

Predicting User Preference Based on Matrix Factorization by Exploiting Music Attributes

Amir Hossein Nabizadeh
LIAAD / INESC TEC, Porto,
Portugal
DCC-FCUP, Universidade do
Porto, Portugal
amirhossein@dcc.fc.up.pt

Alípio Mário Jorge
LIAAD / INESC TEC, Porto,
Portugal
DCC-FCUP, Universidade do
Porto, Portugal
amjorge@fc.up.pt

Suhua Tang
Graduate School of
Informatics and
Engineering, University of
Electro-Communications
shtang@uec.ac.jp

Yi Yu
Digital Content and Media
Sciences Research Division,
National Institute of
Informatics
yiyu@nii.ac.jp

ABSTRACT

With the emergence of online Music Streaming Services (MSS) such as Pandora and Spotify, listening to music online became very popular. Despite the availability of these services, users face the problem of finding among millions of music tracks the ones that match their music taste. MSS platforms generate interaction data such as users' defined playlists enriched with relevant metadata. These metadata can be used to predict users' preferences and facilitate personalized music recommendation. In this work, we aim to infer music tastes of users by using personal playlist information. Characterizing users' taste is important to generate trustable recommendations when the amount of usage data is limited. Here, we propose to predict the users' preferred music feature's value (e.g. *Genre* as a feature has different values like *Pop*, *Rock*, etc.) by modeling, not only usage information, but also music description features. Music attribute information and usage data are typically dealt with separately. Our method FPMF (Feature Prediction based on Matrix Factorization) treats music feature values as virtual users and retrieves the preferred feature values for real target users. Experimental results indicate that our proposal is able to handle the *item cold start* problem and can retrieve preferred music feature values with limited usage data. Furthermore, our proposal can be useful in recommendation explanation scenarios.

CCS Concepts

•Information systems → Recommender systems;

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](http://permissions.acm.org).

C3S2E '16, July 20-22, 2016, Porto, Portugal

© 2016 ACM. ISBN 978-1-4503-4075-5/16/07...\$15.00

DOI: <http://dx.doi.org/10.1145/2948992.2949010>

Keywords

Recommender Systems; Matrix Factorization; Music Recommender; Cold Start

1. INTRODUCTION

Listening to music for leisure or for blocking ambient noise is a routine activity and has become particularly popular with the emergence of online music streaming services (MSS) such as AppleMusic, TIDAL, Spotify, etc. MSS deliver music audio tracks to users without the necessity of downloading a music file. A common issue is how to find preferred music among an extensive list of musical tracks. Music recommender systems have been proposed to help users find music of interest from large collections [1]. These recommenders apply different types of data such as playlists or metadata (user or music metadata) to generate the recommendations. These data are typically generated during the interaction of the user with the system. It is important to be able to generate trustable recommendations also when the amount of usage data is limited (e.g. in case of *Cold start*). Many researchers have addressed this problem by applying different techniques and data [12, 19, 5] (See Section 2).

In this work we exploit the relationship between music features (e.g. Genre, Band, Instrumental, etc.) and users to predict the musical taste of users and recommend relevant music tracks. In our approach, we consider each value of each music feature as a *virtual user* that has listened to the music tracks with the corresponding feature value. For example, the genre "hip-hop" is regarded as a user and the tracks of this genre are the activity of this user. From this feature-music track relationship, we can predict the music a user may like given the user's preferred features. In this study, we propose a method based on Matrix Factorization [2, 10, 11] to recommend the most likely music feature's value to target users (for example, in the case of the feature *Genre*, our method can suggest the values *Pop*, *Rock*, *Jazz*, etc.). After extending the user set with the virtual users, we factorize the extended $User \times Music$ observed matrix using an Incremental Stochastic Gradient Descent (ISGD) algo-

rithm (training phase) [17]. To retrieve the favorite features of a target user, we infer the most likely features by applying the dot product of the target user vector and the pivot temporary item (detail in Section 4). We call our method FPMF (Feature Prediction based on Matrix Factorization).

The main motivations of our study are the following: first, to tackle the *item cold start* issue; second, recommending a music features helps explaining why a specific music has been recommended to a user and this explanation increases the users’ satisfaction and enhances their trust in recommenders [15]; third, our proposal, which is based on Matrix Factorization, can deal with *Sparsity* and *Scalability* issues. Furthermore, since in the proposed method typically the number of values for each music feature is significantly lower than the number of users and musics, characterizing the preferred music feature for a user consumes only an additional small quantity of time and memory.

The remainder of this paper is structured as follows: section 2 introduces related work section 3 explains in detail the algorithm, which is used as the basis for our approach; section 4 describes the FPMF approach section 5 provides a recommendation algorithm; in section 6 we describe the techniques that are applied to evaluate the method and results; and finally, in section 7, we present our conclusions.

2. RELATED WORK

Collaborative filtering approaches (especially using matrix factorization) have been successfully used in prediction and recommendation problems. Researchers such as [10, 6, 18] applied latent factor models based on MF to enhance the rating prediction accuracy of traditional collaborative filtering or to tackle the *cold start* issues.

Zhou et al. [19] proposed functional Matrix Factorization (fMF), a method that solves the problem of initial interview construction within the context of learning user and item profiles. In their method they utilized the explicit answer of users about items to overcome the *cold start* problem. In [8], a collaborative music recommender using an item-based probabilistic model is described, where items are classified into groups and predictions are made for users considering the Gaussian distribution of user ratings. This model is extended by utilizing audio attributes in order to enhance the result. In their proposal they could alleviate the *cold start* issue.

Spiegel et al. designed a hybrid method based on MF, which applied users and items attributes to enhance the prediction accuracy of traditional collaborative filtering [16]. Another study that influenced our work was conducted by Gantner et al. in 2010, who mapped user/item attributes to the latent features of a MF model. Their mapping technique was utilized to generate an attribute aware MF model for item recommendation from implicit, positive-only feedback [3].

3. MATRIX FACTORIZATION

MF discovers latent relations between users and items in a ratings matrix [7, 9, 14]. Suppose that R is a matrix with size $n \times m$ (n user and m items) whose entries are ratings. The matrix R can be decomposed into two matrices using a MF technique as shown in the following equation:

$$R \approx \hat{R} = A \cdot B^T \quad (1)$$

where A represents a user matrix with size $n \times q$ (i.e. each row A_u represents a user u) and B is an item matrix with size $m \times q$ (i.e. each row B_i represents an item i). q represents the number of latent factors which are learned from past responses of users. Interpretation of factors is not intuitive and changes tremendously depending on the choice of q . To predict the rating user u would give to item i we can use the dot product between the respective rows of A and B .

$$\hat{R}_{ui} = A_u \cdot B_i^T$$

To obtain the matrices A and B (eq. 1) we can minimize the regularized squared error for known values in R . This is the training step.

$$\min_{A,B} \sum_{(u,i) \in R} (R_{ui} - A_u \cdot B_i^T)^2 + \lambda(\|A_u\|^2 + \|B_i\|^2) \quad (2)$$

λ is a regularization parameter and the term $\lambda(\|A_u\|^2 + \|B_i\|^2)$ is used to avoid overfitting by penalizing the high values for the parameters.

4. USER PREFERENCE PREDICTION

In this section, we describe our method FPMF in detail and explain how to associate users with the music features they prefer. Without loss of generality we will consider binary features. Any feature f with values v_1, \dots, v_m can be transformed to m binary features $f = v_1, \dots, f = v_m$. We assume a set of n users U , a set of m music tracks I , and a set of k binary features F . Each track $i \in I$ has a set of features f expressed as $f(i)$. User-music interaction occurs whenever a user selects a music (i.e. a positive interaction between user and item is considered as a manifestation of preference).

The usual MF approaches factorize a conceptual $Users \times Music$ interaction matrix (R), where, assuming binary interactions, $R_{u,i} = 1$ if user u selected music i , and $R_{u,i} = 0$ otherwise. To handle binary item features, we extend the R matrix with a set of virtual users as represented in Figure 1. Each virtual user corresponds to a feature in F . The extended matrix \bar{R} has dimensions $(n+k) \times m$. $\bar{R}_{f,i}, f > n$ is such that $\bar{R}_{f,i} = 1$ if music i possesses feature f and zero otherwise.

We factorize the extended matrix \bar{R} using the *ISGD* algorithm described below [17]. Note that only the positive observations ($\bar{R}_{u,i} = 1$) are actually used for training. After factorization we are ready to retrieve the favorite features of a target user. For that, we add a temporary dummy item Z (Z is a vector), which interacts only with the target user (see Figure 1). We then predict the features of this music track given that the user prefers it. Now, we retrieve the interaction of the dummy item with the virtual users (features). This interaction is the retrieved level of feature preference of the target user. At a glance, our proposed method FPMF works as follow:

- Generate \bar{R} based on usage data (R) and music track features.
- Factorize \bar{R} with Algorithm 1 (ISGD).
- Retrieve preferred feature value by Algorithm 2.

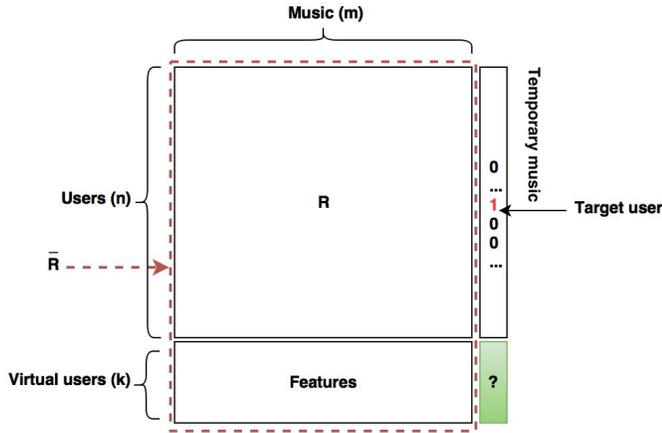


Figure 1: Conceptual model of FPMF : initial ratings matrix R is conceptually extended (\bar{R}) by regarding each binary feature as a virtual user. Non binary features are binarized.

4.1 Incremental Stochastic Gradient Descent (ISGD)

We use the Incremental Stochastic Gradient Descent algorithm (ISGD, Alg. 1) [17] to factorize the generated matrix \bar{R} into *user* and *music* matrices. ISGD is a very efficient, scalable incremental algorithm that only requires a single pass over the data.

Input : \bar{R} , λ =Regularization, η = Learn rate,
 q = # factors
Output: User matrix A and music matrix B

```

for (User  $u$ , Item  $i$ )  $\in$   $\bar{R}$  do
  if user  $u \notin U$  then
     $A_u \leftarrow N(0, 0.1)$ ;  $\triangleright$   $A_u$  size :  $q$ 
    add  $u$  to  $U$ 
  end
  if item  $i \notin I$  then
     $B_i^T \leftarrow N(0, 0.1)$ ;  $\triangleright$   $B_i^T$  size :  $q$ 
    add  $i$  to  $I$ 
  end
   $E_{ui} \leftarrow 1 - A_u \cdot B_i^T$ 
   $A_u \leftarrow A_u + \eta(E_{ui}B_i - \lambda A_u)$ 
   $B_i \leftarrow B_i + \eta(E_{ui}A_u - \lambda B_i)$ 
end

```

Algorithm 1: Incremental SGD [17].

To apply the ISGD algorithm, we need to find appropriate values for the parameters λ , η and q . For that, we have applied cross-validation. MyMediaLite [4] framework has been used to implement ISGD and evaluate results.

4.2 Feature retrieving

After obtaining the factor matrices A and B with ISGD, we apply FPMF (Algorithm 2) to retrieve the preferred feature's value (f) for each user.

To retrieve the value of feature (f) that a given user prefers, we generate a vector (Z) that represents a dummy item. We will now simulate an interaction between the dummy item and the target user, as follows.

First, Z is initialized to zeros. Then it is updated with re-

spect to its target user. After updating, in order to retrieve the preferred value for that user, we maximize the dot product between the lines of A corresponding to each feature and the dummy item Z . The result of each dot product provides the predicted rate (r_f) of the respective feature (f) for that target user. All parameters (λ , η , q) in Algorithm 2 are set the same as Algorithm 1.

Input : Target user u , A and B matrices, q
Output: Retrieved feature's value (f)

```

 $Z \leftarrow \mathbf{0}$  (Size of vector  $Z$ :  $q$ )
 $E_{uz} \leftarrow 1 - A_u \cdot Z^T$ 
 $A_u \leftarrow A_u + \eta(E_{uz}Z - \lambda A_u)$ 
 $Z \leftarrow Z + \eta(E_{uz}A_u - \lambda Z)$ 
Return  $f$  with the highest  $r_f = A_f \cdot Z$ 

```

Algorithm 2: Retrieving music feature's value.

5. RECOMMENDATION

In this section we suggest a content-based recommendation algorithm based on the ideas described above. In this case we use the vector p_u of predicted feature ratings r_f of a user u and recommend the items (music tracks) whose features g are most similar to the preferred features of the user.

In Algorithm 3, p_u are the features preferred by the user, generated by Algorithm 2 for a target user u . In algorithm 2 we only retrieved the feature with the highest rate but here we keep the values for all the features. The vector g represents the features of a music track. It can be a new track with no previous interactions with users or a known one. Similarity is calculated using the *Cosine Similarity* between p_u and each g .

Input : User u , p_u (features' vector of user u), g
(features' vector of an item),
 N_f (number of features)
Output: A set of recommendations for the target user u

```

if  $u \in U$  then
  for each  $g \in I$  do
    
$$\text{Cos}(\theta) = \frac{\sum_{n=1}^{N_f} p_{un} \cdot g_n}{\sqrt{\sum_{n=1}^{N_f} p_{un}^2} \cdot \sqrt{\sum_{n=1}^{N_f} g_n^2}}$$

  end
  Recommend top n music with the highest similarity
end

```

Algorithm 3: Recommendation Algorithm.

6. EVALUATION AND DISCUSSION

To evaluate our proposed method FPMF we used a music data set containing the music tracks playlisted by a collection of users.

6.1 Preprocessing

The original data set contains playlists of users from the Palco-Principal website (<http://www.palcoprincipal.com>), a

social network dedicated to music enthusiasts and artists. It contains 1968 users, 4748 music tracks and 27058 user-item interactions. Music tracks are also characterized with three features: “Band” (1603 values), “Genre” (30 values) and “Instrumental” (binary).

For each feature f we generated three separate datasets comprising the original users and items, plus the virtual users corresponding to the respective binarized feature f and the items they interact with. The resulting datasets are described in Table 1.

Sparsity is the percentage of non-existing user-item interactions, and is defined as follows.

$$Sparsity = 100 - \left(\frac{NumberofInteractions * 100}{NumberofUser * NumberofMusic} \right)$$

Table 1: Description of the data sets.

Feature	Val.	Inter actions	Users	Tracks	Sparsity
Band	1603	54116	3571	4748	99.68%
Genre	30	53888	1997	4731	99.42%
Instr.	2	53882	1960	4724	99.41%

After preprocessing, each dataset consists of a set of pairs in the form $\langle user, music \rangle$ corresponding to positive user-music interactions, followed by a set of pairs in the form $\langle feature\ value, music \rangle$ corresponding to virtual users (f).

6.2 Evaluation

In the following section we describe three sets of experiments. The first assesses the ability of FPMF to predict the favorite features of users, given interaction data. The second one compares FPMF with other methods in this task. In the third experiment we look at the possibility of using FPMF in an item cold start situation.

In these experiments we have used the following metrics to evaluate our proposal:

- *Recall*. Due to the large number of values (f) for Band (1603 values) and Genre (30 values), we used $Recall@N$ with $N \in \{1, 2, 3, 4, 5\}$ for these two features.
- Precision-Recall curve
- Average Precision (AP)

6.3 Results and discussion

In the first set of experiments we assess the ability of the proposed approach to correctly characterize the preferred features of a target user. For that, the $top\ n$ predicted feature’s values for a given user are compared with his/her favorite feature (the most frequent feature value observed in items he/she interacts with). If the favorite feature is included in the $top\ n$ predicted ones, it is considered as a hit. For example, if a user likes *Pop* music, and if *Pop* appears in the second prediction for this user, it means $Recall@1$ of this user is 0 while its $Recall@2$ is 1.

For “Instrumental”, which is a binary feature, the distribution of the two possible values is almost 50 % - 50 % (Table 2). FPMF correctly retrieved 59% of the values (Table

Table 2: Distribution of Instrumental feature.

	# of users	Distribution
Users liked Instrumental	975	49.82%
Users liked Non-Instrumental	982	50.18%
Total number of users	1957	100%

Table 3: Instrumental feature - Recall@1.

Total number of users	1957
Total Number of hits	1146
Recall	59%

3). This represents limited success in finding the interesting “Instrumental” feature’s value for the users.

For the other two features (“Band” and “Genre”) we applied FPMF and compared the results with the baseline algorithm *popular top 5* (recommending popular top 5 Genres or Bands). The results show that FPMF obtains clearly higher values of recall than the *top 5* baseline.

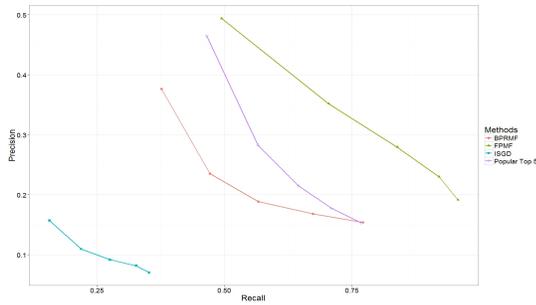
Table 4: Personalized Top5 (Recall@5) vs. Popular Top5.

		Personalized Top 5	Popular Top 5
Genre	Users	1967	1967
	Hits	1884	1472
	Recall	96%	76%
Band	Users	1968	1968
	Hits	1477	430
	Recall	76%	22%

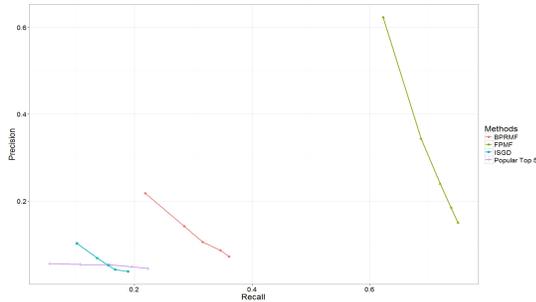
In the second set of experiments we compare the result of FPMF with ISGD and BPRMF. These two methods are trained using the $\langle user, feature's\ value \rangle$ interactions. We have also included *Popular top 5* in the comparison. Figures 2 and 3 show the results with respect to the Precision-Recall curve and $AP@N$ measure for the non-binary features (“Genre”, “Band”).

Figure 4 details the $Recall@5$ values obtained for the “Genre” and “Band”, as well as $Recall@1$ for the “Instrumental” feature. Here we can see the direct comparison of FPMF with the competing methods under these settings.

In order to assess our method with respect to the *Item Cold Start* problem, we added several new artificial music tracks per feature value (two artificial tracks per feature value for “Genre” and “Band” and twenty artificial music per feature value for “Instrumental” since this one is a binary feature), with the form of $\langle feature\ value, track \rangle$, to the extended dataset. This means we had the values of the features for the new tracks, but no data about the interaction with users. We then applied FPMF for learning the model and generated recommendations for each user. Finally, we counted the number of times a new music was correctly recommended to a user (i.e. the feature’s values of a new music is compatible with the user preference). Table 5 shows the percentage of new tracks that were correctly recommended, taking into account the favorite features of users. This indicates that FPMF is able to include previously unseen items in the produced recommendations thus alleviating the item cold start problem.

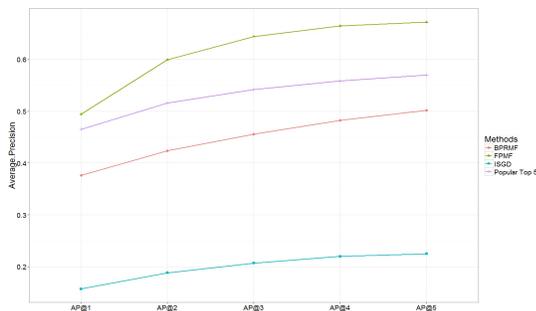


(a) Genre

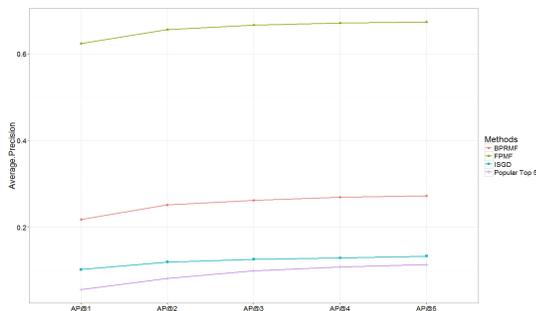


(b) Band

Figure 2: Precision-Recall curves for evaluated methods for the non-binary features “Genre” and “Band”. The number of predicted features ranges in 1..5.



(a) Genre



(b) Band

Figure 3: $AP@N$ curves for evaluated methods for the non-binary features “Genre” and “Band”. The number of predicted features ranges in 1..5.

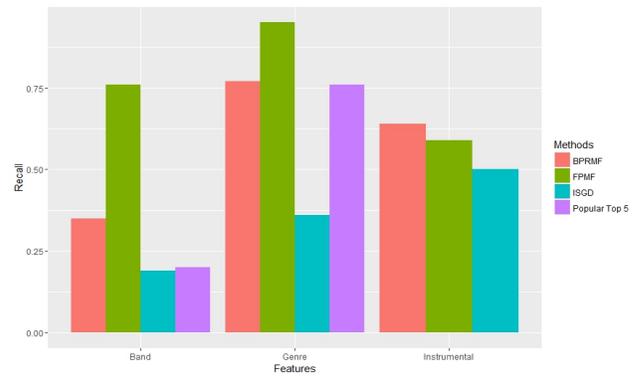


Figure 4: Comparison of FPMF with ISGD[17], BPRMF[13] and Popular top 5. “Band” and “Genre” are appraised by $Recall@5$. “Instrumental” is evaluated by $Recall@1$.

Table 5: Item Cold Start evaluation. “Genre” and “Band” are evaluated by $Recall@5$ and “Instrumental” is appraised by $Recall@1$.

	Recall@5		Recall@1
	Band	Genre	Instrumental
New tracks	54.80%	49%	30%

7. CONCLUSION

In this paper, we propose the method FPMF to retrieve preferred features for users. Our approach works by modeling both usage data and item description features with an off-the-shelf incremental matrix factorization algorithm. The method is implemented by considering feature’s values of items as virtual users and retrieving the preferred feature’s values for target users.

FPMF was successful in the task of retrieving preferred features in a music data set. We have empirically compared the results of FPMF with three other methods (ISGD, BPRMF, Popular top 5). The method can potentially be applied in *item cold start* recommendation scenarios. It can also be used to explain the recommendations (why a specific music is recommended), which can increase the users’ satisfaction and trust. In future, we plan to extend our work for the *user cold start* situation and perform a comprehensive evaluation of this issue.

8. ACKNOWLEDGMENTS

Project “TEC4Growth - Pervasive Intelligence, Enhancers and Proofs of Concept with Industrial Impact/NORTE-01-0145-FEDER-000020” is financed by the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, and through the European Regional Development Fund (ERDF). The first author thanks the National Institute of Informatics (NII) in Tokyo, Japan for the internship and hospitality.

9. REFERENCES

- [1] M. Balabanović and Y. Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.
- [2] M. W. Berry, Z. Drmac, and E. R. Jessup. Matrices, vector spaces, and information retrieval. *SIAM review*,

- 41(2):335–362, 1999.
- [3] Z. Gantner, L. Drumond, C. Freudenthaler, S. Rendle, and L. Schmidt-Thieme. Learning attribute-to-feature mappings for cold-start recommendations. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 176–185. IEEE, 2010.
- [4] Z. Gantner, S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. Mymedialite: A free recommender system library. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 305–308. ACM, 2011.
- [5] Y. Hang, Y. Hang, and W. Xingwei. A cold-start recommendation algorithm based on new user’s implicit information and multi-attribute rating matrix. In *Hybrid Intelligent Systems, 2009. HIS’09. Ninth International Conference on*, volume 2, pages 353–358. IEEE, 2009.
- [6] L. Hu, J. Cao, G. Xu, L. Cao, Z. Gu, and C. Zhu. Personalized recommendation via cross-domain triadic factorization. In *Proceedings of the 22nd international conference on World Wide Web*, pages 595–606. International World Wide Web Conferences Steering Committee, 2013.
- [7] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434. ACM, 2008.
- [8] Q. Li, S. H. Myaeng, and B. M. Kim. A probabilistic music recommender considering user opinions and audio features. *Information processing & management*, 43(2):473–487, 2007.
- [9] A. Micarelli, F. Sciarbone, and M. Marinilli. Web document modeling. In *The adaptive web*, pages 155–192. Springer, 2007.
- [10] A. Mnih and R. Salakhutdinov. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264, 2007.
- [11] S. A. P. Parambath. Matrix factorization methods for recommender systems. umea university, faculty of science and technology, department of computing science, master thesis. 2013.
- [12] S.-T. Park and W. Chu. Pairwise preference regression for cold-start recommendation. In *Proceedings of the third ACM conference on Recommender systems*, pages 21–28. ACM, 2009.
- [13] 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*, pages 452–461. AUAI Press, 2009.
- [14] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Application of dimensionality reduction in recommender system—a case study. Technical report, DTIC Document, 2000.
- [15] R. Sinha and K. Swearingen. The role of transparency in recommender systems. In *CHI’02 extended abstracts on Human factors in computing systems*, pages 830–831. ACM, 2002.
- [16] S. Spiegel, J. Kunegis, and F. Li. Hydra: a hybrid recommender system [cross-linked rating and content information]. In *Proceedings of the 1st ACM international workshop on Complex networks meet information & knowledge management*, pages 75–80. ACM, 2009.
- [17] J. Vinagre, A. M. Jorge, and J. Gama. Fast incremental matrix factorization for recommendation with positive-only feedback. In *User Modeling, Adaptation, and Personalization*, pages 459–470. Springer, 2014.
- [18] Y. Zhang, M. Zhang, Y. Liu, S. Ma, and S. Feng. Localized matrix factorization for recommendation based on matrix block diagonal forms. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1511–1520. International World Wide Web Conferences Steering Committee, 2013.
- [19] K. Zhou, S.-H. Yang, and H. Zha. Functional matrix factorizations for cold-start recommendation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 315–324. ACM, 2011.