Expert Systems With Applications 106 (2018) 244–251

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Tag-aware dynamic music recommendation

Ervine Zheng, Gustavo Yukio Kondo, Stephen Zilora, Qi Yu*

College of Computing and Information Sciences Rochester Institute of Technology Rochester, NY 14623, United States

ARTICLE INFO

Article history:
Received 25 September 2017
Revised 9 April 2018
Accepted 10 April 2018
Available online 11 April 2018

Keywords:
Music recommendation
Matrix factorization
Temporal dynamics
Tag aware

ABSTRACT

We present a tag-aware dynamic music recommendation framework that achieves personalized and accurate music recommendations to users. The proposed framework leverages the available semantic labels (in terms of tags) of music tracks to complement a highly sparse user-item interaction matrix, which effectively addresses the data sparsity issue faced by most music recommendation systems. Music tracks are more accurately represented by aggregating the latent factors derived from both the tag space and the user interaction information. The proposed framework further employs a Gaussian state-space model to capture the evolving nature of users' preferences over time, which helps achieve time-sensitive recommendation of music. A variational approximation is developed to achieve fast inference and learning of model parameters. Experiments conducted using actual music data and comparison with state-of-the-art competitive recommendation algorithms help demonstrate the effectiveness of the proposed framework.

1. Introduction

The market share of digital music has been growing quickly as a result of technological innovation on storage and compression. According to the 2017 IFPI Global Music Report, the digital music category accounted for 50% of the total revenue generated by recorded music in 2016, and the current trend of users choosing digital format over physical format is likely to grow. Today, mobile devices are available to the general public with both the capacity of storing thousands of tracks and also music applications that provide vast collections of music free of charge or at reasonable prices. It is possible for users to access a huge number of tracks and artists, and for the artists to reach a larger audience. Moreover, the digital music market now contains massive amounts of data consisting of information about tracks, artists, users, and interactions thereof. Effectively leveraging the data will help deliver personalized music recommendation to each user based on his/her preference and consequently help promote more artists who are new to the industry.

Collaborative filtering has emerged as an important technique for building modern recommender systems. Evidence across multiple domains show that customers who have interacted (e.g., bought, listened, watched) with particular products tend to choose similar products in the future. This forms the basic working rationale of collaborative filtering based recommender systems. Existing collaborative filtering algorithms can be broadly categorized into memory-based approaches and latent-factor models. The former group relies on the identification of a small number of users/items that are similar to the target user/item and uses their historical interaction information for prediction purpose. These approaches are usually sensitive to the sparsity of the user-item interaction matrix as a more sparse matrix will reduce the chance of finding a high-quality neighborhood of users/items. The latent factor models partly address this issue by leveraging the global structure of the interaction matrix to discover a smaller number of latent factors that capture the inherent attributes of all the items as well as the users' preferences for these attributes.

Given that a typical user plays only a small portion of tracks in the music library and a typical track is played only by a small number of users, data sparsity poses a fundamental issue for music recommendation. Furthermore, most existing recommender systems assume the interaction pattern between users and items does not change. Nonetheless, in reality, users' preference may change and artists' music style may also evolve. It is important to capture those temporal dynamics in order to provide time-sensitive music recommendations to users.

In this paper, we propose a music recommendation framework to address the challenges as outlined above. The framework analyzes both the play counts data over time and the tags assigned to tracks. The play counts are referred to as *implicit data* because it does not require users to rate the tracks they listen to, and thus making it more available than explicit rating data, which is typically used by most recommender systems. Additionally, tag data is integrated into the model in order to indicate the impor-

* Corresponding author.
E-mail addresses: mxz5733@rit.edu (E. Zheng), gyk2849@rit.edu (G.Y. Kondo), stephen.zilora@rit.edu (S. Zilora), qi.yu@rit.edu (Q. Yu).

https://doi.org/10.1016/j.eswa.2018.04.014
0957-4174/

tant semantic features of tracks. The proposed approach combining user-track interaction and tags effectively addresses the data sparsity issue. To capture the changes of user and item features over time, the proposed music recommendation framework systematically models the temporal dynamics by introducing a dynamic framework that integrates low-rank matrix factorization with a linear-Gaussian state-space model. The proposed framework is able to discover the latent factors and their temporal evolution to capture the changing user preference over items of interest. We summarize our major contributions as follows:

- A tag model that is based on Logistic Principle Component Analysis (LPCA) and integrates the semantic information of tracks encoded in the tags.
- A Gaussian state-space model coupled with low-rank matrix factorization that captures the latent factors corresponding to users' evolving music preference over time.
- A variational approximation algorithm to achieve fast inference and learning of model parameters.
- Extensive experiments using actual data from a well-known music dataset (i.e., LastFM) and comparison with state-of-the-art recommendation algorithms to evaluate the effectiveness of the proposed recommendation framework.

The remainder of the paper is organized as follows. We give an overview of related works in Section 2. In Section 3, we present the technical background of general recommendation systems. In Section 4, we discuss in detail the proposed music recommendation framework. In Section 5, we show our experimental results and evaluate the effectiveness of the proposed model. We conclude in Section 6.

2. Related works

Recommendation using implicit data has received increasing attention in recent years. Several ways have been developed to capture the unique property of implicit data. The counts of user-item interaction over a fixed period can be considered an indicator of users' interest on items, and modeled with Poisson distribution (Gopalan, Hofman, & Blei, 2013). Some research works propose binary transformation on counts, arguing that counts indicate the confidence of observing user-item interactions (Hu, Koren, & Volinsky, 2008). Other research works convert implicit counts to ratings by calculating the frequency of a user interacting with an item among all items and implements linear transformation (Sánchez-Moreno, González, Vicente, Batista, & García, 2016). Alternatively, the user-item interactions over time can be viewed as a stochastic counting process (Du, Wang, He, Sun, & Song, 2015).

To incorporate temporal dynamics into recommendation systems, dynamic matrix factorization has been developed which allows latent features to change over time. A typical example is the SVD++ model (Koren, 2010), which includes a bias term to capture user related temporal changes. Several other variants have also been developed by modeling user-item-time tensor (Xiong, Chen, Huang, Schneider, & Carbonell, 2010). While these works successfully model time-specific changes, they do not provide a systematic way to capture the evolving nature of user preferences. We assume that while a user's current preference may be different than a previous one, the new preference will have evolved from and hence relates to the former one. The Markov structure, including Hidden Markov Model (HMM) (Sahoo, Singh, & Mukhopadhyay, 2010) and Linear Dynamical System (LDS) (Gultekin & Paisley, 2014), has been employed to model the time-evolving changes. Similarly, a dynamic Poisson factorization model has been developed (Charlin, Ranganath, McInerney, & Blei, 2015).

Collaborative filtering algorithms usually fail to make accurate recommendations when the observed interaction is only a small portion of the user and item pool, which is referred to as data sparsity problem (Kim, Choi, Han, Man, & Wan, 2014). Various approaches have been developed to address this problem (Wu, Chen, Yu, Han, & Wu, 2015; Zhang, Cheng, Qiu, Zhu, & Lu, 2015) by incorporating additional information into the system. Content-based recommendation uses topic modeling as a way to uncover users' preference for latent topics by mining text such as users' reviews and comments (Liu, Liu, Shen, & Li, 2017).

Except for texts, items' tags can also be used as supplemental meta-data to uncover item attributes. One way to explore the relationship between users, items, and tags is to introduce a three-dimensional tensor as model input with the context feature as the third dimension (Tso-Sutter, Marinho, & Schmidt-Thieme, 2008). However, given that tags are usually arbitrary words assigned by users, using the original tag space significantly increases the computational cost. An alternative way is using tags to measure similarity among different users and augmenting a conventional matrix factorization technique (Ma et al., 2016). Other research works treat tags as item attributes in collaborative filtering algorithms to analyze the collection of user-context relationships for recommendation (Alhamid, Rawashdeh, Dong, Hossain, & El Saddik, 2016). However, the information contained in tags does not cover all of the items' attributes. The proposed framework in this paper integrates tag information and temporal dynamics in user-item interaction for personalized music recommendation. Integration of tag information can more effectively address the sparsity in the user-item interaction matrix, which is common in the music domain (and many other domains as well). By systematically modeling the changing behavior of users, the proposed framework is expected to provide time-sensitive and hence more accurate music recommendation.

3. Preliminaries

In this section, we provide an overview of some fundamental collaborative filtering techniques and discuss how they are applied in recommendation systems.

3.1. Memory-based collaborative filtering

As one the most popular techniques to build recommendation systems, collaborative filtering leverages existing interactions between users and items to predict unknown interactions. The existing interaction records are typically ratings provided by users, which can be represented as a matrix $X_{U\times I}$ encoding the interactions between I items and U users. It should be noted that X is usually sparse as each user may only interact with a small number of items. The goal of collaborative filtering is to leverage the observed entries in X to predict the missing entries (potential items previously unknown to users).

The memory-Based collaborative filtering approaches pick the top K users (or items) similar to a target user (or item) and use the existing interaction data to make recommendations. Memory-based collaborative filtering is further divided into user-based and item-based approaches depending on whether the chosen targets are users or items. The following discussion is based on the user-based approach but it can be similarly applied to the item-based approach. Given user u, user v and a set of items $\mathcal{I}$, we consider the following similarity measures:

- *Cosine Similarity* measures the similarity between two vectors, defined as

$$cos(u,v) = \frac{u \cdot v}{\| u \| \| v \|} = \frac{\sum_{i \in I} x_{u,i} x_{v,i}}{\sqrt{\sum_{i \in I} x_{u,i}^2} \sqrt{\sum_{i \in I} x_{v,i}^2}} \quad (1)$$

- *Pearson Correlation Coefficient* computes the correlations between users u and v, defined as

$$corr(u,v) = \frac{cov(u,v)}{\sigma_u \sigma_v} = \frac{\sum_{i \in I}(x_{u,i} - \bar{x}_u)(x_{v,i} - \bar{x}_v)}{\sqrt{\sum_{i \in I}(x_{u,i} - \bar{x}_u)^2} \sqrt{\sum_{i \in I}(x_{v,i} - \bar{x}_v)^2}} \quad (2)$$

where $x_{u,\,i}$ denotes the interaction between item i and user u, $r_{v,\,i}$ denotes the interaction between item i and user v, and $\overline{x_u}$ and $\overline{x_v}$ denote the means of user u's and v's interactions, respectively.

By leveraging users' existing interactions with items, the users' potential interactions with previously unknown items are predicted as

$$\widehat{x}_{u,i} = \bar{x}_u + \frac{\sum_{v \in S(u)}^{k} sim(u,v)(x_{v,i} - \bar{x}_v)}{\sum_{v \in S(u)}^{K} sim(u,v)} \quad (3)$$

where $\widehat{x}_{u,i}$ is the predicted interaction between user u and item i, $\bar{x}_u$ is the mean of user u's interactions, and $S(u)^K$ is the set of K most similar users to user u.

3.2. Model-based collaborative filtering

Model-based CF approaches use the $X_{U \times I}$ matrix encoding the interactions between I items and U users to build a model to predict missing entries. It usually implements the low-rank matrix factorization technique with latent features. In other words, it decomposes the original matrix into a user matrix $R_{U \times K}$ and an item matrix $C_{I \times K}$, where K is the number of latent features. The (u, k) entry in matrix R denotes the relevancy of latent variable k to user u and the (k, i) entry in matrix C denotes the relevancy of latent variable k to item i. The user-item interaction is predicted by the dot product of user's preference and item's attributes over a number of latent features, and the objective of model-based CF is to approximate X by finding the R and C that best explain X.

It is a common practice for Bayesian treatment to assume a Gaussian prior with zero-mean for the user and item features $R_{u,\,k} \sim N(0,\,\sigma_R)$ $\quad C_{i,\,k} \sim N(0,\,\sigma_C)$, and to assume a Gaussian noise of observation $X_{u,\,i} \sim N(\Sigma_k R_{u,\,k} C_{i,\,k},\,\sigma_X)$. The log-likelihood function is optimized with respect to latent preferences and attributes.

$$L = \sum_{u,i} \ln p(X_{u,i}|R, C, \sigma_x) + \sum_{u,k} \ln p(R_{u,k}|\sigma_R) + \sum_{i,k} \ln p(C_{i,k}|\sigma_C)$$

Maximizing the log-likelihood is equivalent to minimizing the cost function $F(R, C)$ which is the squared Frobenius normal of the difference between the input matrix and the predicted matrix, plus two regularization terms to address over-fitting:

$$F(R,C) = \|O \circ (X - RC)\|^2 + \lambda_R \|R\|^2 + \lambda_C \|C\|^2$$

where $O_{U \times I}$ is an indicator matrix, with $O(u,i) = 1$ if $X_{u,\,i}$ is an observed data entry and $O(u,i) = 0$ if otherwise. The widely used techniques to minimize the cost function $F(R, C)$ include gradient descent and multiplicative update with non-negative constraint.

The proposed music recommendation framework is based on the matrix factorization techniques but makes significant extensions by allowing the discovered latent factors to evolve over time to capture the changes of users and their interactions with items. Tag information is further integrated to more effectively address data sparsity.

4. The music recommendation framework

In this section, we present the proposed music recommendation framework that integrates tag-aware analysis with a state-space model coupled with matrix factorization.

4.1. Modeling the tags

Tags are labels assigned to items and provided by users. Many users tag the tracks they have played to help them mark and categorize tracks. An illustrative example of the most popular tags is provided in Table1. Rock, electronic and alternative are music genres, while female vocalist, singer-songwriter and Japanese describe artists. However, compared with formal labels provided by domain experts, tags are arbitrary words created by public users and thus less structured. The tags assigned to the same track may vary from user to user, and different tags may not be mutually exclusive. Therefore, tags cannot be simply considered as item attributes and should be structured before integrating into the recommendation system.

Dimensionality reduction is one of the ways to structure tags. The tag space is constructed by collecting popular tags assigned to tracks by all users. Tags assigned to each track can be represented as a binary vector, and those vectors are usually sparse with most of the entries as 0s. We compress the original tag space in which 98.1 percent of the entries are 0s into a compact representation by using Logistic Principal Component Analysis (LPCA). Similar to conventional Principal Component Analysis (PCA), LPCA is a way to implement dimensionality reduction. However, the conventional PCA assumes Gaussian noise over a set of observations, while LPCA assumes Bernoulli noise. Since the tag vectors are binary, LPCA is a natural way for compression.

We use an approach to implement LPCA proposed by Schein, Saul, and Ungar (2003), and briefly discuss the algorithm as follows make this paper self-contained. The input of LPCA is the original tag space over a large number of tracks, represented as a matrix $T_{I \times D}$ where D is the total number of popular tags and I is the number of tracks. Each entry in the input matrix is assumed a Bernoulli variable,

$$P(T_{i,d}|p) = p^{T_{i,d}}(1-p)^{1-T_{i,d}} = \sigma(V)^{T_{i,d}} \sigma(-V)^{1-T_{i,d}}$$

where $\sigma(.)$ is the logistic Sigmoid function that satisfies $\sigma(V) = p$ and $V = \ln(p/(1-p))$. By doing so, the original binary matrix is converted into a real matrix. Assume that the tag matrix T can be factorized by the product of two low-rank matrices

$$v_{i,d} = \sum_{k_1} g_{i,k_1} w_{k_1,d}$$

where $G_{N \times K_1}$ is the coefficient matrix, and $W_{K_1 \times D}$ is the basis matrix. The output of LPCA is a compressed coefficient matrix C and the number of output features $K_1 \ll D$. To make inference, a tight lower bound of the log-likelihood is introduced

$$\begin{aligned} \ln p(T|V) \geq & \sum_{i,d} \frac{1 - 2\sigma(u_{i,d})}{4u_{i,d}}(v_{i,d}^2 - u_{i,d}^2) \\ & + (T_{i,d} - \frac{1}{2})v_{i,d} + \ln\sigma(u_{i,d}) - \frac{u_{i,d}}{2} \\ = & Q(U,V) \end{aligned}$$

where $u_{i,d} = v_{i,d}^{old}$ is an auxiliary variable. The coefficient matrix and the basis matrix are updated by letting the partial derivatives $Q(U,V)$ with respect to G and W be 0 iteratively.

While tags contain semantic information of tracks, they may not completely capture all the important properties of tracks. For example, two tracks with similar music styles may be preferred by different users. We propose to capture the user-related properties

Table 1
Popular tags.

Popular tag	Rock	Electronic	Alternative	Female vocalist	Singer-songwriter	Japanese
Total count	3,919,087	2,319,846	2,062,767	1,580,760	820,811	446,106

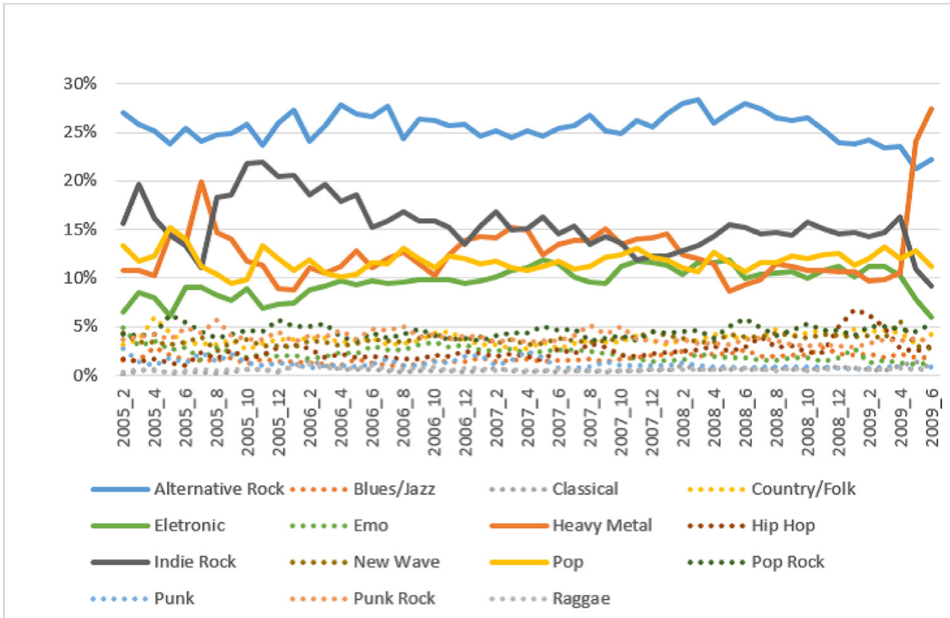

Fig. 1. An example user's preference over time.

of tracks by performing low-rank matrix factorization of the user-item interaction matrix. To accommodate the temporal dynamic of user preferences, we couple matrix factorization with Gaussian state-space model with a first-order Markov structure to properly model the preference evolution of users.

4.2. Modeling the user-item interactions

We first use an example to illustrate the change of users' music preference over time. As shown in Fig. 1, the sample user's favorite music style remained almost the same until April 2009. After that, it significantly changed from alternative rock to heavy metal. The user's preference for Indie rock was strong in late 2005 but diminished afterwards. The preference for electronic music indicates an opposite trend. In addition, the preference for heavy metal seems to oscillate a lot.

To capture these temporal dynamics, we propose a state-space model to simulate the shift of users' preference, users' activity and items' popularity. We represent each user as a user latent vector $\mathbf{r}_u \in \mathbb{R}^K$ to quantify the user's preference, and each item as an item latent vector $\boldsymbol{c}_i \in \mathbb{R}^K$ to quantify the item's attributes, including tag-related and tag-unrelated features. We apply a logarithm transformation $x_{u,i} = \ln(y_{u,i})$ to quantify the user-item interactions.

The shift of user preference is captured by a state-space model with the first-order Markov structure. For user u,

$$r^1_{u,k} \sim N(0, \zeta) \quad r^t_{u,k} \sim N(r^{t-1}_{u,k}, \eta)$$

Items' attributes tend to change more slowly over time than user preference, and thus are assumed stable. Item's vector is the concatenation of the compressed tag-related features and other latent features unexplained by tags. Each item's attribute is represented as a K vector where $K = K_1 + K_2$. The first K_1 entries latent features explained by tags. For the K_2 latent features unexplained by tags of item i,

$$c_{i,k_1} \sim N(\alpha g_{i,k_1}, \beta) \quad c_{i,k_2} \sim N(0, \phi)$$

where α is a scaling factor. Users' bias is modeled with a time-specific bias. For implicit data, a user's activity may be influenced by his monetary and/or time budget that is likely to change over time.

$$b^t_u \sim N(0, \rho)$$

Similarly, the items' bias is modeled with a time-specific bias.

$$p^t_i \sim N(0, \upsilon)$$

Finally, $x^t_{u,i}$ is drawn as

$$x^t_{u,i} \sim N(b^t_u + p^t_i + \sum_k c_{k,i} r^t_{u,k}, \sigma)$$

4.3. Variational approximation

An exact inference of the proposed model may not be efficient because of the coupling of variables. We adopt a variational Bayesian inference technique to achieve fast inference (Bishop, 2006). Let Z denote all latent variables and θ denote all hyper-parameters, the variational method seeks to maximize the lower bound L,

$$\begin{aligned} \ln p(X|\theta) \geq L &= \int dZ q(Z)[\ln p(X, Z|\theta) - \ln q(Z)] \\ &= E_q[\ln p(X, Z|\theta)] + H[q(Z)] \end{aligned}$$

where $q(Z)$ is the variational distribution and $H[q(Z)]$ is the entropy. Consider a variational distribution that can be factorized as

$$q(Z) = \prod_{u,k,t} q(\boldsymbol{r}^t_{u,k}) \prod_{i,k_1} q(\mathbf{c}_{i,k_1}) \prod_{i,k_2} q(\mathbf{c}_{i,k_2}) \prod_{u,t} q(b^t_u) \prod_{i,t} q(p^t_i)$$

Table 2
Summary statistics of data.

Statistics	Minimum	1st Quarter	Median	Mean	3rd Quarter	Maximum
Users per track	1	1	1	4.124	3	346
Tracks per user	2	1158	2720	3997	5176	59,870

For each latent variable Z_i, the optimal variational distribution is given by

$$\begin{aligned}\ln q^*(Z_i) &= E_{q(Z_j, j\neq i)} \ln p(X, Z|\theta) \\ &= E_{q(Z_j, j\neq i)} \ln p(X|Z) + E_{q(Z_j, j\neq i)} \ln p(Z|\theta)\end{aligned}$$

To infer the posteriors parameters of users preference, items attributes, users' budget and items' popularity, Gradient ascent algorithm is applied to optimize the evidence lower bound with respect to those parameters. More specifically, the evidence lower bound of variational approximation is written as follows

$$\begin{aligned}L = &\sum_{u,k} E_q[\ln p(r^1_{u,k}|\zeta)] + \sum_{u,k,t>2} E_q[\ln p(r^t_{u,k}|r^{t-1}_{u,k}, \eta)] \\ &+ \sum_{i,k_1} E_q[\ln p(c_{i,k_1}|\beta) + \sum_{i,k_2} E_q[\ln p(c_{i,k_2}|\phi)] \\ &+ \sum_{u,t} E_q[\ln p(b^t_u|\rho)] + \sum_{i,t} E_q[\ln p(p^t_i|\upsilon)] \\ &+ \sum_{u,i,t} E_q[\ln p(X^t_{u,i,t}|Z)] + H(q(Z))\end{aligned}$$

where the entropy term $H(q(Z))$ is expanded as

$$\begin{aligned}H(q(Z)) = &\sum_{i,k_1} H(q(c_{i,k_1})) + \sum_{i,k_2} H(q(c_{i,k_2})) + \sum_{u,k,t} H(q(r^t_{u,k})) \\ &+ \sum_{i,t} H(q(p^t_i)) + \sum_{u,t} H(q(b^t_u))\end{aligned}$$

The posterior parameters of $r^t_{u,k}$, $c_{i,\,k}$, b^t_u and p^t_i are updated iteratively to maximize L.

5. Experiments

We conducted extensive experiments using Last.fm data set to demonstrate the effectiveness of the proposed dynamic music recommendation framework.

The entire data set consists of a total of 19,150,868 records from 992 users and 176,968 artists. As shown in Table 2, the data set is sparse because at least half of the tracks were played by only one user and three-fourths of the tracks were played by three users or fewer. The interaction data has been processed by keeping only users who listened to no fewer than 100 distinct tracks and tracks that were listened to by no fewer than 25 users. Records where a user played a track only once are removed, because such interactions possibly happen by chance. Tag data is retrieved through API provided by Last.fm Web Services. The frequency of tags used by users is highly skewed. The interaction data has been processed by keeping only popular tags that have been assigned to more than 4 tracks.

The training-testing process is designed to best reflect reality. In practices, recommendation systems make predictions for the future based on historical data. Therefore, we sliced the timeline from 2005 to 2008 into 8 periods, with the granularity set to 6 months. For each period t, all user-item interactions from period 1 to $t-1$ are used for training, and all interactions within t are used for testing. This process is repeated for each t, and the results are averaged. Parameters including ζ, η, β, ϕ, ρ and υ are tuned through 5-fold cross-validation.

5.1. Evaluation measures

For model effectiveness evaluation, we analyze the experiment results in terms of both the deviation of predicted values from ground truth and the errors of ranking sequences. We use Root Mean Squared Error (RMSE) and Normalized Discounted Cumulative Gain (NDCG) averaged across all users. RMSE quantifies the deviation of predicted values from ground truth, while NDCG measures the quality of predicted ranking from ideal ranking.

$$RMSE = \sqrt{\sum_{x_{u,i}\in O} (\hat{x}_{u,i} - x_{u,i})^2/|O|}$$

where O is the observation set for testing.

$$NDCG = \sum_n \frac{rel^{pred}_n}{log_2(1+n)} \Big/ \sum_n \frac{rel^{ideal}_n}{log_2(1+n)}$$

where rel_n is relevancy of nth item in ranking sequence for user u. For implicit data rel is binary and for explicit data rel is the ratings given by users. To penalize the negative feedback, the ratings are linearly mapped to a range of $[-1,1]$. The NDCG is the fraction of Discounted Cumulative Gain (DCG) of recommendation result over the ideal DCG. It should be noted that a user may be in favor of an item categorized as a false positive recommendation. Therefore, the testing set is usually a subset of true relevant items that users may be interested in.

5.2. Algorithms for comparison

We compare the proposed algorithm with a number of representative dynamic recommendation models, which are described below. Optimal model orders were determined by 5-fold cross-validation.

- Time SVD++ (Koren, 2010) applies singular value decomposition and augments data with bias terms to model users' activity and items' popularity, and time-specific factors to capture the change of users' interest. A logarithm transformation is applied because Time SVD++ is not specifically designed for implicit data.
- Collaborative Kalman Filtering (CKF) (Sun, Parthasarathy, & Varshney, 2014) simulates users' preference over time with a Gaussian state-space model. A linear dynamical system is used to make Bayesian inference. Similarly, a logarithm transformation is applied because CKF is not specifically designed for implicit data.
- Dynamic Poisson Factorization (DPF) (Charlin et al., 2015) is a dynamic probabilistic model designed for implicit data. Both user factors and item factors are allowed to evolve over time and simulated with Gaussian state-space. Poisson emission process is applied to model the observations.

5.3. Performance

We report the results of experiments in Figs. 2. As people are more likely to be interested in items on the top end of the full recommendation list, we focus on the quality of recommendation for those items. We vary the size of recommendation list N to study its influence on NDCG, and analyze the RMSE of the top X% predictions against the ground truth.

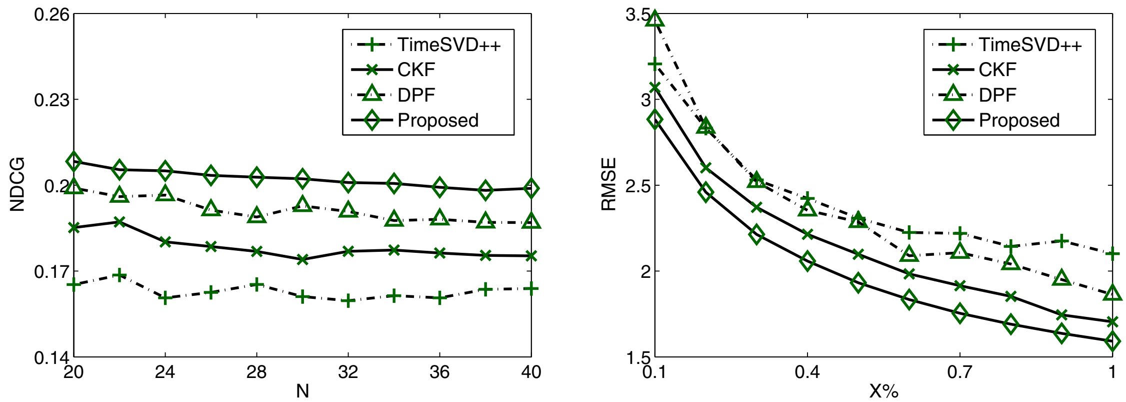

Fig. 2. Evaluation of recommendation result.

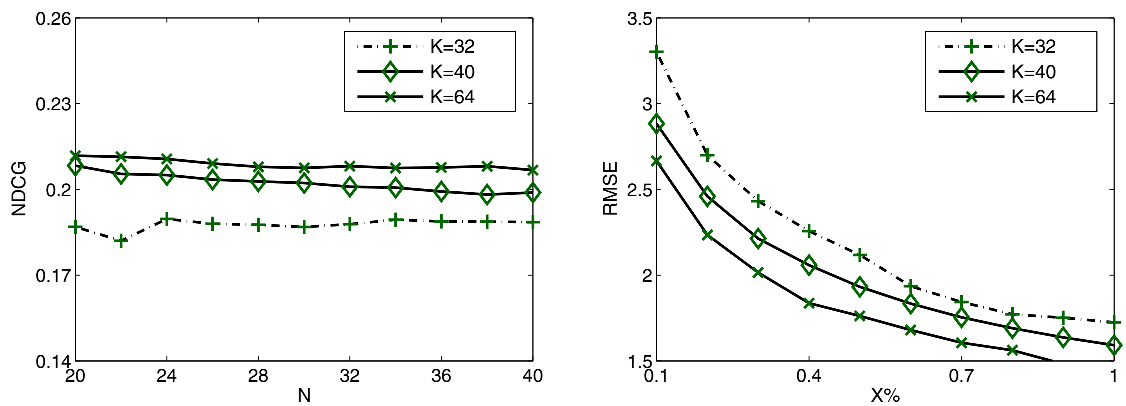

Fig. 3. Influence of model order (K).

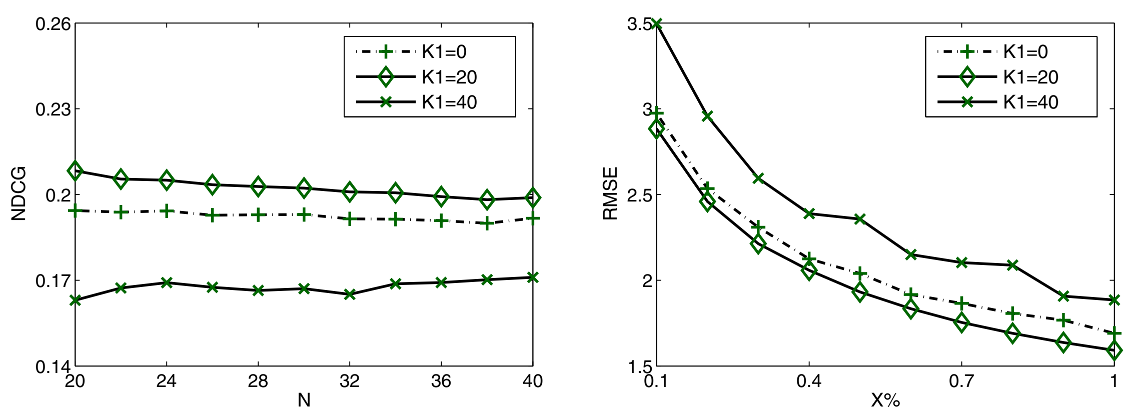

Fig. 4. Influence of tag-related features (K_1).

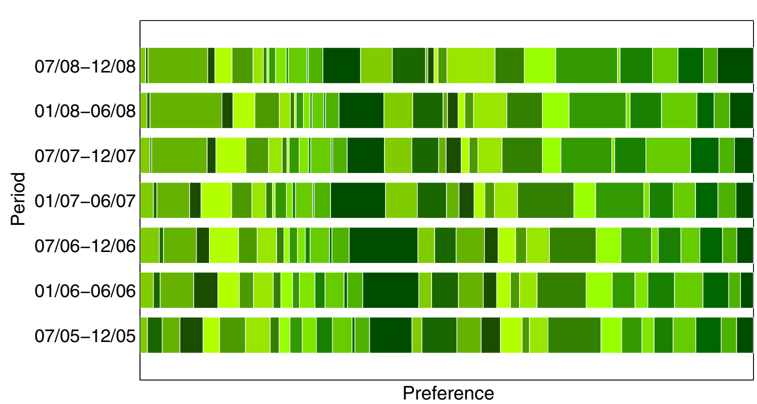

Fig. 5. Preference change of a sample user.

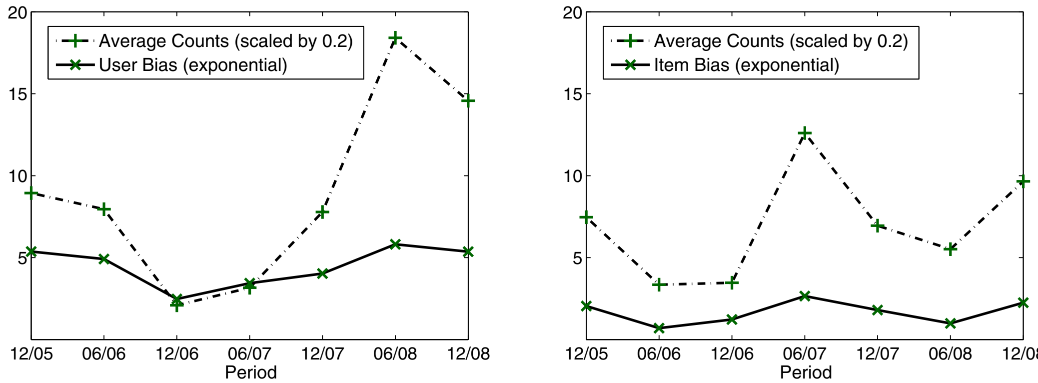

Fig. 6. Play count change of a sample user and a sample track.

Among all the approaches, the proposed model gives the best result in terms of NDCG and RMSE. Possible explanations are:

- Matrix factorization algorithms based on singular value decomposition usually converge at local optima, especially when the input matrix is sparse. In this case, recommendation based solely on user-item interaction data may not be consistent. The proposed algorithm integrates tag information to more accurately model the item attributes, and thus improve the quality of recommendation.
- Many users' play counts change significantly over time. The variation of the total play counts for a given user may be affected by his monetary and/or time budget. Our proposed model is able to capture part of that by modeling those latent factors via a bias term.

5.4. Discussion

In the proposed model, the item vector is represented as a concatenation of tag-related features and other features unexplained by tags. We study the influence of model order. Holding the number of tag-related features static (i.e. $K = 24$), we vary the model order (K) from 32 to 64 as shown in Fig. 3. We also study the influence of tag-related features to investigate its effect on the quality of recommendation. Holding the total number of latent features static (i.e. $K = 40$), we vary the number of tag-related (K_1) features from 0 to 40 as shown in Fig. 4. When $K_1 = 0$, no tag-related feature is used; in this case the model becomes a basic dynamic matrix factorization.

Those results indicate that integrating tag-aware analysis improves the quality of recommendation. However, tags do not represent all of the item attributes, and therefore a model should also incorporate tag-unrelated features and balance between the two types of features.

An illustrative example is presented to show how the proposed model uncovers the change of users' preference and budget, as well as items' popularity. We set the number of tag-related features to 20 and the total number of features to 40. We pick a sample user (ID 25) and a sample track ("Automatic Stop").

Fig. 5 shows how this user's preference for different latent features changes dynamically and is captured by the proposed model. The first 20 are the preference for tag-related latent features (on the left side). Note that latent preference factors are not semantically annotated because they can be the common component of multiple music styles. Fig. 6 shows how the average play counts of the user over relevant tracks and play counts of the track from relevant users change dramatically over time. The user bias is able to capture this user's budget, and the item bias is able to capture this track's popularity.

6. Conclusions and future work

We present a dynamic music recommendation framework that integrates tag information of music tracks with temporal dynamics in user-item interaction for personalized music recommendation. The user-item interaction matrix is complemented by tags, which carry important semantic information of tracks, to address the data sparsity issue. A Gaussian state-space model coupled with low-rank matrix factorization is developed to successfully encode the evolution of users' preferences, making the overall recommendation time-sensitive and hence more accurate. A fast variational inference algorithm is developed to achieve efficient and scalable training of the model. Experiments have been conducted over a large-scale real-world music data set and demonstrate the effectiveness of the proposed music recommendation framework.

An interesting future direction is to apply online learning techniques, because live data sets normally grow continuously in real-time. It is more efficient to update the model of a recommendation system rather than reconstructing the entire model when new data is added. Moreover, it is suggested to generalize the proposed tag-aware recommendation algorithm to domains other than music, because tagging is widely applied in digital products and e-commerce.

References

Alhamid, M. F., Rawashdeh, M., Dong, H., Hossain, M. A., & El Saddik, A. (2016). Exploring latent preferences for context-aware personalized recommendation systems. *IEEE Transactions on Human-Machine Systems, 46*(4), 615–623.

Bishop, C. (2006). *Pattern recognition and machine learning*. Springer.

Charlin, L., Ranganath, R., McInerney, J., & Blei, D. M. (2015). Dynamic poisson factorization. In *Proceedings of the 9th ACM conference on recommender systems* (pp. 155–162). ACM.

Du, N., Wang, Y., He, N., Sun, J., & Song, L. (2015). Time-sensitive recommendation from recurrent user activities. In *Advances in neural information processing systems* (pp. 3492–3500).

Gopalan, P., Hofman, J. M., & Blei, D. M. (2013). Scalable recommendation with poisson factorization. arXiv:1311.1704.

Gultekin, S., & Paisley, J. (2014). A collaborative Kalman filter for time-evolving dyadic processes. In *Data mining (ICDM), 2014 IEEE international conference on* (pp. 140–149). IEEE.

Hu, Y., Koren, Y., & Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. In *Data mining, 2008. icdm'08. eighth IEEE international conference on* (pp. 263–272). IEEE.

Kim, S., Choi, S.-M., Han, Y.-S., Man, K. L., & Wan, K. (2014). Analyzing item features for cold-start problems in recommendation systems. In *Intelligent information hiding and multimedia signal processing (iih-msp), 2014 tenth international conference on* (pp. 167–170). IEEE.

Koren, Y. (2010). Collaborative filtering with temporal dynamics. *Communications of the ACM, 53*(4), 89–97.

Liu, Y., Liu, Y., Shen, Y., & Li, K. (2017). Recommendation in a changing world: Exploiting temporal dynamics in ratings and reviews. *ACM Transactions on the Web (TWEB), 12*(1), 3.

Ma, T., Suo, X., Zhou, J., Tang, M., Guan, D., Tian, Y., et al. (2016). Augmenting matrix factorization technique with the combination of tags and genres. *Physica A: Statistical Mechanics and its Applications, 461*, 101–116.

Sahoo, N., Singh, P. V., & Mukhopadhyay, T. (2010). *A hidden Markov model for collaborative filtering*.

Sánchez-Moreno, D., González, A. B. G., Vicente, M. D. M., Batista, V. F. L., & García, M. N. M. (2016). A collaborative filtering method for music recommendation using playing coefficients for artists and users. *Expert Systems with Applications, 66*, 234–244.

Schein, A. I., Saul, L. K., & Ungar, L. H. (2003). A generalized linear model for principal component analysis of binary data. In *Aistats* (p. 10).

Sun, J. Z., Parthasarathy, D., & Varshney, K. R. (2014). Collaborative kalman filtering for dynamic matrix factorization.. *IEEE Trans. Signal Processing, 62*(14), 3499–3509.

Tso-Sutter, K. H., Marinho, L. B., & Schmidt-Thieme, L. (2008). Tag-aware recommender systems by fusion of collaborative filtering algorithms. In *Proceedings of the 2008 ACM symposium on applied computing* (pp. 1995–1999). ACM.

Wu, J., Chen, L., Yu, Q., Han, P., & Wu, Z. (2015). Trust-aware media recommendation in heterogeneous social networks. *World Wide Web, 18*(1), 139–157.

Xiong, L., Chen, X., Huang, T.-K., Schneider, J., & Carbonell, J. G. (2010). Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *Proceedings of the 2010 SIAM international conference on data mining* (pp. 211–222). SIAM.

Zhang, X., Cheng, J., Qiu, S., Zhu, G., & Lu, H. (2015). Dualds: a dual discriminative rating elicitation framework for cold start recommendation. *Knowledge-Based Systems, 73*, 161–172.