COT: Contextual Operating Tensor for Context-aware Recommender Systems

Qiang Liu, Shu Wu, Liang Wang
Center for Research on Intelligent Perception and Computing, National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences, China
{qiang.liu, shu.wu, wangliang}@nlpr.ia.ac.cn

Abstract

With rapid growth of information on the internet, recommender systems become fundamental for helping users alleviate the problem of information overload. Since contextual information can be used as a significant factor in modeling user behavior, various context-aware recommendation methods are proposed. However, the state-of-the-art context modeling methods treat contexts as other dimensions similar to the dimensions of users and items, and cannot capture the special semantic operation of contexts. On the other hand, some works on multi-domain relation prediction can be used for the context-aware recommendation, but they have problems in generating recommendation under a large amount of contextual information. In this work, we propose Contextual Operating Tensor (COT) model, which represents the common semantic effects of contexts as a contextual operating tensor and represents a context as a latent vector. Then, to model the semantic operation of a context combination, we generate contextual operating matrix from the contextual operating tensor and latent vectors of contexts. Thus latent vectors of users and items can be operated by the contextual operating matrices. Experimental results show that the proposed COT model yields significant improvements over the competitive compared methods on three typical datasets, i.e., Food, Adom and Movielens-1M datasets.

Introduction

With rapid growth of available information on the internet, users are getting in trouble with the problem of information overload. Recommender systems have become important for helping user to select interesting information in many Web applications such as social networks, e-commerce, online reading, review websites and so on. Nowadays, with enhanced ability of systems in collecting information, a great amount of contextual information in recommender systems has been collected. The contextual information in real-world application includes location, time, weather, companion and so on. These kinds of contexts have significant effect on the user behavior. For instance, a man may like to watch cartoon with his children while may like to watch romantic movies with his wife. He may prefer to read novels during weekend while may tend to read professional books during weekdays.

Due to the fundamental effect of contextual information in recommender systems, many different kinds of context modeling methods have been developed. Some existing works (Karatzoglou et al. 2010; Rendle et al. 2011) incorporate the contextual information in a factorization model via treating the context as one or several dimensions which have similar properties as dimensions of users and items. However, treating contexts as same as users and items, these methods cannot model the semantic operation of contexts. Moreover, some works on multi-domain relation prediction (Zhang, Agarwal, and Chen 2011; Jamali and Lakshmanan 2013) can also be implemented for the context-aware recommendation. These methods incorporate transfer matrix to map latent vectors of entities from one domain to another domain. To deal with context-aware recommendation, using transfer matrix, latent vectors can be mapped from one context to another context. However, since there are multiple contexts in the real world (e.g., location, time, companion), using a transfer matrix for each specific context combination, these methods have problem in confronting with large amount of contextual information.

To overcome the shortages mentioned above, we propose a novel context modeling method, which uses contextual operating tensor to capture the operation of contexts. In the research of natural language processing, a noun has semantic information as a latent vector, and an adjective has semantic operation on nouns as an operating matrix (Baroni and Zamparelli 2010). For instance, in the phrase "excellent product", the noun "product" has a latent vector, while the adjective "excellent" has the semantic operating matrix which operate the vector of the noun "product". Thus, the phrase "excellent product" has a new latent vector which represents a positive attitude to the "product". In this work, we assume that context combinations have similar properties of adjectives and can operate the latent characteristics of entities. For instance, when a man is with children, the context of this companion operates his latent interests, and then he may like to watch cartoons. Here, for the context combination in a user-item rating behavior, we use contextual operating matrix to represent the semantic operation. Inspired by (Socher et al. 2013) in simplifying the matrix-vector operation, we use contextual operating tensor to capture common effects of contexts.

Besides, we represent each context combination as a latent vector, and then multiply it with the tensor to generate the operating matrix. The main contributions of this work are listed as follows:

- We model the contextual information as the semantic operation on entities, which presents a novel perspective on modeling of the contextual information.
- We use contextual operating tensor to capture the common semantic effects of contexts, and latent vectors to capture the specific properties of contexts. Then the contextual operating matrix can be generated from them.
- Experiments conducted on three real datasets show that COT is effective and evidently outperforms the state-of-the-art context-aware models.

Related Works

In this section, we review some related works on matrix factorization and state-of-the-art context-aware models.

Matrix Factorization

Matrix Factorization (MF) (Mnih and Salakhutdinov 2007; Koren, Bell, and Volinsky 2009; Koren and Bell 2011) has become the state-of-the-art approach to recommender systems. The basic objective of MF is to factorize a user-item rating matrix into two low rank matrices, each of which represents the latent factors of users or items. The original rating matrix can be approximated via the multiplying calculation.

MF has been extended nowadays. SVD++ (Koren 2008) combines the neighborhood model with matrix factorization model in one prediction function. Treating time information as a special context, timeSVD++ (Koren 2010) and tensor factorization (Xiong et al. 2010) are proposed, which achieve the state-of-the-art performance in the time-aware recommendation. Factorization machine (FM) (Rendle 2010) models all interactions between pairs of variables with the target by using factorized interaction parameters. FM is a flexible model for recommendation, and other state-of-the-art factorization models including SVD++ and timeSVD++ can be implemented using FM by defining the input data or features (Rendle 2012).

Context-aware Recommender Systems

Contextual information has been proved to be useful for recommender systems (Palmisano, Tuzhilin, and Gorgoglione 2008; Adomavicius and Tuzhilin 2011), and many context-aware recommendation methods have been developed. According to the survey of (Adomavicius and Tuzhilin 2011), these methods can be categorized into pre-filtering, post-filtering and context modeling. Employing pre-filtering or post-filtering strategies, conventional methods (Adomavicius et al. 2005; Baltrunas and Ricci 2009; Panniello et al. 2009) utilize contextual information to drive data selection or adjust the resulting set. These ad-hoc methods may work in practice, but they require the supervision and the fine-tuning in all steps of recommendation (Rendle et al. 2011).

The context modeling approaches, which use the contextual information directly in the model, have become popular. Recent works on context modeling have focused on integrating contextual information with user-item rating matrix and building models based on factorization models. Multiverse recommendation (Karatzoglou et al. 2010) represents the user-item rating matrix with contextual information as a user-item-context rating tensor, which is factorized with Tucker decomposition (Tucker 1966). And FM is easily applicable to a wide variety of context by specifying only the input data (Rendle et al. 2011). However, these methods treat contexts as one or several dimensions besides the dimensions of users and items, and cannot capture the special semantic operation of contexts.

Research on multi-domain relation prediction can also be used for the context-aware recommendation. Collective Matrix Factorization (CMF) (Singh and Gordon 2008) factorizes the rating matrix in each domain, and the latent vectors of some entities are shared among different domains. CMF can also be applied in social rating networks (Yang et al. 2011) and attribute-aware relation prediction (Lippert et al. 2008). The work of (Zhang, Agarwal, and Chen 2011) considers user attributes as priors for user latent vectors, and a transfer matrix is used to generate latent vectors from the original ones. Similarly, Heterogeneous Matrix Factorization (HeteroMF) (Jamali and Lakshmanan 2013) generates context-specific latent vectors of entities using a context-dependent transfer matrix and the original latent vectors of entities. However, for context-aware recommendation, with a transfer matrix for each context combination, these methods cannot be implemented for real applications with large amount of contextual information.

Proposed COT Model

In this section, we introduce our proposed contextual operating tensor model. We introduce the notations at first, then present the proposed model thoroughly, and finally describe the process of parameter inference.

Notations

In typical recommender systems, there are users and items denoted by $U = \{u_1, u_2, ...\}$ and $V = \{v_1, v_2, ...\}$. The latent vectors of user i and item j can be denoted by $u_i \in \mathbb{R}^d$ and $v_j \in \mathbb{R}^d$. There are multiple contexts $\mathcal{C}_1, ..., \mathcal{C}_n$, such as location, time, companion and so on. A context value c is a variable of the context $\mathcal{C}$, and a specific combination $\{c_{1,k}, ..., c_{n,k}\}$ is named context combination k. The latent vector of the context value $c_{m,k}$ is denoted as $h_{m,k} \in \mathbb{R}^{d_c}$, then the context combination k can be represented as a latent matrix $H_k = [h_{1,k}, ..., h_{n,k}] \in \mathbb{R}^{d_c \times n}$.

In this work, the rating that user i provides to item j under the context combination k can be written as $r_{i,j,k}$. Note that, in some cases, i and j can be the same entity. For example, in social network, both i and j are users.

Contextual Operating Matrix

In typical matrix factorization methods, latent vectors of users and items are constant with varying contexts. But in

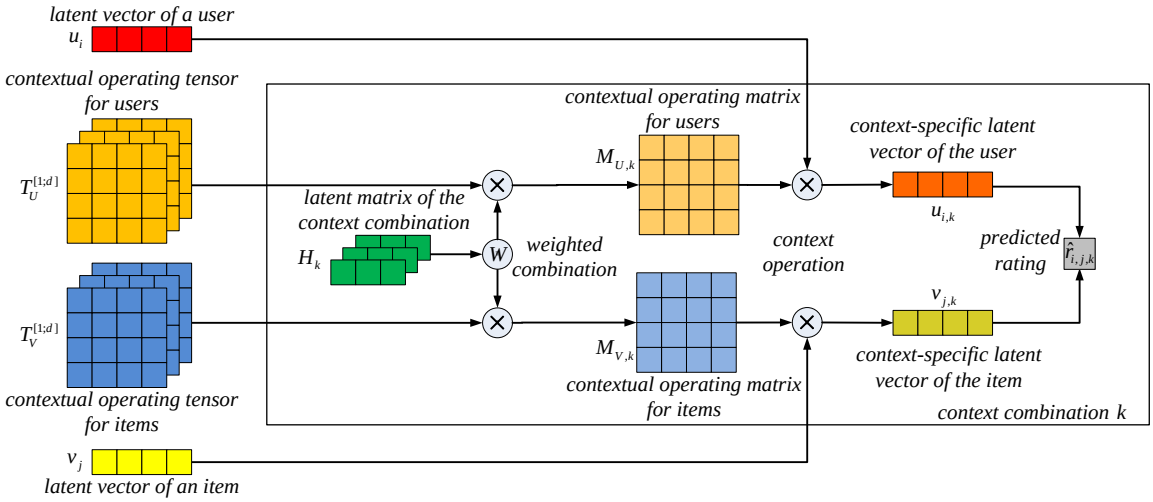


Figure 1: Overview of our proposed COT model. Contextual operating tensors and latent vectors of entities are shown on the left side, and the computational procedure under each context combination is illustrated in the square.

real-world applications, user interests and item properties are changed with different context combinations. Here, we use context-specific latent vectors for users and items under different context combinations. Then, the matrix factorization equation can be rewritten as:

$$\hat{r}_{i,j,k} = \omega_0 + \omega_i + \omega_j + \sum_{m=1}^{n} \omega_{m,k} + u_{i,k}^T v_{j,k} \ , \qquad (1)$$

where ω_0 denotes the global bias, ω_i and ω_j denote the biases of user i and item j, $\omega_{m,k}$ is the bias of the context value $c_{m,k}$, $u_{i,k}$ and $v_{j,k}$ are context-specific latent vectors of user i and item j under the context combination k.

Similar to a phrase of noun and adjective, where the noun has semantic information and the adjective has semantic operation on the noun, in recommender systems, entities have rich semantic information and contexts act like adjectives which have the operation on entities. For example, companion with children can make users would like to watch cartoons, and romantic films become popular during Valentine's Day. We use two contextual operating matrices for a specific context combination. These matrices describe how a context combination affects the properties of entities and how context-specific latent vectors of users and items can be generated from their original ones. Here, we can calculate context-specific latent vectors as:

$$u_{i,k} = M_{U,k} u_i \ , \qquad (2)$$

$$v_{j,k} = M_{V,k} v_j \ , \qquad (3)$$

where $M_{U,j}$ and $M_{V,j}$ are both $d \times d$ matrices, denoting the contextual operating matrices for users and items under the context combination k.

Contextual Operating Tensor

As discussed above, with two contextual operating matrices for each context combination, the number of parameters will grow rapidly as the number of context combinations grows. Besides, different contexts share similar common semantic effects, for example, both weekend and being at home can make you would like to read novels. Context operation can be represented as a unity of common semantic effects and specific properties of contexts. Therefore, it will be wonderful and plausible if we can generate contextual operating matrices from several basic matrices which represent some common semantic effects of contexts. In this way, we can not only reduce the number of parameters to be estimated, but also model the underlying characteristics of contexts. Here, incorporating contextual operating tensors, we have:

$$M_{U,k} = a_k^T T_U^{[1:d]} \ , \qquad (4)$$

$$M_{V,k} = a_k^T T_V^{[1:d]} \ , \qquad (5)$$

where $T_U^{[1:d]}$ and $T_V^{[1:d]}$ are both $d_c \times d \times d$ tensors, denoting the contextual operating tensors for users and items. Representing the context combination k, a_k is a d_c dimensional latent vector, which is a weighted combination of latent vectors of each context value:

$$a_k = H_k W \ , \qquad (6)$$

where W is a n dimensional vector, indicating the weights of each context, and each column of H_k denotes the latent vector of each context value under the context combination k. Then, substituting the Equation (4-6) into the context-specific latent vectors of users and items in Equation (2-3), we have:

$$u_{i,k} = (H_k W)^T T_U^{[1:d]} u_i \ , \qquad (7)$$

$$v_{j,k} = (H_k W)^T T_V^{[1:d]} v_j \ , \qquad (8)$$

Equation (7-8) can be calculated as:

$$u_{i,k} = \begin{bmatrix} (H_k W)^T T_{U,1}^{[1:d]} u_i \\ \cdots \\ (H_k W)^T T_{U,d}^{[1:d]} u_i \end{bmatrix} \ , \qquad (9)$$

$$v_{j,k} = \begin{bmatrix} (H_k W)^T T_{V,1}^{[1:d]} v_j \\ \cdots \\ (H_k W)^T T_{V,d}^{[1:d]} v_j \end{bmatrix} \ , \qquad (10)$$

where $T_{U,m}^{[1:d]}$ and $T_{V,m}^{[1:d]}$ denote the mth slices of the tensor $T_U^{[1:d]}$ and $T_V^{[1:d]}$, and each slice is a $d_c \times d$ matrix.

With such a computation, the contextual operating tensor can relate latent vectors of entities and latent vectors of contexts. Each slice of contextual operating tensor can capture a specific type of common semantic operation on users or items.

After discussing our model above, the overall rating prediction function for COT can be written as:

$$\hat{r}_{i,j,k} = \omega_0 + \omega_i + \omega_j + \sum_{m=1}^{n} \omega_{m,k} + \left[\underbrace{\overbrace{(H_k W)^T T_U^{[1:d]}}^{M_{U,k}} u_i}_{u_{i,k}} \right]^T \underbrace{\overbrace{(H_k W)^T T_V^{[1:d]}}^{M_{V,k}} v_j}_{v_{j,k}} . \quad (11)$$

As shown in Figure 1, under each context combination k, the contextual operating matrix $M_{U,k}$ or $M_{V,k}$ is a certain combination of all types of semantic operation described in contextual operating tensor $T_U^{[1:d]}$ or $T_U^{[1:d]}$. In addition, latent factor of context combination k is represented as a weighted combination of columns in the latent matrix H_k. Therefore, the context-specific latent vectors $u_{i,k}$ and $v_{j,k}$ are the latent vectors u_i and v_j under the context operation captured by $M_{U,k}$ and $M_{V,k}$.

Parameter Inference

We have already introduced our model mathematically in the previous section. Now, to accomplish the parameter inference, we need to minimize the following objective function:

$$\min_{U,V,H,T,W} J = \sum_{\langle i,j,k \rangle \in \Omega} (r_{i,j,k} - \hat{r}_{i,j,k})^2 + \frac{\lambda}{2}(||U||^2 + ||V||^2 + ||H||^2 + ||T||^2 + ||W||^2) , \quad (12)$$

where Ω denotes the train set, and λ is a parameter to control the regularizations, which can be determined using 5-fold cross validation.

The derivations of J with respect to all the parameters can be calculated as:

$$\frac{\partial J}{\partial u_i} = -2 \sum_{\langle i,j,k \rangle \in \Omega} l_{i,j,k} \left(\sum_{m=1}^{d} T_{U,m}^{[1:d]} v_{i,k,m} \right)^T H_k W + \lambda u_i ,$$

$$\frac{\partial J}{\partial v_j} = -2 \sum_{\langle i,j,k \rangle \in \Omega} l_{i,j,k} \left(\sum_{m=1}^{d} T_{V,m}^{[1:d]} u_{j,k,m} \right)^T H_k W + \lambda v_j ,$$

$$\frac{\partial J}{\partial T_{U,m}^{[1:d]}} = -2 \sum_{\langle i,j,k \rangle \in \Omega} l_{i,j,k} H_k W u_i^T v_{j,k,m} + \lambda T_{U,m}^{[1:d]} ,$$

$$\frac{\partial J}{\partial T_{V,m}^{[1:d]}} = -2 \sum_{\langle i,j,k \rangle \in \Omega} l_{i,j,k} H_k W v_j^T u_{i,k,m} + \lambda T_{V,m}^{[1:d]} ,$$

$$\frac{\partial J}{\partial H_k} = -2 \sum_{\langle i,j,k \rangle \in \Omega} l_{i,j,k} Q_{i,j,k} W^T + \lambda H_k ,$$

$$\frac{\partial J}{\partial W} = -2 \sum_{\langle i,j,k \rangle \in \Omega} l_{i,j,k} H_k^T Q_{i,j,k} + \lambda W ,$$

where $u_{i,k,m}$ and $v_{j,k,m}$ denote the mth components of latent vector $u_{i,k}$ and $v_{j,k}$, $l_{i,j,k}$ and $Q_{i,j,k}$ are two intermediate variables which are computed as follows:

$$l_{i,j,k} = (r_{i,j,k} - \hat{r}_{i,j,k}) ,$$

$$Q_{i,j,k} = \left(T_U^{[1:d]} u_i \right) v_{j,k} + \left(T_V^{[1:d]} v_j \right) u_{j,k} .$$

After calculating all the derivations, a solution of J in Equation (12) can be obtained by using stochastic gradient descent, which has been widely used in recommender systems (Koren, Bell, and Volinsky 2009; Koren and Bell 2011).

Experiment

In this section, we empirically investigate the performance of COT. First we describe the datasets and settings in our experiments, then report and analyze the experiment results.

Experiment Datasets

Our experiments are conducted on three real datasets.

- **Food dataset** (Ono et al. 2009) is collected from a restaurant. There are two contexts: *virtuality* describes if the situation in which the user rates is virtual or real, and *hunger* captures how hungry the user is.
- **Adom dataset** (Adomavicius et al. 2005) is collected on a movie website and has rich contextual information. There are five contexts: *companion* captures who the user watches the movie with, *when* shows whether the user watches the movie at weekend, *release* indicates whether the user watches the movie on first release, *rec* captures how will the user recommend the movie, and *where* indicates whether the user watches the movie in the theater.
- **Movielens-1M**[1] is collected from a movie recommender system Movielens[2]. There is no explicit contextual information, but the timestamp can be split into two contexts: *hour* and *day*.

Compared Methods

We compare COT model with four state-of-the-art models.

- **SVD++** (Koren 2008) is a context-unaware model which is used as a baseline in our experiments. We use LibFM[3] to implement the method.
- **Multiverse Recommendation** (Karatzoglou et al. 2010) is a state-of-the-art model and has been shown to outperform other context-aware recommendation models on the Food dataset and the Adom dataset.
- **FM** (Rendle et al. 2011) is applicable to the contextual information by specifying the input data. We also use LibFM as its implementation.

[1] http://grouplens.org/datasets/
[2] http://movielens.org/
[3] http://www.libfm.org/

Table 1: Performance comparison on three datasets and two kinds of splitting, measured by RMSE and MAE ($d = 8$, $d_c = 4$).

	Food Dataset				Adom Dataset				Movielens-1M			
	All Users		Cold Start		All Users		Cold Start		All Users		Cold Start	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
SVD++	1.155	0.948	1.278	1.086	2.782	2.093	3.421	2.436	0.942	0.721	1.248	0.956
Multiverse	1.063	0.841	1.121	0.921	1.833	1.383	2.168	1.556	0.883	0.669	1.025	0.771
FM	1.055	0.845	1.115	0.918	1.852	1.446	2.125	1.563	0.878	0.672	1.001	0.766
HeteroMF	1.072	0.862	1.136	0.932	2.084	1.552	2.384	1.782	0.902	0.686	1.072	0.792
COT	**1.002**	**0.792**	**1.098**	**0.898**	**1.726**	**1.367**	**2.056**	**1.518**	**0.841**	**0.645**	**0.987**	**0.759**

- **HeteroMF** (Jamali and Lakshmanan 2013) uses transfer matrix to model the contextual information. Each specific context combination has a transfer matrix in HeteroMF.

Evaluation Metrics

To measure the performance of rating prediction, we use the most popular metrics, Root Mean Square Error (RMSE) and Mean Average Precision (MAE):

$$RMSE = \sqrt{\frac{\sum_{(i,j,k)\in\Omega_{test}} \left(r_{i,j,k} - \hat{r}_{i,j,k}\right)^2}{n_{test}}}, \qquad (13)$$

$$MAE = \frac{\sum_{(i,j,k)\in\Omega_{test}} |r_{i,j,k} - \hat{r}_{i,j,k}|}{n_{test}}, \qquad (14)$$

where Ω_{test} denotes the test set, and n_{test} denotes the number of ratings in test set. The smaller the value, the better the performance.

Experiment Methodology

To examine the performance on all users and cold start users, we adopt two different ways to split the datasets.

- **All Users**: We randomly sample about 10% of the ratings from the original dataset to create the test set, and the remaining 90% ratings are treated as the train set.
- **Cold Start**: We randomly sample some users from the original dataset then select less than three of their ratings as the train set, and leave all remaining ratings as the test set. The numbers of ratings of each user in test set are randomly decided. Also, the test set covers about 10% of the original dataset, and the train set covers about 90%.

Performance Comparison

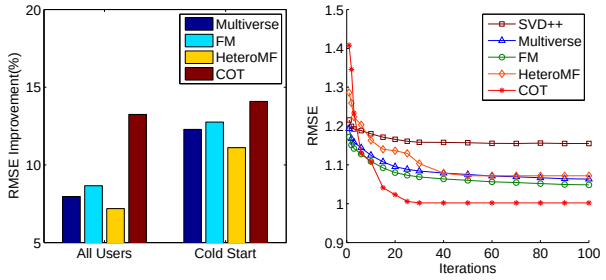
Table 1 illustrates all the experiment results measured by RMSE and MAE on three datasets and two kinds of splitting. This table shows that COT achieves the best results consistently, which demonstrates the effectiveness of using contextual operating tensor to model the contextual information. On all users splitting, comparing with the best performance of other models, COT improves the RMSE values by 5.2%, 6.0% and 4.2% on the Food, Adom and Movelen-1M datasets respectively. On cold start splitting, the improvements become 1.5%, 3.2% and 1.0%. COT improves greatly on the Adom dataset, which proves COT to be particularly helpful for the dataset with rich contextual information. Through all the experiments, context-aware models outperform the context-unaware model SVD++, which demonstrates the importance of the utilization of contextual information in recommender systems. Multiverse Recommendation and FM achieve close performance on all the datasets. On Food and Movielens-1M datasets, HeteroMF performs close to Multiverse Recommendation and FM, but fails on the Adom dataset. This may be because HeteroMF needs to estimate too many transfer matrices with rich contextual information in the Adom dataset.

Moreover, we compute and illustrate the RMSE improvements of the context-aware models comparing with SVD++ in the left part of Figure 2. We can observe that, on all three datasets, the RMSE improvements are larger on cold start splitting than those on all users splitting. This shows that the contextual information is more important in the cold start situation and can be used to compensate for the lack of history information.

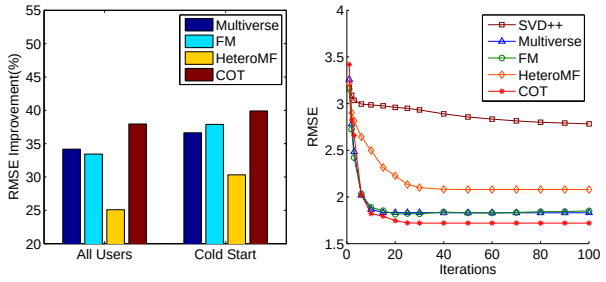
The convergence curves are illustrated in the right part of Figure 2. The convergence curves show that the RMSE of COT becomes stable after about 30 iterations. These indicate that COT has a satisfactory convergence rate and can be trained rapidly and effectively in practical applications.

Analysis of Contexts

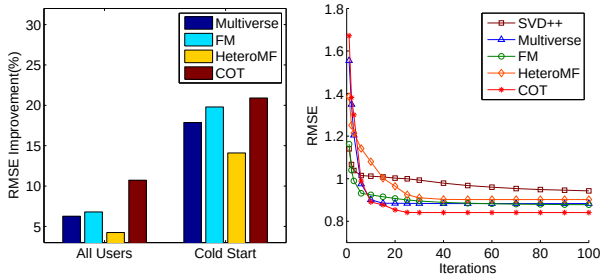
As we discussed in the section of the COT model, each slice of contextual operating tensor represents one kind of common operation. With larger difference among all the slices, the contextual operating tensor is more powerful in modeling the context operation. Similar to the content diversity measuring the difference among content of movies (Nguyen et al. 2014), we use a metric *matrix diversity* measuring the difference among all slices. Matrix diversity is defined as average RMSE of all slice pairs of tensor. Figure 3 illustrates how matrix diversity of the contextual operating tensors changes with increasing number of iterations. The figure shows that values of matrix diversity increase when the number of iterations grows from 1 to about 30. After the iteration 30, the values of matrix diversity become stable. Besides, we can see that COT achieves convergence in Figure 2 at the same time as the values of matrix diversity converged in Figure 3 on three datasets. These evidences



(a) Food Dataset



(b) Adom Dataset



(c) Movielens-1M

Figure 2: Performance comparison on three datasets. The left part illustrates the RMSE improvement of the context-aware models compared to SVD++. The right part illustrates the convergence curves of the models.

indicate that in the training process, when matrix diversity achieves stable results, COT achieves the best performance. Moreover, the final value of matrix diversity on the Adom dataset is larger than those on the other two datasets. This may be because richer contextual information on the Adom dataset has more powerful operating ability in changing the properties of users and items.

The weights of different contexts captured by W on three datasets are illustrated in Figure 4. The figure shows that the context *hunger* is more important than *virtuality* on the Food dataset, and *day* is more important than *hour* on Movielens-1M. These observations follow our intuition. For the Adom dataset, the context *rec* has much higher weight and becomes the dominant context. This may be because the context *rec*, which indicates how will the user recommend the movie, has high relevance with the final rating.

Impact of Parameters

As shown in Figure 5, we study the model performance with varying dimensionalities of entity vector d and context vector d_c on the Food dataset. With increasing d and d_c, the value of RMSE decreases at first, then stays nearly stable after $d = 5$ and $d_c = 3$. The parameter d_c can be selected in a large range, which means that the performance of COT doesn't rely on parameter selection very much. Since the performances of COT with different parameter values selected from these ranges are very similar, in previous subsection, we only illustrate the results with $d = 8$ and $d_c = 4$ on three datasets for simplicity.

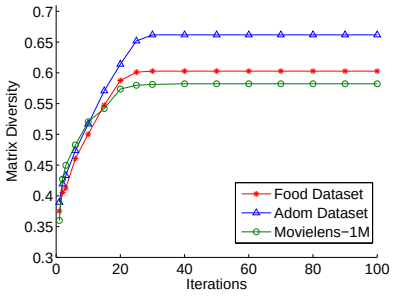


Figure 3: Matrix diversity of tensor T_U and T_V with increasing number of iterations on three datasets.

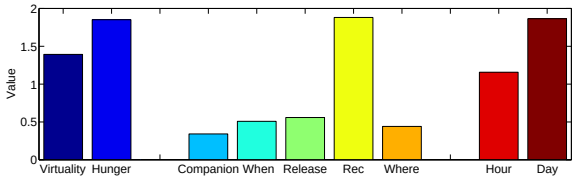


Figure 4: Weights of different contexts on three datasets.

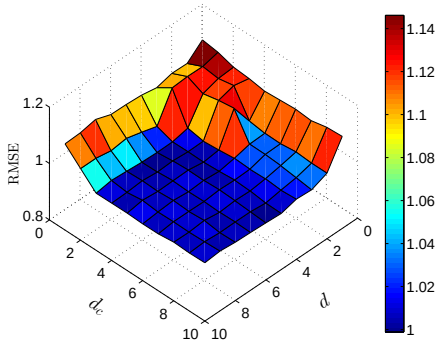


Figure 5: Performance of COT on the Food dataset with varying dimensionalities d and d_c.

Conclusion

In this paper, a novel context-aware recommendation method, i.e. COT, is proposed. In COT, the context is converged to be a vector and the common semantic effects of contexts are captured by a contextual operating tensor. Besides, the semantic operation of a context combination on entities, i.e., users and items, are captured by a contextual operating matrix, which can be generated from the contextual operating tensor and latent vectors of contexts. The experimental results on three real datasets show that COT outperforms the state-of-the-art context-aware models and can well reveal the relationship between contexts and entities.

Acknowledgments

This work is jointly supported by National Basic Research Program of China (2012CB316300), National Natural Science Foundation of China (61403390, 61175003, 61135002) and Hundred Talents Program of CAS.

References

Adomavicius, G., and Tuzhilin, A. 2011. Context-aware recommender systems. In *Recommender systems handbook*. Springer. 217–253.

Adomavicius, G.; Sankaranarayanan, R.; Sen, S.; and Tuzhilin, A. 2005. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems (TOIS)* 23(1):103–145.

Baltrunas, L., and Ricci, F. 2009. Context-based splitting of item ratings in collaborative filtering. In *Proceedings of the third ACM conference on Recommender systems*, 245–248. ACM.

Baroni, M., and Zamparelli, R. 2010. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, 1183–1193. Association for Computational Linguistics.

Jamali, M., and Lakshmanan, L. 2013. Heteromf: recommendation in heterogeneous information networks using context dependent factor models. In *Proceedings of the 22nd international conference on World Wide Web*, 643–654. International World Wide Web Conferences Steering Committee.

Karatzoglou, A.; Amatriain, X.; Baltrunas, L.; and Oliver, N. 2010. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, 79–86. ACM.

Koren, Y., and Bell, R. 2011. Advances in collaborative filtering. In *Recommender Systems Handbook*. Springer. 145–186.

Koren, Y.; Bell, R.; and Volinsky, C. 2009. Matrix factorization techniques for recommender systems. *Computer* 42(8):30–37.

Koren, Y. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 426–434. ACM.

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

Lippert, C.; Weber, S. H.; Huang, Y.; Tresp, V.; Schubert, M.; and Kriegel, H.-P. 2008. Relation prediction in multi-relational domains using matrix factorization. In *Proceedings of the NIPS 2008 Workshop: Structured Input-Structured Output, Vancouver, Canada*.

Mnih, A., and Salakhutdinov, R. 2007. Probabilistic matrix factorization. In *Advances in neural information processing systems*, 1257–1264.

Nguyen, T. T.; Hui, P.-M.; Harper, F. M.; Terveen, L.; and Konstan, J. A. 2014. Exploring the filter bubble: the effect of using recommender systems on content diversity. In *Proceedings of the 23rd international conference on World wide web*, 677–686. International World Wide Web Conferences Steering Committee.

Ono, C.; Takishima, Y.; Motomura, Y.; and Asoh, H. 2009. Context-aware preference model based on a study of difference between real and supposed situation data. In *User Modeling, Adaptation, and Personalization*. Springer. 102–113.

Palmisano, C.; Tuzhilin, A.; and Gorgoglione, M. 2008. Using context to improve predictive modeling of customers in personalization applications. *Knowledge and Data Engineering, IEEE Transactions on* 20(11):1535–1549.

Panniello, U.; Tuzhilin, A.; Gorgoglione, M.; Palmisano, C.; and Pedone, A. 2009. Experimental comparison of pre- vs. post-filtering approaches in context-aware recommender systems. In *Proceedings of the third ACM conference on Recommender systems*, 265–268. ACM.

Rendle, S.; Gantner, Z.; Freudenthaler, C.; and Schmidt-Thieme, L. 2011. Fast context-aware recommendations with factorization machines. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, 635–644. ACM.

Rendle, S. 2010. Factorization machines. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, 995–1000. IEEE.

Rendle, S. 2012. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)* 3(3):57.

Singh, A. P., and Gordon, G. J. 2008. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 650–658. ACM.

Socher, R.; Perelygin, A.; Wu, J. Y.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1631–1642. Association for Computational Linguistics.

Tucker, L. R. 1966. Some mathematical notes on three-mode factor analysis. *Psychometrika* 31(3):279–311.

Xiong, L.; Chen, X.; Huang, T.-K.; Schneider, J. G.; and Carbonell, J. G. 2010. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *SDM*, volume 10, 211–222. SIAM.

Yang, S.-H.; Long, B.; Smola, A.; Sadagopan, N.; Zheng, Z.; and Zha, H. 2011. Like like alike: joint friendship and interest propagation in social networks. In *Proceedings of the 20th international conference on World wide web*, 537–546. ACM.

Zhang, L.; Agarwal, D.; and Chen, B.-C. 2011. Generalizing matrix factorization through flexible regression priors. In *Proceedings of the fifth ACM conference on Recommender systems*, 13–20. ACM.