2016 IEEE First International Conference on Data Science in Cyberspace

Ranking-based Music Recommendation in Online Music Radios

Yao Lu[1,3],Zhi Qiao[2,3],Peng Zhang[1],Li Guo[1]
[1]Institute of Information Engineering, Chinese Academy of Sciences
[2]Institute of Computing Technology of the Chinese Academy of Sciences
[3]University of Chinese Academy of Sciences
luyao9023@iie.ac.cn, qiaozhi@nelmail.iie.ac.cn, {zhangpeng, guoli}@iie.ac.cn

***Abstract*—Online music radios, such as Last.fm and Douban.fm, which provide users with free music, have gained much popularity in recent years. In online music radios, music recommendation plays a central role in recommending the most relevant music to users who are most likely to listen to. Different from traditional on-demand music service, online music radios have only users' listening records instead of users' ratings, which make it difficult to give an accurate recommendation. In this paper, we use a pairwise ranking based collaborative filtering method which performs well on one-class collaborative filtering. We future extend our model by considering users' artist preference, which based on the assumption that users who have listened tracks belonging to the same artist are likely to have similar preferences. The resulting model is able to generalize considerably better for users with few ratings. Experimental results on real-world data sets show the performance of our method.**

***Keywords*-music recommendation; implicit recommendation; collaborative filtering**

I. INTRODUCTION

Recent years have witnessed rapid development of online music radios, such as Last.fm [1] and Douban.fm [2], which provide users with a continuous stream of music. In these applications, users provide signals about their preferences at a very high rate. For example, every listening event is indicative for the user's current preference. Using this signal for music recommendation is challenging on two accounts:

1. Implicit feedback. In traditional on-demend music services, such as Spotify and Deezer, recommendation is usually based on user ratings, for which various algorithms are proposed in order to predict the users' preferences scores accurately. However, the signal that a user provides about his preference is implicit [3] in online music radios, such as listening a track. Usually only a small fraction of the data are positive examples, the non-postive examples are ambiguity. Negative examples and unlabeled positive examples are mixed together and we are unable to distinguish them. For example, we cannot really attribute a user not listening a track to a lack of interest or lack of awareness of the track. As a result, we get only positive feedback rather than both positive feedback and negative feedback.

2. Infrequent users. Another challenge is that the data collected from online music radios is very imbalanced, with infrequent users maybe listening less than 10 tracks, while frequent users maybe listening more than 10,000 tracks. However, most of the existing collaborative filtering algorithms have trouble in making accurate predictions for users who have very few ratings.

To handle the above challenges, we analyze the data collected from Last.fm and find there exists a tendency that users are likely to select tracks created by a small group of artists in their listing behavior. And this tendency reflects users' artist preference. We then combine a ranking based Bayesian latent factor model with users' artist preference. And the resulting model shares information between users who have similar artist preference, which is important to learn infrequent users. Experiments on real-world data verify the performance of the method. The research contributions of the paper are:

1. We develop an algorithm for music recommendation in online music radios that can handle with implicit feedback and performs well for users with few listing records.

2. We observe user's artist preference and empirically demonstrate the importance of this preference in recommendation for infrequent users.

We first provide an overview of related work (sec. 2) and then describe the problem setting and user's artist preference (sec. 3). Then we discuss how artist preference can be used in the ranking model and discuss an efficient algorithm to learn the model from implicit feedback (sec. 4). Finally, we conduct experiments (sec. 5) to investigate our approach and the importance of artist preference.

II. RELATED WORK

There are different approaches to music recommendations, including: (a) Expert-based approaches [4] that are based on human annotation of music data. (b) Social based-approaches [5], [6] that characterize items based on social tags, textual attributes and other kinds of social-based annotations. (c) Content based approaches [7], [8] that analyze the audio content for characterizing tracks. (d) Collaborative Filtering (CF) approaches [9] that analyze listening patterns by many users, in order to establish similarities across users and items. Among the above approaches, CF is widely

978-1-5090-1192-6/16 $31.00 © 2016 IEEE
DOI 10.1109/DSC.2016.102

used in large scale recommendation tasks and have received relatively good performance.

Latent factor models constitute one of the leading CF techniques. They characterize both item and users as vectors in a space inferred from observed data patterns. One of the most successful realizations of latent factor models is based on matrix factorization (MF) [10]. These methods have become very popular in recent years by combining good scalability with predictive accuracy. In addition, MF provides a substantial expressive power that allows modeling specific data characteristics, such as social relations [11], [12], [13] and geographic information [14], [15], [16]. Due to its practicability and flexibility, we extend the latent factor model to music recommendation.

Among MF methods our work is mostly related to those dealing with implicit user feedback, such as [17], [18], [3]. Such methods can be roughly summarized in two manners: rating based methods and ranking based methods. The former learn a latent factor vector of both users and items via minimizing a square loss to evaluate the absolute rating scores, while the latter take pairs of items as basic units and maximize the posterior probability of observed and unobserved pairs. Empirically the ranking based methods achieve much better performance than rating based methods. Due to the success of ranking based methods, some algorithms have been proposed to combine ranking based methods with some auxiliary data, such as social information [19], geographic information [20] and item-side taxonomy information [21]. In this paper, we propose a model to combine ranking based method with users' artist preference. And our model is able to generalize considerably better for users with few ratings.

III. Preliminary

We collected users' listing history data from Last.fm [1], one of the most popular online music radios.

A. Artist Preference

We first analyze users' listing history, and observe users' artist preference. Fig. 1 displays the distribution of the number of songs listened by users as the triangle line and that of artists who created this songs as the square line. The y-axis represents cumulative distribution function (CDF). From this figure we can obtain that about 90 precent of users listened less than 4,000 pieces of tracks created by less than 400 artists , about half of users listened less than 700 songs created by less than 100 artists and about 15 precent of users listened less 10 songs and these songs were created by less than 2 artists.

As the Fig. 1 shows, the number of songs users' listened during this period is much larger than the number of artists who created these songs, which verifies the correctness that users are likely to listen to the tracks that created by their favorite artists.

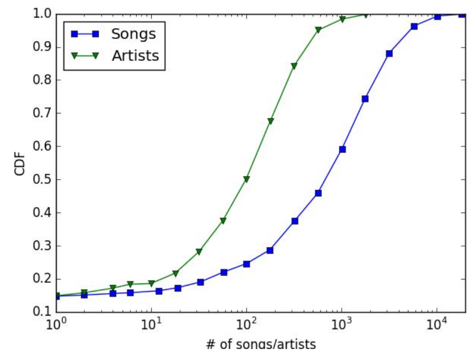

Figure 1. Number of songs and artists

B. Problem Description

For the task of music recommendation, we have three types of entities:U(user), V(track) and S(artist). Let $U = \{u_1, u_2, ..., u_M\}$ denote the set of users and $S = \{s_1, s_2, ..., s_K\}$ denote the set of artists. Let the set of tracks be $V = \{v_1, v_2, ..., v_N\}$. For each track $v_i \in V$, it has a artist s_j.

Formally, the problem is defined as ranking all tracks for each user, according to the dyadic rating score $r(u, v)$, which indicates user u's preference to song v. Hence, predicting $r(u, v)$ plays a central role in music recommendation.

We propose a latent factor model which combines personal preference with artist preference. In order to learn the model a Bayesian personal ranking (BPR) [3] framework is introduced.

IV. Proposed Approach

In this section, we discuss our proposed model, shown in Fig.2. Our model is a Bayesian latent factor model which jointly learns the user, track and artist factor latent spaces. We use matrix factorization to derive latent feature space from user's listening history, and we use a Bayesian personal ranking framework to learn the parameters. First, we will discuss the user personal rating model and further expend it with artist preference, and then we will discuss BPR for our complete model.

Personal rating. The personal rating is similar to the matrix factorization latent model [10], which is proved efficient in recommendations. The basic idea is to embody user i and song j with the low-dimensional latent factor vectors $U_i \in R^k$ and $V_j \in R^k$. Then, the dyadic rating $\hat{r}(u, j)$ of user i to track j is approximated as follows.

$$\hat{r}(u_i, v_j) = U_i^T V_j. \quad (1)$$

Artist preference rating. In user rating model, users with few ratings will have feature vectors that are close to the average user, so the prediction ratings for thoses users will be close to the music average ratings. In the last section,

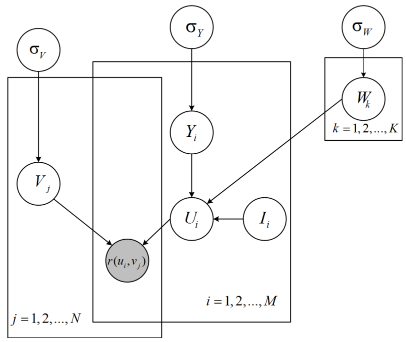

Figure 2. The graphical model for user ratings with artist preference

we have observed user' artist preference. Hence, we introduce an additional way of constraining user-specific feature vectors by taking user' artist preference into consideration and we will show that this method has a strong effect on infrequent users.

Let $W_i \in R^k$ be artist $i's$ latent factor vector. We define the feature vector for user i as:

$$U_i = \alpha Y_i + (1-\alpha)\frac{\sum_{k=1}^{K} I_{ik} W_k}{\sum_{k=1}^{K} I_{ik}} \tag{2}$$

Where I is the observed indicator matrix with I_{ij} taking on value 1 if user i has listened artist $j's$ music and 0 otherwise. Intuitively, Y_i can be seen as user $i's$ preference latent factor vector and the latter part captures the effect of a user having listened a particular artist's music has on the user's feature vector. And $\alpha \in (0,1)$ is the fusion parameter that controls the contribution of the two parts. As a result, users that have listened to tracks created by same (or similar) artists will have similar feature vectors. And users with few ratings will have feature vectors that close to the average users who have similar favorite artists.

The collected data is in one-class form, which is also called implicit feedback. Bayesian Personalized Ranking (BPR)[3] emphases on predicting the dyadic ratings $r(u_i, v_j)$ and top-ranking items with high scores, which can be used to solve the implicit feedback recommendation problem. Based on the BPR optimization criterion, we regard tracks listened by user u as a positive music set denoted as S_u^P, while the remaing songs as the negative set denoted as S_u^N. Then we expect to maximize the objective function that ranks the positive set S_u^P higher than the negative set S_u^N, as in Eq.(3)

$$\max_{\Theta} \prod_{(u_i,v_j,v_k)\in(U,S_u^P,S_u^N)} P(r(u_i,v_j) > r(u_i,v_k)|\Theta) \tag{3}$$

Where $\Theta = (U, V, W)$ is the parameter set in the model.

In order to avoid over-fitting in the learning process, we also enforce Gaussian priors on the latent factor vectors U_i, V_j, W_k. Then give the user/music feedback matrix R, user/artist matrix I, prior parameters $\Phi = (\sigma_Y, \sigma_V, \sigma_W)$, we can obtain the posterior probability $P(\Theta|R, I, \Phi)$ as Eq.(4)

$$\begin{aligned} P(\Theta|R,I,\Phi) = &\prod_{(u_i,v_j,v_k)\in(U,S_u^P,S_u^N)} P(r(u_i,v_j) > r(u_i,v_k)|\Theta) \\ &\cdot \prod_i^M P(Y_i|\sigma_Y) \cdot \prod_j^N P(V_j|\sigma_V) \cdot \prod_k^K P(W_k|\sigma_W) \end{aligned} \tag{4}$$

Where $P(r(u_i,v_j) > r(u_i,v_k)) := L(r(u_i,v_j) - r(u_i,v_k))$, and function L is the logistic function.

Then we plug Eq.(2) into the likelihood, we get Eq.(5)

$$\begin{aligned} &\prod_{(u_i,v_j,v_k)\in(U,S_u^P,S_u^N)} P(r(u_i,v_j) > r(u_i,v_k)|\Theta) \\ &= \prod_{(u_i,v_j,v_k)\in(U,S_u^P,S_u^N)} L(r(u_i,v_j) - r(u_i,v_k)) \\ &= \prod_{(u_i,v_j,v_k)\in(U,S_u^P,S_u^N)} L(\alpha Y_i + (1-\alpha)\frac{\sum_{k=1}^{N} I_{ik} W_k}{\sum_{k=1}^{N} I_{ik}})(v_j - v_k) \end{aligned} \tag{5}$$

The priors are Eq.(6):

$$\begin{aligned} P(\Theta|\Phi) &= \prod_i^M P(Y_i|\sigma_Y) \cdot \prod_j^N P(V_j|\sigma_V) \cdot \prod_k^K P(W_k|\sigma_W) \\ &= \prod_i^M N(Y_i|0, \sigma_Y^2 I) \cdot \prod_j^N N(V_j|0, \sigma_V^2 I) \cdot \prod_k^K N(W_k|0, \sigma_W^2 I) \end{aligned} \tag{6}$$

Where $N(x|\mu, \sigma^2)$ is the probability density function of the Gaussian distribution with mean μ and variance σ^2, and I is the identity matrix.

From Eq.(5) and Eq.(6) we can get the posterior. As we know, maximizing the log-posterior is equivalent to minimizing the loss function with quadratic regularization terms:

$$\begin{aligned} E = &\sum_{(u_i,v_j,v_k)\in(U,S_u^P,S_u^N)} ln(1 + e^{-(\alpha Y_i + (1-\alpha)\frac{\sum_{k=1}^{K} I_{ik} W_k}{\sum_{k=1}^{K} I_{ik}})(v_j - v_k)}) \\ &+ \frac{\lambda_Y}{2}\sum_{i=1}^{M} \|Y_i\|^2 + \frac{\lambda_V}{2}\sum_{j=1}^{N} \|V_j\|^2 + \frac{\lambda_W}{2}\sum_{k=1}^{K} \|W_k\|^2 \end{aligned} \tag{7}$$

Where $\lambda_Y = 1/\sigma_Y^2$,$\lambda_V = 1/\sigma_v^2$, and $\lambda_W = 1/\sigma_W^2$.

A. Parameter Learning

The parameters can be learned by minimizing the above objective function E by using stochastic gradient descent (SGD) algorithm. SGD has fast speed to convergence and high scalability to large-scale data sets.

We first choose the triples (u_i, v_j, v_k) from (U, S_u^P, S_u^N) randomly with replacement. Then based on the objective function in Eq.(7), we have the following gradients:

$$\frac{\partial E}{\partial Y_i} = \frac{-e^{-(r(u_i,v_j) - r(u_i,v_k))}}{1 + e^{-(r(u_i,v_j) - r(u_i,v_k))}}\alpha(v_j - v_k) + \lambda_Y \|Y_i\| \tag{8}$$

$$\frac{\partial E}{\partial V_j} = \frac{-e^{-(r(u_i,v_j)-r(u_i,v_k))}}{1+e^{-(r(u_i,v_j)-r(u_i,v_k))}}(\alpha Y_i + (1-\alpha)\frac{\sum_{k=1}^{K} I_{ik}W_k}{\sum_{k=1}^{K} I_{ik}}) + \lambda_V \|V_j\| \tag{9}$$

$$\frac{\partial E}{\partial V_k} = \frac{e^{-(r(u_i,v_j)-r(u_i,v_k))}}{1+e^{-(r(u_i,v_j)-r(u_i,v_k))}}(\alpha Y_i + (1-\alpha)\frac{\sum_{k=1}^{K} I_{ik}W_k}{\sum_{k=1}^{K} I_{ik}}) + \lambda_V \|V_k\| \tag{10}$$

$$\frac{\partial E}{\partial W_s} = \frac{-e^{-(r(u_i,v_j)-r(u_i,v_k))}}{1+e^{-(r(u_i,v_j)-r(u_i,v_k))}}(1-\alpha)\frac{I_{is}}{\sum_{k=1}^{K} I_{ik}}(v_j - v_k) + \lambda_W \|W_s\| \tag{11}$$

We thus have the update rules used in the SGD algorithm framework:

$$\begin{aligned} Y_i = Y_i - \gamma\frac{\partial E}{\partial Y_i}; \quad V_j = V_j - \gamma\frac{\partial E}{\partial V_j} \\ V_k = V_k - \gamma\frac{\partial E}{\partial V_k}; \quad W_s = W_s - \gamma\frac{\partial E}{\partial W_s} \end{aligned} \tag{12}$$

where $\gamma > 0$ is the learning rate.

The complete steps to learn the model parameters are depicted in Algorithm 1.

Algorithm 1 The algorithm to learn the model parameters.

Input: user-music matrix R, user-artist matrix I, fusion parameter α, learning rate γ, regularization parameters $\lambda_Y, \lambda_W, \lambda_V$, parameter ϵ.
Output: user's personal latent factor Y, artist's latent factor W, music latent factor V

```
Initialize U, V and W with randomly generated vectors
Generate (U, S_u^P, S_u^N) from R
Draw (u_i, v_j, v_k) from(U, S_u^P, S_u^N)
Initialize parameter pQ = 0
Compute AUC value Q for train data
repeat
    pQ = Q
    Calculate the gradients as in Eqs.(8~11)
    Update the parameters as in Eq.(12)
    Compute AUC value Q for train data
until Q − pQ > ϵ
return the parameters U, V and W.
```

V. Experiment

We implement the proposed recommendation model and test on real-world data sets to demonstrate its effectiveness. We latter give experimental results with discussions.

A. Datasets

We use data collected form Last.fm, starting on March 1st and lasting until May 31st, 2015. The dataset contains 1,171,094 user listening records assigned by 864 users on 280,338 tracks which were created by 34,291 artists.

For the datasets, we randomly sample twenty percent of the observed user-item pairs as test data, and the rest as training data; we then randomly divide the training data into 5 parts to do cross validation. We repeat the above procedure for three times, so we have three copies of training data and test data. The experimental results are averaged over the performance on those three copies of test data.

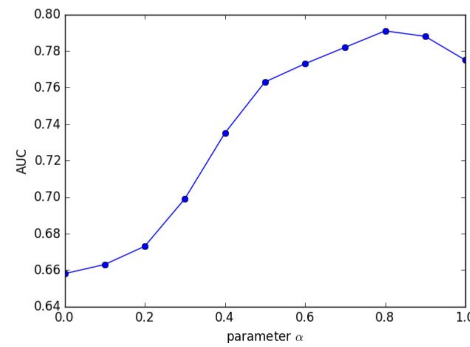

Figure 3. Test on the parameter α

B. Evaluation Metrics

Because users usually only check a few top-ranked items [22], we use P@k(Precision at Position k) to study the recommendation preference. Since AUC is a good measure for highly imbalanced dataset, as in our case where the negative music take a high proportion we also include in our evaluation.

AUC measures the overall results of classification. In this work, the AUC is calculated via E.q(13).

$$AUC = \frac{\sum_{u_i \in U}\sum_{v_j \in S_u^P}\sum_{v_k \in S_u^N} I(r(u_i,v_j) > r(u_i,v_k))}{\sum_{u_i \in U} |S_u^P| \cdot |S_u^N|} \tag{13}$$

where $I(\cdot)$ is a indicator function that equals to 1 if $r(u_i,v_j) > r(u_i,v_k)$ is true, otherwise, 0.

C. Baselines and Parameter Settings

We implement the following baseline methods for comparisons:

- User-based Collaborative Filtering(User-CF). User-CF assumes that similar users share similar preference.
- Item-based Collaborative Filtering(Item-CF)[23]. Item-CF assumes that similar items share similar characteristics.
- Matrix Factorization(MF). We adapt MF algorithm by randomly select negative samples for each user [17].
- Baysian Personal Ranking(BPR)[3]. BPR is a strong baseline, which is shown to be much better than some well-known pointwise methods[19].

The AUC performance on the validation data is used to select the best parameters for MF, BPR and our method. We perform a grid search for parameters and after trying various values for the learning rate and regularization parameters, we

Table I
EXPERIMENT RESULTS.

Method	P@5	P@10	AUC
User-CF	0.114	0.086	0.551
Item-CF	0.037	0.029	0.523
MF	0.096	0.0828	0.532
BPR	0.244	0.221	0.774
Our method	**0.256**	**0.237**	**0.791**

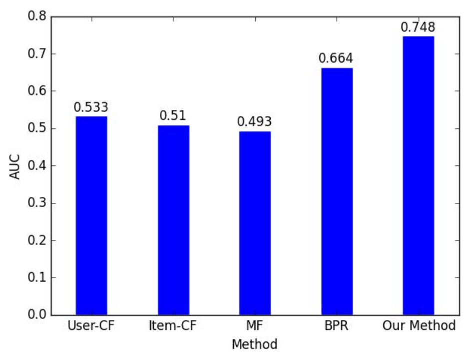

Figure 4. Results on unfrequent users

choose to use a learning rate of 0.09, and a regularization parameter of 0.01 for λ_Y, λ_W, and λ_V. We tune the fusion coefficient α by evaluating how the AUC changes. As the results shown in Fig.3, we get better performance when $\alpha = 0.8$, so we set $\alpha = 0.8$. Finally, we set the number of dimensions to be 32 for the latent factors in the model.

D. Summary of Experimental Results

1) Recommendation for all users: In this first experiment, we investigate the predictive quality of the recommendation approach for all users. The recommendation performance of our method and other baselines are shown in Table 1, from which we can have the following observations,

- Both BPR and our method perform very well in this protocol and largely outperform the rating based algorithms in all metrics, which demonstrates the effectiveness of ranking based recommendation.
- Our method further improves the performance of BPR in all evaluation metrics on our dataset, which shows the effect of artist preference in music recommendation.

2) Recommendation for infrequent users: To test the recommendation for users with few listening records, we select 139 users with less than 10 listening records in train set and show the results of AUC in Fig.4.

We obtain the following observations: First, BPR and our method perform much better than rating based methods. Second, all the methods perform worse than in recommendation for all users protocol. Third, our proposed method has a fewer decrease in performance than BPR. This can be explained by the effect of introducing the users' artist preference, as our model shares information between all users who listened tracks belonging to the same artist.

VI. CONCLUSION

In this paper, we studied the music recommendation problem in online music radios. The technical challenge is to deal with implicit feedback and get better recommendation for users with few listing records. We analyzed the collected data and observed user's artist preference and designed a ranking based algorithm combined with this preference. Experimental results on real-world dataset demonstrated the performance of our method.

ACKNOWLEDGMENT

This work was supported by the 973 project (No. 2013CB329605) and Strategic Leading Science and Technology Projects of CAS (No.XDA06030200).

REFERENCES

[1] Last.fm. [Online]. Available: http://www.last.fm/

[2] Douban.fm. [Online]. Available: http://www.douban.fm/

[3] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press, 2009, pp. 452–461.

[4] M. Castelluccio, "The music genome project," *Strategic Finance*, vol. 88, no. 6, p. 57, 2006.

[5] A. Nanopoulos, D. Rafailidis, P. Symeonidis, and Y. Manolopoulos, "Musicbox: Personalized music recommendation based on cubic analysis of social tags," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, no. 2, pp. 407–412, 2010.

[6] D. Eck, P. Lamere, T. Bertin-Mahieux, and S. Green, "Automatic generation of social tags for music recommendation," in *Advances in neural information processing systems*, 2008, pp. 385–392.

[7] N. Hariri, B. Mobasher, and R. Burke, "Context-aware music recommendation based on latenttopic sequential patterns," in *Proceedings of the sixth ACM conference on Recommender systems*. ACM, 2012, pp. 131–138.

[8] A. Van den Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," in *Advances in Neural Information Processing Systems*, 2013, pp. 2643–2651.

[9] N. Aizenberg, Y. Koren, and O. Somekh, "Build your own music recommender by modeling internet radio streams," in *Proceedings of the 21st international conference on World Wide Web*. ACM, 2012, pp. 1–10.

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

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

[12] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 2011, pp. 287–296.

[13] F. Liu and H. J. Lee, "Use of social network information to enhance collaborative filtering performance," *Expert systems with applications*, vol. 37, no. 7, pp. 4772–4778, 2010.

[14] C. Cheng, H. Yang, I. King, and M. R. Lyu, "Fused matrix factorization with geographical and social influence in location-based social networks," in *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[15] H. Gao, J. Tang, X. Hu, and H. Liu, "Content-aware point of interest recommendation on location-based social networks." in *AAAI*, 2015, pp. 1721–1727.

[16] Z. Qiao, P. Zhang, Y. Cao, C. Zhou, and L. Guo, "Improving collaborative recommendation via location-based user-item subgroup," *Procedia Computer Science*, vol. 29, pp. 400–409, 2014.

[17] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang, "One-class collaborative filtering," in *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. IEEE, 2008, pp. 502–511.

[18] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. Ieee, 2008, pp. 263–272.

[19] L. Du, X. Li, and Y.-D. Shen, "User graph regularized pairwise matrix factorization for item recommendation," in *Advanced Data Mining and Applications*. Springer, 2011, pp. 372–385.

[20] Z. Qiao, P. Zhang, Y. Cao, C. Zhou, L. Guo, and B. Fang, "Combining heterogenous social and geographical information for event recommendation," in *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[21] B. Kanagal, A. Ahmed, S. Pandey, V. Josifovski, J. Yuan, and L. Garcia-Pueyo, "Supercharging recommender systems using taxonomies for learning user purchase behavior," *Proceedings of the VLDB Endowment*, vol. 5, no. 10, pp. 956–967, 2012.

[22] L. Chen and P. Pu, "Users' eye gaze pattern in organization-based recommender interfaces," in *Proceedings of the 16th international conference on Intelligent user interfaces*. ACM, 2011, pp. 311–314.

[23] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*. ACM, 2001, pp. 285–295.