Learning Hierarchical Representation Model for Next Basket Recommendation

Conference Paper · July 2015

DOI: 10.1145/2766462.2767694

CITATIONS
236

READS
1,160

6 authors, including:

Pengfei Wang
Chinese Academy of Sciences
6 PUBLICATIONS **335** CITATIONS
SEE PROFILE

Jiafeng Guo
Chinese Academy of Sciences
200 PUBLICATIONS **5,271** CITATIONS
SEE PROFILE

Yanyan Lan
Chinese Academy of Sciences
144 PUBLICATIONS **3,354** CITATIONS
SEE PROFILE

Shengxian Wan
Chinese Academy of Sciences
7 PUBLICATIONS **872** CITATIONS
SEE PROFILE

Learning Hierarchical Representation Model for Next Basket Recommendation

Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, Xueqi Cheng
CAS Key Lab of Network Data Science and Technology
Institute of Computing Technology, Chinese Academy of Sciences
{wangpengfei,wanshengxian}@software.ict.ac.cn
{guojiafeng,lanyanyan,junxu,cxq}@ict.ac.cn

ABSTRACT

Next basket recommendation is a crucial task in market basket analysis. Given a user's purchase history, usually a sequence of transaction data, one attempts to build a recommender that can predict the next few items that the user most probably would like. Ideally, a good recommender should be able to explore the sequential behavior (i.e., buying one item leads to buying another next), as well as account for users' general taste (i.e., what items a user is typically interested in) for recommendation. Moreover, these two factors may interact with each other to influence users' next purchase. To tackle the above problems, in this paper, we introduce a novel recommendation approach, namely hierarchical representation model (HRM). HRM can well capture both sequential behavior and users' general taste by involving transaction and user representations in prediction. Meanwhile, the flexibility of applying different aggregation operations, especially nonlinear operations, on representations allows us to model complicated interactions among different factors. Theoretically, we show that our model subsumes several existing methods when choosing proper aggregation operations. Empirically, we demonstrate that our model can consistently outperform the state-of-the-art baselines under different evaluation metrics on real-world transaction data.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications-Data Mining

General Terms

Algorithms, Experiments, Performance, Theory

Keywords

Hierarchical Representation Model; Sequential Behavior; General Taste; Next Basket Recommendation

SIGIR 15, August 09 - 13, 2015, Santiago, Chile.

http://dx.doi.org/10.1145/2766462.2767694 .

1. INTRODUCTION

Market basket analysis helps retailers gain a better understanding of users' purchase behavior which can lead to better decisions. One of its most important tasks is next basket recommendation [7, 8, 12, 20]. In this task, usually sequential transaction data is given per user, where a transaction is a set/basket of items (e.g. shoes or bags) bought at one point of time. The target is to recommend items that the user probably want to buy in his/her next visit.

Typically, there are two modeling paradigms for this problem. One is sequential recommender [5, 25], mostly relying on Markov chains, which explores the sequential transaction data by predicting the next purchase based on the last actions. A major advantage of this model is its ability to capture sequential behavior for good recommendations, e.g. for a user who has recently bought a mobile phone, it may recommend accessories that other users have bought after buying that phone. The other is general recommender [1, 23], which discards any sequential information and learns what items a user is typically interested in. One of the most successful methods in this class is the model based collaborative filtering (i.e. matrix factorization models). Obviously, such general recommender is good at capturing the general taste of the user by learning over the user's whole purchase history.

A better solution for next basket recommendation, therefore, is to take both sequential behavior and users' general taste into consideration. One step towards this direction is the factorizing personalized Markov chains (FPMC) model proposed by Steffen Rendle et al. [23]. FPMC can model both sequential behavior (by interaction between items in the last transaction and that in the next basket) and users' general taste (by interaction between the user and the item in the next basket), thus achieves better performance than either sequential or general recommender alone. However, a major problem of FPMC is that all the components are linearly combined, indicating that it makes strong independent assumption among multiple factors (i.e. each component influence users' next purchase independently).

Unfortunately, from our analysis, we show that the independent assumption is not sufficient for good recommendations.

To tackle the above problems, we introduce a novel hierarchical representation model (HRM) for next basket recommendation. Specifically, HRM represents each user and item as a vector in continuous space, and employs a two-layer structure to construct a hybrid representation over user and items from last transaction: The first layer forms the trans-

action representation by aggregating item vectors from last transaction; While the second layer builds the hybrid representation by aggregating the user vector and the transaction representation. The resulting hybrid representation is then used to predict the items in the next basket. Note here the transaction representation involved in recommendation models the sequential behavior, while the user representation captures the general taste in recommendation.

HRM allows us to flexibly use different types of aggregation operations at different layers. Especially, by employing nonlinear rather than linear operations, we can model more complicated interactions among different factors beyond independent assumption. For example, by using a max pooling operation, features from each factor are compared and only those most significant are selected to form the higher level representation for future prediction. We also show that by choosing proper aggregation operations, HRM subsumes several existing methods including markov chain model, matrix factorization model as well as a variation of FPMC model. For learning the model parameters, we employ the negative sampling procedure [27] as the optimization method.

We conducted experiments over three real-world transaction datasets. The empirical results demonstrated the effectiveness of our approach as compared with the state-of-the-art baseline methods.

In total the contributions of our work are as follows:

- We introduce a general model for next basket recommendation which can capture both sequential behavior and users' general taste, and flexibly incorporate different interactions among multiple factors.
- We introduce two types of aggregation operations, i.e. average pooling and max pooling, into our hierarchical model and study the effect of different combinations of these operations.
- Theoretically we show that our model subsumes several existing recommendation methods when choosing proper aggregation operations.
- Empirically we show that our model, especially with nonlinear operations, can consistently outperform state-of-the-art baselines under different evaluation metrics on next basket recommendation.

2. RELATED WORK

Next basket recommendation is a typical application of recommender systems based on implicit feedback, where no explicit preferences (e.g. ratings) but only positive observations (e.g. purchases or clicks) are available [2, 7]. These positive observations are usually in a form of sequential data as obtained by passively tracking users' behavior over a sequence of time, e.g. a retail store records the transactions of customers. In this section, we briefly review the related work on recommendation with implicit feedback from the following three aspects, i.e. sequential recommender, general recommender, and the hybrid model.

Sequential recommender, mainly based on a Markov chain model, utilizes sequential data by predicting users' next action given the last actions [6]. For example, Zimdar et al. [3] propose a sequential recommender based on Markov chains, and investigate how to extract sequential patterns to learn the next state using probablistic decision-tree models. Mobasher et al. [18] study different sequential patterns for recommendation and find that contiguous sequential patterns are more suitable for sequential prediction task than general sequential patterns. Ghim-Eng Yap et al. [29] introduce a new Competence Score measure in personalized sequential pattern mining for next-items recommendation. Shani et al. [24] present a recommender based on Markov decision processes and show that a predictive Markov Chain model is effective for next basket prediction. Chen et al. [5] model playlists as a Markov chain, and propose logistic Markov Embedding to learn the representations of songs for playlist prediction. The main difference of our work to all the previous approaches is the inclusion of users' general taste in recommendation beyond sequential behavior. Besides, the previous sequential recommenders seldom address the interactions among items in sequential factors.

General recommender, in contrast, does not take sequential behavior into account but recommends based on users' whole purchase history. The key idea is collaborative filtering (CF) which can be further categorized into memory-based CF and model-based CF [1, 26]. The memory-based CF provides recommendations by finding k-nearest-neighbour of users or products based on certain similarity measure [16]. While the model-based CF tries to factorize the user-item correlation matrix for recommendation. For example, Lee et al. [12] treat the market basket data as a binary user-item matrix, and apply a binary logistic regression model based on principal component analysis (PCA) for recommendation. Hu et al. [10] conduct the factorization on user-item pairs with least-square optimization and use pair confidence to control the importance of observations. Pan et al. [19] also introduce the weights to user-item pairs, and optimize the factorization with both least-square and hinge-loss criteria. Rendle et al . [22] propose a different optimization criterion, namely Bayesian personalized ranking, which directly optimizes for correctly ranking over item pairs instead of scoring single items. They apply this method to matrix factorization and adaptive KNN to show its effectiveness. General recommender is good at capturing users' general taste, but can hardly adapt its recommendations directly to users' recent purchases without modeling sequential behavior.

Hybrid model, tries to integrate both sequential behavior and users' general taste for a better recommendation. A state-of-the-art method is the FPMC model proposed by Rendle et al. [23]. In their work, a transition cube is constructed where each entry of the cube gives the probability of a user buying next item given he has bought a certain item in the last transaction. By factorizing this cube, they interpret this probability by three pairwise interactions among user, items in the last transaction and items in the next basket. In this way, FPMC models sequential behavior by interaction between items in the last transaction and that in the next basket, as well as users' general taste by interaction between the user and the item in the next basket. It has been shown that such a hybrid model can achieve better performance than either a sequential or general recommender alone.

3. MOTIVATION

Next basket recommendation is the task of predicting what a user most probably would like to buy next when his/her sequential transaction data is given. When tackling this problem, both the sequential and general recommender have their own advantages. The sequential recommender can ful-

Figure 1: Next basket recommendation by linear combination of sequential and general factors. The numbers above the movie denote the recommendation scores produced by the recommender.

ly explore the sequential transaction data to discover the correlation between items in consequent purchases, leading to very responsive recommendation according to users' recent purchase. While the general recommender can leverage users' whole purchase histories to learn the taste of different users, and thus achieve better personalization in recommendation.

As shown in previous work [23], it is better to take both sequential and general factors into account for better recommendation. A simple solution is to use a linear combination over these two factors. Furthermore, when modeling the sequential factor, items in the last transaction are often linearly combined in predicting the next item [23]. Obviously, one major assumption underlying these linear combinations is the independence among multiple factors. That is, both sequential and general factor influence the next purchase independently, and each item in the last transaction influence the next purchase independently as well. Here comes the question: Is the independent assumption among multiple factors sufficient for good recommendation?

To answer the above question, we first consider the independent assumption between the general and sequential factors. Let us take a look at an example shown in Figure 1. Imagine a user in general buys science fiction movies like 'The Matrix' and 'X-men'. In contrast to his usual buying behavior, he recently has become fascinated in Scarlett Johansson and purchased 'Match Point' to watch. A sequential recommender based on recent purchase would recommend movies like 'Lost in Translation' (0.9) and 'Girl with a Pearl Earring' (0.85), which are also dramas performed by Scarlett Johansson. (Note that the number in the parentheses denotes the recommendation score). In contrast, a general recommender which mainly accounts for user's general taste would recommend 'The Dark Knight' (0.95) and 'Inception' (0.8) and other science fiction movies. By taking into account both factors, good recommendations for the user might be the movies like 'Lucy' and 'The Avengers', which are science fiction movies performed by Scarlett Johansson. However, if we linearly combine the two factors, i.e. independent in prediction, we may not obtain the right results as we expected. The reason lies in that a good recommendation under joint consideration of the two factors may not obtain a high recommendation score when calculating from each individual factor. For example, the scores of 'Lucy' (0.3) and 'The Avengers' (0.2) in sequential recommender are low since they do not match well with the genre preference (i.e. drama) based on the last purchase of the user. Their scores are also not very high in general recommender since there are many better and popular movies fitting the science fiction taste. Thus the linear combination cannot boost the good recommendations to the top.

Let us take a further look at sequential factor alone, i.e. recommending next items based on the last transaction. For example, people who have bought pumpkin will probably buy other vegetables like cucumber or tomato next, while people who have bought candy will probably buy other snacks like chocolate or chips next. However, people who have bought pumpkin and candy together will very probably buy Halloween costumes next. Again, we can see that if we simply combine the recommendation results from pumpkin and candy respectively, we may not be able to obtain the right recommendations.

From the above examples, we find that models based on linear combination do have limitations in capturing complicated influence of multiple factors on next purchase. In other words, independent assumption among different factors may not be sufficient for good recommendations. We need a model that is capable of incorporating more complicated interactions among multiple factors. This becomes the major motivation of our work.

4. OUR APPROACH

In this section, we first introduce the problem formalization of next basket recommendation. We then describe the proposed HRM in detail. After that, we talk about the learning and prediction procedure of HRM. Finally, we discuss the connections of HRM to existing methods.

4.1 Formalization

Let $U = \{u_1, u_2, \ldots, u_{|U|}\}$ be a set of users and $I = \{i_1, i_2, \ldots, i_{|I|}\}$ be a set of items, where $|U|$ and $|I|$ denote the total number of unique users and items, respectively. For each user u, a purchase history T^u of his transactions is given by $T^u := (T_1^u, T_2^u, \ldots, T_{t_u-1}^u)$, where $T_t^u \subseteq I$, $t \in [1, t_u - 1]$. The purchase history of all users is denoted as $T := \{T^{u_1}, T^{u_2}, \ldots, T^{u_{|U|}}\}$. Given this history, the task is to recommend items that user u would probably buy at the next (i.e. t_u-th) visit. The next basket recommendation task can then be formalized as creating a personalized total ranking $>_{u,t} \subset I^2$ for user u and t_u-th transaction. With this ranking, we can recommend the top n items to the user.

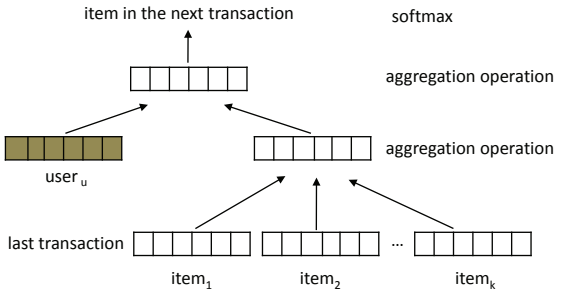

Figure 2: The HRM model architecture. A two-layer structure is employed to construct a hybrid representation over user and items from last transaction, which is used to predict the next purchased items.

4.2 HRM Model

To solve the above recommendation problem, here we present the proposed HRM in detail. The basic idea of our work is to learn a recommendation model that can involve both sequential behavior and users' general taste, and meanwhile modeling complicated interactions among these factors in prediction.

Specifically, HRM represents each user and item as a vector in a continuous space, and employs a two-layer structure to construct a hybrid representation over user and items from last transaction: The first layer forms the transaction representation by aggregating item vectors from last transaction; While the second layer builds the hybrid representation by aggregating the user vector and the transaction representation. The resulting hybrid representation is then used to predict the items in the next basket. The hierarchical structure of HRM is depicted in Figure 2. As we can see, HRM captures the sequential behavior by modeling the consecutive purchases, i.e. constructing the representation of the last transaction from its items for predicting the next purchase. At the same time, by integrating a personalized user representation in sequential recommendation, HRM also models the user's general taste.

More formally, let $V^U = \{\vec{v}_u^U \in \mathbb{R}^n | u \in U\}$ denote all the user vectors and $V^I = \{\vec{v}_i^I \in \mathbb{R}^n | i \in I\}$ denote all the item vectors. Note here V^U and V^I are model parameters to be learned by HRM. Given a user u and two consecutive transactions T_{t-1}^u and T_t^u, HRM defines the probability of buying next item i given user u and his/her last transaction T_{t-1}^u via a softmax function:

$$p(i \in T_t^u | u, T_{t-1}^u) = \frac{exp(\vec{v}_i^I \cdot \vec{v}_{u,t-1}^{Hybrid})}{\sum_{j=1}^{|I|} exp(\vec{v}_j^I \cdot \vec{v}_{u,t-1}^{Hybrid})} \quad (1)$$

where $\vec{v}_{u,t-1}^{Hybrid}$ denotes the hybrid representation obtained from the hierarchical aggregation which is defined as follows

$$\vec{v}_{u,t-1}^{Hybrid} := f_2(\vec{v}_u^U, f_1(\vec{v}_l^I \in T_{t-1}^u))$$

where $f_1(\cdot)$ and $f_2(\cdot)$ denote the aggregation operation at the first and second layer, respectively.

One advantage of HRM is that we can introduce various aggregation operations in forming higher level representation from lower level. In this way, we can model different interactions among multiple factors at different layers, i.e. interaction among items forming the transaction representation at the first layer, as well as interaction between user and transaction representations at the second layer. In this work, we study two typical aggregation operations as follows.

- *average pooling*: To aggregate a set of vector representations, average pooling construct one vector by taking the average value of each dimension. Let $V = \{\vec{v}_l \in \mathbb{R}^n | l = 1, \ldots, |V|\}$ be a set of input vectors to be aggregated, average pooling over V can be formalized as

$$f_{avg}(V) = \frac{1}{|V|} \sum_{l=1}^{|V|} \vec{v}_l$$

  Obviously, average pooling is a linear operation, which assumes the independence among input representations in forming higher level representation.

- *max pooling*: To aggregate a set of vector representations, max pooling constructs one vector by taking the maximum value of each dimension, which can be formalized as

$$f_{max}(V) = \begin{bmatrix} max(\vec{v}_1[1], \ldots, \vec{v}_{|V|}[1]) \\ max(\vec{v}_1[2], \ldots, \vec{v}_{|V|}[2]) \\ \vdots \\ max(\vec{v}_1[n], \ldots, \vec{v}_{|V|}[n]) \end{bmatrix}$$

  where $\vec{v}_l[k]$ denotes the k-th dimension in $\vec{v}_l$. In Contrary to average pooling, max pooling is a nonlinear operation which models interactions among input representations, i.e. features from each input vector are compared and only those most significant features will be selected to the next level. Take the movie recommender mentioned in Section 3.1 for example, we suppose vector representations are used for both sequential and general factors. If there are two dimensions capturing the genre and actor/actress preference respectively, max pooling then selects the most significant feature in each dimension (e.g. science fiction and Scarlett Johansson) in aggregating the two vectors.

Note that there are other ways to define the aggregation operations, e.g. top-k average pooling or Hadamard product. We may study these operations in the future work. Besides, one may also consider to introduce nonlinear hidden layers as in deep neural network [4]. However, we resort to simple models since previous work has demonstrated that such models can learn accurate representations from very large data set due to low computational complexity [17, 27].

Since there are two-layer aggregations in HRM, we thus can obtain four versions of HRM based on different combinations of operations, namely HRM$_{AvgAvg}$, HRM$_{MaxAvg}$, HRM$_{AvgMax}$, and HRM$_{MaxMax}$, where the two abbreviations in subscript denote the first and second layer aggregation operation respectively. For example, HRM$_{AvgMax}$ denotes the model that employs average pooling at the first layer and max pooling at second layer.

As we can see, these four versions of HRM actually assume different strength of interactions among multiple factors. By only using average pooling, HRM$_{AvgAvg}$ assume independence among all the factors. We later show that HRM$_{AvgAvg}$ can be viewed as some variation of FPMC.

Both HRM_{AvgMax} and HRM_{MaxAvg} introduce partial interactions, either among the items in last transaction or between the user and transaction representations. Finally, by using nonlinear operations at both layers, HRM_{MaxMax} assumes full interactions among all the factors.

4.3 Learning and Prediction

In learning, HRM maximizes the log probability defined in Equation (1) over the transaction data of all users as follows

$$\ell_{HRM} = \sum_{u \in U} \sum_{T_t^u \in T^u} \sum_{i \in T_t^u} \log p(i \in T_t^u | u, T_{t-1}^u) - \lambda \|\Theta\|_F^2$$

where λ is the regularization constant and Θ are the model parameters (i.e. Θ=$\{V^U$,$V^I\}$). As defined in Section 4.1, the goal of next basket recommendation is to derive a ranking $>_{u,t}$ over items. HRM actually defines the ranking as

$$i >_{u,t} i' :\Leftrightarrow p(i \in T_t^u | u, T_{t-1}^u) > p(i' \in T_t^u | u, T_{t-1}^u)$$

and attempts to derive such ranking by maximizing the buying probability of next items over the whole purchase history.

However, directly optimizing the above objective function is impractical because the cost of computing the full softmax is proportional to the size of items $|I|$, which is often extremely large. Therefore, we adopt the negative sampling technique [21, 27] for efficient optimization, which approximates the original objective ℓ_{HRM} with the following objective function

$$\ell_{NEG} = \sum_{u \in U} \sum_{T_t^u \in T^u} \sum_{i \in T_t^u} \Big(\log \sigma(\vec{v}_i^I \cdot \vec{v}_{u,t-1}^{Hybrid}) + k \cdot \mathbb{E}_{i' \sim P_I} [\log \sigma(-\vec{v}_{i'}^I \cdot \vec{v}_{u,t-1}^{Hybrid})] \Big) - \lambda \|\Theta\|_F^2$$

where $\sigma(x) = 1/(1+e^{-x})$, k is the number of "negative" samples, and i' is the sampled item, drawn according to the noise distribution P_I which is modeled by empirical unigram distribution over items. As we can see, the objective of HRM with negative sampling aims to derive the ranking $>_{u,t}$ in a discriminative way by maximizing the probability of observed item i and meanwhile minimizing the probability of unobserved item i's.

We then apply stochastic gradient descent algorithm to maximize the new objective function for learning the model. Moreover, when learning the nonlinear models, we also adopt Dropout technique to avoid overfitting. In our work, we simply set a fixed drop ratio (50%) for each unit.

With the learned user and item vectors, the next basket recommendation with HRM is as follows. Given a user u and his/her last transaction $T_{t_u-1}^u$, for each candidate item $i \in I$, we calculate the probability $p(i \in I | u, T_{t_u-1}^u)$ according to Equation (1). We than rank the items according to their probabilities, and select the top n results as the final recommendations to the user.

4.4 Connection to Previous Models

In this section, we discuss the connection of the proposed HRM to previous work. We show that by choosing proper aggregation operations, HRM subsumes several existing methods including Markov chain model, matrix factorization model as well as a variation of FPMC model.

4.4.1 HRM vs. Markov Chain Model

To show that HRM can be reduced to a certain type of Markov chain model, we first introduce a special aggregation operation, namely select-copy operation. When aggregating a set of vector representations, the select-copy operation select one of the vectors according to some criterion, and copy it as the aggregated one. Now we apply this operation to both levels of HRM. Specifically, when constructing the transaction representation from item vectors, the operation randomly selects one item vector and copies it. When combining the user and transaction representations, the operation always selects and copies the transaction vector. We refer the HRM with this model architecture as $\text{HRM}_{CopyItem}$. The new objective function of $\text{HRM}_{CopyItem}$ using negative sampling is as follows:

$$\ell_{CopyItem} = \sum_{u \in U} \sum_{T_t^u \in T^u} \sum_{i \in T_t^u} \Big(\log \sigma(\vec{v}_i^I \cdot \vec{v}_s^I) + k \cdot \mathbb{E}_{i' \sim P_I} [\log \sigma(-\vec{v}_{i'}^I \cdot \vec{v}_s^I)] \Big) - \lambda \|\Theta\|_F^2$$

where $\vec{v}_s^I$ denotes the vector of randomly selected item in last transaction.

Similar as the derivation in [21], we can show that the solution of $\text{HRM}_{CopyItem}$ follows that

$$\vec{v}_i^I \cdot \vec{v}_s^I = PMI(v_i^I, v_s^I) - \log k$$

which indicates that $\text{HRM}_{CopyItem}$ is actually a factorized Markov chain model (FMC) [23], which factorizes a transition matrix between items from two consecutive transactions with the association measured by *shifted PMI* (i.e. $PMI(x,y) - \log k$). When $k = 1$, the transition matrix becomes a PMI matrix.

In fact, if we employ noise contrastive estimation [27] for optimization, the solution then follows that:

$$\vec{v}_i^I \cdot \vec{v}_s^I = \log P(v_i^I | v_s^I) - \log k$$

which indicates the transition matrix factorized by $\text{HRM}_{CopyItem}$ become a (shifted) log-conditional-probability matrix.

4.4.2 HRM vs. Matrix Factorization Model

Now we only apply the select-copy operation to the second layer (i.e. aggregation over user and transaction representations), and this time we always select and copy user vector. We refer this model as $\text{HRM}_{CopyUser}$. The corresponding objective function using negative sampling is as follows:

$$\ell_{CopyUser} = \sum_{u \in U} \sum_{T_t^u \in T^u} \sum_{i \in T_t^u} \Big(\log \sigma(\vec{v}_i^I \cdot \vec{v}_u^U) + k \cdot \mathbb{E}_{i' \sim P_I} [\log \sigma(-\vec{v}_{i'}^I \cdot \vec{v}_u^U)] \Big) - \lambda \|\Theta\|_F^2$$

Again, we can show that $\text{HRM}_{CopyUser}$ has the solution in the following form:

$$\vec{v}_u^U \cdot \vec{v}_i^I = PMI(v_u^U, v_i^I) - \log k$$

In this way, $\text{HRM}_{CopyUser}$ reduces to a matrix factorization model, which factorizes a user-item matrix where the association between a user and a item is measured by shifted PMI.

4.4.3 HRM vs. FPMC

FPMC conducts a tensor factorization over the transition cube constructed from the transition matrices of all users. It is optimized under the Bayesian personalized ranking (BPR) criterion and the objective function using MAP-estimator is

Table 1: Statistics of the datasets used in our experiments.

dataset	users $\|U\|$	items $\|I\|$	transactions T	avg.transaction size	avg.transaction per user
Ta-Feng	9238	7982	67964	7.4	5.9
BeiRen	9321	5845	91294	9.7	5.8
T-Mall	292	191	1805	5.6	1.2

as follows [23]:

$$\ell_{FPMC} = \sum_{u \in U} \sum_{T_t^u \in T^u} \sum_{i \in T_t^u} \sum_{i' \notin T_t^u} \log \sigma(\hat{x}_{u,t,i} - \hat{x}_{u,t,i'}) - \lambda \|\Theta\|_F^2 \quad (2)$$

where $\hat{x}_{u,t,i}$ denotes the prediction model

$$\begin{aligned} \hat{x}_{u,t,i} &:= \hat{p}(i \in T_t^u | u, T_{t-1}^u) \\ &:= \vec{v}_u^U \cdot \vec{v}_i^I + \frac{1}{|T_{t-1}^u|} \sum_{l \in T_{t-1}^u} (\vec{v}_i^I \cdot \vec{v}_l^I) \end{aligned} \quad (3)$$

To see the connection between HRM and FPMC, we now set the aggregation operation as average pooling at both layers and apply negative sampling with $k = 1$. We denote this model as $\text{HRM}_{AvgAvgNEG1}$ and its objective function is as follows

$$\begin{aligned} \ell_{AvgAvgNEG1} &= \sum_{u \in U} \sum_{T_t^u \in T^u} \sum_{i \in T_t^u} \Big(\log \sigma(\vec{v}_i^I \cdot \vec{v}_{u,t-1}^{Hybrid}) \\ &\quad + \mathbb{E}_{i' \sim P_I} [\log \sigma(-\vec{v}_{i'}^I \cdot \vec{v}_{u,t-1}^{Hybrid})] \Big) - \lambda \|\Theta\|_F^2 \\ &= \sum_{u \in U} \sum_{T_t^u \in T^u} \sum_{i \in T_t^u} \sum_{i' \notin T_t^u} \Big(\log \sigma(\vec{v}_i^I \cdot \vec{v}_{u,t-1}^{Hybrid}) \\ &\quad + \log \sigma(-\vec{v}_{i'}^I \cdot \vec{v}_{u,t-1}^{Hybrid}) \Big) - \lambda \|\Theta\|_F^2 \end{aligned} \quad (4)$$

where

$$\vec{v}_{u,t-1}^{Hybrid} = \frac{1}{2} (\vec{v}_u^U + \frac{1}{|T_{t-1}^u|} \sum_{l \in T_{t-1}^u} \vec{v}_l^I) \quad (5)$$

With Equation (3) and (5), we can rewrite Equation (4) as follows

$$\begin{aligned} \ell_{AvgAvgNEG1} &= \sum_{u \in U} \sum_{T_t^u \in T^u} \sum_{i \in T_t^u} \sum_{i' \notin T_t^u} \Big(\log \sigma(\hat{x}_{u,t,i}) \\ &\quad + \log \sigma(-\hat{x}_{u,t,i'}) \Big) - \lambda \parallel \Theta \parallel_F^2 + C \\ &= \sum_{u \in U} \sum_{T_t^u \in T^u} \sum_{i \in T_t^u} \sum_{i' \notin T_t^u} \Big(\log \sigma(\hat{x}_{u,t,i}) \\ &\quad + \log(1 - \sigma(\hat{x}_{u,t,i'})) \Big) - \lambda \parallel \Theta \parallel_F^2 + C \end{aligned} \quad (6)$$

Based on the above derivations, we can see that both $\text{HRM}_{AvgAvgNEG1}$ and FPMC share the same prediction model denoted by Equation (3), but optimize with slightly different criteria. FPMC tries to maximize the pairwise rank, i.e. an observed item i ranks higher than an unobserved item i', by defining the pairwise probability using a logistic function as shown in Equation (2). While $\text{HRM}_{AvgAvgNEG1}$ also optimizes this pairwise rank by maximizing the probability of item i and minimizing the probability of item i', each defined in a logistic form as shown in Equation (6). In fact, we can also adopt BPR criterion to define the objective function of HRM_{AvgAvg}, and obtain the same model as FPMC.

Based on all the above analysis, we can see that the proposed HRM is actually a very general model. By introducing different aggregation operations, we can produce multiple recommendation models well connected to existing methods. Moreover, HRM also allows us to explore other prediction functions as well as optimization criteria, showing large flexibility and promising potential.

5. EVALUATION

In this section, we conduct empirical experiments to demonstrate the effectiveness of our proposed HRM on next basket recommendation. We first introduce the dataset, baseline methods, and the evaluation metrics employed in our experiments. Then we compare the four versions of HRM to study the effect of different combinations of aggregation operations. After that, we compare our HRM to the state-of-the-art baseline methods to demonstrate its effectiveness. Finally, we conduct some analysis on our optimization procedure, i.e. negative sampling technique.

5.1 Dataset

We evaluate different recommenders based on three real-world transaction datasets, i.e. two retail datasets Ta-Feng and BeiRen, and one e-commerce dataset T-Mall.

- The Ta-Feng[1] dataset is a public dataset released by RecSys conference, which covers products from food, office supplies to furniture. It contains 817,741 transactions belonging to 32,266 users and 23,812 items.
- The BeiRen dataset comes from BeiGuoRenBai[2], a large retail enterprise in China, which records its supermarket purchase history during the period from Jan. 2013 to Sept. 2013. It contains 1,123,754 transactions belonging to 34,221 users and 17,920 items.
- The T-Mall[3] dataset is a public online e-commerce dataset released by Taobao[4], which records the online transactions in terms of brands. It contains 4298 transactions belonging to 884 users and 9,531 brands.

We first conduct some pre-process on these transaction datasets similar as [23]. For both Ta-Feng and BeiRen dataset, we remove all the items bought by less than 10 users and users that has bought in total less than 10 items. For the T-Mall dataset, which is relatively smaller, we remove all the items bought by less than 3 users and users that has bought in total less than 3 items. The statistics of the three datasets after pre-processing are shown in Table 1.

Finally, we split all the datasets into two non overlapping set, i.e. a training set and a testing set. The testing set contains only the last transaction of each user, while all the remaining transactions are put into the training set.

[1] http://recsyswiki.com/wiki/Grocery_shopping_datasets
[2] http://www.brjt.cn/
[3] http://102.alibaba.com/competition/addDiscovery/index.htm
[4] http://www.taobao.com

Table 2: Performance comparison among four versions of HRM over three datasets

(a) Performance comparison on Ta-Feng

Models	d=50			d=100			d=150			d=200		
	F1-score	Hit-ratio	NDCG	F1-score	Hit-ratio	NDCG	F1-score	Hit-ratio	NDCG	F1-score	Hit-ratio	NDCG
HRM_{AvgAvg}	0.051	0.240	0.073	0.060	0.276	0.082	0.063	0.283	0.080	0.063	0.286	0.086
HRM_{MaxAvg}	0.059	0.275	0.080	0.064	0.279	0.087	0.065	0.290	0.083	0.067	0.298	0.086
HRM_{AvgMax}	0.057	0.262	0.080	0.064	0.288	0.085	0.065	0.289	0.082	0.068	0.293	0.090
HRM_{MaxMax}	**0.062**	**0.282**	**0.089**	**0.065**	**0.293**	**0.088**	**0.068**	**0.298**	**0.085**	**0.070**	**0.312**	**0.093**

(b) Performance comparison on BeiRen

Models	d=50			d=100			d=150			d=200		
	F1-score	Hit-ratio	NDCG	F1-score	Hit-ratio	NDCG	F1-score	Hit-ratio	NDCG	F1-score	Hit-ratio	NDCG
HRM_{AvgAvg}	0.100	0.463	0.119	0.107	0.475	0.128	0.112	0.505	0.137	0.113	0.509	0.137
HRM_{MaxAvg}	0.105	0.485	0.131	0.113	0.498	0.138	0.115	0.509	0.139	0.115	0.505	0.141
HRM_{AvgMax}	0.106	0.494	0.131	0.114	0.512	0.140	0.115	0.510	0.141	0.115	0.510	0.140
HRM_{MaxMax}	**0.111**	**0.501**	**0.134**	**0.115**	**0.515**	**0.144**	**0.117**	**0.516**	**0.146**	**0.118**	**0.515**	**0.145**

(c) Performance comparison on T-Mall

Models	d=10			d=15			d=20			d=25		
	F1-score	Hit-ratio	NDCG	F1-score	Hit-ratio	NDCG	F1-score	Hit-ratio	NDCG	F1-score	Hit-ratio	NDCG
HRM_{AvgAvg}	0.052	0.154	0.119	0.055	0.139	0.146	0.061	0.180	0.146	0.063	0.186	0.151
HRM_{MaxAvg}	0.062	0.186	0.133	0.063	0.148	0.157	0.066	0.196	0.154	0.068	0.202	0.158
HRM_{AvgMax}	0.061	0.186	0.133	0.063	0.148	0.153	0.064	0.191	0.157	0.066	0.196	0.159
HRM_{MaxMax}	**0.065**	**0.191**	**0.142**	**0.066**	**0.197**	**0.163**	**0.070**	**0.207**	**0.163**	**0.071**	**0.212**	**0.168**

5.2 Baseline Methods

We evaluate our model by comparing with several state-of-the-art methods on next-basket recommendation:

- TOP: The top popular items in training set are taken as recommendations for each user.
- MC: A Markov chain model (i.e. sequential recommender) which predicts the next purchase based on the last transaction of the user. The prediction model is as follows:

$$p(i \in T^u_{t_u} | T^u_{t_u-1}) := \frac{1}{|T^u_{t_u-1}|} \sum_{l \in T^u_{t_u-1}} p(i \in T^u_{t_u} | l \in T^u_{t_u-1})$$

  The transition probability of buying an item based on the last purchase is estimated from the training set.
- NMF: A state-of-the-art model based collaborative filtering method [14]. Here Nonnegative Matrix Factorization is applied over the user-item matrix, which is constructed from the transaction dataset by discarding the sequential information. For implementation, we adopt the publicly available codes from NMF:DTU Toolbox[5].
- FPMC: A state-of-the-art hybrid model on next basket recommendation [23]. Both sequential behavior and users' general taste are taken into account for prediction.

For NMF, FPMC and our HRM[6] methods, we run several times with random initialization by setting the dimensionality $d \in \{50, 100, 150, 200\}$ on Ta-Feng and BeiRen datasets, and $d \in \{10, 15, 20, 25\}$ on T-Mall dataset. We compare the best results of different methods and demonstrate the results in the following sections.

[5]http://cogsys.imm.dtu.dk/toolbox/nmf/

[6]http://www.bigdatalab.ac.cn/benchmark/bm/bd?code=HRM

5.3 Evaluation Metrics

The performance is evaluated for each user u on the transaction $T^u_{t_u}$ in the testing dataset. For each recommendation method, we generate a list of N items (N=5) for each user u, denoted by $R(u)$, where $R_i(u)$ stands for the item recommended in the i-th position. We use the following quality measures to evaluate the recommendation lists against the actual bought items.

- F1-score: F1-score is the harmonic mean of precision and recall, which is a widely used measure in recommendation [9, 15, 23]:

$$\text{Precison}(T^u_{t_u}, R(u)) = \frac{|T^u_{t_u} \bigcap R(u)|}{|R(u)|}$$

$$\text{Recall}(T^u_{t_u}, R(u)) = \frac{|T^u_{t_u} \bigcap R(u)|}{|T^u_{t_u}|}$$

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- Hit-Ratio: Hit-Ratio is a All-but-One measure used in recommendation [13, 28]. If there is at least one item in the test transaction also appears in the recommendation list, we call it a *hit*. The Hit-Ratio is calculated in the following way:

$$\text{Hit-Ratio} = \frac{\sum_{u \in U} I(T^u_{t_u} \bigcap R(u) \neq \phi)}{|U|}$$

  where $I(\cdot)$ is an indicator function and ϕ denotes the empty set. Hit-Ratio focuses on the *recall* of a recommender system, i.e. how many people can obtain at least one correct recommendation.
- NDCG@k: Normalized Discounted Cumulative Gain (NDCG) is a ranking based measure which takes into

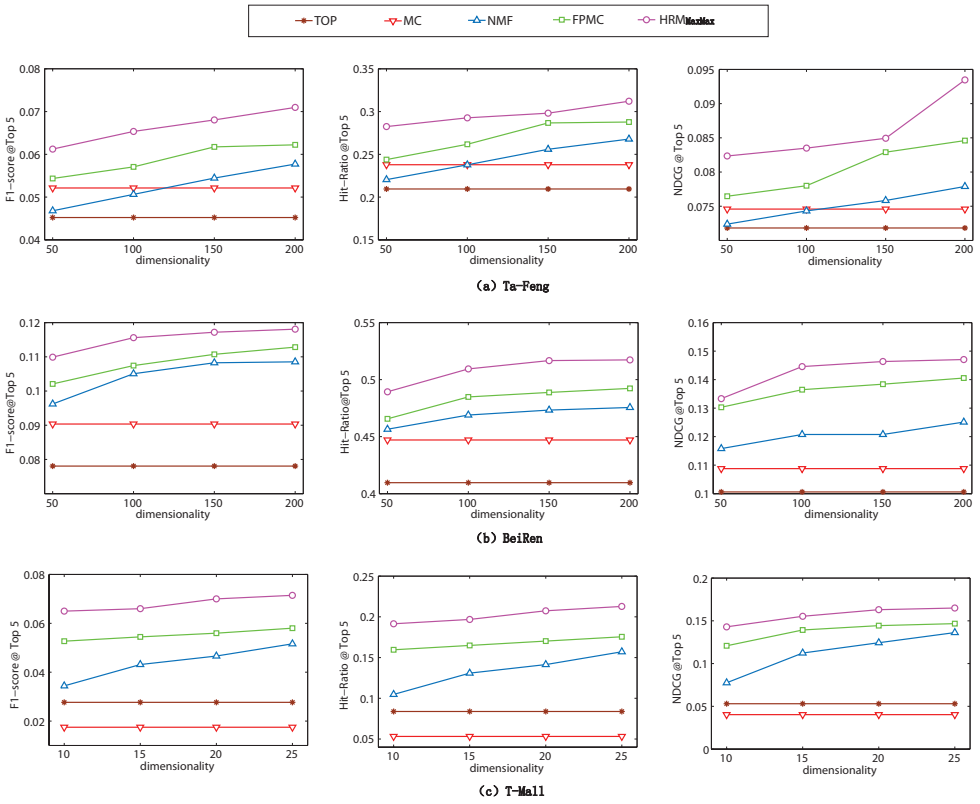

Figure 3: Performance comparison of HRM among TOP,MC,NMF, and FPMC over three datasets. The dimensionality is increased from 50 to 200 on Ta-Feng and BeiRen, and 10 to 25 on T-Mall.

account the order of recommended items in the list[11], and is formally given by:

$$NDCG@k = \frac{1}{N_k}\sum_{j=1}^{k}\frac{2^{I(R_j(u)\in T^u_{t_u})}-1}{log_2(j+1)}$$

where $I(\cdot)$ is an indicator function and N_k is a constant which denotes the maximum value of NDCG@k given $R(u)$.

5.4 Comparison among Different HRMs

We first empirically compare the performance of the four versions of HRM, referred to as HRM$_{AvgAvg}$, HRM$_{MaxAvg}$, HRM$_{AvgMax}$, HRM$_{MaxMax}$. The results over three datasets are shown in Table 2.

As we can see, HRM$_{AvgAvg}$, which only uses average pooling operations in aggregation, performs the worst among the four models. It indicates that by assuming independence among all the factors, we may not be able to learn a good recommendation model. Both HRM$_{MaxAvg}$ and HRM$_{AvgMax}$ introduce partial interactions by using max pooling either at the first or the second layer, and obtain better results than HRM$_{AvgAvg}$. Take the Ta-Feng dataset as an example, when compared with HRM$_{AvgAvg}$ with dimensionality set as 50, the relative performance improvement by HRM$_{MaxAvg}$ and HRM$_{AvgMax}$ is around 13.6% and 9.8%, respectively. Besides, we also find that there is no consistent dominant between these two partial-interaction models, indicating that interactions at different layers may both help the recommendation in their own way. Finally, by applying max pooling at both layers (i.e. full interactions), HRM$_{MaxMax}$ can outperform the other three variations in terms of all the three evaluation measures. The results demonstrate the advantage of modeling interactions among multiple factors in next basket recommendation.

5.5 Comparison against Baselines

We further compare our HRM model to the state-of-the-art baseline methods on next basket recommendation. Here we choose the best performed HRM$_{MaxMax}$ as the representative for clear comparison. The performance results over Ta-Feng, BeiRen, and T-Mall are shown in Figure 3.

We have the following observations from the results. (1) Overall, the Top method is the weakest. However, we find that the Top method outperforms MC on the T-Mall dataset. This might be due to the fact that the items in T-Mall dataset are actually brands. Therefore, the distributions of top popular brands on both training and testing datasets are very close, which accords with the assumption of the Top method and leads to better performance. (2) The NMF method outperforms the MC method in most cases. A major reason might be that the transition matrix estimated in the

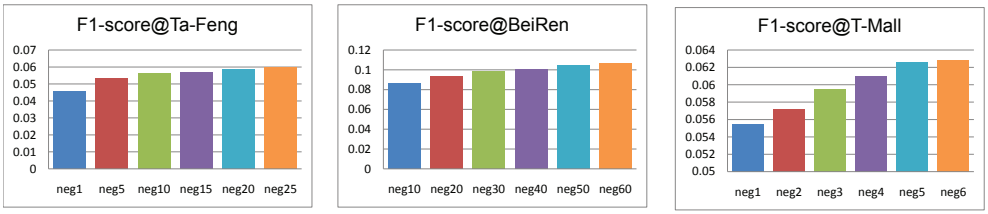

Figure 4: Performance variation in terms of F1-score against the number of negative samples over three datasets with HRM_{MaxMax}. The number of negative samples is increased from 1 to 25 on Ta-Feng, 10 to 60 on BeiRen, and from 1 to 6 on T-Mall.

Table 3: Performance comparison on Ta-Feng over different user groups with dimensionality set as 50.

user activeness	method	F1-score	Hit-Ratio	NDCG@5
Inactive	Top	0.036	0.181	0.054
	MC	0.042	0.206	0.058
	NMF	0.037	0.198	0.046
	FPMC	0.043	0.216	0.060
	HRM_{MaxMax}	**0.048**	**0.236**	**0.062**
Medium	Top	0.051	0.230	0.084
	MC	0.059	0.262	0.088
	NMF	0.052	0.234	0.072
	FPMC	0.059	0.263	0.087
	HRM_{MaxMax}	**0.068**	**0.299**	**0.097**
Active	Top	0.045	0.207	0.074
	MC	0.050	0.212	0.075
	NMF	0.056	0.223	0.075
	FPMC	0.054	0.224	0.080
	HRM_{MaxMax}	**0.062**	**0.246**	**0.087**

MC method are rather sparse, and directly using it for recommendation may not work well. One way to improve the performance of the MC method is to factorize the transition matrix to alleviate the sparse problem [23]. (3) By combining both sequential behavior and users' general taste, FPMC can obtain better results than both MC and NMF. This result is quite consistent with the previous finding in [23]. (4) By further introducing the interactions among multiple factors, the proposed HRM_{MaxMax} can consistently outperform all the baseline methods in terms of all the measures over the three datasets. Take the Ta-Feng dataset as an example, when compared with second best performed baseline method (i.e. FPMC) with dimensionality set as 200, the relative performance improvement by HRM_{MaxMax} is around 13.1%, 11.1%, and 12.5% in terms of F1-score, Hit-Ratio and NDCG@5, respectively.

To further investigate the performance of different methods, we split the users into three groups (i.e., inactive, medium and active) based on their activeness and conducted the comparisons on different user groups. Take the Ta-Feng dataset as an example, a user is taken as inactive if there are less than 5 transactions in his/her purchase history, and active if there are more than 20 transactions in the purchase history. The remaining users are taken as medium. In this way, the proportions of inactive, medium and active are 40.8%, 54.5%, and 4.7% respectively. Here we only report the comparison results on Ta-Feng dataset under one dimensionality (i.e. $d = 50$) due to the page limitation. In fact, similar conclusions can be drawn from other datasets. The results are shown in Table 3.

From the results we can see that, not surprisingly, the Top method is still the worst on all the groups. Furthermore, we find that MC works better than NMF on both inactive and medium users in terms of all the measures; While on active users, NMF can achieve better performance than MC. The results indicate that it is difficult for NMF to learn a good user representation with few transactions for recommendation. By combining both sequential behavior and users' general taste linearly, FPMC obtains better performance than MC on inactive and active users, and performs better than NMF on inactive and medium users. However, we can see the improvements are not very consistent on different user groups. Finally, HRM_{MaxMax} can achieve the best performance on all the groups in terms of all the measures. It demonstrates that modeling interactions among multiple factors can help generate better recommendations for different types of users.

5.6 The Impact of Negative Sampling

To learn the proposed HRM, we employ negative sampling procedure for optimization. One parameter in this procedure is the number of negative samples we draw each time, denoted by k. Here we investigate the impact of the sampling number k on the final performance. Since the item size is different over the three datasets, we tried different ranges of k accordingly. Specifically, we tried $k \in \{1, 5, 10, 15, 20, 25\}$ on Ta-Feng, $k \in \{10, 20, 30, 40, 50, 60\}$ on BeiRen, and $k \in \{1, 2, 3, 4, 5, 6\}$ on T-Mall, respectively. We report the test performance of HRM_{MaxMax} in terms of F1-score against the number of negative samples over the three datasets in Figure 4. Here we only show the results on one dimension over each dataset (i.e. $d = 50$ on Ta-Feng and BeiRen and $d = 10$ on T-Mall) due to the space limitation.

From the results we find that: (1) As the sampling number k increases, the test performance in terms of F1-score increases too. The trending is quite consistent over the three datasets. (2) As the sampling number k increases, the performance gain between two consecutive trials decreases. For example, on Ta-Feng dataset, when we increase k from 20 to 25, the relative performance improvement in terms of F1-score is about 0.0011%. It indicates that if we continue to sample more negative samples, there will be less performance improvement but larger computational complexity. Therefore, in our performance comparison experiments, we set k as 25, 60, 6 on Ta-Feng, BeiRen and T-Mall, respectively.

6. CONCLUSION

In this paper, we propose a novel hierarchical representation model (HRM) to predict what users will buy in next

basket. Our model can well capture both sequential behavior and users' general taste in recommendation. What is more important is that HRM allows us to model complicated interactions among multiple factors by using different aggregation operations over the representations of these factors. We conducted experiments on three real-world transaction datasets, and demonstrated that our approach can outperform all the state-of-the-art baseline methods consistently under different evaluation metrics.

For the future work, we would like to try other aggregation operations in our HRM. We also want to analyze what kind of interactions are really effective in next basket prediction. Moreover, we would like to study how to integrate other types of information into our model, e.g. the transaction timestamp, which may introduce even more complicated interactions with the existing factors.

7. ACKNOWLEDGMENTS

This research work was funded by 973 Program of China under Grant No.2014CB340406, No.2012CB316303, 863 Program of China under Grant No.2014AA015204, Project supported by the National Natural Science Foundation of China under Grant No.61472401, No.61433014, No.61425016, and No.61203298. We would like to thank the anonymous reviewers for their valuable comments.

8. REFERENCES

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734–749, June 2005.

[2] T. R. Andreas Mild. An improved collaborative filtering approach for predicting cross-category purchases based on binary market basket data. *Journal of Retailing and Consumer Services*, 2003.

[3] C. M. Andrew Zimdars, David Maxwell Chickering. Using temporal data for making recommendations. *The Conference on Uncertainty in Artificial Intelligence*, 2001.

[4] E. Arisoy, T. N. Sainath, B. Kingsbury, and B. Ramabhadran. Deep neural network language models. In *Proceedings of the NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT*, WLM '12, pages 20–28, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[5] S. Chen, J. L. Moore, D. Turnbull, and T. Joachims. Playlist prediction via metric embedding. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 714–722, New York, NY, USA, 2012. ACM.

[6] A. G. Chetna Chand, Amit Thakkar. Sequential pattern mining: Survey and current research challenges. *International Journal of Soft Computing and Engineering*, 2012.

[7] A. Christidis, K. Exploring customer preferences with probabilistic topics models. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2010.

[8] M. Gatzioura, A. ;Sanchez Marre. A case-based recommendation approach for market basket data. *Intelligent Systems, IEEE*, 2014.

[9] D. Godoy and A. Amandi. User profiling in personal information agents: A survey. *Knowl. Eng. Rev.*, 20(4):329–361, Dec. 2005.

[10] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ICDM '08, pages 263–272, Washington, DC, USA, 2008. IEEE Computer Society.

[11] K. Järvelin and J. Kekäläinen. Ir evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, pages 41–48, New York, NY, USA, 2000. ACM.

[12] S. K. Jong-Seok Lee, Chi-Hyuck JunčňJaewook Lee. Classification-based collaborative filtering using market basket data. *Expert Systems with Applications*, 2005.

[13] G. Karypis. Evaluation of item-based top-n recommendation algorithms. In *Proceedings of the Tenth International Conference on Information and Knowledge Management*, CIKM '01, pages 247–254, New York, NY, USA, 2001. ACM.

[14] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In T. Leen, T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 556–562. MIT Press, 2001.

[15] W. Lin, S. A. Alvarez, and C. Ruiz. Efficient adaptive-support association rule mining for recommender systems. *Data Min. Knowl. Discov.*, 6(1):83–105, Jan. 2002.

[16] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, Jan. 2003.

[17] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[18] T. Mobasher, B. ; Sch. of Comput. Sci. Using sequential and non-sequential patterns in predictive web usage mining tasks. *The IEEE International Conference on Data Mining series*, 2002.

[19] R. Pan and M. 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*, KDD '09, pages 667–676, New York, NY, USA, 2009. ACM.

[20] Y. L. Pengfei Wang, Jiafeng Guo. Modeling retail transaction data for personalized shopping recommendation. In *23rd International Conference on Information and Knowledge Management*, 2014.

[21] T. M. Quoc V. Le. distributed representations of sentences and documents. *The 31st International Conference on Machine Learning*, 2014.

[22] 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*, UAI '09, pages 452–461, Arlington, Virginia, United States, 2009. AUAI Press.

[23] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 811–820, New York, NY, USA, 2010. ACM.

[24] G. Shani, R. I. Brafman, and D. Heckerman. An mdp-based recommender system. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, UAI'02, pages 453–460, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

[25] R. Srikant and R. Agrawal. Mining sequential patterns: Generalizations and performance improvements. In *Proceedings of the 5th International Conference on Extending Database Technology: Advances in Database Technology*, EDBT '96, pages 3–17, London, UK, UK, 1996. Springer-Verlag.

[26] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Adv. in Artif. Intell.*, 2009:4:2–4:2, Jan. 2009.

[27] K. C. G. C. J. D. Tomas Mikolov, Ilya Sutskever. Distributed representations of words and phrases and their compositionality. *Neural Information Processing Systems Foundation*, 2013.

[28] L. Xiang, Q. Yuan, S. Zhao, L. Chen, X. Zhang, Q. Yang, and J. Sun. Temporal recommendation on graphs via long- and short-term preference fusion. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 723–732, New York, NY, USA, 2010. ACM.

[29] G.-E. Yap, X.-L. Li, and P. S. Yu. Effective next-items recommendation via personalized sequential pattern mining. In *Proceedings of the 17th International Conference on Database Systems for Advanced Applications - Volume Part II*, DASFAA'12, pages 48–64, Berlin, Heidelberg, 2012. Springer-Verlag.