactions are

50%, 60%, 50%, and 90% for all datasets, respectively. Compared with other methods, NRLMF is able to achieve comparable or even better recommendation results across all the datasets. These observations indicate that the proposed algorithm is very effective for finding novel DTIs, thus it may help biologists or clinicians significantly reduce the cost of biological test.

5.5 Summary

This chapter introduces a computational approach, namely neighborhood regularized logistic matrix factorization (NRLMF), to predicting potential interactions between drugs and targets. The novelty of NRLMF lies in integrating logistic matrix factorization with neighborhood regularization for drug-target interaction recommendation. In NRLMF, we model the interaction probability for each drug-target pair using logistic matrix factorization. As the observed interacting drug-target pairs are experimentally verified, they are more trustworthy than the unknown pairs. We propose to assign higher importance levels to interaction pairs and lower importance levels to unknown pairs. In addition, we further improve the recommendation accuracy by neighborhood regularization, which considers the neighborhood influences from most similar drugs and most similar targets.

The optimization problem of NRLMF is solved using an alternating gradient descent optimization algorithm, the time complexity of which is $O(iter \cdot r \cdot m \cdot n)$, where $iter$ denotes the number of iterations. However, the time complexity of the solutions to the other two matrix factorization methods, *i.e.*, KBMF2K and CMF, are $O(iter \cdot (r \cdot m^3 + r \cdot n^3 + r^3))$ and $O(iter \cdot (r^2 \cdot (m + n)^2 + r^3 \cdot (m + n)))$, respectively. Therefore, NRLMF is more efficient than previous matrix factorization based DTI recommendation methods, *i.e.*, KBMF2K and CMF.

To evaluate the performance of NRLMF, we conducted extensive experiments on four benchmark datasets. The experimental results demonstrated that NRLMF usually outperformed five state-of-the-art methods under three different cross-validation

settings, in terms of AUC and AUPR. In addition, we confirmed the practical prediction ability of NRLMF by mapping with the latest version of four online biological databases, including ChEMBL, DrugBank, KEGG, and Matador.

Chapter 6

Conclusions and Future Work

This thesis presents the recommendation approaches proposed for three emerging application scenarios. Specifically, the following recommendation tasks have been investigated in this dissertation: (a) personalized location recommendation in location-based social networks, (b) top- N recommendation with users' implicit feedback, and (c) drug-target interaction recommendation for drug discovery. This chapter first concludes this thesis by highlighting the main contributions and then discusses the potential research directions for future work.

6.1 Summary of Contributions

The main contributions made in this thesis are as follows:

- **Personalized Location Recommendation.** The objective of personalized location recommender systems is to suggest a set of interesting locations that may be preferred by a particular user. Therefore, it is able to enhance users' experiences in location-based social networks. This thesis proposes a novel personalized recommendation approach, namely IRenMF (Instance-Region Neighborhood Matrix Factorization), to location recommendation by exploiting two levels of geographical neighborhood characteristics from the location perspective. By incorporating these two levels of neighborhood characteristics into

the learning of latent factors of users and locations, IRenMF has a more accurate modelling of users' preferences on locations. To solve the optimization problem of IRenMF, a novel alternating optimization algorithm has been proposed, with which IRenMF achieves stable recommendation accuracy. The experiments on real data show that IRenMF leads to significant improvements over classical matrix factorization approaches (*i.e.*, WRMF [HKV08, PZC⁺08] and BPRMF [RFGST09]) and other state-of-the-art location recommendation models (*i.e.*, GeoCF [YYLL11] and MGMMF [CYKL12]). On average, IRenMF outperforms WRMF, GeoCF, and MGMMF by 38.63%, 17.90%, and 116.49% respectively, in terms of Precision@10.

- **Top- N Recommendation with Users'Implicit Feedback.** In most application scenarios, the recommender systems are built based on users' implicit feedback. The recommendation task is usually formulated as a top- N recommendation problem. This thesis introduces an adaptive recommendation algorithm, named AdaBPR (Adaptive Boosting Personalized Ranking), which exploits boosting to improve the accuracy of top- N recommendation based on users' implicit feedback. AdaBPR creates an ensemble of a set of homogeneous component recommenders built by the same collaborative filtering algorithm, to predict users' preferences on items. Empirical results on three real datasets demonstrate the effectiveness of the proposed algorithm in comparison with a few strong baseline methods. On average, the proposed AdaBPR model outperforms other competitors BPRMF [RFGST09], GBPR [PC13b], and AdaMF [WSZ14] by 17.86%, 12.42%, and 18.48%, respectively, in terms of NDCG@10.
- **Drug-Target Interaction Recommendation.** The computational approaches developed for recommending potential drug-target interactions can help to direct the experimental verification and thus significantly improve the drug discovery efficiency. This thesis extends the personalized recommendation method

(*i.e.*, logistic matrix factorization [Joh14]) for the drug-target interaction recommendation task and proposes a novel recommendation method, namely NRLMF (Neighborhood Regularized Logistic Matrix Factorization). The novelty of NRLMF comes from integrating logistic matrix factorization with neighborhood regularization to predict the interaction probability of a given drug-target pair. Specifically, both drugs and targets are mapped into a shared latent space, and the drug-target interactions are modeled using the linear combinations of the drug-specific and target-specific latent vectors. In addition, higher importance level is assigned to the positive observations (*i.e.*, observed interacting drug-target pairs), while lower level is for negative observations (*i.e.*, unknown pairs). Moreover, the neighborhood regularization based on the drug similarities and target similarities is utilized to further improve the recommendation ability of the proposed model. To evaluate the performances of NRLMF, an extensive set of experiments have been performed on four benchmark datasets, compared with five state-of-the-art DTI recommendation methods. The promising results further validate the empirical efficacy of the proposed algorithm. Under the cross-validation setting CVS1 (*i.e.*, cross-validation on drug-target pairs), NRLMF achieves 20.97%, 29.18%, 18.87%, 29.38%, and 4.73% better average AUPR values than other baseline methods NetLapRLS [XWZW10], KBMF2K [Gön12], BLM-NII [MKY⁺13], WNN-GIP [vLM13], and CMF [ZDMZ13] respectively.

6.2 Future Work

This section lists the potential directions for future research work, which mainly focus on two aspects, *i.e.*, the model improvement and potential extensions.

6.2.1 Personalized Location Recommendation

The location recommendation model proposed in this thesis can be improved from the following aspects. Firstly, the auxiliary information associated with locations (*e.g.*,

opening hours and services provided) will be incorporated to improve the recommendation accuracy. These auxiliary information can be treated as the observed features of locations and used to measure the similarities between a pair of neighborhood locations. The modeling of instance-level neighborhood characteristics in IRenMF can be improved via considering both the similarity defined based on observed features of locations and the similarity defined based on users' check-in activities at locations. Secondly, more complex region structures (*e.g.*, hierarchical structure) associated with users' check-in data will be exploited to improve the modeling of region-level geographical characteristics. In location-based social networks, the locations are usually organized using the hierarchical graph structure (see Figure 6.1 as an example), which has been demonstrated to be effective in modeling users' preferences in [LLAM13]. This hierarchical structure can also be incorporated into the IRenMF

Figure 6.1: An example of the category tree associated with locations.

model by using hierarchical group lasso regularization [KX10, JOB10]. Thirdly, as shown in Figure 6.2, users may leave tips to locations to show their comments and suggestions. These tips include contextual information about users' preferences on locations. The recent work [WB11, ML13] provides potential ideas for jointly modeling users' preferences and the latent topics of users' tips. In addition, the time-aware location recommendation has been studied in recent studies [YCM⁺13, GTHL13, YCS14]. The objective of this recommendation task is to suggest a list of locations for a user to visit at a given time. In the future, the proposed IRenMF model will be extended for time-aware location recommendation tasks, considering the temporal information

Figure 6.2: An example of the tips given by foursquare users.

associated with users' check-ins. The tensor factorization technique, which has been successfully used for context-aware recommendation [KABO10], should be a potential solution to this recommendation problem. A simple extension for IRenMF is to replace the weighted regularized matrix factorization model with the weighted tensor factorization model proposed in [KBB11].

Moreover, in location-based social networks, users are connected with their online social friends. The social relationship between users have been demonstrated to be beneficial to different recommendation problems [KLM10]. It has also been found that users' check-in activities in location-based social networks are usually influenced by their social neighbors [CML11,SNM11]. The recent studies [YYL10,YYLL11,BZM12,ZC15] indicate that the social influences from online friends are useful in improving the location recommendation accuracy. The key of integrating social networks among users into the location recommendation tasks is to explore the underlying structure of users' social networks and the influence on the preferences of two social friends. The social regularization [MZL⁺11] and social voting [YSGL12] strategies are two potential approaches that can be used to extend IRenMF to incorporate social influences for location recommendation.

In addition, the locations and activities are naturally related. Besides the recommendations of interesting locations, users are also interested in the recommendations of interesting activities. For example, they would wonder where to go for food, and meanwhile they also want to know what they can do at a particular location. The first question corresponds to the location recommendation tasks with an activity query, and the second question corresponds to the activity recommendation problem with a location query. In location-based social networks, such as Foursquare, the category information of a particular location and the tips associated to it clearly specify users' activities at that location. Thus, it enables us to study the correlations between users' activities and the locations. The recent work [ZZXY10, ZCZ⁺10, LLL11] begins to collectively explore activity recommendation and location recommendation. The location and activity recommendation tasks are integrated in a unified knowledge mining process. The key step of this recommendation task is to construct the user-location relationship, which is usually represented using a location-activity matrix, where the rows stand for locations and columns stand for activities. However, this location-activity matrix is very sparse, as many locations have no indications of activities. To address this issue, the observed features and the correlation between activities can be used to contribute to the inference of location-activity relationships [ZZXY10, ZCZ⁺10].

6.2.2 Top- N Recommendation with Users' Implicit Feedback

The proposed boosting recommendation algorithm, *i.e.*, AdaBPR, creates an ensemble of a set of homogenous component recommenders for better top- N recommendation accuracy. Currently, in the AdaBPR framework, the component recommenders are created by optimizing the ranking metric AUC (Area Under the ROC Curve). The future work will explore constructing component recommenders that optimize more complex ranking metrics, *e.g.*, MAP (Mean Average Precision) [SKB⁺12a] and MRR (Mean Reciprocal Rank) [SKB⁺12b], to improve the recommendation accuracy. However, the direct optimization of MAP and MRR is usually very time-consuming.

An alternative solution is to optimize the Weighted Approximately Ranked Pairwise (WARP) loss, which has been demonstrated to be effective for top- N recommendation tasks, *e.g.*, personalized tweet recommendation [HDD13] and personalized music recommendation [WWY13]. The WARP loss focuses more on the top- N positions in the ranked item list, which are usually cared by users. Different choices of the parameters used in the definition of WARP result in the optimization of different ranking metrics, *e.g.*, AUC, MAP, precision@ N . The optimization problem for WARP loss is usually solved by using a stochastic gradient descent algorithm based on stochastically sampling. The optimization process can be further accelerated by the truncated sample scheme proposed in [LL14]. Another interesting direction for future work is to extend AdaBPR to incorporate contextual information (*e.g.*, spatial and temporal information) of each user-item interaction for recommendation. In this scenario, a component recommender may be built by using factorization machine [Ren10] to model a specific kind of context interaction. In addition, the proposed AdaBPR framework will be extended to build live recommender systems that can evolve on-the-fly over time. For this recommendation task, the component recommenders may be created using the online AUC maximization (OAM) method proposed in [ZJYH11], which can directly optimize the ranking metric AUC under the online learning settings.

6.2.3 Drug-Target Interaction Recommendation

In this thesis, the proposed drug-target interaction recommendation approach, *i.e.*, NRMF, only considers one type of similarity from drugs and targets. The drug similarities are calculated based on the chemical structures of compounds, and the target similarities are computed based on the amino acid sequences of the target proteins. However, there are various types of side information associate with drugs and targets. Thus, multiple types of drug similarities and target similarities can be computed, for example, the chemical structure similarity and ATC similarity for drugs and the genomic sequence similarity, protein-protein interaction network similarity, and gene ontology similarities for targets. Moreover, previous work [ZDMZ13] has demonstrated

that the drug-target interaction recommendation accuracy can be significantly improved by considering multiple types of similarities defined based on different kinds of side information. As a part of future work, NRLMF will be extended to exploit multiple kinds of similarities via coupling the logistic matrix factorization technique with the multiple kernel learning technique [GA11]. Moreover, the drug-target interaction recommendation is indeed a top- N recommendation problem, which can be solved by ranking-based matrix factorization models. Another potential direction for future work is to utilize the proposed AdaBPR model for predicting potential drug-target interactions. For this task, the AdaBPR model is learned by optimizing the loss function defined based on the ranking accuracy for all drug-target interaction pairs.

Appendix A

Proof for Theorems

A.1 The Proof for Theorem 1

Take the sub-gradient of $\psi(\mathbf{L})$ over $\mathbf{L}_{(g)}^d$:

$$\frac{\partial \psi(\mathbf{L})}{\partial \mathbf{L}_{(g)}^d} = \tau(\mathbf{L}_{(g)}^d - \mathbf{Z}_{(g)}^d) + \lambda_2 \mathbf{L}_{(g)}^d + \lambda_3 \omega_g \frac{\partial \|\mathbf{L}_{(g)}^d\|_2}{\partial \mathbf{L}_{(g)}^d}, \quad (\text{A.1})$$

where $\frac{\partial \|\mathbf{L}_{(g)}^d\|_2}{\partial \mathbf{L}_{(g)}^d} = \frac{\mathbf{L}_{(g)}^d}{\|\mathbf{L}_{(g)}^d\|_2}$, if $\mathbf{L}_{(g)}^d \neq \mathbf{0}$, and $\frac{\partial \|\mathbf{L}_{(g)}^d\|_2}{\partial \mathbf{L}_{(g)}^d} = \{\alpha \in \mathbb{R}^{n_{(g)} \times 1} \mid \|\alpha\|_2 \leq 1\}$, otherwise.

When $\mathbf{L}_{(g)}^d \neq \mathbf{0}$, plugging $\frac{\partial \|\mathbf{L}_{(g)}^d\|_2}{\partial \mathbf{L}_{(g)}^d} = \frac{\mathbf{L}_{(g)}^d}{\|\mathbf{L}_{(g)}^d\|_2}$ into Eq. (A.1), we have

$$(\tau + \lambda_2 + \frac{\lambda_3 \omega_g}{\|\mathbf{L}_{(g)}^d\|_2}) \mathbf{L}_{(g)}^d = \tau \mathbf{Z}_{(g)}^d. \quad (\text{A.2})$$

Taking the vector ℓ_2 norm on both sides of Eq. (A.2):

$$\|\mathbf{L}_{(g)}^d\|_2 = \frac{\tau}{\tau + \lambda_2} \|\mathbf{Z}_{(g)}^d\|_2 - \frac{\lambda_3 \omega_g}{\tau + \lambda_2} \quad (\text{A.3})$$

Since $\mathbf{L}_{(g)}^d \neq \mathbf{0}$, the ℓ_2 norm of $\mathbf{L}_{(g)}^d$ should be a positive real value, which requires $\|\mathbf{Z}_{(g)}^d\|_2 > \frac{\lambda_3 \omega_g}{\tau}$. If the condition $\|\mathbf{Z}_{(g)}^d\|_2 > \frac{\lambda_3 \omega_g}{\tau}$ holds, by plugging Eq. (A.3) into

Eq. (A.2), we can get

$$\mathbf{L}_{(g)}^d = \frac{\mathbf{Z}_{(g)}^d (\|\mathbf{Z}_{(g)}^d\|_2 - \frac{\lambda_3 \omega_g}{\tau})}{(1 + \frac{\lambda_2}{\tau}) \|\mathbf{Z}_{(g)}^d\|_2}. \quad (\text{A.4})$$

If $\|\mathbf{Z}_{(g)}^d\|_2 \leq \frac{\lambda_3 \omega_g}{\tau}$, plugging $\mathbf{L}_{(g)}^d = \mathbf{0}$ into Eq. (A.1), we obtain

$$\frac{\partial \|\mathbf{L}_{(g)}^d\|_2}{\partial \mathbf{L}_{(g)}^d} = \frac{\tau}{\lambda_3 \omega_g} \mathbf{Z}_{(g)}^d. \quad (\text{A.5})$$

As $\frac{\tau}{\lambda_3 \omega_g} \mathbf{Z}_{(g)}^d \leq 1$, $\frac{\tau}{\lambda_3 \omega_g} \mathbf{Z}_{(g)}^d$ is a valid sub-gradient of $\|\mathbf{L}_{(g)}^d\|_2$ over $\mathbf{L}_{(g)}^d$, when $\mathbf{L}_{(g)}^d = \mathbf{0}$. $\square$

A.2 The Proof for Theorem 2

Set $Z_t = \sum_{(u,i) \in \mathcal{D}} \beta_u e^{-E[\pi(u,i,f^t)]} = \sum_{(u,i) \in \mathcal{D}} \beta_u e^{-E[\pi(u,i,f^{t-1} + \alpha_t h^{(t)})]}$, then we can get

$$\begin{aligned} Z_t &= \sum_{(u,i) \in \mathcal{D}} \beta_u e^{-E[\pi(u,i,f^{t-1})]} e^{-\alpha_t E[\pi(u,i,h^{(t)})]} e^{-\delta_{ui}^t} \\ &\leq \sum_{(u,i) \in \mathcal{D}} \beta_u e^{-\delta_{min}^t} e^{-E[\pi(u,i,f^{t-1})]} e^{-\alpha_t E[\pi(u,i,h^{(t)})]} \\ &= e^{-\delta_{min}^t} \sum_{(u,i) \in \mathcal{D}} \beta_u e^{-E[\pi(u,i,f^{t-1})]} e^{-\alpha_t E[\pi(u,i,h^{(t)})]} \\ &= e^{-\delta_{min}^t} Z_{t-1} \sum_{(u,i) \in \mathcal{D}} \frac{e^{-E[\pi(u,i,f^{t-1})]}}{Z_{t-1}} e^{-\alpha_t E[\pi(u,i,h^{(t)})]} \\ &= e^{-\delta_{min}^t} Z_{t-1} \sum_{(u,i) \in \mathcal{D}} w_{ui}^t e^{-\alpha_t E[\pi(u,i,h^{(t)})]}, \end{aligned} \quad (\text{A.6})$$

where $\delta_{ui}^t = E[\pi(u,i,f^{t-1} + \alpha_t h^{(t)})] - E[\pi(u,i,f^{t-1})] - \alpha_t E[\pi(u,i,h^{(t)})]$, $\delta_{min}^t = \min_{(u,i) \in \mathcal{D}} \delta_{ui}^t$, and $w_{ui}^t = \frac{e^{-E[\pi(u,i,f^{t-1})]}}{Z_{t-1}}$. Moreover, if $E[\pi(u,i,h^{(t)})] \in [-1, 1]$ then,

$$Z_t \leq \underbrace{e^{-\delta_{min}^t} Z_{t-1} \sum_{(u,i) \in \mathcal{D}} w_{ui}^t \left(\frac{1 + E[\pi(u,i,h^{(t)})]}{2} e^{-\alpha_t} + \frac{1 - E[\pi(u,i,h^{(t)})]}{2} e^{\alpha_t} \right)}_{L_t}. \quad (\text{A.7})$$

Take the gradient of L_t over α_t :

$$\frac{\partial L_t}{\partial \alpha_t} = \sum_{(u,i) \in \mathcal{D}} w_{ui}^t \left(\frac{1 - E[\pi(u, i, h^{(t)})]}{2} e^{\alpha_t} - \frac{1 + E[\pi(u, i, h^{(t)})]}{2} e^{-\alpha_t} \right). \quad (\text{A.8})$$

By setting $\frac{\partial L_t}{\partial \alpha_t} = 0$, we can get

$$\alpha_t = \frac{1}{2} \ln \frac{\sum_{(u,i) \in \mathcal{D}} w_{ui}^t (1 + E[\pi(u, i, h^{(t)})])}{\sum_{(u,i) \in \mathcal{D}} w_{ui}^t (1 - E[\pi(u, i, h^{(t)})])}. \quad (\text{A.9})$$

Set $\varphi(t) = \sum_{(u,i) \in \mathcal{D}} w_{ui}^t E[\pi(u, i, h^{(t)})]$ and $\phi(t) = \frac{1+\varphi(t)}{2}$, then $e^{\alpha_t} = \sqrt{\frac{\phi(t)}{1-\phi(t)}}$. Eq. (A.7) is converted to:

$$\begin{aligned} Z_t &\leq e^{-\delta_{min}^t} Z_{t-1} \left(\phi(t) \sqrt{\frac{1-\phi(t)}{\phi(t)}} + (1-\phi(t)) \sqrt{\frac{\phi(t)}{1-\phi(t)}} \right) \\ &= e^{-\delta_{min}^t} Z_{t-1} \sqrt{4\phi(t)(1-\phi(t))} \leq Z_1 \prod_{k=2}^t e^{-\delta_{min}^k} \sqrt{4\phi(k)(1-\phi(k))}. \end{aligned} \quad (\text{A.10})$$

For Z_1 , we can get the following equation:

$$\begin{aligned} Z_1 &\leq e^{-\delta_{min}^1} \left(\sum_{(u,i) \in \mathcal{D}} \beta_u \right) \left[\sum_{(u,i) \in \mathcal{D}} \frac{\beta_u}{\sum_{(u,i) \in \mathcal{D}} \beta_u} e^{-\alpha_1 E[\pi(u, i, h^{(1)})]} \right] \\ &\leq e^{-\delta_{min}^1} \left(\sum_{(u,i) \in \mathcal{D}} \beta_u \right) \sum_{(u,i) \in \mathcal{D}} w_{ui}^1 \left[\frac{1 + E[\pi(u, i, h^{(1)})]}{2} e^{-\alpha_1} + \frac{1 - E[\pi(u, i, h^{(1)})]}{2} e^{\alpha_1} \right] \\ &= e^{-\delta_{min}^1} \left(\sum_{(u,i) \in \mathcal{D}} \beta_u \right) \sqrt{4\phi(1)(1-\phi(1))}. \end{aligned} \quad (\text{A.11})$$

By plugging Eq. (A.11) into Eq. (A.10), we can obtain

$$Z_t \leq \left(\sum_{(u,i) \in \mathcal{D}} \beta_u \right) \prod_{k=1}^t e^{-\delta_{min}^k} \sqrt{4\phi(k)(1-\phi(k))}. \quad (\text{A.12})$$

Therefore,

$$\begin{aligned}
 \frac{1}{|U|} \sum_{u \in U} \frac{1}{|V_u^+|} \sum_{i \in V_u^+} E[\pi(u, i, f)] &\geq \frac{1}{|U|} \sum_{u \in U} \beta_u \left(1 - e^{-E[\pi(u, i, f^t)]}\right) \\
 &= \frac{1}{|U|} \left(\sum_{u \in U} \beta_u - \sum_{u \in U} \beta_u e^{-E[\pi(u, i, f^t)]} \right) \\
 &= \frac{\sum_{u \in U} \beta_u}{|U|} \left(1 - \prod_{k=1}^t e^{-\delta_{min}^k \sqrt{1 - \varphi(k)^2}} \right).
 \end{aligned} \tag{A.13}$$

□

Appendix B

List of Publications

- **Yong Liu**, Min Wu, Chunyan Miao, Peilin Zhao, Xiao-Li Li. Neighborhood Regularized Logistic Matrix Factorization for Drug-Target Interaction Prediction. **PLoS Computational Biology**, 12(2): e1004760, 2016.
- **Yong Liu**, Peilin Zhao, Aixin Sun, Chunyan Miao. A Boosting Algorithm for Item Recommendation with Implicit Feedback. Proceedings of the 24th International Joint Conference on Artificial Intelligence (**IJCAI 2015**), Pages 1792-1798. Buenos Aires, Argentina, July 25-31, 2015.
- **Yong Liu**, Wei Wei, Aixin Sun, Chunyan Miao. Exploiting Geographical Neighborhood Characteristics for Location Recommendation. Proceedings of the 23rd ACM International Conference on Information and Knowledge Management (**CIKM 2014**), Pages 739-748. Shanghai, China, November 3-7, 2014.
- Jianping Mei, Han Yu, **Yong Liu**, Zhiqi Shen, Chunyan Miao. A Social Trust Model Considering Trustees' Influence. Proceedings of 17th International Conference on Principles and Practice of Multi-Agent Systems(**PRIMA 2014**), Pages 357-364. Gold Coast, Australia. December 1-5, 2014.
- Longke Hu, Aixin Sun, **Yong Liu**. Your Neighbors Affect Your Ratings: On Geographical Neighborhood Influence to Rating Prediction. Proceedings of the

37th International ACM SIGIR Conference on Research and Development in Information Retrieval (**SIGIR 2014**), Pages 345-354. Gold Coast, Australia, July 6-11, 2014.

- Xin Liu, **Yong Liu**, Karl Aberer, Chunyan Miao. Personalized Point-of-Interest Recommendation by Mining Users' Preference Transition. Proceedings of the 22nd ACM International Conference on Information and Knowledge Management (**CIKM 2013**), Pages 733-738. San Francisco, CA, USA. October 27-November 1, 2013.

Bibliography

- [AS11] Frederik Armknecht and Thorsten Strufe. An efficient distributed privacy-preserving recommendation system. In *Proceedings of the 10th IFIP Annual Mediterranean Ad Hoc Networking Workshop*, pages 65–70, 2011.
- [AT05] Gediminas Adomavicius and Alexander 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.
- [BC12] Suhrid Balakrishnan and Sumit Chopra. Collaborative ranking. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining*, pages 143–152, 2012.
- [BEF13] Jim Blomo, Martin Ester, and Marty Field. Recsys challenge 2013. In *Proceedings of the 7th ACM Conference on Recommender Systems*, pages 489–490, 2013.
- [BHK98] John S Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pages 43–52, 1998.
- [BK07] Robert M Bell and Yehuda Koren. Lessons from the netflix prize challenge. *ACM SIGKDD Explorations Newsletter*, 9(2):75–79, 2007.

- [BKG⁺05] Arindam Banerjee, Chase Krumpelman, Joydeep Ghosh, Sugato Basu, and Raymond J Mooney. Model-based overlapping clustering. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pages 532–537, 2005.
- [BMEWL11] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*, pages 591–596, 2011.
- [BS11] Betim Berjani and Thorsten Strufe. A recommendation system for spots in location-based online social networks. In *Proceedings of the 4th ACM Workshop on Social Network Systems*, 2011.
- [BY09] Kevin Bleakley and Yoshihiro Yamanishi. Supervised prediction of drug–target interactions using bipartite local models. *Bioinformatics*, 25(18):2397–2403, 2009.
- [BZM12] Jie Bao, Yu Zheng, and Mohamed F Mokbel. Location-based and preference-aware recommendation using sparse geo-social networking data. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pages 199–208, 2012.
- [BZWM15] Jie Bao, Yu Zheng, David Wilkie, and Mohamed Mokbel. Recommendations in location-based social networks: a survey. *GeoInformatica*, 19(3):525–565, 2015.
- [CDW12] Bin Chen, Ying Ding, and David J Wild. Assessing drug target association using semantic linked data. *PLoS Computational Biology*, 8(7):e1002574, 2012.

- [CG99] Yung-Hsin Chien and Edward I George. A bayesian model for collaborative filtering. In *Proceedings of the 7th International Workshop on Artificial Intelligence and Statistics*, 1999.
- [CGD⁺09] Jilin Chen, Werner Geyer, Casey Dugan, Michael Muller, and Ido Guy. Make new friends, but keep the old: recommending people on social networking sites. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 201–210, 2009.
- [Che13] Xi Chen. *Learning with sparsity: structures, optimization and applications*. PhD thesis, Carnegie Mellon University, 2013.
- [CHW01] Sonny Han Seng Chee, Jiawei Han, and Ke Wang. Rectree: An efficient collaborative filtering method. *Data Warehousing and Knowledge Discovery*, pages 141–151, 2001.
- [CKT10] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the 4th ACM Conference on Recommender Systems*, pages 39–46, 2010.
- [CLH⁺13] Murat Can Cobanoglu, Chang Liu, Feizhuo Hu, Zoltán N. Oltvai, and Ivet Bahar. Predicting drug–target interactions using probabilistic matrix factorization. *Journal of Chemical Information and Modeling*, 53(12):3399–3409, 2013.
- [CLJ⁺12] Feixiong Cheng, Chuang Liu, Jing Jiang, Weiqiang Lu, Weihua Li, Guixia Liu, Weixing Zhou, Jin Huang, and Yun Tang. Prediction of drug-target interactions and drug repositioning via network-based inference. *PLoS Computational Biology*, 8(5):e1002503, 2012.
- [CML11] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. Friendship and mobility: User movement in location-based social networks. In *Proceedings of*

- the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1082–1090, 2011.
- [CSHS12] Justin Cranshaw, Raz Schwartz, Jason Hong, and Norman Sadeh. The livelihoods project: Utilizing social media to understand the dynamics of a city. In *Proceedings of the 6th International AAAI Conference on Weblogs and Social Media*, page 58, 2012.
- [CXZ⁺14] Chen Cheng, Fen Xia, Tong Zhang, Irwin King, and Michael R. Lyu. Gradient boosting factorization machines. In *Proceedings of the 8th ACM Conference on Recommender Systems*, pages 265–272, 2014.
- [CYKL12] Chen Cheng, Haiqin Yang, Irwin King, and Michael R Lyu. Fused matrix factorization with geographical and social influence in location-based social networks. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, pages 17–23, 2012.
- [CYLK13] Chen Cheng, Haiqin Yang, Michael R Lyu, and Irwin King. Where you like to go next: Successive point-of-interest recommendation. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pages 2605–2611, 2013.
- [DH02] Nigel Duffy and David Helmbold. Boosting methods for regression. *Machine Learning*, 47(2-3):153–200, 2002.
- [DHS11] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.
- [DK04] Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems*, 22(1):143–177, 2004.

- [DLS11] Liang Du, Xuan Li, and Yi-Dong Shen. User graph regularized pairwise matrix factorization for item recommendation. In *Advanced Data Mining and Applications*, pages 372–385. Springer, 2011.
- [DTMZ13] Hao Ding, Ichigaku Takigawa, Hiroshi Mamitsuka, and Shanfeng Zhu. Similarity-based machine learning methods for predicting drug–target interactions: a brief review. *Briefings in Bioinformatics*, page bbt056, 2013.
- [DZ15] Yan-Fen Dai and Xing-Ming Zhao. A survey on the computational approaches to identify drug targets in the postgenomic era. *BioMed research international*, 2015, 2015.
- [FS95] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Computational Learning Theory*, pages 23–37, 1995.
- [GA11] Mehmet Gönen and Ethem Alpaydm. Multiple kernel learning algorithms. *The Journal of Machine Learning Research*, 12:2211–2268, 2011.
- [GBB⁺12] Anna Gaulton, Louisa J. Bellis, A. Patricia Bento, Jon Chambers, Mark Davies, Anne Hersey, Yvonne Light, Shaun McGlinchey, David Michalovich, Bissan Al-Lazikani, and John P. Overington. ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Research*, 40(D1):D1100–D1107, 2012.
- [GJP14] Vladimir Gligorijević, Vuk Janjić, and Nataša Pržulj. Integration of molecular network data reconstructs gene ontology. *Bioinformatics*, 30(17):i594–i600, 2014.

- [GK14] Mehmet Gonen and Samuel Kaski. Kernelized bayesian matrix factorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(10):2047–2060, 2014.
- [GKD⁺08] Stefan Günther, Michael Kuhn, Mathias Dunkel, Monica Campillos, Christian Senger, Evangelia Petsalaki, Jessica Ahmed, Eduardo Garcia Urdiales, Andreas Gewiess, Lars Juhl Jensen, Reinhard Schneider, Roman Skoblo, Robert B. Russell, Philip E. Bourne, Peer Bork, and Robert Preissner. Supertarget and matador: resources for exploring drug-target relationships. *Nucleic Acids Research*, 36(suppl 1):D919–D922, 2008.
- [GM05] Thomas George and Srujana Merugu. A scalable collaborative filtering framework based on co-clustering. In *Proceedings of the 5th IEEE International Conference on Data Mining*, pages 625–628, 2005.
- [Gön12] Mehmet Gönen. Predicting drug–target interactions from chemical and genomic kernels using bayesian matrix factorization. *Bioinformatics*, 28(18):2304–2310, 2012.
- [GTHL13] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. Exploring temporal effects for location recommendation on location-based social networks. In *Proceedings of the 7th ACM Conference on Recommender Systems*, pages 93–100, 2013.
- [HAX⁺12] TaeHyun Hwang, Gowtham Atluri, MaoQiang Xie, Sanjoy Dey, Changjin Hong, Vipin Kumar, and Rui Kuang. Co-clustering phenotype–genome for phenotype classification and disease gene discovery. *Nucleic Acids Research*, 40(19):e146, 2012.
- [HDD13] Liangjie Hong, Aziz S Doumith, and Brian D Davison. Co-factorization machines: modeling user interests and predicting individual decisions

- in twitter. In *Proceedings of the 6th ACM International Conference on Web Search and Data Mining*, pages 557–566, 2013.
- [HKBR99] Jonathan L Herlocker, Joseph A Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 230–237, 1999.
- [HKV08] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Proceedings of the 8th IEEE International Conference on Data Mining*, pages 263–272, 2008.
- [HNV06] Tzvetan Horozov, Nitya Narasimhan, and Venu Vasudevan. Using location for personalized poi recommendations in mobile environments. *International Symposium on Applications and the Internet*, 2006.
- [Hof01] Thomas Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1-2):177–196, 2001.
- [Hof03] Thomas Hofmann. Collaborative filtering via gaussian probabilistic latent semantic analysis. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pages 259–266, 2003.
- [Hof04] Thomas Hofmann. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems*, 22(1):89–115, 2004.
- [HOGK03] Masahiro Hattori, Yasushi Okuno, Susumu Goto, and Minoru Kanehisa. Development of a chemical structure comparison method for integrated analysis of chemical and genomic information in the metabolic pathways. *Journal of the American Chemical Society*, 125(39):11853–11865, 2003.

- [HP99] Thomas Hofmann and Jan Puzicha. Latent class models for collaborative filtering. In *International Joint Conference on Artificial Intelligence*, volume 16, pages 688–693, 1999.
- [HSL14] Longke Hu, Aixin Sun, and Yong Liu. Your neighbors affect your ratings: on geographical neighborhood influence to rating prediction. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 345–354, 2014.
- [HTFF09] Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer-Verlag, 2009.
- [Hua11] Junzhou Huang. *Structured sparsity: theorems, algorithms and applications*. PhD thesis, Rutgers, The State University of New Jersey, 2011.
- [HZ10] Junzhou Huang and Tong Zhang. The benefit of group sparsity. *The Annals of Statistics*, 38(4):1978–2004, 2010.
- [HZM11] Junzhou Huang, Tong Zhang, and Dimitris Metaxas. Learning with structured sparsity. *The Journal of Machine Learning Research*, 12:3371–3412, 2011.
- [JCS04] Rong Jin, Joyce Y Chai, and Luo Si. An automatic weighting scheme for collaborative filtering. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 337–344, 2004.
- [JNG⁺13] Xiaotian Jiang, Zhendong Niu, Jiamin Guo, Ghulam Mustafa, Zihan Lin, Baomi Chen, and Qian Zhou. Novel boosting frameworks to improve the performance of collaborative filtering. In *Proceedings of the 5th Asian Conference on Machine Learning*, pages 87–99, 2013.

- [JOB10] Rodolphe Jenatton, Guillaume Obozinski, and Francis Bach. Structured sparse principal component analysis. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, 2010.
- [Joh14] Christopher C. Johnson. Logistic matrix factorization for implicit feedback data. *NIPS 2014 Workshop on Distributed Machine Learning and Matrix Computations*, 2014.
- [JOV09] Laurent Jacob, Guillaume Obozinski, and Jean-Philippe Vert. Group lasso with overlap and graph lasso. In *Proceedings of the 26th International Conference on Machine Learning*, pages 433–440, 2009.
- [JV08] Laurent Jacob and Jean-Philippe Vert. Protein-ligand interaction prediction: an improved chemogenomics approach. *Bioinformatics*, 24(19):2149–2156, 2008.
- [KABO10] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the 4th ACM conference on Recommender Systems*, pages 79–86, 2010.
- [KAP⁺12] Bhargav Kanagal, Amr Ahmed, Sandeep Pandey, Vanja Josifovski, Jeff Yuan, and Lluís García-Pueyo. Supercharging recommender systems using taxonomies for learning user purchase behavior. *Proceedings of the VLDB Endowment*, 5(10):956–967, 2012.
- [KBB11] Alexandros Karatzoglou, Linas Baltrunas, and Matthias Böhmer. Collaborative context-aware preference learning. In *NIPS Workshop on Choice Models and Preference Learning*, 2011.
- [KBS13] Alexandros Karatzoglou, Linas Baltrunas, and Yue Shi. Learning to rank for recommender systems. In *Proceedings of the 7th ACM Conference on Recommender Systems*, pages 493–494, 2013.

- [KBV09] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.
- [Ken09] Terry Kenakin. *A pharmacology primer: theory, application and methods*. Academic Press, 2009.
- [KGH⁺06] Minoru Kanehisa, Susumu Goto, Masahiro Hattori, Kiyoko F. Aoki-Kinoshita, Masumi Itoh, Shuichi Kawashima, Toshiaki Katayama, Michihiro Araki, and Mika Hirakawa. From genomics to chemical genomics: new developments in kegg. *Nucleic Acids Research*, 34(suppl 1):D354–D357, 2006.
- [KGS⁺12] Minoru Kanehisa, Susumu Goto, Yoko Sato, Miho Furumichi, and Mao Tanabe. Kegg for integration and interpretation of large-scale molecular data sets. *Nucleic Acids Research*, 40(gkr988):D109–D114, 2012.
- [KLB⁺09] Sarah L Kinnings, Nina Liu, Nancy Buchmeier, Peter J Tonge, Lei Xie, and Philip E Bourne. Drug discovery using chemical systems biology: repositioning the safe medicine comtan to treat multi-drug and extensively drug resistant tuberculosis. *PLoS Computational Biology*, 5(7):e1000423, 2009.
- [KLJ⁺11] Craig Knox, Vivian Law, Timothy Jewison, Philip Liu, Son Ly, Alex Frolkis, Allison Pon, Kelly Banco, Christine Mak, Vanessa Neveu, Yannick Djoumbou, Roman Eisner, An Chi Guo, and David S. Wishart. Drugbank 3.0: a comprehensive resource for ‘omics’ research on drugs. *Nucleic Acids Research*, 39(suppl 1):D1035–D1041, 2011.
- [KLM10] Irwin King, Michael R Lyu, and Hao Ma. Introduction to social recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, pages 1355–1356, 2010.

- [KLZ12] Su Mon Kywe, Ee-Peng Lim, and Feida Zhu. A survey of recommender systems in twitter. In *Social Informatics*, pages 420–433. Springer, 2012.
- [KNK13] Santosh Kabbur, Xia Ning, and George Karypis. Fism: factored item similarity models for top-n recommender systems. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 659–667, 2013.
- [Kor08] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 426–434, 2008.
- [Kor09] Yehuda Koren. The bellkor solution to the netflix grand prize. *Netflix Prize Documentation*, 81, 2009.
- [KRA⁺07] Michael J Keiser, Bryan L Roth, Blaine N Armbruster, Paul Ernsberger, John J Irwin, and Brian K Shoichet. Relating protein pharmacology by ligand chemistry. *Nature biotechnology*, 25(2):197–206, 2007.
- [KRSJ14] Berk Kapicioglu, David S Rosenberg, Robert E Schapire, and Tony Jebara. Collaborative ranking for local preferences. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics*, pages 466–474, 2014.
- [KS11] Yehuda Koren and Joe Sill. Ordrec: an ordinal model for predicting personalized item rating distributions. In *Proceedings of the 5th ACM Conference on Recommender Systems*, pages 117–124, 2011.
- [KSX09] Mladen Kolar, Le Song, and Eric P Xing. Sparsistent learning of varying-coefficient models with structural changes. In *Advances in Neural Information Processing Systems*, pages 1006–1014, 2009.

- [KX10] Seyoung Kim and Eric P Xing. Tree-guided group lasso for multi-task regression with structured sparsity. In *Proceedings of the 27th International Conference on Machine Learning*, pages 543–550, 2010.
- [LAJ11] Yvonne Y Li, Jianghong An, and SJ Jones. A computational approach to finding novel targets for existing drugs. *PLoS Computational Biology*, 7(9):e1002139–e1002139, 2011.
- [LBK⁺14] Joonseok Lee, Samy Bengio, Seungyeon Kim, Guy Lebanon, and Yoram Singer. Local collaborative ranking. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 85–96, 2014.
- [LFYX13] Bin Liu, Yanjie Fu, Zijun Yao, and Hui Xiong. Learning geographical preferences for point-of-interest recommendation. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1043–1051, 2013.
- [LG08] Yves Landry and Jean-Pierre Gies. Drugs and their molecular targets: an updated overview. *Fundamental & clinical pharmacology*, 22(1):1–18, 2008.
- [LHWZ13] Jing Lu, Steven Hoi, Jialei Wang, and Peilin Zhao. Second order online collaborative filtering. In *Proceedings of the 5th Asian Conference on Machine Learning*, pages 325–340, 2013.
- [LHZC10] Yanen Li, Jia Hu, ChengXiang Zhai, and Ye Chen. Improving one-class collaborative filtering by incorporating rich user information. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pages 959–968, 2010.
- [LL14] Daryl Lim and Gert Lanckriet. Efficient learning of mahalanobis metrics for ranking. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1980–1988, 2014.

- [LLAM13] Xin Liu, Yong Liu, Karl Aberer, and Chunyan Miao. Personalized point-of-interest recommendation by mining users' preference transition. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, pages 733–738, 2013.
- [LLL11] Kenneth Wai-Ting Leung, Dik Lun Lee, and Wang-Chien Lee. Clr : A collaborative location recommendation framework based on co-clustering. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 305–314, 2011.
- [LR96] Thomas Lengauer and Matthias Rarey. Computational methods for biomolecular docking. *Current opinion in structural biology*, 6(3):402–406, 1996.
- [LSEM12] Justin J Levandoski, Mohamed Sarwat, Ahmed Eldawy, and Mohamed F Mokbel. Lars: A location-aware recommender system. In *Proceedings of the 28th IEEE International Conference on Data Engineering*, pages 450–461, 2012.
- [LSY03] Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, 2003.
- [LWSM14] Yong Liu, Wei Wei, Aixin Sun, and Chunyan Miao. Exploiting geographical neighborhood characteristics for location recommendation. In *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management*, pages 739–748, 2014.
- [LZC09] Jiao Li, Xiaoyan Zhu, and Jake Yue Chen. Building disease-specific drug-protein connectivity maps from molecular interaction networks and pubmed abstracts. *PLoS Computational Biology*, 5(7):e1000450, 2009.

- [LZSM15] Yong Liu, Peilin Zhao, Aixin Sun, and Chunyan Miao. A boosting algorithm for item recommendation with implicit feedback. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pages 1792–1798, 2015.
- [LZX⁺14] Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. Geomf: joint geographical modeling and matrix factorization for point-of-interest recommendation. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 831–840, 2014.
- [LZZW15] Kai Lu, Yi Zhang, Lanbo Zhang, and Shuxin Wang. Exploiting user and business attributes for personalized business recommendation. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 891–894, 2015.
- [Mar04] Benjamin Marlin. Modeling user rating profiles for collaborative filtering. *Advances in Neural Information Processing Systems*, 16:627–634, 2004.
- [Mer99] Arnd Kohrs-Bernard Merialdo. Clustering for collaborative filtering applications. *Intelligent Image Processing, Data Analysis & Information Retrieval*, 3, 1999.
- [MKY⁺13] Jian-Ping Mei, Chee-Keong Kwoh, Peng Yang, Xiao-Li Li, and Jie Zheng. Drug–target interaction prediction by learning from local information and neighbors. *Bioinformatics*, 29(2):238–245, 2013.
- [ML13] Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*, pages 165–172, 2013.

- [MP00] Koji Miyahara and Michael J Pazzani. Collaborative filtering with the simple bayesian classifier. In *PRICAI 2000 Topics in Artificial Intelligence*, pages 679–689. Springer, 2000.
- [MP02] Koji Miyahara and Michael J Pazzani. Improvement of collaborative filtering with the simple bayesian classifier. *Information Processing Society of Japan*, pages 3429–3437, 2002.
- [MS07] Andriy Mnih and Ruslan Salakhutdinov. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, pages 1257–1264, 2007.
- [MS08] Andriy Mnih and Ruslan Salakhutdinov. Probabilistic matrix factorization. *Advances in neural information processing systems*, 20:1257–1264, 2008.
- [MZL⁺11] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. Recommender systems with social regularization. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*, pages 287–296, 2011.
- [NJW02] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems*, 2:849–856, 2002.
- [NK11] Xia Ning and George Karypis. Slim: Sparse linear methods for top-n recommender systems. In *Proceedings of the 11th IEEE International Conference on Data Mining*, pages 497–506, 2011.
- [NK12] Xia Ning and George Karypis. Sparse linear methods with side information for top-n recommendations. In *Proceedings of the 6th ACM Conference on Recommender Systems*, pages 155–162, 2012.

- [NS07] Nobuyoshi Nagamine and Yasubumi Sakakibara. Statistical prediction of protein–chemical interactions based on chemical structure and mass spectrometry data. *Bioinformatics*, 23(15):2004–2012, 2007.
- [OALH06] John P Overington, Bissan Al-Lazikani, and Andrew L Hopkins. How many drug targets are there? *Nature reviews Drug discovery*, 5(12):993–996, 2006.
- [PAP⁺15] Tapio Pahikkala, Antti Airola, Sami Pietilä, Sushil Shakyawar, Agnieszka Szwejda, Jing Tang, and Tero Aittokallio. Toward more realistic drug–target interaction predictions. *Briefings in Bioinformatics*, 16:325–337, 2015.
- [PC13a] Weike Pan and Li Chen. Cofiset: Collaborative filtering via learning pairwise preferences over item-sets. In *Proceedings of the SIAM International Conference on Data Mining*, pages 641–649, 2013.
- [PC13b] Weike Pan and Li Chen. Gbpr: Group preference based bayesian personalized ranking for one-class collaborative filtering. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pages 2691–2697, 2013.
- [PK13] Ulrich Paquet and Noam Koenigstein. One-class collaborative filtering with random graphs. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 999–1008, 2013.
- [PS09] Rong Pan and Martin Scholz. Mind the gaps: weighting the unknown in large-scale one-class collaborative filtering. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 667–676, 2009.
- [PZC⁺08] Rong Pan, Yunhong Zhou, Bin Cao, Nathan Nan Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. One-class collaborative filtering. In

- Proceedings of the 8th IEEE International Conference on Data Mining*, pages 502–511, 2008.
- [Ren10] Steffen Rendle. Factorization machines. In *Proceedings of the 10th IEEE International Conference on Data Mining*, pages 995–1000, 2010.
- [RF14] Steffen Rendle and Christoph Freudenthaler. Improving pairwise learning for item recommendation from implicit feedback. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, pages 273–282, 2014.
- [RFGST09] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, pages 452–461, 2009.
- [RFST10] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, pages 811–820, 2010.
- [RIS⁺94] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens : an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, pages 175–186, 1994.
- [SBHM10] Vikas Sindhwani, Serhat S Bucak, Jianying Hu, and Aleksandra Moksilovic. One-class matrix completion with low-density factorizations. In *Proceedings of the 10th IEEE International Conference on Data Mining*, pages 1055–1060, 2010.
- [SCE⁺04] Ida Schomburg, Antje Chang, Christian Ebeling, Marion Gremse, Christian Heldt, Gregor Huhn, and Dietmar Schomburg. Brenda, the

- enzyme database: updates and major new developments. *Nucleic Acids Research*, 32(suppl 1):D431–D433, 2004.
- [SG11] Guy Shani and Asela Gunawardana. Evaluating recommendation systems. In *Recommender Systems Handbook*, pages 257–297. Springer, 2011.
- [SJ03a] Luo Si and Rong Jin. Flexible mixture model for collaborative filtering. In *Proceedings of the 20th International Conference on Machine Learning*, volume 20, pages 704–711, 2003.
- [SJ03b] Nathan Srebro and Tommi Jaakkola. Weighted low-rank approximations. In *Proceedings of the 20th International Conference on Machine Learning*, pages 720–727, 2003.
- [SK06] Xiaoyuan Su and Taghi M Khoshgoftaar. Collaborative filtering for multi-class data using belief nets algorithms. In *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence*, pages 497–504, 2006.
- [SK09] Xiaoyuan Su and Taghi M Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009:4, 2009.
- [SKB⁺12a] Yue Shi, Alexandros Karatzoglou, Linas Baltrunas, Martha Larson, Alan Hanjalic, and Nuria Oliver. Tfmap: optimizing map for top-n context-aware recommendation. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 155–164, 2012.
- [SKB⁺12b] Yue Shi, Alexandros Karatzoglou, Linas Baltrunas, Martha Larson, Nuria Oliver, and Alan Hanjalic. Climf: learning to maximize reciprocal rank with collaborative less-is-more filtering. In *Proceedings of the 6th ACM Conference on Recommender Systems*, pages 139–146, 2012.

- [SKKR01] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, pages 285–295, 2001.
- [SLH10] Yue Shi, Martha Larson, and Alan Hanjalic. List-wise learning to rank with matrix factorization for collaborative filtering. In *Proceedings of the 4th ACM Conference on Recommender Systems*, pages 269–272, 2010.
- [SLH14] Yue Shi, Martha Larson, and Alan Hanjalic. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Computing Surveys*, 47(1):3, 2014.
- [SM08] Ruslan Salakhutdinov and Andriy Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. *Proceedings of the 25th International Conference on Machine learning*, pages 880–887, 2008.
- [SM11] Salvatore Scellato and Cecilia Mascolo. Measuring user activity on an online location-based social network. In *2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 918–923, 2011.
- [SNM11] Salvatore Scellato, Anastasios Noulas, and Cecilia Mascolo. Exploiting place features in link prediction on location-based social networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1046–1054, 2011.
- [SRJ04] Nathan Srebro, Jason Rennie, and Tommi S Jaakkola. Maximum-margin matrix factorization. *Advances in Neural Information Processing Systems*, 17:1329–1336, 2004.

- [SS99] Robert E. Schapire and Yoram Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.
- [Ste10] Harald Steck. Training and testing of recommender systems on data missing not at random. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 713–722, 2010.
- [TBK⁺14] Amit Tiroshi, Shlomo Berkovsky, Mohamed Ali Kaafar, David Vallet, Terence Chen, and Tsvi Kuflik. Improving business rating predictions using graph based features. In *Proceedings of the 19th International Conference on Intelligent User Interfaces*, pages 17–26, 2014.
- [TS06] Yuichiro Takeuchi and Masanori Sugimoto. Cityvoyager: an outdoor recommendation system based on user location history. In *Ubiquitous Intelligence and Computing*, pages 625–636. Springer, 2006.
- [TY10] Kim-Chuan. Toh and Sangwoon Yun. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Pacific Journal of Optimization*, 6(15):615–640, 2010.
- [UBG09] Nicolas Usunier, David Buffoni, and Patrick Gallinari. Ranking with ordered weighted pairwise classification. In *Proceedings of the 26th International Conference on Machine Learning*, pages 1057–1064, 2009.
- [UF98] Lyle H Ungar and Dean P Foster. Clustering methods for collaborative filtering. In *AAAI Workshop on Recommendation Systems*, pages 114–129, 1998.
- [vLM13] Twan van Laarhoven and Elena Marchiori. Predicting drug-target interactions for new drug compounds using a weighted nearest neighbor profile. *PloS One*, 8(6):e66952, 2013.

- [vLNM11] Twan van Laarhoven, Sander B. Nabuurs, and Elena Marchiori. Gaussian interaction profile kernels for predicting drug–target interaction. *Bioinformatics*, 27(21):3036–3043, 2011.
- [WB11] Chong Wang and David M Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 448–456, 2011.
- [WHZL13] Jialei Wang, Steven C.H. Hoi, Peilin Zhao, and Zhi-Yong Liu. Online multi-task collaborative filtering for on-the-fly recommender systems. In *Proceedings of the 7th ACM Conference on Recommender Systems*, pages 237–244, 2013.
- [WKG⁺08] David S. Wishart, Craig Knox, An Chi Guo, Dean Cheng, Savita Shrivastava, Dan Tzur, Bijaya Gautam, and Murtaza Hassanali. Drugbank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic Acids Research*, 36(suppl 1):D901–D906, 2008.
- [WKLS07] Markus Weimer, Alexandros Karatzoglou, Quoc Viet Le, and Alex Smola. Maximum margin matrix factorization for collaborative ranking. *Advances in neural information processing systems*, pages 1–8, 2007.
- [WSZ⁺13] Kejian Wang, Jiazhi Sun, Shufeng Zhou, Chunling Wan, Shengying Qin, Can Li, Lin He, and Lun Yang. Prediction of drug-target interactions for drug repositioning only based on genomic expression similarity. *PLoS Computational Biology*, 9(11):e1003315, 2013.
- [WSZ14] Yanghao Wang, Hailong Sun, and Richong Zhang. Adamf: Adaptive boosting matrix factorization for recommender system. In *Proceedings of the 15th International Conference on Web-Age Information Management*, pages 43–54. Springer, 2014.

- [WTM13] Hao Wang, Manolis Terrovitis, and Nikos Mamoulis. Location recommendation in location-based social networks using user check-in data. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 374–383, 2013.
- [WWY13] Jason Weston, Ron J Weiss, and Hector Yee. Nonlinear latent factorization by embedding multiple user interests. In *Proceedings of the 7th ACM Conference on Recommender Systems*, pages 65–68, 2013.
- [XL07] Jun Xu and Hang Li. Adarank: a boosting algorithm for information retrieval. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 391–398, 2007.
- [XLY⁺05] Gui-Rong Xue, Chenxi Lin, Qiang Yang, WenSi Xi, Hua-Jun Zeng, Yong Yu, and Zheng Chen. Scalable collaborative filtering using cluster-based smoothing. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 114–121, 2005.
- [XWZW10] Zheng Xia, Ling-Yun Wu, Xiaobo Zhou, and Stephen TC Wong. Semi-supervised drug-protein interaction prediction from heterogeneous biological spaces. *BMC Systems Biology*, 4(suppl 2):S6, 2010.
- [YAG⁺08] Yoshihiro Yamanishi, Michihiro Araki, Alex Gutteridge, Wataru Honda, and Minoru Kanehisa. Prediction of drug–target interaction networks from the integration of chemical and genomic spaces. *Bioinformatics*, 24(13):i232–i240, 2008.
- [YCM⁺13] Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat Thalmann. Time-aware point-of-interest recommendation. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 363–372, 2013.

- [YCS14] Quan Yuan, Gao Cong, and Aixin Sun. Graph-based point-of-interest recommendation with geographical and temporal influences. In *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management*, pages 659–668, 2014.
- [YKM⁺14] Yoshihiro Yamanishi, Masaaki Kotera, Yuki Moriya, Ryusuke Sawada, Minoru Kanehisa, and Susumu Goto. Dinies: drug–target interaction network inference engine based on supervised analysis. *Nucleic Acids Research*, 42(W1):W39–W45, 2014.
- [YL06] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.
- [YLS⁺11] Shuang-Hong Yang, Bo Long, Alexander J Smola, Hongyuan Zha, and Zhaohui Zheng. Collaborative competitive filtering: learning recommender using context of user choice. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 295–304, 2011.
- [YSC⁺13] Hongzhi Yin, Yizhou Sun, Bin Cui, Zhiting Hu, and Ling Chen. Lcars: a location-content-aware recommender system. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 221–229, 2013.
- [YSGL12] Xiwang Yang, Harald Steck, Yang Guo, and Yong Liu. On top-k recommendation using social networks. In *Proceedings of the 6th ACM Conference on Recommender Systems*, pages 67–74, 2012.
- [YTY⁺14] Yuan Yao, Hanghang Tong, Guo Yan, Feng Xu, Xiang Zhang, Boleslaw K Szymanski, and Jian Lu. Dual-regularized one-class collaborative filtering. In *Proceedings of the 23rd ACM International Con-*

- ference on Information and Knowledge Management*, pages 759–768, 2014.
- [YYL10] Mao Ye, Peifeng Yin, and Wang-Chien Lee. Location recommendation for location-based social networks. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 458–461, 2010.
- [YYLL11] Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik-Lun Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 325–334, 2011.
- [YZC13] Jihang Ye, Zhe Zhu, and Hong Cheng. What’s your next move: User activity prediction in location-based social networks. In *Proceedings of the SIAM International Conference on Data Mining*, 2013.
- [YZX12] Jing Yuan, Yu Zheng, and Xing Xie. Discovering regions of different functions in a city using human mobility and pois. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 186–194, 2012.
- [ZC13] Jia-Dong Zhang and Chi-Yin Chow. igslr: personalized geo-social location recommendation: a kernel density estimation approach. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 334–343, 2013.
- [ZC15] Jia-Dong Zhang and Chi-Yin Chow. Geosoca: Exploiting geographical, social and categorical correlations for point-of-interest recommendations. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 443–452, 2015.

- [ZCZY13] Weinan Zhang, Tianqi Chen, Jun Wang, and Yong Yu. Optimizing top-n collaborative filtering via dynamic negative item sampling. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 785–788, 2013.
- [ZCZ⁺10] Vincent Wenchen Zheng, Bin Cao, Yu Zheng, Xing Xie, and Qiang Yang. Collaborative filtering meets mobile recommendation: A user-centered approach. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, volume 10, pages 236–241, 2010.
- [ZDMZ13] Xiaodong Zheng, Hao Ding, Hiroshi Mamitsuka, and Shanfeng Zhu. Collaborative matrix factorization with multiple similarities for predicting drug-target interactions. *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1025–1033, 2013.
- [Zhe12] Yu Zheng. Tutorial on location-based social networks. In *Proceedings of the 21st International Conference on World Wide Web*, volume 12, 2012.
- [ZJYH11] Peilin Zhao, Rong Jin, Tianbao Yang, and Steven C.H. Hoi. Online auc maximization. In *Proceedings of the 28th International Conference on Machine Learning*, pages 233–240, 2011.
- [ZPX⁺14] Hao Zhong, Weike Pan, Congfu Xu, Zhi Yin, and Zhong Ming. Adaptive pairwise preference learning for collaborative recommendation with implicit feedbacks. In *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management*, pages 1999–2002, 2014.
- [ZRY09] Peng Zhao, Guilherme Rocha, and Bin Yu. The composite absolute penalties family for grouped and hierarchical variable selection. *The Annals of Statistics*, 37(6A):3468–3497, 2009.

- [ZWFM06] Sheng Zhang, Weihong Wang, James Ford, and Fillia Makedon. Learning from incomplete ratings using non-negative matrix factorization. In *Proceedings of the 6th SIAM Conference on Data Mining*, pages 548–552, 2006.
- [ZWSP08] Yunhong Zhou, Dennis Wilkinson, Robert Schreiber, and Rong Pan. Large-scale parallel collaborative filtering for the netflix prize. In *Algorithmic Aspects in Information and Management*, pages 337–348. Springer, 2008.
- [ZX11] Yu Zheng and Xing Xie. Location-based social networks: Locations. In *Computing with spatial trajectories*, pages 277–308. Springer, 2011.
- [ZZXM09] Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma. Mining interesting locations and travel sequences from gps trajectories. *Proceedings of the 18th International Conference on World Wide Web*, pages 791–800, 2009.
- [ZZXY10] Vincent W Zheng, Yu Zheng, Xing Xie, and Qiang Yang. Collaborative location and activity recommendations with gps history data. In *Proceedings of the 19th International Conference on World Wide Web*, pages 1029–1038, 2010.