Journal of Computer Science and Cybernetics, V.31, N.2 (2015), 133–147
DOI: 10.15625/1813-9663/31/2/5860

FACTORIZATION FORECASTING APPROACH FOR USER MODELING

NGUYEN THAI-NGHE[1] AND LARS SCHMIDT-THIEME[2]

[1] *Can Tho University, Vietnam; ntnghe@cit.ctu.edu.vn*
[2] *University of Hildesheim, Germany; schmidt-thieme@ismll.de*

Abstract. User modeling is a task which customizes and adapts the systems to meet users' specific needs. The user modeling is widely used in many areas. For example, in e-commerce, it is used for modeling consumers' preferences (behaviours) then predicting their future preferences to recommend suitable products to them. In e-learning (e.g., intelligent tutoring systems - ITS), the user modeling is used to model the learners (students) to track/predict their performance/knowledge.

In this work, an approach which integrates forecasting model into matrix factorization model to take into account sequential/temporal effects in user modeling since users' need/knowledge may change overtime is introduced. The model as well as how to use stochastic gradient descent to learn this model, then resulting with an algorithm are thoroughly presented. The proposed model is validated using several data sets which are extracted from both e-commerce and e-learning areas. Experimental results on these data sets show that the proposed approach performs nicely. This could be a promising approach for user modeling.

Keywords. User modeling, matrix factorization, factorization forecasting, sequential effect, recommender systems, intelligent tutoring systems

1. INTRODUCTION

User modeling is an interesting topic which has been used in many areas [7] such as Adaptive hypermedia systems, Intelligent tutoring systems (ITS), Expert systems, Recommender systems (RS), etc[1].

For example, in Adaptive hypermedia systems, the user modeling is used to display contents and hyperlinks that are chosen on basis of users' specific characteristics.

In e-commerce, the user modeling is used for modeling consumers' preferences/ behaviors then predicting their future preferences to produce suitable recommendations [16, 17].

Recommender System is a type of information filtering system which is used to predict user preference on an item which had not been seen in the past (item could be song, movie, video clip, paper, etc). For example, in an online shopping system such as Amazon, to maximize the user shopping capability, the system usually takes into account which user likes which item based on the past behaviors of the user (these behaviors could be the users' rate, number of clicks, browsing time,.. on the items). Using these behaviors, the system can automatically predict the next items which the user may prefer and then recommend them to that user. Besides e-commerce area, Recommender System is now used

[1] en.wikipedia.org/wiki/User_modeling

in many other areas such as in entertainment: Music recommendation (e.g., www.last.fm), movie recommendation (e.g., www.netflix.com), video clip recommendation (e.g., www.youtube.com). There are many published works in this area including state-of-the-art techniques such as Matrix Factorization [12]. Other works can be found in [17].

Another used area of the user modeling is in e-learning such as intelligent tutoring systems (ITS) in which their aim is to help students in a specific field of study. In this area, the user modeling is used to model the learners'(students') performance, to track/predict their knowledge, and to recommend learning resource such as books, papers, web links, etc. to the learners [4, 20, 24]. The tutoring system can adapt to specific student by presenting appropriate exercises/examples as well as offering hints/help where the student is most likely to need them.

This work focuses on two main areas which are recommender systems (RS) and intelligent tutoring systems (ITS). In these two areas, many works have been published. Typical works in RS and ITS can be found in [17] and [13], respectively.

For improving model performance in user modeling, time (or sequence) is an important factor and should be taken into account. For example, in the recommender systems, user preferences (or activities) may change overtime. In the tutoring systems, the learner's knowledge may also accumulate/improve overtime (that is what we expect in education since the students may gain experience overtime). Thus, sequential/temporal effect is an important information for the models.

In this work, an approach, which is extended from previous work in [22], that integrates forecasting model into matrix factorization model to take into account the sequential/temporal effect in user modeling is thoroughly introduced.

2. PROBLEM FORMULATION

In this work, the method which uses historical data about user activities/behaviors to predict the user activities/behaviors in the future is proposed.

The user activity/behavior may have different name/meaning depending on the systems. For example, in recommender systems, they could be user rating, user click, etc.; In tutoring systems, the user activity/behavior could be represented by student performance, grading, score, etc. To simplify the terms, from this point forward, **user feedback** is called instead of user activities/ behaviors/ performances/..

More formally, let $\mathbf{U}$ be the set of users (u be a user), $\mathbf{I}$ be the set of items (i be an item), $\mathbf{T}$ be the set of times, and $\mathbf{R}$ be the set of feedback on the items by the users.

Let

$$\mathcal{D}^{train} \subseteq (\mathbf{U} \times \mathbf{I} \times \mathbf{T} \times \mathbf{R})$$

and

$$\mathcal{D}^{test} \subseteq (\mathbf{U} \times \mathbf{I} \times \mathbf{T} \times \mathbf{R})$$

be the train data set and test data set, respectively.

Then the problem of predicting the user feedback is, given $\mathcal{D}^{train}$ to find

$$\hat{r} : \mathbf{U} \times \mathbf{I} \times \mathbf{T} \to \mathbb{R}$$

such that a measure $\mathcal{E}(\hat{r}, r)$ will be satisfied a certain condition, where r is the the true feedback, i.e.,

$$r : \mathbf{U} \times \mathbf{I} \times \mathbf{T} \to \mathbf{R}, \qquad (u, i, t) \mapsto r$$

For example, if $\mathcal{E}$ is an error measure, e.g., root mean square error (RMSE), it needs to be minimum.

$$RMSE = \sqrt{\frac{\sum_{(u,i,r,t) \in \mathcal{D}^{test}} (r - \hat{r}_{(u,i,t)})^2}{|\mathcal{D}^{test}|}}$$

The time can be exploited by two different ways:

1. **Concrete time**, which represents specific points of time, as used in the literature [6]. This kind of time is usually used in context-aware recommender systems, e.g., weekend, weekday, Christmas day, etc [1,8,21].

2. **Relative time**, which describes sequence (order) of the data, e.g., the sequence of solving problem in tutoring systems. This kind of time is usually used in forecasting techniques or in modeling sequential data [3,14].

This work focuses on the **relative time**. Thus, the formulation of the train set and the test set is changed, denoting

$$\mathcal{D}^{train} \subseteq (\mathbf{U} \times \mathbf{I} \times \mathbf{R})^*$$

and

$$\mathcal{D}^{test} \subseteq (\mathbf{U} \times \mathbf{I} \times \mathbf{R})^*$$

3. FACTORIZATION MODELS

In this section, first, the current state-of-the-art model in recommender systems, which is matrix factorization [12], is summarized. Then, an extended model which is called tensor factorization is presented. These models belong to the group of latent factor models.

3.1. Matrix Factorization

Matrix factorization is the task of approximating a matrix **X** by the product of two smaller matrices **W** and **H** such that **X** can be re-constructed from these two smaller matrices [12], .i.e.

$$\mathbf{X} \approx \mathbf{W}\mathbf{H}^T$$

An illustration of matrix decomposition is presented in Figure 1

In the context of recommender systems, the matrix **X** is the partially observed ratings matrix; $\mathbf{W} \in \mathbb{R}^{|\mathbf{U}| \times K}$ is a matrix where each row u is a vector containing the K latent factors ($K << |\mathbf{U}|, K << |\mathbf{I}|$) describing the user u and $\mathbf{H} \in \mathbb{R}^{|\mathbf{I}| \times K}$ is a matrix where each row i is a vector containing the K factors describing the item i.

Let w_{uk} and h_{ik} be the elements of **W** and **H** (**w** and **h** are their vectors, respectively), respectively, then the rating given by a user u to an item i is predicted by:

$$\hat{r}_{ui} = \mathbf{w} \cdot \mathbf{h}^T = \sum_{k=1}^{K} w_{uk} h_{ik} \tag{1}$$

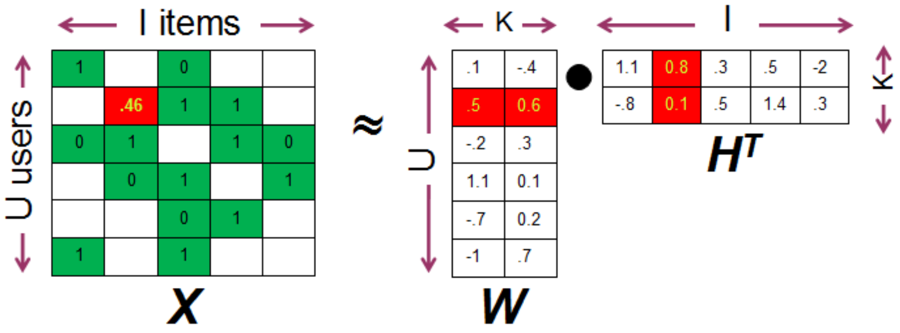


Figure 1: An illustration of matrix factorization

where **W** and **H** are the latent matrices (model parameters) and can be learned by optimizing an objective function given a criterion such as Root Mean Squared Error (RMSE).

$$\text{RMSE} = \sqrt{\frac{\sum_{(u,i,r)\in\mathcal{D}^{test}}(r_{ui} - \hat{r}_{ui})^2}{|\mathcal{D}^{test}|}} \tag{2}$$

3.2. Training Phase

Using matrix factorization, training the model is to find the optimal parameters **W** and **H**. One approach is that, first, these two matrices are initialized with some random values, e.g., from the normal distribution $\mathcal{N}(0, \sigma^2)$ with mean = 0 and standard deviation $\sigma^2 = 0.01$, and compute the error (objective) function, for example

$$\mathcal{O}^{MF} = \sum_{(u,i,u)\in\mathcal{D}^{train}} e_{ui}^2 \tag{3}$$

where

$$e_{ui}^2 = (r_{ui} - \hat{r}_{ui})^2 = (r_{ui} - \sum_{k=1}^{K} w_{uk}h_{ik})^2 \tag{4}$$

then try to minimize this error function by updating the values of **W** and **H** iteratively, e.g., using gradient descent [19].

To minimize the error function in equation (3), it is needed to know for each data point in which direction to update the value of w_{uk} and h_{ik}. Thus, gradients of the function (4) are computed:

$$\frac{\partial}{\partial w_{uk}} e_{ui}^2 = -2e_{ui}h_{ik} = -2(r_{ui} - \hat{r}_{ui})h_{ik} \tag{5}$$

$$\frac{\partial}{\partial h_{ik}} e_{ui}^2 = -2e_{ui}w_{uk} = -2(r_{ui} - \hat{r}_{ui})w_{uk} \tag{6}$$

After having the gradients, the values of w_{uk} and h_{ik} are updated in the direction opposite to the gradient:

$$w'_{uk} = w_{uk} - \beta\frac{\partial}{\partial w_{uk}} e_{ui}^2 = w_{uk} + 2\beta e_{ui}h_{ik} = w_{uk} + 2\beta(r_{ui} - \hat{r}_{ui})h_{ik} \tag{7}$$

$$h'_{ik} = h_{ik} - \beta \frac{\partial}{\partial h_{ik}} e_{ui}^2 = h_{ik} + 2\beta e_{ui} w_{uk} = h_{ik} + 2\beta (r_{ui} - \hat{r}_{ui}) w_{uk} \tag{8}$$

where β is a learning rate ($0 \leq \beta < 1$).

The values of $\mathbf{W}$ and $\mathbf{H}$ are iteratively updated until the error converges on its minimum ($\mathcal{O}^{\text{MF}}_{Iter_{(n-1)}} - \mathcal{O}^{\text{MF}}_{Iter_n} < \epsilon$) or reaching a predefined number of iterations.

Regularization term: To prevent over-fitting, one can modify the error function (4) by adding a term which controls the magnitudes of the factor vectors such that $\mathbf{W}$ and $\mathbf{H}$ would give a good approximation of $\mathbf{X}$ without having to contain large numbers. The error function now becomes:

$$\mathcal{O}^{MF} = (r_{ui} - \hat{r}_{ui})^2 + \lambda \left(||\mathbf{W}||_F^2 + ||\mathbf{H}||_F^2 \right) \tag{9}$$

$$= (r_{ui} - \sum_{k=1}^{K} w_{uk} h_{ik})^2 + \lambda \left(||\mathbf{W}||_F^2 + ||\mathbf{H}||_F^2 \right) \tag{10}$$

where λ is a regularization term ($0 \leq \lambda < 1$) which is used to prevent overfitting and $||.||^2$ is a Frobenius norm [2]. For example,

$$||\mathbf{W}||_F = \sqrt{\sum_{u=1}^{|U|} \sum_{k=1}^{K} |w_{uk}|^2}$$

With this new error function, the values of w_{uk} and h_{ik} are updated by

$$w'_{uk} = w_{uk} + \beta(2e_{ui}h_{ik} - \lambda w_{uk}) = w_{uk} + \beta(2(r_{ui} - \hat{r}_{ui})h_{ik} - \lambda w_{uk}) \tag{11}$$

$$h'_{ik} = h_{ik} + \beta(2e_{ui}w_{uk} - \lambda h_{ik}) = h_{ik} + \beta(2(r_{ui} - \hat{r}_{ui})w_{uk} - \lambda h_{ik}) \tag{12}$$

Algorithm 1 describes details of training a matrix factorization model using stochastic gradient descent (the stochastic gradient descent is used for all algorithms in our work since it has been shown that the computing cost of stochastic gradient descent has a huge advantage for large-scale problems [11]).

First, the parameters $\mathbf{W}$ and $\mathbf{H}$ are initialized randomly from the normal distribution $\mathcal{N}(0, \sigma^2)$ with mean is 0 and standard deviation $\sigma^2 = 0.01$, as in lines 2-3. While the stopping condition is not met, e.g., reaching the maximum number of predefined iterations or converging ($\mathcal{O}^{\text{MF}}_{Iteration_{(n-1)}} - \mathcal{O}^{\text{MF}}_{Iteration_n} < \epsilon$), the latent factors are updated iteratively. For example, in each iteration, an instance in the training set (u, i, p) is randomly selected, then the prediction for this user and item is computed, as in lines 5-9. Then the error in this iteration is estimated and the values of $\mathbf{W}$ and $\mathbf{H}$ are updated as in lines 11-14.

3.3. Prediction Phase

After the training phase, there are the two optimal latent factors $\mathbf{W}$ and $\mathbf{H}$, the remaining task is straightforward. The rating of user u in for a given item i is predicted easily by equation (1).

Please note that, for the new users or the new items, those are in the test set but not in the train set, the global average (average rating of all users in the training set) can be simply returned. This is a cold-start problem [18], however, this problem will not be under discussion in this study.

[2] http://en.wikipedia.org/wiki/Matrix_norm#Frobenius_norm

Algorithm 1 Learn a matrix factorization using stochastic gradient descent with K latent factors, β learning rate, λ regularization term, and a stopping criterion

```
procedure MF-SGD($\mathcal{D}^{train}$, $K$, $\beta$, $\lambda$, stopping condition)
Let $u \in U$ be a user, $i \in I$ an item, $r \in R$ a rating score
Let $W[|U|][K]$ and $H[|I|][K]$ be latent factors of users and items
    $W \leftarrow \mathcal{N}(0, \sigma^2)$
    $H \leftarrow \mathcal{N}(0, \sigma^2)$
    while (Stopping criterion is NOT met) do
        Draw randomly $(u, i, r)$ from $\mathcal{D}^{train}$
        $\hat{r} \leftarrow 0$
        for $k \leftarrow 1, \ldots, K$ do
            $\hat{r} \leftarrow \hat{r} + W[u][k] * H[i][k]$
        end for
        $e_{ui} = r - \hat{r}$
        for $k \leftarrow 1, \ldots, K$ do
            $W[u][k] \leftarrow W[u][k] + \beta * (e_{ui} * H[i][k] - \lambda * W[u][k])$
            $H[i][k] \leftarrow H[i][k] + \beta * (e_{ui} * W[u][k] - \lambda * H[i][k])$
        end for
    end while
    return $\{W, H\}$
end procedure
```

3.4. Tensor Factorization

Tensor Factorization is a general form of matrix factorization. A tensor is also known as a cube and its modes is also called dimensions. A two-mode tensor is a matrix and a three-mode tensor is thus the cube [6,10].

Given the three-mode tensor $\mathcal{Z}$ with its size $|\mathbf{U}| \times |\mathbf{I}| \times |\mathbf{T}|$, where the first and the second mode describe the user and the item, respectively; the third mode describes the time. Then $\mathcal{Z}$ can be written as a sum of rank-1 tensors by using the Tucker tensor [23] or by using the CANDECOM-PARAFAC tensor [9] as in the following:

$$\mathcal{Z} \approx \sum_{k=1}^{K} \lambda_k \ \mathbf{w}_k \circ \mathbf{h}_k \circ \mathbf{q}_k \tag{13}$$

where λ_k is a scalar vector; $\circ$ is an outer product; and each vector $\mathbf{w}_k \in \mathbb{R}^{|\mathbf{U}|}$, $\mathbf{h}_k \in \mathbb{R}^{|\mathbf{I}|}$, and $\mathbf{q}_k \in \mathbb{R}^{|\mathbf{T}|}$ describes the latent factors of user, item, and time, respectively. An illustration of tensor decomposition is presented in Figure 2.

The tensor factorization (or tensor decomposition) approach has been used in many other areas such as recommender systems, topic modeling, link prediction, and more [6,10,15]. However, on the time mode, most of them are used for aforementioned **concrete time**.

The idea is adopted from this tensor factorization approach to model the temporal/sequential effects in user modeling by using the **relative time** which is usually used in forecasting problems.

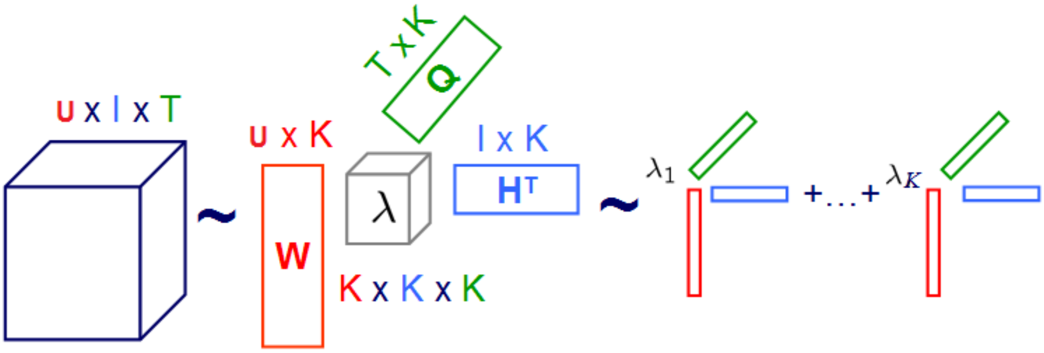


Figure 2: A tensor is decomposed into three low-rank matrices

4. FACTORIZATION FORECASTING APPROACH

In this section, the model that incorporates the forecasting technique into the latent factor model is introduced. Similar to Matrix Factorization [12] which is presented in previous section, here, a tensor is decomposed into three smaller matrices so that the original tensor can be re-constructed from these matrices, as presented in Figure 2.

There are several ways to decompose the tensor as presented in [10]. In this work, three smaller matrices are obtained by optimizing an objective function using stochastic gradient descent approach. The objective function for optimizing is presented as

$$\mathcal{O}^{FF} = \sum_{(u,i,r)\in\mathcal{D}^{train}} e^2_{uiT} + \lambda\left(||\mathbf{W}||^2_F + ||\mathbf{H}||^2_F + ||\mathbf{H}'||^2_F + ||\mathbf{Q}||^2_F + \mathbf{b}^2_u + \mathbf{b}^2_i\right) \quad (14)$$

$$= \sum_{(u,i,r)\in\mathcal{D}^{train}} \left(r_{uiT} - \hat{r}_{uiT}\right)^2 + \lambda\left(||\mathbf{W}||^2_F + ||\mathbf{H}||^2_F + ||\mathbf{H}'||^2_F + ||\mathbf{Q}||^2_F + \mathbf{b}^2_u + \mathbf{b}^2_i\right) \quad (15)$$

In this objective function, difference to matrix factorization where there are only two latent matrices **W** and **H** representing for the latent factors of the user and item, respectively. Here, two more matrices are in use. The first matrix is **Q** which is the time latent matrix, as presented in Figure 2. The second matrix is **H'** which take into account the information of the previous item in the sequence since in our previous work shows that this work well for sequential data [22].

Moreover, in the objective function, μ, $\mathbf{b}_u$, and $\mathbf{b}_i$ which are global average, user bias, and item bias are included, as shown in [12], respectively.

The global average μ is determined by

$$\mu = \frac{\sum_{(u,i,r)\in\mathcal{D}^{train}} r}{|\mathcal{D}^{train}|} \quad (16)$$

The user bias $\mathbf{b}_u$ is determined by:

$$\mathbf{b}_u = \frac{\sum_{(u',i,r)\in\mathcal{D}^{train}|u'=u} (r-\mu)}{|\{(u',i,r)\in\mathcal{D}^{train}|u'=u\}|} \quad (17)$$

Algorithm 2 Learn a factorization forecasting model using stochastic gradient descent with K latent factors, β learning rate, λ regularization weight, L history length, and stopping condition

```
procedure TFF($\mathcal{D}^{train}$, $K$, $\beta$, $\lambda$, $L$, stopping condition)
    $\{\mathbf{W}, \mathbf{H}, \mathbf{H'}, \mathbf{Q}\} \leftarrow \mathcal{N}(0, \sigma^2)$
    $\mu \leftarrow \frac{\sum_{r \in \mathcal{D}^{train}} r}{|\mathcal{D}^{train}|}$
    for each user $u$ do
        $\mathbf{b}_u \leftarrow \frac{\sum_i (r_{ui} - \mu)}{|\mathcal{D}_u^{train}|}$
    end for
    for each item $i$ do
        $\mathbf{b}_i \leftarrow \frac{\sum_u (r_{ui} - \mu)}{|\mathcal{D}_i^{train}|}$
    end for
    while (stopping condition is NOT met) do
        Draw randomly $(u, i, r_{uiT})$ at row $T$ from $\mathcal{D}^{train}$
                                        ▷ $T$ is considered as current time in the sequence
        $$\hat{r}_{uiT} \leftarrow \mu + b_u + b_i + \sum_{k=1}^{K} \left( w_{uk} h_{ik} \left( \frac{\sum_{t=1}^{L} h'_{(T-t)k} \cdot q_{tk} \cdot r^u_{T-t}}{L} \right) \right)$$
        $e_{uiT} \leftarrow r_{uiT} - \hat{r}_{uiT}$
        $\mu \leftarrow \mu + \beta \cdot e_{uiT}$
        $\mathbf{b}_u \leftarrow \mathbf{b}_u + \beta \cdot (e_{uiT} - \lambda \cdot \mathbf{b}_u)$
        $\mathbf{b}_i \leftarrow \mathbf{b}_i + \beta \cdot (e_{uiT} - \lambda \cdot \mathbf{b}_i)$
        for $k \leftarrow 1, \dots, K$ do
            $$w_{uk} \leftarrow w_{uk} - \beta \left( \frac{\partial \mathcal{O}^{\mathrm{FF}}}{\partial w_{uk}} \right)$$
            $$h_{ik} \leftarrow h_{ik} - \beta \left( \frac{\partial \mathcal{O}^{\mathrm{FF}}}{\partial h_{ik}} \right)$$
            for $t \leftarrow 1, \dots, L$ do
                $$h'_{(T-t)k} \leftarrow h'_{(T-t)k} - \beta \left( \frac{\partial \mathcal{O}^{\mathrm{FF}}}{\partial h'_{(T-t)k}} \right)$$
                $$q_{tk} \leftarrow q_{tk} - \beta \left( \frac{\partial \mathcal{O}^{\mathrm{FF}}}{\partial q_{tk}} \right)$$
            end for
        end for
    end while
    return $\{\mathbf{W}, \mathbf{H}, \mathbf{H}', \mathbf{Q}, \mathbf{b}_u, \mathbf{b}_i, \mu\}$
end procedure
```

The item bias $\mathbf{b}_i$ is determined by:

$$\mathbf{b}_i = \frac{\sum_{(u,i',r)\in\mathcal{D}^{train}|i'=i}(r-\mu)}{|\{(u,i',r)\in\mathcal{D}^{train}\}|i'=i|} \tag{18}$$

In the objective function (15), the second term

$$\lambda\left(||\mathbf{W}||_F^2 + ||\mathbf{H}||_F^2 + ||\mathbf{H'}||_F^2 + ||\mathbf{Q}||_F^2 + \mathbf{b}_u^2 + \mathbf{b}_i^2\right)$$

is the regularization which is used to prevent over-fitting, and the first term e_{uiT}^2 is the squared error, which is determined by

$$e_{uiT}^2 = (r_{uiT} - \hat{r}_{uiT})^2$$

where r_{uiT} is the true feedback value of user u on item i at time T, and $\hat{r}_{uiT}$ is the prediction value, determined by

$$\hat{r}_{uiT} = \mu + \mathbf{b}_u + \mathbf{b}_i + \sum_{k=1}^{K} w_{uk} h_{ik} \Phi_{Tk} \tag{19}$$

where K is the number of latent factors; T is the (current) time in the sequence to predict; and Φ_{Tk} is determined by

$$\Phi_{Tk} = \frac{\sum_{t=1}^{L} h'_{(T-t)k} \cdot q_{tk} \cdot r_{T-t}^{u}}{L} \tag{20}$$

For simplification purpose, Moving Average forecasting[3] with a period L on the time mode is in use, however, other forecasting techniques could also be applied in the similar way.

In Φ_{Tk} equation, q_{tk} is the time latent factor; r_{T-t}^{u} is the true feed back of user u at the previous time in a sequence; L is the history length as using in the Moving Average method and $h'_{(T-t)k}$ is the latent factor of the previous item in the sequence.

In this approach, the ideas in [14, 22] are used for both e-commerce and e-learning areas. For example, in e-commerce area, [14] have used matrix factorization with Markov chains to 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. The authors proposed using previous "basket of items" to predict the next "basket of items" with high probabilities that the users might want to buy.

In the intelligent tutoring system environment, a natural fact is that the performance of the learners not only depend on the recent knowledge (e.g., the knowledge in the previous problems or lessons, which act as "previous basket of items") but also depend on the cumulative knowledge in the past that the learners have studied [22]. Thus, for modeling the sequential/temporal effects, the researchers propose integrating the forecasting technique into the latent factor model.

For learning the model (in equation (15)) using stochastic gradient descent, the model parameters ($\mathbf{W}$, $\mathbf{H}$, $\mathbf{Q}$,..) are iteratively updated using the following gradients

[3] Moving average is an unweighted mean of previous n data points in the sequence

Data set	**#User**	**#Item**	**#feedback**
Algebra 2009-2010 (Algebra)	3,310	1,422,200	8,918,054
Bridge-to-Algebra 2009-2010 (Bridge)	6,043	888,834	20,012,498
Assistments	8,519	35,978	1,011,079
Movielens-100k	943	1,682	100,000
Movielens-1M	6,040	3,900	1,000,209

Table 1: Data set information

$$\frac{\partial \mathcal{O}^{FF}}{\partial w_{uk}} = -2e_{uiT}h_{ik}\Phi_{Tk} + \lambda w_{uk} \tag{21}$$

$$\frac{\partial \mathcal{O}^{FF}}{\partial h_{ik}} = -2e_{uiT}w_{uk}\Phi_{Tk} + \lambda h_{ik} \tag{22}$$

$$\frac{\partial \mathcal{O}^{FF}}{\partial h'_{(T-t)k}} = -2e_{uiT}w_{uk}h_{ik}\left(\frac{\sum_{t=1}^{L} q_{tk} \cdot r^{u}_{T-t}}{L}\right) + \lambda h'_{(T-t)k} \tag{23}$$

$$\frac{\partial \mathcal{O}^{FF}}{\partial q_{tk}} = -2e_{uiT}w_{uk}h_{ik}\left(\frac{\sum_{t=1}^{L} h'_{(T-t)k} \cdot r^{u}_{T-t}}{L}\right) + \lambda q_{tk} \tag{24}$$

Finally, it comes up with a learning algorithm as presented in Algorithm 2 which briefly summarizes the training process of the proposed factorization forecasting model.

In the Algorithm 2, first, the parameters (**W,H,H',Q**) are initialized randomly from the normal distribution $\mathcal{N}(0, \sigma^2)$ with mean is 0 and standard deviation $\sigma^2 = 0.01$, as in line 2. Then, the values of the global average, user bias, and item bias are computed as in lines 3-9. While the stopping condition is not met, e.g., not reaching the predefined number of iterations or not converging ($\mathcal{O}^{\text{FF}}_{Iteration_{(n-1)}} - \mathcal{O}^{\text{FF}}_{Iteration_n} < \epsilon$), the latent factors and the biased terms are updated iteratively.

After the training phase, the parameters are obtained. Then, one can easily predict the value of the user feedback using equation (19).

5. EXPERIMENTS

This section presents several benchmark data sets which are collected from real systems that the proposed method can be applied. Those are belong to educational environment (intelligent tutoring systems - ITS) and entertainment environment (recommender systems - RS).

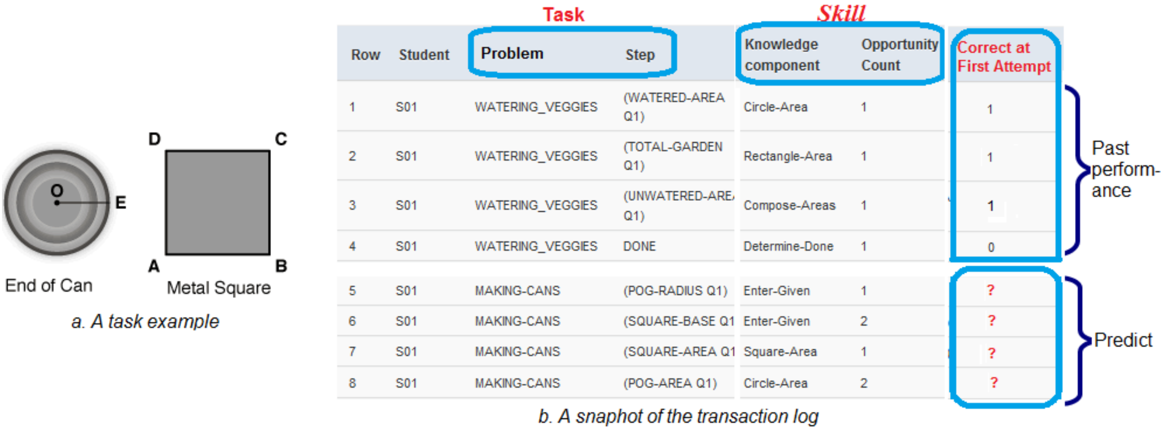
5.1. Practical issues and Data sets

5.1.1. Data sets from the ITS

In this environment, data sets from the KDD Challenge 2010[4] and the ASSISTments Platform[5] are used for experiments. These data sets represent the log files of interactions between the students

[4] http://pslcdatashop.web.cmu.edu/KDDCup/

[5] http://teacherwiki.assistment.org/wiki/Assistments_2009-2010_Full_Dataset


D
C
O
E
A
B
End of Can
Metal Square
Task
Skill
Past perform- ance
Predict

a. A task example

Row	Student	Problem	Step	Knowledge component	Opportunity Count	Correct at First Attempt
1	S01	WATERING_VEGGIES	(WATERED-AREA Q1)	Circle-Area	1	1
2	S01	WATERING_VEGGIES	(TOTAL-GARDEN Q1)	Rectangle-Area	1	1
3	S01	WATERING_VEGGIES	(UNWATERED-ARE/ Q1)	Compose-Areas	1	1
4	S01	WATERING_VEGGIES	DONE	Determine-Done	1	0
5	S01	MAKING-CANS	(POG-RADIUS Q1)	Enter-Given	1	?
6	S01	MAKING-CANS	(SQUARE-BASE Q1	Enter-Given	2	?
7	S01	MAKING-CANS	(SQUARE-AREA Q1	Square-Area	1	?
8	S01	MAKING-CANS	(POG-AREA Q1)	Circle-Area	2	?

b. A snaphot of the transaction log

Figure 3: Predicting student performance: A scenario (Picture source: pslcdatashop.web.cmu.edu/KDDCup)

and the tutoring systems. While the students solve problems in the tutoring system, their activities, success and progress indicators are logged as individual rows in the data sets. The user feedback in these data sets are the student performance scores (0: incorrect; 1: correct). This also is the target prediction task.

Figure 3a presents an example of the task[6]. Given the circle and the square as in this figure, the task for students could be *"What is the remaining area of the square after removing the circular area?"* [22]

To solve this task (question), students could do some smaller subtasks which is called as solving-step. Each step may be required one or more skills (or it can be called as "knowledge components"), for example:
- Step 1: Calculate the circle area (the required skills for this step are the value of π, square, multiplication, and finally putting them together $area_1 = \pi * (OE)^2$)
- Step 2: Calculate the square area (skill: $area_2 = (AB)^2$)
- Step 3: Calculate the remaining (skill: $area_2 - area_1$)

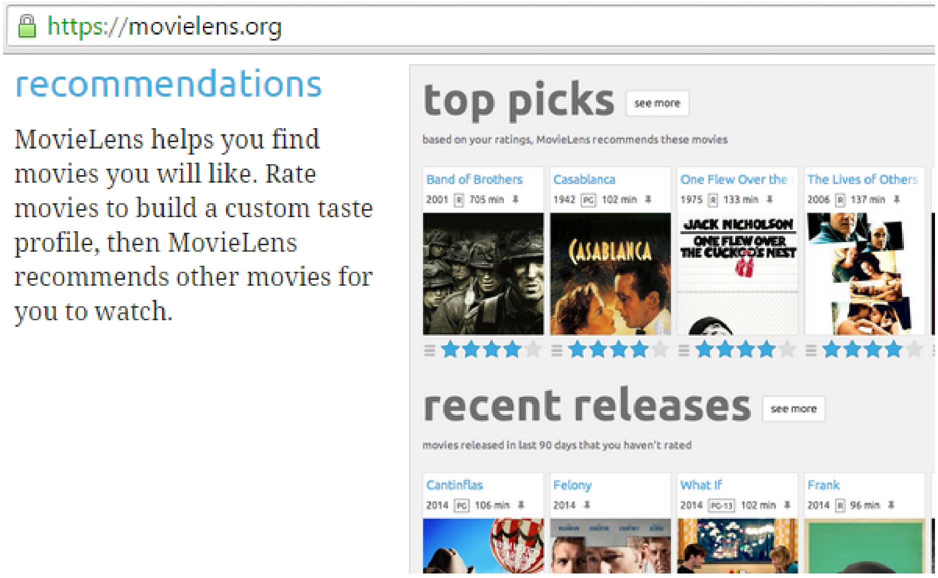
Each solving-step is recorded as a transaction. Figure 3b presents a snapshot of the transactions. Based on the past performance, students' next performance (e.g. correct/incorrect) in solving the tasks will be predicted.

5.1.2. Data sets from the RS

The Movielens[7] data sets are also used for experiments. These data sets are extracted from a movie recommender system (Figure 4) which are widely used in RS area.

Detailed information about these data sets are presented in Table ??.

[6] Source: https://pslcdatashop.web.cmu.edu/KDDCup
[7] http://grouplens.org/datasets/movielens/



https://movielens.org

recommendations

MovieLens helps you find movies you will like. Rate movies to build a custom taste profile, then MovieLens recommends other movies for you to watch.

top picks

based on your ratings, MovieLens recommends these movies

recent releases

movies released in last 90 days that you haven't rated

Figure 4: A snapshot from Movielens system

5.2. Evaluation metric and Baselines

The most popular metric - root mean squared error (RMSE) - is used to evaluate the models.

$$RMSE = \sqrt{\frac{\sum_{(u,i,r,t)\in\mathcal{D}^{test}}(r - \hat{r}_{(u,i,t)})^2}{|\mathcal{D}^{test}|}}$$

To understand how the proposed approach improves to the other methods, the RMSE of several methods is reported such as *global average*, *user average*, and the current state-of-the-art in recommender systems which is *Matrix Factorization - MF* [12]. Moreover, the RMSE of the state-of-the-art in user (student) modeling in the ITS, which is Bayesian Knowledge Tracing [2, 5], is also reported as well.

5.3. Empirical results

The RMSE results are presented in Figures 5 and 6. It can be observed that the Factorization Forecasting (FF) approach performs nicely compared to the other methods including the current state-of-the-art (Matrix Factorization - MF) in recommender systems (user modeling).

Table 2 presents the RMSE of the proposed methods and the well-known Bayesian Knowledge Tracing (BKT) [2, 5]. The results also show that the factorization forecasting (FF) approach have improvements compared to the BKT model.

Data set	BKT	FF
Algebra	0.30561	**0.30159**
Bridge	0.30649	**0.28700**
Assistments	0.46919	**0.43964**

Table 2: RMSE of BKT vs. FF Approach

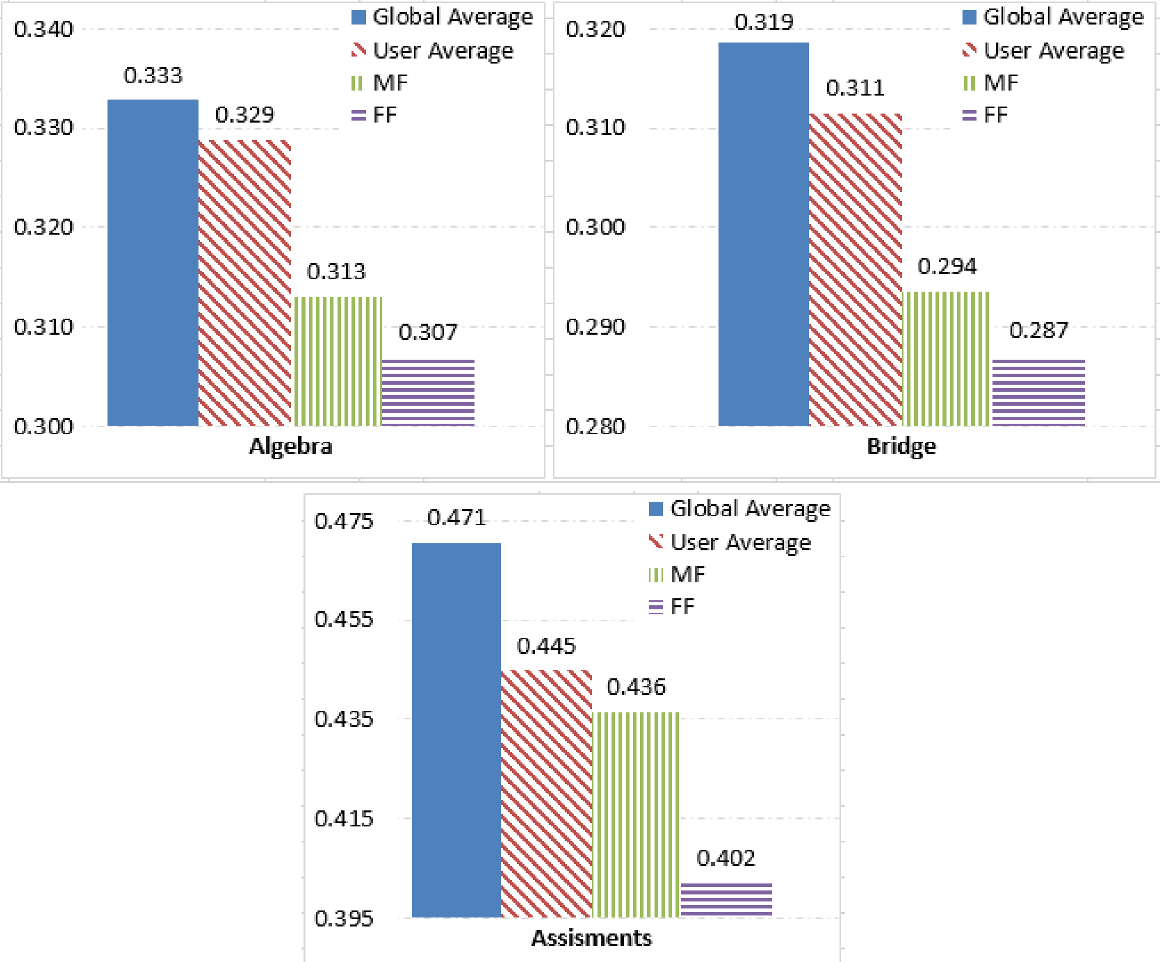


Figure 5: RMSE on ITS data (Algebra, Bridge and Assistments data set)

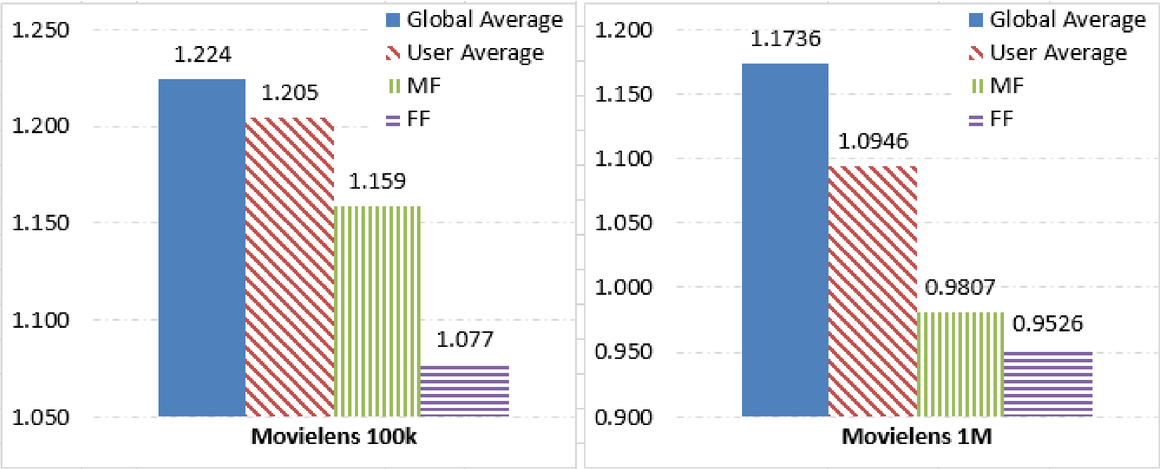


Figure 6: RMSE on recommender system data (Movielens data sets)

6. CONCLUSIONS

This work introduces an approach which integrates forecasting model into matrix factorization model to take into account the sequential/temporal effects in user modeling. Experimental results on several data sets which are extracted from both e-commerce/entertainment and e-learning areas show that the proposed approach performs nicely, thus, this approach could be promising for user modeling area.

The researchers continue to improve the model such as including multi-relational concepts to the model as well as to compare the proposed approach with other advanced methods such as TimeSVD++ [11].

REFERENCES

[1] G. Adomavicius and A. Tuzhilin, "Context-aware recommender systems," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Springer, 2011, pp. 217–253.

[2] R. S. Baker, A. T. Corbett, and V. Aleven, "More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing," in *Proceedings of the 9th International Conference on Intelligent Tutoring Systems*, ser. ITS '08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 406–415.

[3] Y. Bengio, "Markovian models for sequential data," *Neural Computing Surveys*, vol. 2, pp. 129–162, 1999.

[4] A. T. Corbett, K. R. Koedinger, and J. R. Anderson, "Intelligent tutoring systems," in *Handbook of human-computer interaction*, M. G. Helander, T. K. Landauer, and P. V. Prabhu, Eds. Amsterdam: Elsevier Science B. V., 1997, pp. 849–874.

[5] A. T. Corbett and J. R. Anderson, "Knowledge tracing: Modeling the acquisition of procedural knowledge," *User Modeling and User-Adapted Interaction*, vol. 4, pp. 253–278, 1995.

[6] D. M. Dunlavy, T. G. Kolda, and E. Acar, "Temporal link prediction using matrix and tensor factorizations," *ACM Trans. Knowl. Discov. Data*, vol. 5, pp. 10:1–10:27, February 2011.

[7] G. Fischer, "User modeling in humancomputer interaction," *User Modeling and User-Adapted Interaction*, vol. 11, no. 1-2, pp. 65–86, 2001.

[8] Z. Gantner, S. Rendle, and L. Schmidt-Thieme, "Factorization models for context-/time-aware movie recommendations," in *Proceedings of the Workshop on Context-Aware Movie Recommendation*, ser. CAMRa '10. New York, NY, USA: ACM, 2010, pp. 14–19.

[9] R. A. Harshman, "Foundations of the PARAFAC procedure: Models and conditions for an explanatory multi-modal factor analysis," *UCLA Working Papers in Phonetics*, vol. 16, no. 1, p. 84, 1970.

[10] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, September 2009.

[11] Y. Koren, "Factor in the neighbors: Scalable and accurate collaborative filtering," *ACM Transactions on Knowledge Discovery from Data*, vol. 4, no. 1, pp. 1–24, 2010.

[12] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Computer Society Press*, vol. 42, no. 8, pp. 30–37, 2009.

[13] N. Manouselis, H. Drachsler, K. Verbert, and O. C. Santos, *Recommender Systems for Technology Enhanced Learning: Research Trends and Applications.* Springer, 2014.

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

[15] S. Rendle and L. Schmidt-Thieme, "Pairwise interaction tensor factorization for personalized tag recommendation," in *Proceedings of the 3th ACM International Conference on Web Search and Data Mining (WSDM 2010).* New York, USA: ACM, 2010, pp. 81–90.

[16] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "Grouplens: an open architecture for collaborative filtering of netnews," in *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, ser. CSCW '94. New York, NY, USA: ACM, 1994, pp. 175–186.

[17] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds., *Recommender Systems Handbook.* Springer, 2011.

[18] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in Artificial Intelligence*, vol. 2009, p. 4, 2009.

[19] G. Takács, I. Pilászy, B. Németh, and D. Tikk, "On the Gravity recommendation system," in *Proceddings of KDD Cup Workshop at SIGKDD'07, 13th ACM Int. Conf. on Knowledge Discovery and Data Mining*, San Jose, CA, USA, 2007, pp. 22–30.

[20] N. Thai-Nghe, L. Drumond, T. Horváth, , A. Nanopoulos, and L. Schmidt-Thieme, "Matrix and tensor factorization for predicting student performance," in *Proceedings of the 3rd International Conference on Computer Supported Education (CSEDU 2011). Best Student Paper Award*, Noordwijkerhout, the Netherlands, 2011, pp. 69 – 78.

[21] N. Thai-Nghe, T. Horváth, and L. Schmidt-Thieme, "Context-aware factorization for personalized students task recommendation," in *Proceedings of the international workshop on personalization approaches in learning environments*, vol. 732, 2011, pp. 13–18.

[22] ——, "Factorization models for forecasting student performance," in *Pechenizkiy, M., Calders, T., Conati, C., Ventura, S., Romero , C., and Stamper, J. (Eds.). Proceedings of the 4th International Conference on Educational Data Mining (EDM 2011)*, Eindhoven, the Netherlands, 2011, pp. 11–20.

[23] L. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, pp. 279–311, 1966.

[24] B. P. Woolf, *Building Intelligent Interactive Tutors, Student-Centered Strategies for Revolutionizing E-Learning.* Elsevier, Morgan Kaufmann, 2008.

Received on January 24 - 2015
Revised on June 10 - 2015