

# Learning Context-aware Latent Representations for Context-aware Collaborative Filtering

Xin Liu  
Institute for Infocomm Research  
Singapore  
liu-x@i2r.a-star.edu.sg

Wei Wu  
Institute for Infocomm Research  
Singapore  
wwu@i2r.a-star.edu.sg

## ABSTRACT

In this paper, we propose a generic framework to learn context-aware latent representations for context-aware collaborative filtering. Contextual contents are combined via a function to produce the context influence factor, which is then combined with each latent factor to derive latent representations. We instantiate the generic framework using biased Matrix Factorization as the base model. A Stochastic Gradient Descent (SGD) based optimization procedure is developed to fit the model by jointly learning the weight of each context and latent factors. Experiments conducted over three real-world datasets demonstrate that our model significantly outperforms not only the base model but also the representative context-aware recommendation models.

## Categories and Subject Descriptors

H.3.3 [Information Search and Retrieval]: Information filtering; H.4 [Information Systems Applications]: Miscellaneous

## General Terms

Algorithms, Performance

## Keywords

Context-aware; collaborative filtering; recommendation; latent factor models

## 1. INTRODUCTION

Rich contextual information that is pervasively available in many online applications provides an important information source to accurately model users' preference for collaborative filtering (CF) tasks such as rating prediction and top-N recommendation [1, 11, 5, 8, 3, 2]. For instance, in Point-of-Interest (POI) recommendation, the information such as time, weather and emotion may influence a user's preference on a certain POI, and hence influences the recommendations generated for this user.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [Permissions@acm.org](mailto:Permissions@acm.org).  
*SIGIR'15*, August 09 - 13, 2015, Santiago, Chile.  
© 2015 ACM. ISBN 978-1-4503-3621-5/15/08 ...\$15.00.  
DOI: <http://dx.doi.org/10.1145/2766462.2767775>.

Conventional context-aware CF models either separately process contexts, e.g., using contexts to partition input data (pre-processing) or adjust the order of recommended item list (post-processing) [1, 8] or jointly learn the context influence and other parameters during model building [10, 14, 13]. In this work, we follow the joint learning approach, which provides strong mathematical foundation for context modeling. Recent such models are designed based on latent factor models such as Matrix Factorization (MF). However, most of these approaches assume users, items and contexts share the same latent space (i.e., the relations among the three entities are captured by the inner product of their latent factors), which may not always make sense in complex real-world applications. Moreover, treating users, items and contexts as the same level entities may over-estimate the influence of contextual information, thus weakening the impacts of other latent aspects.

In order to address these issues, we propose a generic context-aware framework for latent factor models. The contextual information is firstly processed to generate user-related and item-related context influence factors. Such influence factors are then directly combined with users' and items' latent factors to derive the context-aware latent representations, which can be used to calculate users' context-aware preference on items. Such an approach not only handles user-related and item-related contexts for users' and items' latent factors respectively, but also properly integrates context influence into latent factors to balance the impacts of contexts.

We instantiate the proposed framework using biased MF model, a popular MF based model for rating prediction. We develop a Stochastic Gradient Descent (SGD) based optimization procedure to learn the context-aware latent representations by jointly estimating context related parameters and users' and items' latent factors. Experiments using three real-world datasets demonstrate that our context-aware biased MF evidently outperforms the conventional biased MF, as well as a representative context-aware model called Factorization Machines (FM) [11].

## 2. RELATED WORK

Recent research focuses on integrating contexts into latent factor models. By modeling feedback information as a user-item-context tensor, Karatzoglou et al. [5] proposed a multiverse recommendation model, where Tucker decomposition is applied to factorize the tensor. However, the type of contexts is restricted to be categorical. Similar ideas are presented in [15], which was particularly designed for top-N

recommendation by directly optimizing Mean Average Precision (MAP). Another line of research focuses on Factorization Machines [11, 13], which can handle diverse types of contexts for context-aware CF. The basic idea is to assign same dimension latent vectors to users, items and contexts and model all interactions between pairs of the three entities with respect to the target. An improvement that selects useful contexts for FM using gradient booting was introduced in [4]. In [10], the authors improve FM by breaking the limits of linearly combining latent factors. A non-linear probabilistic algorithm for context-aware recommendation was proposed using Gaussian processes. As mentioned in introduction section, such approaches suffer from the issue that restricts users, items and contexts to share the same latent space, which may not capture the real relations among the three entities.

There are also some models proposed to handle specific types of context such as social information [9] or time [7]. However, these methods cannot be directly applied to process general contextual information.

### 3. A GENERIC CONTEXT-AWARE FRAMEWORK

In this section, we present a generic framework for context-aware collaborative filtering. We denote user set by  $\mathcal{U} = \{u_1, u_2, \dots\}$ , where the real-valued latent factor vector (with the dimensionality of  $D$ ) of user  $u$  is represented by  $\mathbf{u}$ , and item set by  $\mathcal{V} = \{v_1, v_2, \dots\}$ , where the real-valued latent factor vector (with the dimensionality of  $D$ ) of item  $v$  is represented by  $\mathbf{v}$ . The outcome of the interaction between user  $u$  and item  $v$  is denoted by  $y_{u,v}$ , which can be explicit (e.g., 5-point scale rating) or implicit (e.g., clicked an advertisement). Tensor  $Y \in \mathcal{R}^{|\mathcal{U}| \times |\mathcal{V}| \times K}$  records all observed interactions between users and items under the corresponding  $K$  types of contexts.

The research problem of context-aware collaborative filtering is formulated as *given observed users' feedback to items and the associated contextual information, model the interaction between feedback data and context data to predict users' feedback to unobserved items under certain context situations*. Mathematically, we want to find context-aware latent representations  $\varphi$  for user  $u$  and item  $v$  to derive  $u$ 's feedback to  $v$ :

$$y_{u,v} = f(\varphi(\mathbf{u}, \theta_u), \varphi(\mathbf{v}, \theta_v)), \quad (1)$$

where  $\theta_x$  ( $x = u$  or  $v$ ) denotes the context influence factor derived for the user or the item.  $\varphi(\cdot, \cdot)$  is a function that combines the user's or item's latent factor and the corresponding context influence factor for latent representations.  $f(\cdot, \cdot)$  is a function that formulates users' and items' latent representations for predicting users' feedback score to items<sup>1</sup>.

The framework is generic from two aspects: (1) any form of function (linear or non-linear)  $\varphi(\cdot, \cdot)$  can be applied to combine latent factor and context influence factor to derive context-aware latent representations, and (2) the idea can be applied to any latent factor model such as basic MF, SVD++ [6], even social recommendation models like SoReg

<sup>1</sup>For most latent factor models like MF, such a function calculates the inner product of the latent factors of the corresponding user and item

	Gender		Age	Occupation			Mood		
$C_u$	male	female	age	doctor	engineer	student	happy	Sad	.....
	0	1	25	0	0	1	1	0	.....
	Genres						Site		
$C_v$	Comedy	Crime	Drama	Fantasy	Mystery	Action	Cinema	home	.....
	1	0	0	0	0	1	1	0	.....

Figure 1: An example of user-related and item-related contextual information in the context of movie recommendation. Dashed rectangle indicates continuous-valued context and regular rectangle indicates categorical-valued context (binary variable where 1 indicates “belong to” and 0 otherwise).

[9]. In the next section, we instantiate this generic framework using biased MF to demonstrate how it works.

## 4. CONTEXT-AWARE BIASED MATRIX FACTORIZATION

In order to better model the influence of contexts on users' and items' latent factors, we divide all contexts into user-related contexts that may influence users' behavior (e.g., in movie recommendation, users' age, gender, time of watching, mood, etc.) and item-related contexts that may characterize items (e.g., in movie recommendation, movies' genres, showed in cinema or on TV, etc.). For an interaction between user  $u$  and item  $v$ , the associated user-related contextual information is denoted by  $C_a$  which consists of  $K_a$  types of contexts<sup>2</sup>. Similarly, the associated item-related contextual information, which consists of  $K_b$  types of contexts, is denoted by  $C_b$ . To support both continuous valued contexts and categorical valued contexts, following FM [11],  $C_a/C_b$  is represented as a real-valued variables vector (see Figure 1 as an example).

We use a sigmoid function to linearly combine contexts to derive the context influence factor:

$$\theta_{x,d} = \frac{1}{1 + e^{-(\eta_{x,d} + \sum_{i=1}^{|C_x|} w_{i,d} \gamma_i)}}, \quad (2)$$

where  $x = a$  or  $b$ , indicating the context influence factor is derived for users or items, and  $d$  denotes the index of the latent factor in the vector.  $\gamma_i$  denotes the  $i$ th context value and  $w_i$  is the corresponding weight,  $\eta$  is the bias to adjust the context influence to the corresponding latent factor. Intuitively, each latent factor of each user or item expects a unique influence factor (i.e., the unique context influence parameters including the bias and context weights for contexts combination) for context-aware latent representation. However, this significantly increases model complexity in terms of parameter update computation and storage. To make the model applicable in practice, we propose to maintain  $D$  sets of context influence parameters for the  $D$  latent factors of *all* users and *all* items respectively, that is, we introduce totally  $2 \times D \times (|C_a| + |C_b| + 1 + 1)$  context influence parameters. We argue that the introduced complexity is minor in that it won't increase with the increasing number of observations but only rely on latent vector dimensionality and the size of the context variable vector, which are typically small (compared to the number of latent factors) in most application scenario.

<sup>2</sup>Note that user-related contexts and item-related contexts may overlap.

We can then derive the latent representations that combine<sup>3</sup> latent factors and context influence factors for user  $u$  and item  $v$ :

$$\varphi_{u,d} = \mathbf{u}_d + \theta_{u,d}, \quad (3)$$

$$\varphi_{v,d} = \mathbf{v}_d + \theta_{v,d}, \quad (4)$$

where  $d$  indicates the  $d$ th latent factor of  $u$ 's or  $v$ 's latent vector. After obtaining latent representations, following the design of biased MF, user  $u$ 's feedback to item  $v$  can be predicted as:

$$\hat{y}_{u,v} = \mu + b_u + b_v + \sum_{d=1}^D \varphi_{u,d} \varphi_{v,d}, \quad (5)$$

where  $\mu$  is the global mean,  $b_u$  and  $b_v$  are user bias and item bias respectively. The loss function is formulated as:

$$L = \frac{1}{2} \sum_{(u,v) \in \Omega} (y_{u,v} - \hat{y}_{u,v})^2 + \frac{\lambda_1}{2} \left( \sum_u \|\mathbf{u}\|^2 + \sum_v \|\mathbf{v}\|^2 + \sum_u b_u^2 + \sum_v b_v^2 \right) + \frac{\lambda_2}{2} \sum_{d=1}^D (\eta_{a,d}^2 + \sum_{i=1}^{|C_a|} w_{i,d}^2 + \eta_{b,d}^2 + \sum_{j=1}^{|C_b|} w_{j,d}^2), \quad (6)$$

where  $\Omega$  indicates the training set. The second and third terms are used to avoid overfitting, where  $\lambda_1$  and  $\lambda_2$  are regularization terms for latent factors (and biases), and context influence parameters respectively,  $d$  indicates the  $d$ th latent factor of latent vector.

To fit the proposed context-aware biased MF, we develop a SGD based optimization procedure which iteratively updates all parameters until the loss converges or pre-defined iterations have reached:

$$\Theta_f \leftarrow \Theta_f - \alpha_f \frac{\partial L}{\partial \Theta_f}, \quad (7)$$

$$\Theta_c \leftarrow \Theta_c - \alpha_c \frac{\partial L}{\partial \Theta_c}, \quad (8)$$

where  $\Theta_f$  indicates latent factors, users' and items' biases, and  $\Theta_c$  indicates context influence parameters (i.e., context weights and biases).  $\alpha_f$  and  $\alpha_c$  are the corresponding learning rates. The gradient of  $L$  with respect to each user's or item's latent factor as well as the bias is computed as follows:

$$\frac{\partial L}{\partial \mathbf{u}_d} = (\hat{y}_{u,v} - y_{u,v})(\mathbf{v}_d + \theta_{v,d}) + \lambda_1 \mathbf{u}_d. \quad (9)$$

$$\frac{\partial L}{\partial \mathbf{v}_d} = (\hat{y}_{u,v} - y_{u,v})(\mathbf{u}_d + \theta_{u,d}) + \lambda_1 \mathbf{v}_d. \quad (10)$$

$$\frac{\partial L}{\partial b_u} = (\hat{y}_{u,v} - y_{u,v}) + \lambda_1 b_u. \quad (11)$$

$$\frac{\partial L}{\partial b_v} = (\hat{y}_{u,v} - y_{u,v}) + \lambda_1 b_v. \quad (12)$$

The gradient of  $L$  with respect to each context influence parameter (i.e., the weight of each context and context bias for users and items) is computed as follows:

<sup>3</sup>We also tried other combination methods such as multiplication, but experimental results show that addition generates better performance.

$$\frac{\partial L}{\partial w_{i,d}} = (\hat{y}_{u,v} - y_{u,v})(\mathbf{v}_d + \theta_{v,d}) \theta_{u,d} (1 - \theta_{u,d}) \gamma_i + \lambda_2 w_{i,d}, \quad (13)$$

where  $i$  indicates the  $i$ th context value in the user-related context vector  $C_a$ .

$$\frac{\partial L}{\partial w_{j,d}} = (\hat{y}_{u,v} - y_{u,v})(\mathbf{u}_d + \theta_{u,d}) \theta_{v,d} (1 - \theta_{v,d}) \gamma_j + \lambda_2 w_{j,d}, \quad (14)$$

where  $j$  indicates the  $j$ th context value in the item-related context vector  $C_b$ .

$$\frac{\partial L}{\partial \eta_a} = (\hat{y}_{u,v} - y_{u,v})(\mathbf{v}_d + \theta_{v,d}) \theta_{u,d} (1 - \theta_{u,d}) + \lambda_2 \eta_a. \quad (15)$$

$$\frac{\partial L}{\partial \eta_b} = (\hat{y}_{u,v} - y_{u,v})(\mathbf{u}_d + \theta_{u,d}) \theta_{v,d} (1 - \theta_{v,d}) + \lambda_2 \eta_b. \quad (16)$$

Once all parameters have been learned, user  $u$ 's feedback to item  $v$  can be predicted by using Eq. 5.

## 5. EVALUATION

### 5.1 Experimental setup

We evaluate the performance of the proposed model using three real-world datasets: (1) Movielens-100K<sup>4</sup>, which consists of 100,000 ratings (ranging from 1 to 5) from 943 users on 1682 movies. The user-related contexts include age, gender, occupation; the movie-related contexts include genres. (2) Douban<sup>5</sup> book data [16], which records 1,097,148 ratings from 33,523 users on 381,767 books. The user-related contexts include the number of friends, the number of "wish"<sup>6</sup> issued and the number of ratings provided; the book-related contexts include the number of "wish" received and the number of ratings got. (3) Douban music data [16], which records 1,387,216 ratings from 29,287 users on 257,288 music items. The user-related and item-related contexts are the same with those used in Douban book data.

We compare the proposed context-aware biased MF with conventional biased MF and a representative context-aware model FM. Note that FM and our model share the same context set. The latent factors are initialized following uniform distribution [0, 1E-4]. The parameters of latent factor models such as latent vector dimensionality, regularization term, learning rate, etc. are determined by cross-validation.

We randomly divide each dataset into training set that consists of 80% data and test set that contains 20% data. We measure the performance using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), which are conventional metrics for measuring rating prediction accuracy. Each experiment is conducted 10 times, and we report the averaged results. Standard errors are shown (in Table 1) to indicate the confidence of the results.

### 5.2 Experimental results

Figure 2 shows when Movielens-100K data is used, how the performance of our model varies with different latent

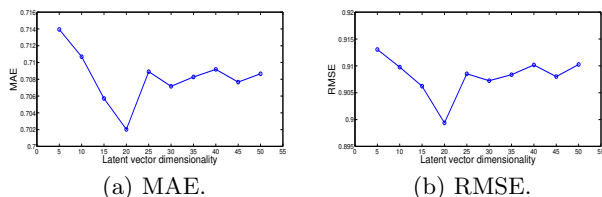
<sup>4</sup><http://grouplens.org/datasets/movielens/>

<sup>5</sup>Douban (<http://www.douban.com/>) is one of the largest Chinese based social platforms for sharing reviews and recommendations for books, movies and music. A social network is also provided to connect users.

<sup>6</sup>A user, although has not read a book, may still express her interest by indicating "wish"

**Table 1: Performance comparison**

	MovieLens-100K		Douban book		Douban music	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
Our model	0.7020 $\pm$ 0.0012	0.8993 $\pm$ 0.0015	0.5782 $\pm$ 0.0010	0.7378 $\pm$ 0.0013	0.5163 $\pm$ 0.0011	0.6588 $\pm$ 0.0016
Biased MF	0.7245 $\pm$ 0.0015	0.9185 $\pm$ 0.0014	0.6045 $\pm$ 0.0013	0.7733 $\pm$ 0.0016	0.5401 $\pm$ 0.0014	0.6911 $\pm$ 0.0016
FM	0.7160 $\pm$ 0.0010	0.9112 $\pm$ 0.0013	0.5886 $\pm$ 0.0012	0.7584 $\pm$ 0.0014	0.5342 $\pm$ 0.0014	0.6755 $\pm$ 0.0017



**Figure 2: Performance with different latent dimensionality.**

vector dimensionality. We set  $\alpha_f = 0.012$ ,  $\alpha_c = 0.0016$ ,  $\lambda_1 = 0.0125$ ,  $\lambda_2 = 0.000003$ . We observe that MAE and RMSE are improved when the latent vector dimensionality increases. The best results are achieved when the dimensionality is 20, after which MAE and RMSE increase and become relatively stable. In the same way, we set latent dimensionality to 30 for Douban data ( $\alpha_f = 0.005$ ,  $\alpha_c = 0.00005$ ,  $\lambda_1 = 0.01$ ,  $\lambda_2 = 0.0001$ ), and 35 for Douban music data ( $\alpha_f = 0.005$ ,  $\alpha_c = 0.00005$ ,  $\lambda_1 = 0.04$ ,  $\lambda_2 = 0.0001$ ).

Table 1 summarizes the performance of all models when different datasets are used. By modeling pair-wise interactions among contexts, users and items, which are all represented by latent vectors, FM evidently outperforms biased MF, indicating the importance of contextual information in collaborative filtering. In all cases, our context-aware model outperforms FM (and biased MF), demonstrating the advantage of directly integrating contextual information into latent factors to learn context-aware latent representations. The results also reveal the limitation of restricting contexts to share the same latent space with users and items.

## 6. CONCLUSION

In this paper, we propose a generic framework to integrate contextual information into latent factor models. Context influence factor, which is derived by combining contexts, is directly combined with latent factor to generate content-aware latent representations. Such a framework is instantiated by using biased MF model. SGD based optimization procedure is developed to fit the context-aware biased MF model. Experimental results demonstrate the advantages of our model by comparing with the base model and FM.

An immediate next step is to instantiate the generic framework by using CF models for implicit feedback data such as Bayesian Personalized Ranking (BPR) [12] to provide context-aware top-N recommendation, which is useful in more application scenarios. Another direction is to investigate more sophisticated ways (e.g., non-linear methods) to combine contexts to better model context influence factor.

## 7. REFERENCES

- [1] Gediminas Adomavicius, Ramesh Sankaranarayanan, Shahana Sen, and Alexander Tuzhilin. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Trans. Inf. Syst.*, 23(1):103–145, 2005.
- [2] Deepak Agarwal, Bee-Chung Chen, and Bo Long. Localized factor models for multi-context recommendation. In *Proceedings of the 17th SIGKDD*, 2011.
- [3] Tianqi Chen, Hang Li, Qiang Yang, and Yong Yu. General functional matrix factorization using gradient boosting. In *Proceedings of the 30th ICML*, 2013.
- [4] Chen Cheng, Fen Xia, Tong Zhang, Irwin King, and Michael R. Lyu. Gradient boosting factorization machines. In *Proceedings of the 8th RecSys*, 2014.
- [5] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the 4th RecSys*, 2010.
- [6] Yehuda Koren. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proceedings of the 14th SIGKDD*, 2008.
- [7] Yehuda Koren. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th SIGKDD*, 2009.
- [8] Xin Liu and Karl Aberer. Soco: A social network aided context-aware recommender system. In *Proceedings of the 22nd WWW*, 2013.
- [9] Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu, and Irwin King. Recommender systems with social regularization. In *Proceedings of the 4th WSDM*, 2011.
- [10] Trung V. Nguyen, Alexandros Karatzoglou, and Linas Baltrunas. Gaussian process factorization machines for context-aware recommendations. In *Proceedings of the 37th SIGIR*, 2014.
- [11] Steffen Rendle. Factorization machines with libFM. *ACM Trans. Intell. Syst. Technol.*, 3(3):57:1–57:22, May 2012.
- [12] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th UAI*, 2009.
- [13] Steffen Rendle, Zeno Gantner, Christoph Freudenthaler, and Lars Schmidt-Thieme. Fast context-aware recommendations with factorization machines. In *Proceedings of the 34th SIGIR*, 2011.
- [14] Yue Shi, Alexandros Karatzoglou, Linas Baltrunas, Martha Larson, and Alan Hanjalic. Cars2: Learning context-aware representations for context-aware recommendations. In *Proceedings of the 23rd CIKM*, 2014.
- [15] Yue Shi, Alexandros Karatzoglou, Linas Baltrunas, Martha Larson, Alan Hanjalic, and Nuria Oliver. Tfmap: Optimizing map for top-n context-aware recommendation. In *Proceedings of the 35th SIGIR*, 2012.
- [16] Erheng Zhong, Wei Fan, Junwei Wang, Lei Xiao, and Yong Li. Comsoc: Adaptive transfer of user behaviors over composite social network. In *Proceedings of the 18th SIGKDD*, 2012.