Factorizing Personalized Markov Chains for Next-Basket Recommendation

Steffen Rendle*
Department of Reasoning for Intelligence
The Institute of Scientific and Industrial Research
Osaka University, Japan
rendle@ar.sanken.osaka-u.ac.jp

Christoph Freudenthaler
Information Systems and Machine Learning Lab
Institute for Computer Science
University of Hildesheim, Germany
freudenthaler@ismll.uni-hildesheim.de

Lars Schmidt-Thieme
Information Systems and Machine Learning Lab
Institute for Computer Science
University of Hildesheim, Germany
schmidt-thieme@ismll.uni-hildesheim.de

ABSTRACT

Recommender systems are an important component of many websites. Two of the most popular approaches are based on matrix factorization (MF) and Markov chains (MC). MF methods learn the general taste of a user by factorizing the matrix over observed user-item preferences. On the other hand, MC methods model sequential behavior by learning a transition graph over items that is used to predict the next action based on the recent actions of a user. In this paper, we present a method bringing both approaches together. Our method is based on personalized transition graphs over underlying Markov chains. That means for each user an own transition matrix is learned – thus in total the method uses a transition cube. As the observations for estimating the transitions are usually very limited, our method factorizes the transition cube with a pairwise interaction model which is a special case of the Tucker Decomposition. We show that our factorized personalized MC (FPMC) model subsumes both a common Markov chain and the normal matrix factorization model. For learning the model parameters, we introduce an adaption of the Bayesian Personalized Ranking (BPR) framework for sequential basket data. Empirically, we show that our FPMC model outperforms both the common matrix factorization and the unpersonalized MC model both learned with and without factorization.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning—*Parameter Learning*

General Terms

Algorithms, Experimentation, Measurement, Performance

Keywords

Basket Recommendation, Markov Chain, Matrix Factorization

*Steffen Rendle is currently on leave from the Machine Learning Lab, University of Hildesheim, Germany.

WWW 2010, April 26–30, 2010, Raleigh, North Carolina, USA.
ACM 978-1-60558-799-8/10/04.

1. INTRODUCTION

A core technology of many recent websites are recommender systems. They are used for example to increase sales in e-commerce, clicking rates on websites or visitor satisfaction in general. In this paper, we deal with the problem setting where sequential basket data is given per user. An obvious example is an online shop where a user buys items (e.g. books or CDs). In these applications, usually several items are bought at the same time, i.e. we have a set/basket of items at one point of time. The target is now to recommend items to the user that he might want to buy in his next visit.

Recommender systems based on a Markov chain (MC) model utilize such sequential data by predicting the users next action based on the last actions. Therefore a transition matrix is estimated that gives the probability of buying an item based on the last purchases of the user. The transition matrix of the MC models is assumed to be the same over all users. The personalization is made by applying the (general) transition matrix on the user's last actions. On the other hand, one of the most successful model classes are factorization methods (MF) based on matrix or tensor decomposition. The best approaches [3, 4] for the 1M$ Netflix challenge[1] are based on this model class. Also on the ECML/PKDD discovery challenge[2] for tag recommendation, a factorization model based on tensor decomposition has outperformed the other approaches [8]. These models learn the general taste of the user disregarding sequential information. Both MF and MC have their advantages: MF uses all data to learn the general taste of the user whereas MC can capture sequence effects in time by using a non-personalized transition matrix, i.e. the transition matrix is learned over all data of all users.

In this paper, we present a model based on an underlying MC where the transitions are user-specific. We model a transition cube where each slice is a user-specific transition matrix of an underlying MC on the users basket history. With this personalization, we bring together both advantages of MC and MF: (1) the sequential data is captured by the transition matrix and (2) as all transition matrices are user-specific, the user-taste over all data is captured. Besides introducing personalized MCs, the central contribution of

[1] http://www.netflixprize.com/
[2] http://www.kde.cs.uni-kassel.de/ws/dc09

this work is the estimation of the transition tensor. Because of the sparsity of the data, it is not possible to get good estimates of the personalized transition matrices by using standard counting approaches (which are Maximum Likelihood Estimators) on a complete parametrization. Instead, we model the transition tensor by a factorization model. This allows to propagate information among similar users, similar items and similar transitions. By using a factorization model based on pairwise interactions, it is possible to deal with high sparsity. We show that this model subsumes both the MF model and the unpersonalized MC model. For learning the factorization parameters, the Bayesian Personalized Ranking (BPR) framework [7] is extended to basket data.

In our evaluation chapter, we apply our method to an anonymized real-world dataset of an e-commerce website. We show that our proposed method FPMC outperforms MF and MC.

In total the contributions are as follows:

- We introduce personalized Markov chains relying on personalized transition matrices. This allows to capture both sequential effects and long term user-taste. We show that this is a generalization of both standard MC and MF models.
- To deal with the sparsity for the estimation of transition probabilities, we introduce a factorization model that can be applied both to personalized and normal transition matrices. This factorization approach results in less parameters and due to generalization to a better quality than full parametrized models.
- We empirically show that our model outperforms other state-of-the-art methods on sequential data.

2. RELATED WORK

Markov chains or recommender systems have been studied by several researchers. Zimdars et al. [10] describe a sequential recommender based on Markov chains. They investigate how to extract sequential patterns to learn the next state with a standard predictor – e.g. a decision tree. Mobasher et al. [5] use pattern mining methods to discover sequential patterns which are used for generating recommendations. Shani et al. [9] introduce a recommender based on Markov decision processes (MDP) and also a MC based recommender. To enhance the maximum likelihood estimates (MLE) of the MC transition graphs, they describe several heuristic approaches like clustering and skipping. Instead of improving the MLE estimates with heuristics, we use a factorization model that is learned for optimal ranking instead of transition MLE. In total, the main difference of our work to all the previous approaches is the use of personalized transition graphs which bring together the benefits of sequential, i.e. time-aware, MC with time-invariant user taste. Furthermore factorizing transition probabilities and optimizing the parameters for ranking is new.

On the other hand, most of the recommender systems do not take sequential patterns into account and recommend based on the whole user history. Besides a very large literature on rating prediction (i.e. regression) emerging from the Netflix contest (e.g. [3, 4]), item recommendation from implicit feedback has started to get more into the focus [2, 6, 7]. Item recommendation is a harder prediction problem than rating prediction, as only positive observations are made and standard sparse regression and classification methods like the ones from the Netflix contest can not directly be applied. Three recent methods for item recommendation are based on the matrix factorization model that factorizes the matrix of user-item correlations. Both Hu et al. [2] and Pan and Scholz [6] optimize the factorization on user-item pairs (u, i) where observed pairs are treated as positive and non-observed ones as negative. Hu et al. [2] use a least-square optimization where case weights are used for controlling the importance of observations. Pan and Scholz [6] also use case weights but with several optimization criteria like hinge-loss and least-square. Because non-observation of an item does not mean that the user does not want to select it in the future, Rendle et al. [7] take another optimization approach by learning over observation pairs (u, i, j), e.g. a user u prefers item i over item j. All these methods have been shown to outperform standard approaches like k-nearest-neighbour based on Pearson similarity. In this work, we will bring the advantages of these MF models together with MC models.

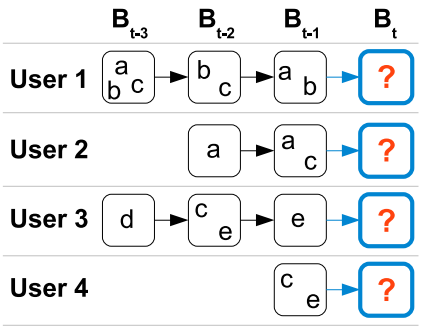

Figure 1: Sequential basket data with four users and five items $\{a, b, c, d, e\}$. The task is to recommend items at time t given a basket history $B_{t-1}, B_{t-2}, \ldots$.

3. ITEM RECOMMENDATION FROM SEQUENTIAL SET DATA

Item recommendation is the task of suggesting a specific user a personalized list of items (e.g. products, songs). This can be seen as creating a personalized ranking on the items. Usually, recommender systems rely on statistical models that use the event history (e.g. purchases, listening) of users on items to generate recommendations. Time and thus sequential behavior is an important additional information that is tracked in almost any real-world application. Secondly, we consider the problem setting with set data – e.g. in online-shopping usually a basket of products is bought at the same time. In total, our setting is item recommendation from sequential set data. An example of such data can be found in figure 1.

3.1 Sequential vs. General Recommender

The most common approach to generate recommendations is to discard any sequential information and learn what items a user is typically interested in. On the other hand, recommendations of sequential methods (mostly relying on Markov chains) are based only on the last user events by learning what an arbitrary user buys next when he has bought a certain item in the recent past. Both methods have their

strengths and disadvantages. Imagine a user that in general buys movies like 'Star Trek' and 'Star Wars'. In contrast to his usual buying behavior, he recently has purchased 'Titanic' and 'Dirty Dancing' to watch with his girlfriend. After that a MC based recommender of length 2 would only recommend movies like 'Notting Hill' and other romantic movies. In contrast, a global personalized recommender would correctly account for the general taste of the user and recommend also movies like 'Back To the Future', 'Alien' or other science fiction movies. But there are also examples where sequential recommenders have advantages: E.g. good recommendations for a user that has recently bought a digital camera are accessories that other users have bought after buying that camera – this is exactly what a Markov chain model does. Global personalized recommender would not adapt directly to the recent purchase (the digital camera) but would recommend items this user likes in general.

3.2 Formalization

Before describing our approach to solve this problem, we introduce the notation of this paper. Let $U = \{u_1, \dots, u_{|U|}\}$ be a set of users and $I = \{i_1, \dots, i_{|I|}\}$ a set of items. For each user u, a purchase history $\mathcal{B}^u$ of his baskets is known: $\mathcal{B}^u := (B_1^u, \dots B_{t_u-1}^u)$ with $B_t^u \subseteq I$. The purchase history of all users is $\mathcal{B} := \{\mathcal{B}^{u_1}, \dots \mathcal{B}^{u_{|U|}}\}$.

Given this history, the task is to recommend items to a user the next time t the user visits the shop. Note that we deal not with absolute time points (i.e. 1st January 2010) but with relative ones regarding a user, e.g. the first, second, etc. basket of a user. The item recommendation task can be formalized in creating a personal ranking

$$<_{u,t} \subset I^2$$

over all pairs of items for user u for his t-th basket. With this ranking, we can recommend the user the top n items.

4. FACTORIZING PERSONALIZED MARKOV CHAINS (FPMC)

First, we introduce MC for sequential set data and extend this to personalized MCs. We discuss the weakness of Maximum Likelihood Estimates for the transition cubes. To solve this, we introduce factorized transition cubes where information among transitions is propagated. We conclude this section by combining both ideas into FPMCs.

4.1 Personalized Markov Chains for Sets

First, we describe how to model the unpersonalized MC for sets with a reasonable state space. Then we show how to estimate the parameters for this unpersonalized MC with the maximum likelihood estimator (MLE). Afterwards, the extension of both the model and the estimation to personalized MCs is simple. Finally, we will show the limitations of full parametrized transition graphs (i.e. one parameter per transition) and the MLE method for personalized Markov chains.

4.1.1 Markov Chains for Sets

In general, a Markov chain of order m is defined as

$$p(X_t = x_t | X_{t-1} = x_{t-1}, \dots, X_{t-m} = x_{t-m}) \quad (1)$$

where $X_t, \dots, X_{t_m}$ are random variables and x_{t-m} their realizations. In a recommender application without sets, the random variables are defined over I – i.e. realizations are single items $i \in I$. But in our case, the variables are defined over $\mathcal{P}(I)$ as the realizations are whole baskets B and thus the size of the state space is $2^{|I|}$. Obviously, defining a long chain over the whole state space is not feasible for sets. To handle this huge state space, we make two simplifications: (1) we use chains of length $m = 1$ and (2) the transition probabilities are simplified.

An unpersonalized Markov chain of order $m = 1$ for the basket problem is:

$$p(B_t | B_{t-1}) \quad (2)$$

In recommender scenarios without sets, usually longer chains (e.g. $m = 3$) are preferable [9] because a history with size $m = 1$ contains only one item. In our case with sets, even a chain with length $m = 1$ is reasonable because it relies already on many items (all items of the basket) – e.g. in the application of our evaluation there are about 10 items on average (see table 1).

Markov chains of length $m = 1$ are described by their stochastic transition matrix A over the state space. In our case the state space over sets is $\mathcal{P}(I)$ and thus the dimensionality of the transition matrix would be $2^{|I|} \times 2^{|I|}$. Thus, instead of modeling transition over baskets, we model transitions over $|I|$ binary variables that describe a set/ basket:

$$a_{l,i} := p(i \in B_t | l \in B_{t-1}) \quad (3)$$

Using this representation has the following implications:

- The state space is now I and thus the size of the transition matrix A is $|I|^2$ – by factorization, we will later reduce the number of parameters needed to represent this space from $|I|^2$ to $2\,k\,|I|$ where k is the number of latent dimensions used in the factorization model.
- The elements of the state space are $i \in B$ which are binary variables, thus $p(i \in B_t | l \in B_{t-1}) + p(i \notin B_t | l \in B_{t-1}) = 1$. Note that the transition matrix A is no longer stochastic, because $\sum_{i \in I} a_{l,i} \neq 1$.

For item recommendation, we are interested in the probability of purchasing an item given the last basket of a user. This can be defined as the mean over all transition probabilities from purchases of the last basket to this item:

$$p(i \in B_t | B_{t-1}) := \frac{1}{|B_{t-1}|} \sum_{l \in B_{t-1}} p(i \in B_t | l \in B_{t-1}) \quad (4)$$

And the full Markov chain over baskets can be expressed by:

$$p(B_t | B_{t-1}) \propto \prod_{i \in B_t} p(i | B_{t-1}) \quad (5)$$

Note that we are looking for a ranked list of items and thus are not interested in the full Markov chain (eq. (5)), but in sortable single-item probabilities (eq. (4)).

4.1.2 Estimation of Transition Probabilities

To make predictions using the Markov chain in eq. (4), the transition probabilities $a_{l,i}$ have to be estimated. The maximum likelihood estimator for $a_{l,i}$ given the data $\mathcal{B}$ is:

$$\hat{a}_{l,i} = \hat{p}(i \in B_t | l \in B_{t-1}) = \frac{\hat{p}(i \in B_t \wedge l \in B_{t-1})}{\hat{p}(l \in B_{t-1})} = \\ = \frac{|\{(B_t, B_{t-1}) : i \in B_t \wedge l \in B_{t-1}\}|}{|\{(B_t, B_{t-1}) : l \in B_{t-1}\}|} \quad (6)$$

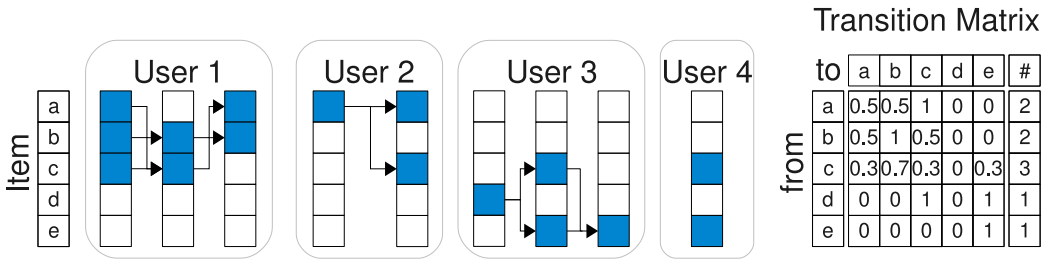

Figure 2: Non-personalized Markov chain: The transition matrix contains the MLE estimates for the probability $p(i \in B_t | l \in B_{t-1})$ using the data of figure 1. The column # states how many observations were used to estimate this transition. In this example, the users 1 and 2 as well as 3 and 4 share a similar taste for items a, c and items c, e respectively. Thus, one would expect to find d before b in the recommendation list for user 4, but the MC would recommend b as best unknown item.

An example for non-personalized MLE can be seen in figure 2. Here, the buying history for the four users of figure 1 are translated into transitions A of eq. (4). The transition matrix can then be applied to predict which items should be recommended given the last basket. E.g. for user 4, the probabilities would be:

$$p(a \in B_t | \{c, e\}) = 0.5(0.3 + 0.0) = 0.15$$
$$p(b \in B_t | \{c, e\}) = 0.5(0.7 + 0.0) = 0.35$$
$$p(c \in B_t | \{c, e\}) = 0.5(0.3 + 0.0) = 0.15$$
$$p(d \in B_t | \{c, e\}) = 0.5(0.0 + 0.0) = 0.00$$
$$p(e \in B_t | \{c, e\}) = 0.5(0.3 + 1.0) = 0.65$$

As the user has already bought item c and e, the best recommendation of unknown items would be b and then a. Looking only at the items this and similar users have bought in the past, one would expect, that item d might be a better recommendation.

4.1.3 Personalized Markov Chains for Sets

Until now, the MC has been defined unpersonalized, i.e. independently of the user. Next, we extend this to a personalized MC per user:

$$p(B_t^u | B_{t-1}^u) \tag{7}$$

Again, we represent each MC by the transitions over items, but now user-specific:

$$a_{u,l,i} := p(i \in B_t^u | l \in B_{t-1}^u) \tag{8}$$

And thus also the prediction depends only on the user's transitions:

$$p(i \in B_t^u | B_{t-1}^u) := \frac{1}{|B_{t-1}^u|} \sum_{l \in B_{t-1}^u} p(i \in B_t^u | l \in B_{t-1}^u) \tag{9}$$

Also MLE can be applied analogously but now the transitions for user u are only estimated from his history $\mathcal{B}^u$ – that means u is not a free variable anymore:

$$\hat{a}_{u,l,i} = \hat{p}(i \in B_t^u | l \in B_{t-1}^u) = \frac{\hat{p}(i \in B_t^u \wedge l \in B_{t-1}^u)}{\hat{p}(l \in B_{t-1}^u)}$$
$$= \frac{|\{(B_t^u, B_{t-1}^u) : i \in B_t^u \wedge l \in B_{t-1}^u\}|}{|\{(B_t^u, B_{t-1}^u) : l \in B_{t-1}^u\}|} \tag{10}$$

That means for each user we have an own transition matrix A^u which in total gives a transition tensor $\mathcal{A} \in [0, 1]^{|U| \times |I| \times |I|}$.

Figure (3) shows the personalized transition matrix of our example. Many of the parameters cannot be estimated because there is no observation in the data. Also the transitions that are estimated are based only on a small number of observations that means they are unreliable. At first glance, using personalized MCs seems to be unreasonable. We will discuss next what are the reasons for the poor estimations and show how to fix it.

4.1.4 Limitations of MLE and Full Parametrization

The problem of unreliable transition probabilities both for unpersonalized and even more for personalized MCs lies in the fact that they work with a full parametrized transition graph (e.g. matrix and tensor respectively) and the way of parameter estimation. Full parametrization means we have $|I|^2$ and $|U||I|^2$ respectively independent parameters for describing the transitions. Note that MLE estimates each transition parameter $a_{l,i}$ independently from the others, i.e. none of the cooccurrences (l, i) will contribute to another transition probability estimator (l, j) but only to $p(i \in B_t | l \in B_{t-1})$. This is even worse for personalized MCs as a triple (u, l, i) does not contribute to the estimate of (u', l, i). In addition, the important properties of MLE (e.g. Gaussian distribution, unbiased estimator, minimal variance under all unbiased estimators) only exist in asymptotic theory. In cases of less data they suffer from underfitting. Since in our scenario the data is extremely sparse, Maximum Likelihood Estimators easily fail.

To get more reliable estimates for the transitions, we factorize the transition cube which breaks the independence of the parameters and the estimation. This way, each transition is influenced by similar users, similar items and similar transitions because information propagates through this model. In our evaluation, we show that this way (1) better transition graphs than MLE can be generated for the non-personalized setting and (2) that personalized MCs outperform both non-personalized factorized MC and non-personalized full parametrized MLE MCs.

4.2 Factorizing Transition Graphs

In the following, we will derive a factorization model for the transition cube $\mathcal{A}$. That means we model the unobserved transition tensor $\mathcal{A}$ by a low rank approximation $\hat{\mathcal{A}}$. The advantage of this approach over a full parametrization is that it can handle sparsity and generalizes to unobserved

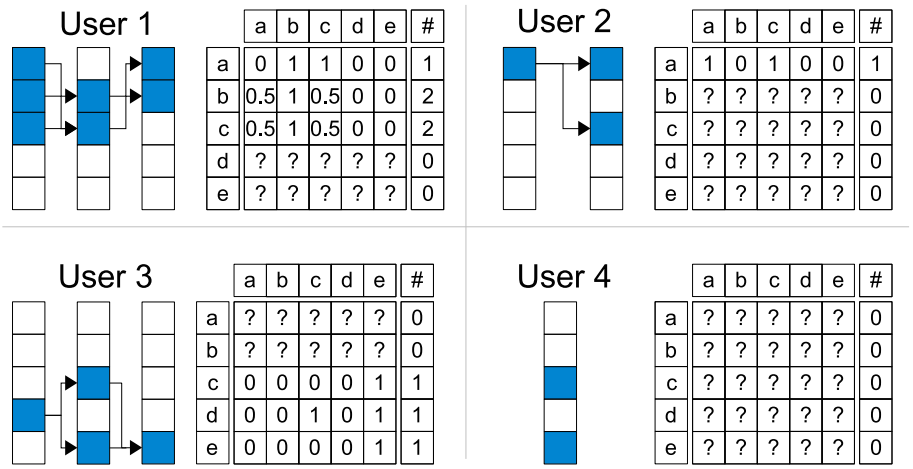

Figure 3: Personalized Markov chains: For each user an individual transition matrix is given. The transition matrices contain the MLE estimates for the probability $p(i \in B_t^u | l \in B_{t-1}^u)$. Entries with ? are missing values as there is no data to estimate the probabilities. Obviously, estimating the personalized transition matrices directly results in very poor transitions as each estimate is not reliable. This problem will be solved later by factorizing the transitions.

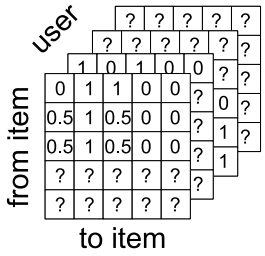

Figure 4: Personalized transition cube: Stacking all transition matrices of the individual users leads to a transition cube. Instead of a fully parametrized cube which is very sparse, a factored cube is used to generate better transition estimates.

data because information propagates through the model – i.e. parameters influence each other.

4.2.1 Factorization of the Transition Cube

A general linear factorization model for estimating the tensor $\mathcal{A}$ is the Tucker Decomposition (TD):

$$\hat{\mathcal{A}} := \mathcal{C} \times_U V^U \times_L V^L \times_I V^I \tag{11}$$

where $\mathcal{C}$ is a core tensor and V^U is the feature matrix for the users, V^L is the feature matrix for the items in the last transition (outgoing nodes) and V^I is the feature matrix for the items to predict (ingoing nodes). They have the following structure:

$$\mathcal{C} \in \mathbb{R}^{k_U, k_L, k_I}, \quad V^U \in \mathbb{R}^{|U| \times k_U}, \tag{12}$$

$$V^L \in \mathbb{R}^{|I| \times k_L}, \quad V^I \in \mathbb{R}^{|I| \times k_I} \tag{13}$$

with the factorization dimensions k_U, k_L and k_I.

The Tucker Decomposition subsumes other factorization models like the Canonical Decomposition (CD) aka parallel factor analysis (PARAFAC). The parallel factor model assumes a diagonal core tensor, i.e.

$$c_{f_u, f_i, f_j} = \begin{cases} 1, & \text{if } f_u = f_i = f_j \\ 0, & \text{else} \end{cases} \tag{14}$$

with equal factorization dimensionality: $k_U = k_L = k_I$.

As the observed transitions for $\mathcal{A}$ are very sparse, we use a special case of CD that models pairwise interactions:

$$\hat{a}_{u,l,i} := \langle v_u^{U,I}, v_i^{I,U} \rangle + \langle v_i^{I,L}, v_l^{L,I} \rangle + \langle v_u^{U,L}, v_l^{L,U} \rangle \tag{15}$$

or equivalently:

$$\hat{a}_{u,l,i} := \sum_{f=1}^{k_{U,I}} v_{u,f}^{U,I} \, v_{i,f}^{I,U} + \sum_{f=1}^{k_{I,L}} v_{i,f}^{I,L} \, v_{l,f}^{L,I} + \sum_{f=1}^{k_{U,L}} v_{u,f}^{U,L} \, v_{l,f}^{L,U} \tag{16}$$

This model directly models the pairwise interaction between all three modes of the tensor, i.e. between U and I, U and J as well as J and I. In total for each mode (i.e. user U, item I, item J), we have two factorization matrizes:

1. For the interaction between U and I: $V^{U,I} \in \mathbb{R}^{|U| \times k_{U,I}}$ modelling the user features and $V^{I,U} \in \mathbb{R}^{|I| \times k_{U,I}}$ for the last item i.

2. For the interaction between I and L: $V^{I,L} \in \mathbb{R}^{|I| \times k_{I,L}}$ for the next item i and $V^{L,I} \in \mathbb{R}^{|I| \times k_{I,L}}$ for the last item l.

3. For the interaction between U and L: $V^{U,L} \in \mathbb{R}^{|U| \times k_{U,L}}$ for the user features and $V^{L,U} \in \mathbb{R}^{|I| \times k_{U,L}}$ for the features of the last item l.

An advantage of this model over TD is that the prediction and learning complexity is much lower than for TD [8]. Furthermore even though TD and PARAFAC subsume the

pairwise interaction model, with standard regularization estimation procedures have problems identifying such a model [8].

In section 5 we describe how to optimize the model parameters (factorization matrices) for item recommendation.

4.2.2 Factorization of the Transition Matrix

The proposed model for factorizing transition cubes can also be applied to estimate a transition matrix A (see formula (3)) for cases where no personalization of the transition graph is desired. By skipping the user-interactions in equation (15), a factorization model for normal transition graphs is obtained:

$$\hat{a}_{l,i} := \langle v_i^{I,L}, v_l^{L,I} \rangle \tag{17}$$

Also the parameter estimation method in section 5 can be used for optimizing the factorization matrices.

4.3 Summary of FPMC

Bringing together the personalized set MC (eq. 9) with the factorized transition cube (eq. 15) results in the factorized personalized Markov chain (FPMC):

$$p(i \in B_t^u | B_{t-1}^u) = \frac{1}{|B_{t-1}^u|} \sum_{l \in B_{t-1}^u} p(i \in B_t^u | l \in B_{t-1}^u) \tag{18}$$

We model $p(i \in B_t^u | l \in B_{t-1}^u)$ with the factorization cube $\hat{\mathcal{A}}$:

$$\begin{aligned} \hat{p}(i \in B_t^u | B_{t-1}^u) &= \frac{1}{|B_{t-1}^u|} \sum_{l \in B_{t-1}^u} \hat{a}_{u,l,i} \\ &= \frac{1}{|B_{t-1}^u|} \sum_{l \in B_{t-1}^u} (\langle v_u^{U,I}, v_i^{I,U} \rangle + \langle v_i^{I,L}, v_l^{L,I} \rangle \\ &\qquad + \langle v_u^{U,L}, v_l^{L,U} \rangle) \end{aligned} \tag{19}$$

And as the factorization (U, I) is independent of l, we can remove it from the sum:

$$\begin{aligned} \hat{p}(i \in B_t^u | B_{t-1}^u) &= \langle v_u^{U,I}, v_i^{I,U} \rangle \\ &\quad + \frac{1}{|B_{t-1}^u|} \sum_{l \in B_{t-1}^u} \left(\langle v_i^{I,L}, v_l^{L,I} \rangle + \langle v_u^{U,L}, v_l^{L,U} \rangle \right) \end{aligned} \tag{20}$$

In the next section, we apply this model to the task of item recommendation. We will show that in this case, the model can be simplified even more because the interaction between U and L vanishes.

Besides better generalization of factorization models compared to a full parametrized transition cube, a further advantage is that less parameters are needed. Instead of $|U| \cdot |I|^2$ parameters in a full parametrized cube or $|I|^2$ in a full parametrized matrix, the factorization model only needs $2 \cdot k_{I,L} \cdot |I|$ parameters for the non-personalized model and $2 \cdot k_{I,L} \cdot |I| + k_{U,I} \cdot (|U| + |I|)$ parameters for the personalized model. This is especially important for applications with a high number of items where a full parametrization with $|I|^2$ parameters might not be feasible.

5. ITEM RECOMMENDATION FROM SEQUENTIAL SET DATA WITH FPMC

So far, a factorization model for personalized Markov chains has been introduced. In the following, we will apply this model to the task of item recommendation. That means, the model parameters should be optimized for ranking. First, we derive S-BPR which is a general optimization criterion for item recommendation from sequential set data. This optimization criterion is not limited to our FPMC model and can be applied also to other models like kNN or standard MF. Secondly, we apply S-BPR to FPMC and show how the model can be simplified in the case of item recommendation using S-BPR. Afterwards we present a stochastic gradient descent learning algorithm based on bootstrap sampling for optimizing the model parameters with S-BPR.

5.1 Optimization Criterion S-BPR

As described in section (3), the goal of item recommendation from sequential basket data is to derive a ranking $>_{u,t}$ over the items. To model the ranking, we assume there is an estimator $\hat{x} : U \times T \times I \to \mathbb{R}$ – e.g. the buying probability of the personalized Markov Chain – which is used to define the ranking:

$$i >_{u,t} j \ :\Leftrightarrow \ \hat{x}_{u,t,i} >_{\mathbb{R}} \hat{x}_{u,t,j} \tag{21}$$

As $>_{\mathbb{R}}$ is a total order on (a closed subset of) the real numbers $\mathbb{R}$, also $>_{u,t}$ will be a total order. Thus $\hat{x}_{u,t,i}$ is able to generate a personalized ranking[3] for a specific time t on the items I.

Next, we derive the sequential BPR (S-BPR) optimization criterion analogously to the general BPR approach [7]. The best ranking $>_{u,t} \subset I^2$ for user u at time t can be formalized as:

$$p(\Theta | >_{u,t}) \propto p(>_{u,t} | \Theta) \, p(\Theta)$$

where Θ are the model parameters – in our case the parameters are $\Theta = \{V^{U,I}, V^{I,U}, V^{L,I}, V^{I,L}, V^{U,L}, V^{L,U}\}$.

Assuming independence of baskets and users, this leads to the maximum a posterior (MAP) estimator of the model parameters:

$$\underset{\Theta}{\operatorname{argmax}} \prod_{u \in U} \prod_{B_t \in \mathcal{B}^u} p(>_{u,t} | \Theta) \, p(\Theta) \tag{22}$$

Expanding $>_{u,t}$ for all item-pairs $(i, j) \in I^2$ and using the same assumptions as in [7], the probability of $p(>_{u,t} | \Theta)$ can be rewritten as:

$$\prod_{u \in U} \prod_{B_t \in \mathcal{B}^u} \prod_{i \in B_t} \prod_{j \notin B_t} p(i >_{u,t} j | \Theta) \tag{23}$$

Next we use the model definition of eq. (21) to express $p(i >_{u,t} j | \Theta)$:

$$\begin{aligned} p(i >_{u,t} j | \Theta) &= p(\hat{x}_{u,t,i} >_{\mathbb{R}} \hat{x}_{u,t,j} \,|\, \Theta) && (24) \\ &= p(\hat{x}_{u,t,i} - \hat{x}_{u,t,j} >_{\mathbb{R}} 0 \,|\, \Theta) && (25) \end{aligned}$$

The Θ can be skipped as they are the model parameters for $\hat{x}$ – i.e. $\hat{x} = \hat{x}(\Theta)$. And we define $p(z > 0) := \sigma(z) = \frac{1}{1+e^{-z}}$ using the logistic function σ:

$$p(i >_{u,t} j | \Theta) = \sigma(\hat{x}_{u,t,i} - \hat{x}_{u,t,j}) \tag{26}$$

Furthermore, we assume Gaussian priors on the model parameters: $\theta \sim N(0, \frac{1}{\lambda_\theta})$.

[3] In case of identities $\hat{x}_{u,t,i} = \hat{x}_{u,t,j}$ a random order between these two is chosen.

In total this leads to the MAP-estimator for sequential BPR:

$$\begin{aligned}&\operatorname*{argmax}_{\Theta} \ln p(>_{u,t}|\Theta)\, p(\Theta)\\ =&\operatorname*{argmax}_{\Theta} \ln \prod_{u\in U}\prod_{B_t\in\mathcal{B}^u}\prod_{i\in B_t}\prod_{j\notin B_t}\sigma(\hat{x}_{u,t,i}-\hat{x}_{u,t,j})p(\Theta)\\ =&\operatorname*{argmax}_{\Theta} \sum_{u\in U}\sum_{B_t\in\mathcal{B}^u}\sum_{i\in B_t}\sum_{j\notin B_t}\ln\sigma(\hat{x}_{u,t,i}-\hat{x}_{u,t,j})-\lambda_\Theta||\Theta||_F^2\end{aligned} \tag{27}$$

where λ_Θ is the regularization constant corresponding to σ_Θ.

5.2 Item Recommendation with FPMC

For item recommendation with FPMC, we express $\hat{x}$ by the FPMC model and apply S-BPR. We will show that one of the pairwise effects of FPMC vanishes which leads to a more compact model.

First, we use FPMC to express $\hat{x}$:

$$\begin{aligned}&\hat{x}'_{u,t,i} := \hat{p}(i\in B_t^u|B_{t-1}^u)\\ =&\langle v_u^{U,I}, v_i^{I,U}\rangle + \frac{1}{|B_{t-1}^u|}\sum_{l\in B_{t-1}^u}\left(\langle v_i^{I,L}, v_l^{L,I}\rangle + \langle v_u^{U,L}, v_l^{L,U}\rangle\right)\end{aligned}$$

LEMMA 1 (INVARIANCE OF (U,L) DECOMPOSITION). *For ranking of items and optimization with S-BPR, the FPMC model is invariant to the (U,L) decomposition, i.e. $\hat{x}'$ is invariant to $\hat{x}$ with:*

$$\hat{x}_{u,t,i} := \langle v_u^{U,I}, v_i^{I,U}\rangle + \frac{1}{|B_{t-1}^u|}\sum_{l\in B_{t-1}^u}\langle v_i^{I,L}, v_l^{L,I}\rangle \tag{28}$$

PROOF. Let $>'$ be the ranking generated by $\hat{x}'$ and $>$ the ranking of $\hat{x}$ according to eq. (21). Two things have to be shown: (1) both models ($\hat{x}'$ and $\hat{x}$) lead to the same ranking and (2) learning both models with S-BPR leads to the same parameters Θ. Both proofs rely on the fact that:

$$\forall u,t,i,j:\ \hat{x}'_{u,t,i}-\hat{x}'_{u,t,j} = \hat{x}_{u,t,i}-\hat{x}_{u,t,j} \tag{29}$$

This holds because the additional term $\sum_{l\in B_{t-1}^u}\langle v_u^{U,L}, v_l^{L,U}\rangle$ in $\hat{x}'_{u,t,\cdot}$ is independent of i and j given u and t and thus vanishes on subtraction. Now it is easy to show the equivalence of the rankings for all u,t,i,j:

$$\begin{aligned}&(i >'_{u,t} j) \Leftrightarrow (\hat{x}'_{u,t,i} >_{\mathbb{R}} \hat{x}'_{u,t,j}) \Leftrightarrow (\hat{x}'_{u,t,i}-\hat{x}'_{u,t,j} >_{\mathbb{R}} 0)\\ \overset{eq.29}{\Leftrightarrow}&(\hat{x}_{u,t,i}-\hat{x}_{u,t,j} >_{\mathbb{R}} 0) \Leftrightarrow (\hat{x}_{u,t,i} >_{\mathbb{R}} \hat{x}_{u,t,j} \Leftrightarrow i >_{u,t} j)\end{aligned}$$

(2) The equivalence of model parameters under S-BPR optimization (eq. (27)) follows directly from eq. (29). □

Thus for item recommendation with FPMC the simpler model $\hat{x}$ from eq. (28) should be used.

5.2.1 Expressiveness

Next, we will show the analogies of the simplified FPMC model to standard matrix factorization (MF) and a factorized Markov chain (FMC). First, we will recollect the definitions of MF and FMC. In our notation, the standard Matrix factorization model for item recommendation [2, 6, 7] is:

$$\hat{x}_{u,t,i}^{\mathrm{MF}} = \langle v_u^{U,I}, v_i^{I,U}\rangle \tag{30}$$

where $\hat{x}$ is independent of the sequential behaviour, i.e. independent of t.

Factorizing an unpersonalized Markov chain using equation (4) and (17) leads to:

$$\hat{x}_{u,t,i}^{\mathrm{FMC}} := \frac{1}{|B_{t-1}|}\sum_{l\in B_{t-1}}\langle v_i^{I,L}, v_l^{L,I}\rangle \tag{31}$$

Thus FPMC (eq. (28)) is a linear combination of both models:

$$\hat{x}_{u,t,i}^{\mathrm{FPMC}} = \hat{x}_{u,t,i}^{\mathrm{MF}} + \hat{x}_{u,t,i}^{\mathrm{FMC}} \tag{32}$$

This means FPMC can generalizes both models: By setting the factorization dimensionality of (U,I) to zero ($k_{U,I}=0$) a pure FMC is obtained and analogously setting $k_{I,L}=0$ leads to a pure MF model.

It is important to note, that even though the model equation for FPMC in the case of item recommendation can be expressed by a combination of a MF and a FMC model, it is different from a simple ensemble of a single MF with a single FMC model because in our case the model parameters are learned jointly. Thus the learned model parameters jointly represent the personalized Markov chain instead of just pure user-item interactions and a global MC. This gets more obvious in the general case of FPMC where the model equation cannot be expressed by a linear combination of MC and FMC. Examples are (1) optimizing for another objective criterion (e.g. least-square) where the (U,L) decomposition cannot be dropped because here the invariance to the objective (Lemma 1) does not hold like in S-BPR. And (2) using another factorization model for $\mathcal{A}$ in FPMC than pairwise interaction (e.g. PARAFAC or TD) also leads to a different model equation even for item recommendation with S-BPR.

5.3 Learning Algorithm

Next, we adapt the BPR-learning algorithm to S-BPR and apply it to FPMC. As FPMC subsumes MF and FMC, both of these models can also be optimized for S-BPR with the provided algorithm.

Trying to optimize S-BPR directly is time consuming, because the number of (u,t,i,j) quadruples is huge, i.e. $O(|S|\,|I|)$ where $S := \{(u,t,i)|u\in U,\ B_t^u\in\mathcal{B}^u,\ i\in B_t^u\}$. Thus standard gradient descent and also basket-wise stochastic gradient descent methods will converge very slowly (see [7] for more details) and are not applicable for problems of reasonable size. Instead, we follow [7, 8] and draw the quadruples independently by bootstrapping and perform stochastic gradient descent on these bootstrap samples. This learning method has been shown to be efficient for two related problem classes: standard item recommendation [7] and tag recommendation [8].

The complete algorithm is shown in figure 5. In each iteration a quadruple (u,t,i,j) is drawn consisting of an item i in the basket B_t^u of user u at time t and an item j that is not in this basket. Then gradient descent on S-BPR using this quadruple is performed. The gradients of S-BPR with respect to a model parameter θ and a given (u,t,i,j) are:

$$\begin{aligned}&\frac{\partial}{\partial\theta}\left(\ln\sigma(\hat{x}_{u,t,i}-\hat{x}_{u,t,j})-\lambda_\theta\,\theta^2\right)\\ =&(1-\sigma(\hat{x}_{u,t,i}-\hat{x}_{u,t,j}))\frac{\partial}{\partial\theta}(\hat{x}_{u,t,i}-\hat{x}_{u,t,j})-2\,\lambda_\theta\,\theta\end{aligned}$$

procedure LEARNSBPR-FPMC(S)
  draw $V^{U,I}, V^{I,U}, V^{I,L}, V^{L,I}$ from $N(0, \sigma^2)$
  repeat
    draw (u, t, i) uniformly from S
    draw j uniformly from $(I \setminus B_t^u)$
    $\delta \leftarrow (1 - \sigma(\hat{x}_{u,t,i} - \hat{x}_{u,t,j}))$
    for $f \in \{1, \dots, k_{U,I}\}$ **do**
      $v_{u,f}^{U,I} \leftarrow v_{u,f}^{U,I} + \alpha \left(\delta \left(v_{i,f}^{I,U} - v_{j,f}^{I,U} \right) - \lambda_{U,I} \, v_{u,f}^{U,I} \right)$
      $v_{i,f}^{I,U} \leftarrow v_{i,f}^{I,U} + \alpha \left(\delta \, v_{u,f}^{U,I} - \lambda_{I,U} \, v_{i,f}^{I,U} \right)$
      $v_{j,f}^{I,U} \leftarrow v_{j,f}^{I,U} + \alpha \left(-\delta \, v_{u,f}^{U,I} - \lambda_{I,U} \, v_{j,f}^{I,U} \right)$
    end for
    $\eta \leftarrow \frac{1}{|B_{t-1}^u|} \sum_{l \in B_{t-1}^u} v_{l,f}^{L,I}$
    for $f \in \{1, \dots, k_{I,L}\}$ **do**
      $v_{i,f}^{I,L} \leftarrow v_{i,f}^{I,L} + \alpha \left(\delta \, \eta - \lambda_{I,L} \, v_{i,f}^{I,L} \right)$
      $v_{j,f}^{I,L} \leftarrow v_{j,f}^{I,L} + \alpha \left(-\delta \, \eta - \lambda_{I,L} \, v_{j,f}^{I,L} \right)$
      for $l \in B_{t-1}^u$ **do**
        $v_{l,f}^{L,I} \leftarrow v_{l,f}^{L,I} + \alpha \left(\delta \frac{v_{i,f}^{I,L} - v_{j,f}^{I,L}}{|B_{t-1}^u|} - \lambda_{L,I} \, v_{l,f}^{L,I} \right)$
      end for
    end for
  until convergence
  return $V^{U,I}, V^{I,U}, V^{I,L}, V^{L,I}$
end procedure

Figure 5: Optimizing FPMC for S-BPR with learning rate α and regularization parameters $\lambda_{U,I}, \lambda_{I,U}, \lambda_{I,L}, \lambda_{L,I}$.

with

$$\frac{\partial}{\partial v_{u,f}^{U,I}}(\hat{x}_{u,t,i} - \hat{x}_{u,t,j}) = v_{i,f}^{I,U} - v_{j,f}^{I,U}$$
$$\frac{\partial}{\partial v_{i,f}^{I,U}}(\hat{x}_{u,t,i} - \hat{x}_{u,t,j}) = v_{u,f}^{U,I}$$
$$\frac{\partial}{\partial v_{j,f}^{I,U}}(\hat{x}_{u,t,i} - \hat{x}_{u,t,j}) = -v_{u,f}^{U,I}$$
$$\frac{\partial}{\partial v_{l,f}^{L,I}}(\hat{x}_{u,t,i} - \hat{x}_{u,t,j}) = \frac{1}{|B_{t-1}^u|}(v_{i,f}^{I,L} - v_{j,f}^{I,L})$$
$$\frac{\partial}{\partial v_{i,f}^{I,L}}(\hat{x}_{u,t,i} - \hat{x}_{u,t,j}) = \frac{1}{|B_{t-1}^u|} \sum_{l \in B_{t-1}^u} v_{l,f}^{L,I}$$
$$\frac{\partial}{\partial v_{j,f}^{I,L}}(\hat{x}_{u,t,i} - \hat{x}_{u,t,j}) = -\frac{1}{|B_{t-1}^u|} \sum_{l \in B_{t-1}^u} v_{l,f}^{L,I}$$

The complexity of the algorithm is $O(\#\text{it} \, (k_{U,I} + k_{I,L} \, \overline{|B|}))$ where $\overline{|B|}$ the average basket size in $\mathcal{B}$ and #it is the number of iterations.

6. EVALUATION

We empirically compare the recommender quality of our proposed factorized MC methods (factorized personalized Markov chain FPMC and factorized Markov chain FMC) to non-factorized Markov chain ('MC dense'), matrix factorization (MF) and the most-popular baseline (MP) – i.e. ranking all items by how often they have been bought in the past. Note that this comparison includes the strong baseline method BPR-MF [7]. As MF ($k_{I,L} = 0$) and FMC ($k_{U,I} = 0$) are a special case of FPMC, we use the FPMC learning algorithm for all three methods.

6.1 Dataset

We evaluate our recommender on anonymized purchase data of an online drug store[4]. The dataset we used is a 10-core subset, i.e. every user bought in total at least 10 items ($\sum_{B \in \mathcal{B}^u} |B|) > 10$ and vice versa each item was bought by at least 10 users. The statistics of the dataset can be found in table 1. We also created a dense subset of the 10-core dataset to study the effect of sparsity on the methods.

6.2 Evaluation Metrics

We evaluated by splitting the dataset S into two non overlapping sets: a training set[5] S_{train} and a testing set S_{test}. This split is done by putting the last basket for each user into S_{test} and the remaining ones into S_{train}. The recommenders were trained on S_{train} and then the performance on S_{test} is measured. We removed those users from the evaluation that have bought less then 10 different items in the past (i.e. S_{train}). Secondly, for each user we removed all items from the test baskets (and the corresponding predictions) that this user has already bought in the past – this is because we want to recommend to the user items that are new/ unknown to him. Note that this makes the prediction task much harder and explains the low f-measure of all methods in figure 6. Otherwise just rerecommending already bought items would be a simple but very successful strategy for non-durable products in drug stores like toothbrushes or cleaner. However, this is not the task of recommender systems because they should help the user to discover new things.

The quality is measured for each user u on the basket B_u in the test dataset. Therefore we rank all items with our methods and let $\hat{r}_u : I \leftrightarrow \{1, \dots, |I|\}$ be the (bijective) mapping from an item i to its (predicted) rank. We use the following quality measures to evaluate the estimated ranking against the actual bought items:

- Half-life-utility (HLU) aka 'Breese score' [1]:

$$\text{HLU}(B, \hat{r}_u) := 100 \, \frac{\sum_{r=1}^{|I|} \delta(\hat{r}_u^{-1}(r) \in B) \, 2^{-\frac{r-1}{\alpha-1}}}{\sum_{r=1}^{|B|} 2^{-\frac{r-1}{\alpha-1}}}$$

  Where we set the half-life parameter α to 5. We report the average HLU over all test baskets.

- Precision and recall of the top-N list:

$$Top(\hat{r}_u, N) := \{\hat{r}_u^{-1}(1), \dots, \hat{r}_u^{-1}(N)\}$$
$$\text{Prec}(B, \hat{r}_u, N) := \frac{|\text{Top}(\hat{r}_u, N) \cap B|}{N}$$
$$\text{Rec}(B, \hat{r}_u, N) := \frac{|\text{Top}(\hat{r}_u, N) \cap B|}{|B|}$$

  We report the f-measure (harmonic mean) over the average precision and average recall over all test baskets using top-5 list.

[4] http://www.rossmannversand.de/

[5] Hyperparameter search is done by removing for each user the last basket of S_{train} and using these baskets for the validation set.

Table 1: Characteristics of the datasets in our experiments in terms of number of users, items, baskets and triples (u, i, t) where t is the sequential time of the basket. The dense dataset is a subset of the sparse one containing the 10,000 users with most purchases and the 1000 most purchased items.

dataset	users $\|U\|$	items $\|I\|$	baskets	avg. basket size	avg. baskets per user	triples
Drug store 10-core (sparse)	71,602	7,180	233,476	11.3	3.2	2,635,125
Drug store (dense)	10,000	1,002	90,655	9.2	9.0	831,442

Table 2: Properties of the MC transition matrix estimated by the counting scheme. For the sparse dataset, only 12% of the entries of the transition matrix are non-zero and non-missing. For the dense subset, 88% are filled.

dataset	total	missing values	non-zero	zero
Drug store 10-core (sparse)	51,552,400 (100%)	1,041,100 (2.0%)	6,234,371 (12.1 %)	44,276,929 (85.9%)
Drug store (dense)	1,004.004 (100%)	0 (0.0%)	889,419 (88.6 %)	114,585 (11.4%)

- Area under the ROC curve:

$$\mathrm{AUC}(B, \hat{r}_u) := \frac{1}{|B| \cdot |I \setminus B|} \sum_{i \in B} \sum_{j \in I \setminus B} \delta(\hat{r}_u(i) < \hat{r}_u(j))$$

  We report the average AUC over all test baskets.

The runtime of model training linearly depends on the number of features. With our implementation, training of the largest models ($k = 128$) took about 4 hours for MF, 31 hours for FMC and 34 hours for FPMC on the larger (sparse) dataset.

6.3 Results

In figure 6 you can see the quality on the sparse and dense online-shopping dataset. For the factorization methods we run each method with $k_{U,I} = k_{I,L} \in \{8, 16, 32, 64, 128\}$ factorization dimension. The x-axis of the diagrams reflects this increasing dimensionality. As expected all methods outperform the most-popular baseline clearly on both datasets and all quality measures. Secondly, with reasonable factorization dimensions (e.g. 32) all the factorization methods outperform the standard MC method. And in total, the factorized personalized MC (FPMC) outperforms all other methods.

6.3.1 MC vs. FMC

First, we want to discuss the advantage of factorization over a dense transition model by comparing MC with non-personalized FMC. The results indicate that learning a factorized transition matrix leads to better estimates than usual counting schemes. Factorization has two advantages (1) it can densify a sparse transition matrix and (2) it prevents overfitting of the estimates by using a low-rank approximation. The sparseness of the transition matrix estimated by counting schemes can be seen in table 2. In the dense setting also the transition matrix is filled in 88% whereas on the sparse dataset this drops to 12%. Comparing the quality on the sparse and dense setting in figure 6, one can see that the advantages of FMC over MC are much higher in the sparse setting than in the dense one. But even in the dense setting where also MC's transition matrix is almost completely filled, FMC outperforms MC because the factorization prevents overfitting by using less parameters.

6.3.2 MF vs. FMC vs. FPMC

Comparing the factorized Markov chain with the matrix factorization, one can see that in the dense setting MF seems to outperform MC whereas in the sparse one MC is superior. The reason could be that in the dense setting there is much more information per user, thus the MF method using all the users purchase information has advantages over the MC model that only relies on the last purchases. And the other way around, MC has advantages on the sparse dataset. FPMC that combines the advantages of both methods outperforms them on both datasets.

7. CONCLUSION

In this paper, we have introduced a recommender method based on personalized Markov chains over sequential set data. Instead of using the same transition matrix for all users, this method uses an individual transition matrix for each user which in total results in a transition cube. As direct estimation (e.g. by Maximum Likelihood) over a full parametrized transition cube leads to very poor estimates, we introduce a factorization model that gives a low-rank approximation to the transition cube. The advantages of this approach is that each transition is influenced by transitions of similar users, similar items and similar transitions. Thus the quality of the final transition graph is much higher than that of a full parametrized model. Secondly, we apply factorized personalized Markov chains (FPMC) to the task of item recommendation with sequential set data by extending the BPR framework [7]. Additionally, we show that FPMC subsumes the popular matrix factorization model and a non-personalized factorized Markov chain. Due to the expressiveness of FPMC it combines the advantages of both the state-of-the-art global personalized approach (MF) and the sequential MC method. Empirically, we show on real-world data that FPMC outperforms MF, FMC and normal MC both on sparse and dense data.

Acknowledgments

We would like to thank Artus Krohn-Grimberghe for preparing the data set. Steffen Rendle is supported by a research fellowship of the Japan Society for the Promotion of Science (JSPS). This work is partially co-funded through the European Commission FP7 project MyMedia (www.mymediaproject.org) under the grant agreement no. 215006. This work is co-funded by the European Regional Development Fund project LEFOS (www.ismll.uni-hildesheim.de) under the grant agreement no. 62700.

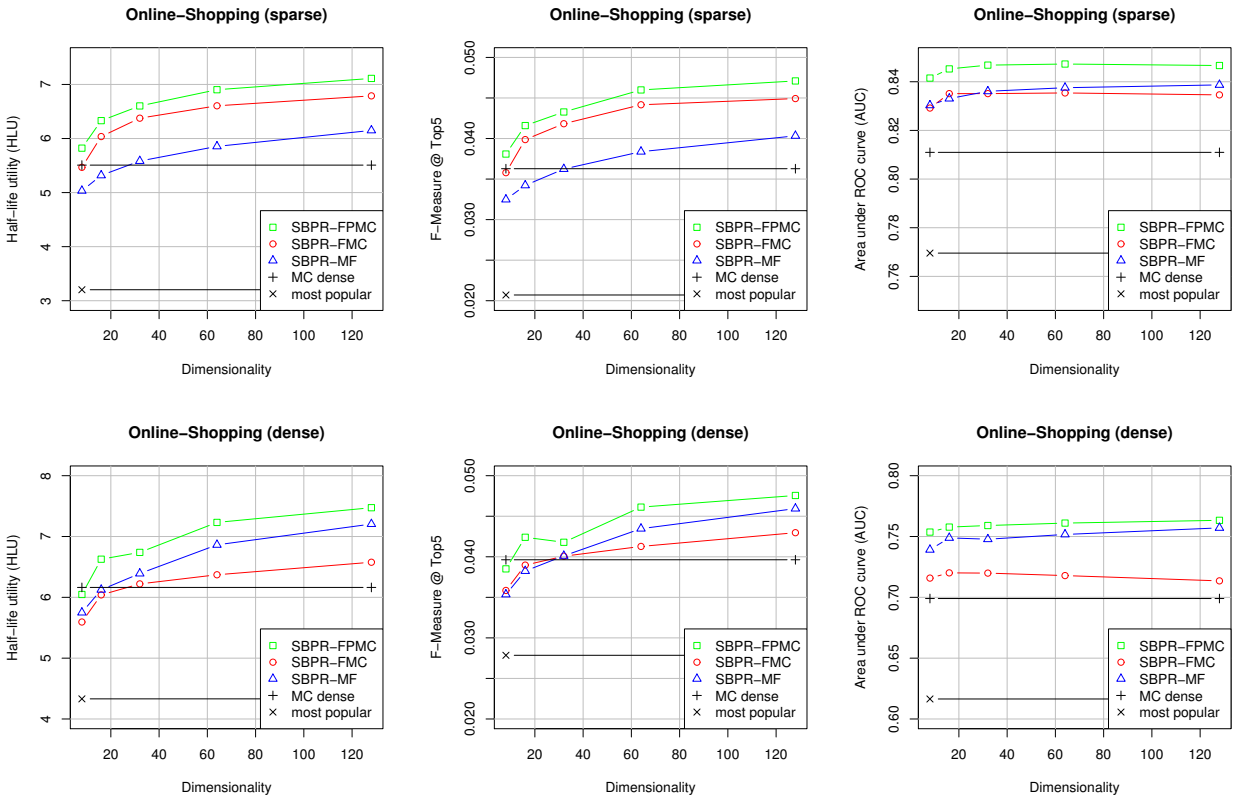

Figure 6: Comparison of factorized personalized Markov chains (FPMC) to a factorized Markov chain (FMC), matrix factorization (MF) [7], a standard dense Markov chain (MC dense) learned with Maximum Likelihood and the baseline 'most-popular'. The factorization dimensionality is increased from 8 to 128.

8. REFERENCES

[1] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pages 43–52, San Francisco, 1998. Morgan Kaufmann.

[2] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *IEEE International Conference on Data Mining (ICDM 2008)*, pages 263–272, 2008.

[3] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434, New York, NY, USA, 2008. ACM.

[4] Y. Koren. Collaborative filtering with temporal dynamics. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 447–456, New York, NY, USA, 2009. ACM.

[5] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa. Using sequential and non-sequential patterns in predictive web usage mining tasks. In *ICDM '02: Proceedings of the 2002 IEEE International Conference on Data Mining*, page 669, Washington, DC, USA, 2002. IEEE Computer Society.

[6] R. Pan and M. Scholz. Mind the gaps: weighting the unknown in large-scale one-class collaborative filtering. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 667–676, New York, NY, USA, 2009. ACM.

[7] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI 2009)*, 2009.

[8] S. Rendle and L. Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining (WSDM 2010)*. ACM, 2010.

[9] G. Shani, D. Heckerman, and R. I. Brafman. An mdp-based recommender system. *Journal of Machine Learning Research*, 6:1265–1295, 2005.

[10] A. Zimdars, D. M. Chickering, and C. Meek. Using temporal data for making recommendations. In *UAI '01: Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, pages 580–588, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.