Translation-based Recommendation: A Scalable Method for Modeling Sequential Behavior

Ruining He, Wang-Cheng Kang, Julian McAuley
UC San Diego
{r4he, wckang, jmcauley}@ucsd.edu

Abstract

Modeling the complex interactions between users and items is at the core of designing successful recommender systems. One key task consists of predicting users' personalized sequential behavior, where the challenge mainly lies in modeling 'third-order' interactions between a user, her previously visited item(s), and the next item to consume. In this paper, we propose a unified method, *TransRec*, to model such interactions for large-scale sequential prediction. Methodologically, we embed items into a 'transition space' where users are modeled as *translation* vectors operating on item sequences. Empirically, this approach outperforms the state-of-the-art on a wide spectrum of real-world datasets.

1 Introduction

In order to predict *sequential* user actions like the next product to purchase, movie to watch, or place to visit, it is essential (and challenging) to model the *third-order* interactions between a user u, the item(s) i she recently consumed, and the item j to visit next. Not only does the model need to handle the complexity of the interactions themselves, but also the scale and inherent sparsity of real-world data.

Traditional recommendation methods usually excel at modeling two-way (i.e., pairwise) interactions. This includes Matrix Factorization (MF) techniques [Koren *et al.*, 2009] that make use of inner products to model the compatibility between user-item pairs (i.e., user preferences). Likewise, (first-order) Markov Chain (MC) models [Serfozo, 2009] capture transition relationships between pairs of adjacent items in sequences (i.e., sequential dynamics), often by way of factorizing the transition matrix in favor of generalization ability. For the task of sequential recommendation, researchers have made use of scalable tensor factorization methods, such as Factorized Personalized Markov Chains (FPMC) [Rendle *et al.*, 2010]. FPMC models third-order relationships between u, i, and j by the *summation* of two pairwise relationships: one for the compatibility between u and the next item j, and another for the sequential continuity between the previous item i and the next item j. Ultimately, this is a combination of MF and MC (see Section 3.2 for details).

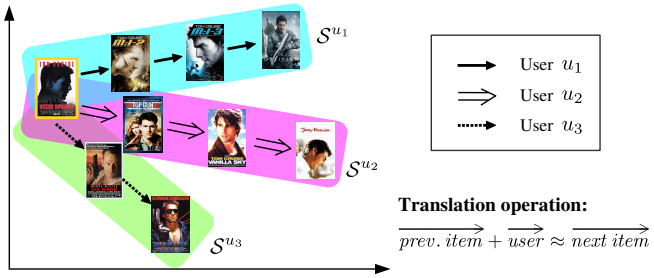

Figure 1: *TransRec*: Items (movies) are embedded into a 'transition space' where each user is modeled by a *translation* vector. The transition of a user from one item to another is captured by a user-specific translation operation.

Recently, there have been two lines of work that aim to improve FPMC. Personalized metric embedding methods replace the inner products in FPMC with Euclidean distances, where the metricity assumption—especially the triangle inequality—enables the model to generalize better [Wu *et al.*, 2013; Moore *et al.*, 2013; Feng *et al.*, 2015]. However, these works still adopt a framework that models the user preference component and sequential continuity component separately, which may be disadvantageous as the two components are inherently correlated. Another line of work [Wang *et al.*, 2015] makes use of operations like average/max pooling to *aggregate* the representations of the user u and the previous item i, before their compatibility with the next item j is measured. These works partially address the issue of modeling the dependence between the two key components, though are hard to interpret and can not benefit from the generalization ability of metric embeddings.

In this paper, we aim to tackle the above issues by introducing a new framework called *Translation-based Recommendation* (*TransRec*). The key idea behind *TransRec* is presented in Figure 1: Items are embedded as points in a (latent) 'transition space'; each user is represented as a 'translation vector' in the same space. Then, the third-order interactions mentioned earlier are captured by a personalized translation operation: the coordinates of the previous item i, plus the translation vector of u determine (approximately) the coordinates of the next item j, i.e., $\vec{\gamma}_i + \vec{t}_u \approx \vec{\gamma}_j$. Finally, we model the compatibility of the triplet (u, i, j) with a distance function

$d(\vec{\gamma}_i + \vec{t}_u, \vec{\gamma}_j)$. At prediction time, recommendations can be made via nearest-neighbor search centered at $\vec{\gamma}_i + \vec{t}_u$.

The advantages of such an approach are three-fold: (1) It naturally models third-order interactions with only a *single* component; (2) It also enjoys the generalization benefits of the implicit metricity assumption; and (3) It can easily handle large sequences (e.g. millions of instances) due to its simple form. Empirically, we conduct comprehensive experiments on a wide range of large, real-world datasets (which are publicly available), and quantitatively demonstrate the superior recommendation performance achieved by *TransRec*.

2 Related Work

General recommendation. Traditional approaches to recommendation ignore sequential signals in the system. Such systems focus on modeling user preferences, and typically rely on Collaborative Filtering (CF) techniques, especially Matrix Factorization (MF) [Ricci *et al.*, 2011]. For implicit feedback data (like purchases, clicks, and thumbs-up), pointwise (e.g. [Hu *et al.*, 2008; Pan *et al.*, 2008; Ning and Karypis, 2011]) and pairwise methods (e.g. [Rendle *et al.*, 2009; Rendle and Schmidt-Thieme, 2010]) based on MF have been proposed.

Sequential recommendation. Scalable sequential models usually rely on Markov Chains (MC) to capture sequential patterns (e.g. [Rendle *et al.*, 2010; Wang *et al.*, 2015; Feng *et al.*, 2015]). Rendle *et al.* proposed to factorize the third-order 'cube' that represents the transitions made by users among items. The resulting model, Factorized Personalized Markov Chains (FPMC), can be seen as a combination of MF and MC and achieves good performance for next-basket recommendation.

There are also works that have adopted metric embeddings for the recommendation task, leading to better generalization ability. For example, Chen *et al.* introduced Logistic Metric Embeddings (LME) for music playlist generation [Chen *et al.*, 2012], where the Markov transitions among different songs are encoded by the distances among them. Recently, Feng *et al.* further extended LME to model personalized sequential behavior and used pairwise ranking for predicting next points-of-interest [Feng *et al.*, 2015]. On the other hand, Wang *et al.* recently introduced the Hierarchical Representation Model (HRM), which extends FPMC by applying aggregation operations (like max/average pooling) to model more complex interactions [Wang *et al.*, 2015]. We will give more details of these works in Section 3.2.

Our work differs from the above in that we introduce a *translation*-based structure which naturally models the third-order interactions between a user, the previous item, and the next item for personalized Markov transitions.

Knowledge bases. Although different from recommendation, there has been a large body of work on knowledge bases that focuses on modeling multiple, complex relationships between various entities. Recently, partially motivated by the findings made by word2vec [Mikolov *et al.*, 2013], translation-based methods (e.g. [Bordes *et al.*, 2013; Lin *et al.*, 2015; Wang *et al.*, 2014]) have achieved state-of-the-art accuracy and scalability, in contrast to those achieved by traditional embedding methods relying on tensor decomposition or collective matrix factorization (e.g. [Nickel *et al.*, 2011; Nickel *et al.*, 2012; Singh and Gordon, 2008]). Our work is inspired by those findings, and we tackle the challenges from modeling large-scale, personalized, and complex sequential data.

Recurrent recommender networks. Recently, Recurrent Neural Networks (RNN) are introduced into recommender systems to capture temporal dynamics [Hidasi *et al.*, 2016; Wu *et al.*, 2017; Jing and Smola, 2017]. For instances, Recurrent Recommender Networks (RRN) split time into segments with a time granularity (e.g. two months), model temporal evolution of users and items via RNNs, and estimate the rating of an item given by a user at any time segment [Wu *et al.*, 2017]. Other than using the specific timestamps, another line of work considers short session-based data, and seeks to use RNNs to capture the sequential dynamics within a session [Hidasi *et al.*, 2016]. Our method is compact and efficient since it only considers item transitions depending on the last visited item which is the most significant factor affecting user's next action (especially on sparse dataset). However, it's promising to investigate long-term dependencies via RNNs for next item recommendation.

3 The Translation-based Model

Problem Formulation. We refer to the objects that users ($\mathcal{U}$) interact with in the system as items ($\mathcal{I}$), e.g. products, movies, or places. The *sequential*, or 'next-item,' prediction task we are tackling is formulated as follows. For each user $u \in \mathcal{U}$ we have a sequence of items $\mathcal{S}^u = (\mathcal{S}^u_1, \mathcal{S}^u_2, \cdots, \mathcal{S}^u_{|\mathcal{S}^u|})$ that u has interacted with. Given the sequence set from all users $\mathcal{S} = \{\mathcal{S}^{u_1}, \mathcal{S}^{u_2}, \cdots, \mathcal{S}^{u_{|\mathcal{U}|}}\}$, our objective is to predict the next item to be 'consumed' by each user and generate recommendation lists accordingly.

3.1 The Proposed Model

We aim to build a model that (1) naturally captures personalized sequential behavior, and (2) easily scales to large, real-world datasets. Methodologically, we learn a transition space $\Phi = \mathbb{R}^K$, where each item i is represented with a point/vector $\vec{\gamma}_i \in \Phi$. $\vec{\gamma}_i$ can be *latent*, or transformed from certain explicit features of the item i, e.g. the output of a neural network. In this paper we model $\vec{\gamma}_i$ as a latent vector.

To model *personalized* sequential behavior, we represent each user u with a *translation* vector $\vec{t}_u \in \Phi$ to capture u's inherent intent or 'long-term preferences' that influence her to make transitioning decisions.[1] In particular, if u transitions from item i to item j, then we want $\vec{\gamma}_i + \vec{t}_u \approx \vec{\gamma}_j$, which means $\vec{\gamma}_j$ should be a nearest neighbor of $\vec{\gamma}_i + \vec{t}_u$ in Φ according to some distance metric $d(x, y)$, e.g. $\mathcal{L}_2$ distance. Note that we are uncovering a metric space where (1) *neighborhood* captures the notion of similarity and (2) *translation* encapsulates various semantically complex transition relationships amongst items. In both cases, the inherent triangle inequality

[1] For sparse data, $\vec{t}_u$ could potentially be modeled as $\vec{t}_u = \vec{t} + \vec{T}_u$, where t is a global vector and $\vec{T}_u$ is regularized towards 0.

assumption plays an important role in helping the model to generalize, as it does in canonical metric learning scenarios.

Finally, the probability that a given user u transitions from the previous item i to the next item j is predicted by

$$\begin{aligned} &Pr(j \mid u, i) \propto \beta_j - d(\vec{\gamma}_i + \vec{t}_u, \vec{\gamma}_j), \\ &\text{subject to} \quad \vec{\gamma}_i \in \Psi \subseteq \Phi, \quad \text{for } i \in \mathcal{I}. \end{aligned} \tag{1}$$

Ψ is a subspace in Φ, e.g. a unit ball, a technique which has been shown to be helpful for mitigating 'curse of dimensionality' issues (e.g. [Bordes *et al.*, 2013; Wang *et al.*, 2014; Lin *et al.*, 2015]). In the above equation a single bias term β_j is added to capture overall item popularity.

Ranking Optimization. Given a user and the associated historical sequence, the ultimate goal of the task is to rank the ground-truth item j higher than all other items ($j' \in \mathcal{I} \setminus j$). Therefore it is a natural choice to optimize the pairwise ranking between j and j' by (e.g.) Sequential Bayesian Personalized Ranking (S-BPR) [Rendle *et al.*, 2010]. To this end, we optimize the total order $>_{u,i}$ given the user u and the previous item i in the sequence:

$$\widehat{\Theta} = \arg\max_{\Theta} \sum_{u \in \mathcal{U}} \sum_{j \in \mathcal{S}^u} \sum_{j' \notin \mathcal{S}^u} \ln \sigma(\widehat{p}_{u,i,j} - \widehat{p}_{u,i,j'}) - \Omega(\Theta),$$

where i is the item preceding j in $\mathcal{S}^u$, $\widehat{p}_{u,i,j}$ is a shorthand for the prediction in Eq. (1), Θ is the parameter set $\{\beta_{i \in \mathcal{I}}, \vec{\gamma}_{i \in \mathcal{I}}, \vec{t}_{u \in \mathcal{U}}\}$, and $\Omega(\Theta)$ is an $\mathcal{L}_2$ regularizer.

Learning the Parameters. Item embeddings $\vec{\gamma}_{i \in \mathcal{I}}$ and $\vec{t}_{u \in \mathcal{U}}$ are randomly initialized to be unit vectors. $\beta_{i \in \mathcal{I}}$ are initialized to be zero. The objective function (Eq. (3.1)) is maximized by stochastic gradient ascent: First, we uniformly sample a user u from $\mathcal{U}$. Then, a 'positive' item j and a 'negative' item j' are uniformly sampled from $\mathcal{S}^u \setminus \mathcal{S}^u_1$ and $\mathcal{I} \setminus \mathcal{S}^u$ respectively. Next, parameters are updated via stochastic gradient ascent. Finally, we re-normalize $\vec{\gamma}_i$, $\vec{\gamma}_j$, and $\vec{\gamma}_{j'}$ to be vectors in Ψ. The above steps are repeated until convergence or until the accuracy plateaus on the validation set.

3.2 Connections to Existing Models

Knowledge Graphs. Our method is inspired by recent advances in knowledge graph completion, e.g. [Bordes *et al.*, 2013; Wang *et al.*, 2014; Lin *et al.*, 2015; Yang *et al.*, 2015; Trouillon *et al.*, 2016], where the objective is to model multiple types of relations between pairs of entities. One state-of-the-art technique (see e.g. [Bordes *et al.*, 2013]) embeds entities as points and relations as *translation* vectors such that the relationship between two entities is captured by the corresponding translation operation. In recommendation settings, items are analogous to 'entities' in knowledge graphs. Our key idea is to represent each user as one particular type of 'relation' such that it captures the personalized reasons a user transitions from one item to another.

Sequential Models. State-of-the-art sequential prediction models are typically based on (personalized) Markov Chains. FPMC [Rendle *et al.*, 2010] is a seminal model whose predictor consists of two key components: (1) the inner product of user and item factors (capturing users' inherent preferences), and (2) the inner product of the factors of the previous and next item (capturing sequential dynamics). FPMC is essentially the combination of MF and factorized MC:

$$Pr(j \mid u, i) \propto \langle \vec{M}_u, \vec{N}_j \rangle + \langle \vec{P}_i, \vec{Q}_j \rangle, \tag{2}$$

where user embeddings $\vec{M}_u$ and item embeddings $\vec{N}_j$, $\vec{P}_i$, $\vec{Q}_j$ are parameters learned from the data.

Recently, Personalized Ranking Metric Embedding (PRME) [Feng *et al.*, 2015] was proposed to improve FPMC by learning two metric spaces: one for measuring user-item affinity and another for sequential continuity. Predictions are made according to

$$Pr(j \mid u, i) \propto -(\alpha \cdot \|\vec{M}_u - \vec{N}_j\|_2^2 + (1 - \alpha) \cdot \|\vec{P}_i - \vec{P}_j\|_2^2),$$

which replaces inner products in FPMC by distances. As argued in [Chen *et al.*, 2012; Feng *et al.*, 2015; Hsieh *et al.*, 2017], the underlying metricity assumption brings better generalization ability. However, like FPMC, PRME still has to learn two closely *correlated* components in a separate manner, using a hyperparameter α to balance them.

Another recent work, Hierarchical Representation Model (HRM) [Wang *et al.*, 2015], tries to extend FPMC by using an *aggregation* operation (max/average pooling) to blend users' preferences ($\vec{M}_u$) and their recent activities ($\vec{N}_i$):

$$Pr(j \mid u, i) \propto \langle \text{aggregation}(\vec{M}_u, \vec{N}_i), \vec{N}_j \rangle. \tag{3}$$

Although the predictor can be seen as modeling the third-order interactions with a single component, the aggregation is hard to interpret and does not reap the benefits of using metric embeddings as PRME does.

TransRec also falls into the category of Markov Chain models; however, it applies a novel *translation*-based structure in a metric space, which enjoys the benefits of using a single, interpretable component as well as a metric space.

4 Experiments

We include a wide range of publicly available datasets varying in domain, size, data sparsity, and variability/complexity. Data and code are available at `http://cseweb.ucsd.edu/~jmcauley/`.

Amazon. Reviews and timestamps on seven large product categories from *Amazon.com* [McAuley *et al.*, 2015]. This dataset spans May 1996 to July 2014 and is notable for its high sparsity and variability.

Epinions. This dataset was collected by [Zhao *et al.*, 2014] from *Epinions.com*, a popular online consumer review website. The reviews span January 2001 to November 2013.

Foursquare. A large number of check-ins of users at different venues from December 2011 to April 2012. This dataset was collected by [Levandoski *et al.*, 2012] and is widely used for evaluating next point-of-interest prediction methods.

Flixter. A large, dense movie rating dataset from *Flixter.com*. The timespan is from November 2005 to November 2009.

Google Local. A new dataset we introduce from *Google* which contains 11.4M reviews and ratings from 4.5M users on 3.1M local businesses.

For each of the above datasets, we discard users and items with fewer than 5 associated actions in the system. In cases

Table 1: Ranking results on different datasets (higher is better). The number of latent dimensions K for all comparison methods is set to 10. The best performance in each case is underlined. The last column shows the percentage improvement of *TransRec* over the best baseline.

Dataset	Metric	PopRec	BPR-MF	FMC	FPMC	HRM_{avg}	HRM_{max}	PRME	$\text{TransRec}_{\mathcal{L}_1}$	$\text{TransRec}_{\mathcal{L}_2}$	%Improv.
Epinions	*AUC*	0.4576	0.5523	0.5537	0.5517	0.6060	0.5617	0.6117	0.6063	<u>0.6133</u>	0.3%
	Hit@50	3.42%	3.70%	3.84%	2.93%	3.44%	2.79%	2.51%	3.18%	<u>4.63%</u>	20.6%
Google	*AUC*	0.5391	0.8188	0.7619	0.7740	0.8640	0.8102	0.8252	0.8359	<u>0.8691</u>	0.6%
	Hit@50	0.32%	4.27%	3.54%	3.99%	3.55%	4.59%	5.07%	6.37%	<u>6.84%</u>	34.9%
Amazon	*AUC*	0.6717	0.7320	0.7214	0.7302	0.7600	0.7436	0.7490	0.7659	<u>0.7772</u>	2.26%
	Hit@50	3.22%	4.51%	4.06%	4.13%	6.32%	4.93%	5.67%	7.16%	<u>7.23%</u>	14.4%
Foursquare	*AUC*	0.9168	0.9511	0.9463	0.9479	0.9559	0.9523	0.9565	0.9631	<u>0.9651</u>	0.9%
	Hit@50	55.60%	60.03%	63.00%	64.53%	60.75%	61.60%	65.32%	66.12%	<u>67.09%</u>	2.7%
Flixter	*AUC*	0.9459	0.9722	0.9568	0.9718	0.9695	0.9687	0.9728	0.9727	<u>0.9750</u>	0.2%
	Hit@50	11.92%	21.58%	22.23%	33.11%	32.34%	30.88%	<u>40.81%</u>	35.52%	35.02%	-13.0%

where star-ratings are available, we take all of them as users' positive feedback. Afterwards we end up with 1.11M users, 1.09M items, and 15.5M actions. The average number of actions per user/item in our data ranges from 3.23 to 310.61.

4.1 Comparison Methods

PopRec: This is a naïve baseline that ranks items according to their popularity.

Bayesian Personalized Ranking (BPR-MF) [Rendle *et al.*, 2009]: BPR-MF is a state-of-the-art item recommendation model which takes Matrix Factorization as the underlying predictor. It ignores sequential signals in the system.

Factorized Markov Chain (FMC): Captures the 'global' sequential dynamics by factorizing the item-to-item transition matrix (shared by all users), but does not capture personalized behavior.

Factorized Personalized Markov Chain (FPMC) [Rendle *et al.*, 2010]: Uses a predictor that combines Matrix Factorization and factorized Markov Chains so that personalized Markov behavior can be captured (see Eq. (2)).

Personalized Ranking Metric Embedding (PRME) [Feng *et al.*, 2015]: PRME models personalized Markov behavior by the summation of two Euclidean distances (see Eq. (3.2)).

Hierarchical Representation Model (HRM) [Wang *et al.*, 2015]: HRM extends FPMC by using aggregation operations to model more complex interactions (see Eq. (3)). We compare against HRM with both max pooling and average pooling, denoted by HRM_{max} and HRM_{avg} respectively.

Translation-based Recommendation (*TransRec*): Our method, which unifies user preferences and sequential dynamics with translations. In experiments we try both $\mathcal{L}_1$ and squared $\mathcal{L}_2$ distance for our predictor (see Eq. (1)).

4.2 Evaluation Methodology

For each dataset, we partition the sequence $\mathcal{S}^u$ for each user u into three parts: (1) the most recent one $\mathcal{S}^u_{|\mathcal{S}^u|}$ for test, (2) the second most recent one $\mathcal{S}^u_{|\mathcal{S}^u|-1}$ for validation, and (3) all the rest for training. Hyperparameters in all cases are tuned by grid search with the validation set. Finally, we report the performance of each method on the test set in terms of the following ranking metrics:

Area Under the ROC Curve (AUC):

$$AUC = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{1}{|\mathcal{I} \setminus \mathcal{S}^u|} \sum_{j' \in \mathcal{I} \setminus \mathcal{S}^u} \mathbf{1}(R_{u,g_u} < R_{u,j'}),$$

Hit Rate at position 50 (Hit@50):

$$Hit@50 = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \mathbf{1}(R_{u,g_u} \leq 50),$$

where g_u is the 'ground-truth' item associated with user u at the most recent time step, $R_{u,i}$ is the rank of item i for user u (smaller is better), and $\mathbf{1}(b)$ is an indicator function that returns 1 if the argument b is $true$; 0 otherwise. Intuitively, AUC counts the fraction of times that rank desired items higher than irelevant items, and reflect overall recommendation performance. Hit@50 measures Top-N ranking performance, which considers whether the 'ground-truth' item is ranked among the top-50 items.

4.3 Performance and Quantitative Analysis

Results are collated in Table 1 (datasets are ranked in ascending order of item density). Due to the sparsity of most of the datasets in consideration, the number of dimensions K of all latent vectors in all cases is set to 10 for simplicity. The main findings are summarized as follows:

BPR-MF and FMC achieve considerably better results than the popularity-based baseline in most cases. FPMC and HRM are essentially combinations of MF and FMC. FPMC beats BPR-MF and FMC mainly on relatively dense datasets like *Foursquare* and *Flixter*, and loses on sparse datasets—possibly due to the large number of parameters it introduces. HRM achieves strong results amongst all baselines in many cases, presumably from the aggregation operations.

PRME replaces the inner products in FPMC by distance functions. It beats FPMC in most cases, though sometimes loses to HRM due to different modeling strategies. Note that like FPMC, PRME turns out to be quite strong at handling dense datasets like *Foursquare* and *Flixter*. We speculate that the two models could benefit from the considerable amount of additional parameters they use when data is dense.

TransRec outperforms other methods in nearly all cases. The improvements seem to be correlated with: (1) **Variability:** *TransRec* achieves large improvements on *Google*, the

dataset with the largest vocabulary of items in our collection, including all kinds of restaurants, bars, shops (etc.) as well as a global user base. (2) **Sparsity:** *TransRec* beats all baselines especially on comparatively sparser datasets like *Epinions* and *Google*. The only exception is in terms of $Hit@50$ on *Flixter*, the *densest* dataset in consideration. We speculate that *TransRec* is at a disadvantage by using fewer parameters (than PRME) especially when K is set to a small number (10). In practice, we achieved comparable results with the strongest baseline when increasing the dimensionality of all models to 100.

5 Conclusion

We introduced a scalable translation-based method, *TransRec*, for modeling the semantically complex personalized sequential dynamics in recommender systems. We analyzed the connections between *TransRec* and existing methods, and demonstrated its suitability for modeling third-order interactions between users, their previously consumed items, and their next item. Superior results achieved on a spectrum of large, real-world datasets suggest that translation-based architectures are a promising avenue for recommendation problems.

References

[Bordes *et al.*, 2013] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *NIPS*, 2013.

[Chen *et al.*, 2012] Shuo Chen, Joshua L. Moore, Douglas Turnbull, and Thorsten Joachims. Playlist prediction via metric embedding. In *KDD*, 2012.

[Feng *et al.*, 2015] Shanshan Feng, Xutao Li, Yifeng Zeng, Gao Cong, Yeow Meng Chee, and Quan Yuan. Personalized ranking metric embedding for next new poi recommendation. In *IJCAI*, 2015.

[Hidasi *et al.*, 2016] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. In *ICLR*, 2016.

[Hsieh *et al.*, 2017] Cheng-Kang Hsieh, Longqi Yang, Yin Cui, Tsung-Yi Lin, Serge J. Belongie, and Deborah Estrin. Collaborative metric learning. In *WWW*, 2017.

[Hu *et al.*, 2008] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, 2008.

[Jing and Smola, 2017] How Jing and Alexander J Smola. Neural survival recommender. In *WSDM*. ACM, 2017.

[Koren *et al.*, 2009] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 2009.

[Levandoski *et al.*, 2012] Justin J. Levandoski, Mohamed Sarwat, Ahmed Eldawy, and Mohamed F. Mokbel. LARS: A location-aware recommender system. In *ICDE*, 2012.

[Lin *et al.*, 2015] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, 2015.

[McAuley *et al.*, 2015] Julian J. McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. Image-based recommendations on styles and substitutes. In *SIGIR*, 2015.

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

[Moore *et al.*, 2013] Joshua L. Moore, Shuo Chen, Douglas Turnbull, and Thorsten Joachims. Taste over time: the temporal dynamics of user preferences. In *ISMIR*, 2013.

[Nickel *et al.*, 2011] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *ICML*, 2011.

[Nickel *et al.*, 2012] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. Factorizing yago: scalable machine learning for linked data. In *WWW*, 2012.

[Ning and Karypis, 2011] Xia Ning and George Karypis. SLIM: Sparse linear methods for top-n recommender systems. In *ICDM*, 2011.

[Pan *et al.*, 2008] Rong Pan, Yunhong Zhou, Bin Cao, Nathan N. Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. One-class collaborative filtering. In *ICDM*, 2008.

[Rendle and Schmidt-Thieme, 2010] Steffen Rendle and Lars Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *WSDM*, 2010.

[Rendle *et al.*, 2009] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, 2009.

[Rendle *et al.*, 2010] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *WWW*, 2010.

[Ricci *et al.*, 2011] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul Kantor. *Recommender systems handbook*. Springer US, 2011.

[Serfozo, 2009] Richard Serfozo. *Basics of applied stochastic processes*. Springer Science & Business Media, 2009.

[Singh and Gordon, 2008] Ajit P. Singh and Geoffrey J. Gordon. Relational learning via collective matrix factorization. In *KDD*, 2008.

[Trouillon *et al.*, 2016] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *ICML*, 2016.

[Wang *et al.*, 2014] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *AAAI*, 2014.

[Wang *et al.*, 2015] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. Learning hierarchical representation model for nextbasket recommendation. In *SIGIR*, 2015.

[Wu *et al.*, 2013] Xiang Wu, Qi Liu, Enhong Chen, Liang He, Jingsong Lv, Can Cao, and Guoping Hu. Personalized next-song recommendation in online karaokes. In *RecSys*, 2013.

[Wu *et al.*, 2017] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J. Smola, and How Jing. Recurrent recommender networks. In *WSDM*, 2017.

[Yang *et al.*, 2015] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *ICLR*, 2015.

[Zhao *et al.*, 2014] Tong Zhao, Julian McAuley, and Irwin King. Leveraging social connections to improve personalized ranking for collaborative filtering. In *CIKM*, 2014.