Attention based collaborative filtering

Mingsheng Fu, Hong Qu*, Dagmawi Moges, Li Lu

School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, PR China

A R T I C L E   I N F O

Article history:
Received 14 December 2017
Revised 7 May 2018
Accepted 17 May 2018
Available online 22 May 2018

Communicated by Dr Xin Luo

Keywords:
Recommender system
Collaborative filtering
Attention model
Deep learning

A B S T R A C T

Neighborhood-based collaborative filtering is a method of high significance among recommender systems, with advantages of simplicity and justifiability. However, recently it is receiving less popularity due to its low prediction accuracy in contrast with model-based collaborative filtering systems, but model-based methods also suffer from a drawback worthy of attention that is they cannot effectively explain the reason behind their estimation. In order to develop a system with both high accuracy and justifiability, we propose a novel neighborhood-based collaborative filtering method inspired by the natural mechanism of attention. Our method can adaptively find neighborhood items to the prediction in user history without any pre-defined function with respect item correlations. Then the estimation are made based on these relationships. Experiments on several benchmarks are carried out to verify the performance of the proposed method, and the result shows that our method beats all previous state-of-the-art methods on MovieLens 10M and Netflix in addition to being able to justify the prediction obtained.

© 2018 Elsevier B.V. All rights reserved.

1. Introduction

Recommendation systems (RS) play an increasingly important role in modern online services. In RS, estimating the rating of the new item is an important task, where collaborative filtering (CF) is the most popular method due to their relatively good prediction performance. In CF methods, the prediction is generated from previous user-item interaction such as users historical ratings or preferences. Generally CF methods can be grouped in two classes: neighborhood-based and model-based methods.

In model-based methods, the user-item ratings are employed to learn a predictive model such as latent factorization (LF) [16,19,28] and neural networks [9,12,34], then the learned model is applied to generate an estimation without directly using the historical rating information. Thus, LF is most popular in RS. There are a number of methods proposed to improve the prediction accuracy or reduce the time cost of LF. For example, Ref. [26] incorporates second-order solvers instead traditional one-order solvers to raise prediction accuracy. The prediction accuracy also can be enhanced by ensemble LF-based methods with several diversifying and aggregating strategies [27]; studies in [24,25] improve the time effectiveness of LF.

In contrast with model-based methods, in neighborhood-based CF, the user-item ratings are directly used for recommendation relying on the relationships between users or items. Among various neighborhood-based CF methods, Item-oriented methods are more favorable due to their better scalability and accuracy [18]. In item-oriented methods, unknown rating of a user for an item is estimated from the known ratings of the user for similar items to it. These similarities of items usually are obtained by some pre-defined or heuristic approaches like cosine similarity and Pearson correlation coefficient [18].

However, recent investigations show model-based methods superior to neighborhood ones in terms of prediction accuracy [18,23,30]. The relatively poor performance of neighborhood based method is mainly due to the correlations of items or users are computed by the hand crafted or pre-defined functions [17]. Therefore, to address this drawback, more advanced and sophisticated functions are proposed to measure these correlations [4,5,36]. On the other hand, several methods are also suggested to automatically learn the weights with respect to the correlations [3,18]. Nevertheless, their performances still cannot compete with the state-of-the-art model-based methods.

Even though we want high prediction accuracy in RS, good prediction accuracy alone does not guarantee an effective and satisfying experience for users [14]. Only showing the recommendation result can make it hard for the users to determine if they can actually trust the RS [13], and model-based methods are almost a "black box" to efficiently explain their estimation. In contrast to model-based CF, one of important advantage of neighborhood-based CF is its interpretability [1]. For example, in item-oriented methods, the set of similar items to the objective item and their corresponding known ratings can be used to explain the reason of
prediction [30]. This can help the user better understand the recommendation and its relevance thus giving-off a better experience.

Recently, neural networks have been widely used in various areas [20,31,32,39]. The attention based model is a popular deep learning method which has been successfully implemented in several applications such as question answering (QA) [38], neural machine translation (NMT) [2] and speech recognition [8]. The attention based model is inspired by the nature mechanism of selective attention of human, which plays fundamental role in human visual system [15,29]. The selective attention mechanism helps to select the most pertinent piece of information, rather than using all available information.

Generally, addressing attention mechanism to the learning systems can significantly increase the accuracy of estimation. More importantly, the obtained intermediate attention score can provide fruitful interpretations for the prediction. For example, in QA, the attention can show the correlation of generated answer with the corresponding context [38]. In NMT, the attention can reveal the align of the source word and the target word [2].

Although deep learning technology had been addressed to recommender system [10,11,41], there are very limited researches on employing attention mechanism to RS. For personalized tag recommendation, Xiao et al. [42] proposed Attentional Factorization Machines (AFM) which introduces attention into Factorization Machines [33]. For multimedia recommendation, Chen et al. [7] proposed a content-based model with the attention mechanism, which makes recommendation via the values of attention of user to the items in the history of the user. In these model, the calculation of attention involves additional multimedia content information of the item such as image, video. However, this model cannot fit very well with the scenario without the additional content information.

In this paper, a novel item-oriented attention-based CF method is proposed for collaborative filtering. Our model holds the perspective of item correlation. That is it can adaptively capture the relationship between the target item and the items in user history without involving additional content information, and from these relationships rating prediction will be obtained. The captured attention score can be interpreted as the similarity of items, and these attention scores are able to reveal the reasons for the forecast. To further enhance the accuracy of attention-based CF, a particular regularization approach named “attention-dropout” is proposed, in which is a special extension of traditional dropout [37] to address the mechanism of attention and it can effectively reduce the over-fitting caused by overly exaggerated attention. Leveraging attention mechanism, the rating prediction accuracy can be significantly improved in addition to maintaining the simplicity and the justifiability of neighborhood-based CF methods.

To verify our methods, numerous experiments are conducted on three real-world benchmarks: Movielens 1M, Movielens 10M and Netflix. The experimental results show that our model is superior to similar methods in terms of RMSE, and it is competitive with the state-of-the-art model-based methods in prediction accuracy. Especially, it outperforms all previous state-of-the-art methods on Movielens 10M and Netflix. Moreover, our model also can justify the prediction is obtained.

The main contributions of the work presented in this paper are as follows.

1. We introduce a new neighborhood-based method using the mechanism of attention, which we consider to be a pure CF-based model obtained only from the ratings in contrast with the previous content-based model using attention.
2. The attention in our method is not directly implemented on the content information rather it is applied on the computations of the embeddings, which can provide better prediction accuracy as shown in the experiments. Through a number of experiments validating the performance of our method on three popular benchmarks, the experimental results show our method can exceed previous state-of-the-art methods providing additional justifiability.

The remainder of this paper is organized as follows. Firstly we introduce the details of our method in Section 2. The experimental results are reported in Section 3. Finally, we conclude this paper in Section 4.

2. Proposed method

To show the detail of our model clearly, we will gradually construct its various components in this section. Firstly, we briefly review the basic neighborhood based CF. Then, we introduce the detail of attention-based CF and attention-dropout. Finally, we show how to optimize the overall model.

2.1. Preliminaries

Given a user-item rating matrix \( R \in \mathbb{R}^{m \times n} \) with \( m \) users and \( n \) items, the element \( R_{uj} \) in \( R \) is the rating. Suppose that there is a set \( D \) which constitutes the training set, the objective of collaborative filtering is to predict unknown ratings depending on \( D \). In a typical item-oriented neighborhood based CF [30], an unknown ratings \( \hat{R}_{uj} \) of the user \( u \) and item \( i \) can be estimated from known ratings made by \( u \) on similar items, and it can be denoted as follows,

\[
\hat{R}_{uj} = \sum_{j \in N(u,i)} c_{ij} R_{uj},
\]

where \( S_i^k(u, i) \) is a set including \( k \) most similar items to \( i \) and known rating \( R_{uj} \). Here \( c_{ij} \) is the weight with respect to \( i \) and \( j \).

Usually, \( c_{ij} \) is the normalized similarity between \( i \) and \( j \), and it can be obtained as follows,

\[
c_{ij} = \frac{s_{ij}}{\sum_{j \in S_i^k(u,i)} s_{ij}},
\]

where \( s_{ij} \) is the similarity between items \( i \) and \( j \) obtained from \( D \), which usually are measured by Pearson correlation coefficient of items.

The advantages of the method is simplicity and justifiability. However, it has poor prediction performance, since the hand-crafted methods to obtain similarity are arbitrary and they cannot discover the underlying correlations of items.

To overcome the drawbacks of traditional neighborhood based CF methods, an end-to-end attention based recommender system is proposed in following section.

2.2. Attention-based collaborative filtering

To predict the rating that user \( u \) gave to an item \( i \) in our attention-based model, an estimation is produced from the weighted average of known ratings of user \( u \), and the weight for a certain known rating \( R_{uj} \) which is the value of attention between items \( i \) and \( j \). Thus, the beginning scheme of attention-based model can be denoted as:

\[
\hat{R}_{uj} = \sum_{j \in S_i^k(u,i)} a_{ij} \hat{R}_{uj},
\]

where \( S_i^k(u) \) is the set of \( k \) items in the history of user \( u \), and \( a_{ij} \) is the attention value of the historical item \( j \) in relation to the target item \( i \).

The attention value \( a_{ij} \) of item \( i \) to item \( j \) is generated from the corresponding \( l \)-dimensional embedding vectors \( e_i \) and \( e_j \), which
are initialized randomly with Gaussian distribution. The calculation of attention $a_{ij}$ can be denoted as,

$$a_{ij} = \text{Softmax}(z_{ij}),$$

yielded by

$$z_{ij} = \exp e_i^T \hat{e}_j,$$

where the embeddings $e_i$ and $\hat{e}_j$ can be learned by minimizing the difference between $r_{u,i}$ and $r_{u,j}$. Note that both $e_i$ and $\hat{e}_j$ represent the same item $i$, but $e_i$ and $\hat{e}_j$ are different embeddings where $e_i$ denotes target item and $\hat{e}_j$ denotes historical item respectively.

Formally, $\text{Softmax}$ can be denoted as follows.

$$\text{Softmax}(z_{ij}) = \frac{z_{ij}}{\sum_{k \in N^u(i)} z_{ik}}.$$ (6)

Most distinction between attention-based model with the traditional neighborhood-based methods is that the learned attention weights $a_{ij}$ can automatically discover the underlying correlations between items. Another difference lies on the set of known ratings. In the traditional method, top-$k$ similar items set $S^u(i)$ is needed to be selected first. But $N^u(u)$ can be constructed from any $k$ items without the pre-procedure of selecting similar items. The way of computing the values of attention in our method is similar to [38], and the set $N^u(u)$ can be seen as the memory of the user.

In the current scheme, the prediction is not precise enough, because there is unavoidable error between the actual and estimated ratings in the model. Calculating for the deviation $r_{u,i} - \hat{r}_{u,i}$ of the actual rating $r_{u,i}$ and the estimated rating $\hat{r}_{u,i}$ from Eq. (3), can be rewritten as

$$r_{u,i} - \hat{r}_{u,i} = \sum_{j \in N^u(u)} a_{ij} (r_{u,i} - r_{u,j}),$$

due to

$$\sum_{j \in N^u(u)} a_{ij} = 1.$$ (7)

Suppose that the weights with respect to the attention have been learned well and let $N^u(u)$ be the set containing all items whose ratings are not equal to $r_{u,i}$. Eq. (8) can be further rewritten as

$$\sum_{j \in N^u(u)} a_{ij} (r_{u,i} - r_{u,j}) = \sum_{k \in N^u(u)} a_{ik} (r_{u,i} - r_{u,k}) + \sum_{j \in N^u(u)} a_{ij} (r_{u,i} - r_{u,j}),$$

since $r_{u,i}$ must be equal to $r_{u,j}$.

Usually, $N^u(u)$ is not empty. Thus, there is unavoidable error in Eq. (3) according to Eq. (8). Especially, this error will be magnified when $a_{ij}$ is high.

To reduce the bias problem in Eq. (3), we introduce the other bias terms as follows.

Let $b^y_{i,j} = r_{u,i} - r_{u,j}$ be the bias between the actual rating of target item $i$ and the known rating of item $j$ in user $u$'s history. Obviously, there is an unbiased estimator denoted by the following equation:

$$\hat{r}_{u,i} = \sum_{j \in N^u(u)} a_{ij} (r_{u,i} + b^y_{ij}).$$ (9)

However, the real bias $b^y_{i,j}$ cannot be observed, thus an estimated bias $\tilde{b}^y_{i,j}$ is used to approximate the actual value, and the scheme of attention-based model is updated to the following equation.

$$\hat{r}_{u,i} = \sum_{j \in N^u(u)} a_{ij} (r_{u,i} + \tilde{b}^y_{ij}).$$ (10)

In Eq. (10), $\tilde{b}^y_{i,j}$ can be viewed as the composition of the biases of user part and item part as follows,

$$\tilde{b}^y_{i,j} = \hat{b}_{u,i} - \hat{b}_{u,j},$$ (11)

where $\hat{b}_{u,i}$ and $\hat{b}_{u,j}$ are the bias of user $u$ to item $j$ and the bias between items $i$ and $j$.

Intuitively, $\hat{b}_{u,i}$ can be seen as the interaction between user $u$ and item $i$, and $\hat{b}_{u,j}$ can be seen as the interaction between items $i$ and $j$. The most direct way to represent these interactions is by inner product of the respective embeddings, which are denoted in the following equation.

$$\hat{b}_{u,i} = (p_u + q_i)^T \hat{q}_j,$$ (12)

where $p_u$ is the user embedding, $q_i$ is the item embedding for target and $\hat{q}_j$ is the item embedding for history.

Consequently, the overall bias $\tilde{b}^y_{i,j}$ can be yielded by Eq. (13) according to Eq. (11) and Eq. (12) as follows.

$$\tilde{b}^y_{i,j} = (p_u + q_i)^T \hat{q}_j,$$ (13)

In addition, Eq. (10) associated with Eq. (13) can be rewritten as

$$\hat{r}_{u,i} = \sum_{j \in N^u(u)} a_{ij} r_{u,j} + \sum_{j \in N^u(u)} a_{ij} \hat{b}_{u,j} + \sum_{j \in N^u(u)} a_{ij} \hat{b}_{i,j},$$

in which $\sum_{j \in N^u(u)} a_{ij} r_{u,j}$ can be seen as the neighborhood-based baseline estimate and the last two parts are the offsets from the user and the history items respectively.

The other improvement made in our model is in accordance to the fact that some items are more noticeable than others preferentially and vice versa. To simulate this behavior, additional prior attentions $b_j$ for a certain item $j$ are placed to Eq. (5).

$$z_{ij} = e_i^T \hat{e}_j + b_j,$$ (14)

where $b_j$ also needs to be learnt.

The overall model is shown in Fig. 1 which is separated into 2 parts, attention-part and bias-part.

2.3. Attention-dropout

In practice, the attention-based CF can efficiently estimate unknown rating and keep the simplicity and justifiability, but it still manifests a problem of over-fitting which is mainly caused by the overly exaggerated attention on some items. To reduce the over-fitting with respect to attention, attention-dropout is proposed which can lead to some items are ignored completely in recommending and then avoid the item receive excessive concern for fitting the training set.

Suppose that the operation of dropout can be denoted as follows

$$y = \text{dropout}(x, p)$$

where $x$ is the input of dropout and $y$ is its output and $p$ is the dropout probability.

As mentioned before, the attentions with respect to $n$ points can be obtained by feeding corresponding features of points into Softmax.

Suppose that the inputs and outputs of softmax with respect to $n$ points are vectors $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_n]$ and $\beta = [\beta_1, \beta_2, \ldots, \beta_n]$ respectively, the value of the attention of point $i$ is $\beta_i$. Let $\hat{\alpha} =$
\[ \hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_n \] be the intermediate features of Softmax where \( \hat{\alpha}_i = \exp \alpha_i \), and then each output of Softmax \( \beta_i \) can be obtained by \( \frac{\hat{\alpha}_i}{\sum_{i=1}^{n} \hat{\alpha}_i} \). Thus, the outputs of Softmax can be used to represent an attentional distribution from each \( \beta_i \in [0, 1] \) and \( \sum_{i=1}^{n} \beta_i = 1 \).

In attention-dropout, let \( \hat{\alpha}_i = [\hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_n] \) be the features after applying dropout operation on \( \hat{\alpha} \) where each \( \hat{\alpha}_i = \text{dropout}(\hat{\alpha}_i, p) \).

Further, the vector of output of Softmax with dropout is denoted as \( \hat{\beta} = [\hat{\beta}_1, \hat{\beta}_2, \ldots, \hat{\beta}_n] \), and \( \hat{\beta}_i = \hat{\alpha}_i / \sum_{i=1}^{n} \hat{\alpha}_i \).

Obviously, attention-dropout does not break the probability characteristics of Softmax since each \( \hat{\beta}_i \in [0, 1] \) and \( \sum_{i=1}^{n} \hat{\beta}_i = 1 \).

In [40], similar method is proposed to address the issue of overfitting, which directly drops some elements in the outputs of softmax \( \beta \). However, the problem of this method is that it breaks the probability characteristics of Softmax since the sum of un-dropped elements in \( \beta \) cannot be equal to 1.

Leveraging attention-dropout, the attention of dropped points are must equal to 0 resulting in the points that are completely ignored, but the attentions of the others will be increased. Note that attention-dropout can be generally equivalent to directly drop corresponding items in \( N^u \).

### 2.4. Optimization

Let \( \theta \) be the set of learning parameters of our model including \( \epsilon, \tilde{\epsilon}, p, q, b, \tilde{q} \). Model parameters are jointly learnt by minimizing mean square error associated with Eq. (10), Eq. (13) and Eq. (14). This is minimized via AdaGrad with the adoption of L2 regularization with respect to the embeddings that helps reduce over-fitting. The Mean Square Error (MSE) is adopted as the cost function, and the cost function \( J \) can be denoted as follows.

\[
\min_{\theta} J = \sum_{(u,i) \in D} (r_{ui} - \hat{r}_{ui})^2 + L2, \tag{15}
\]

where \( L2 = \lambda (|\epsilon|_2 + |q|_2 + |p|_2 + \sum_{j \in N^u} (|\tilde{\epsilon}|_2 + |\tilde{q}|_2)) \).

To minimize the cost function \( J \), the parameters can be updated by

\[
\theta = \theta - \eta \frac{\partial J}{\partial \theta}
\]

where \( \eta \) is the learning rate.

Especially, the derivations of the cost function \( J \) considering all parameters can be calculated as (we ignore the L2 terms in here for clarity):

\[
\frac{\partial J}{\partial \epsilon_i} = \psi \sum_{j \in N^u} (r_{ui,j} - \hat{r}_{ui,j}) \left( \tilde{\epsilon}_j Z - \sum_{k \in N^u} z_{i,k} \tilde{\epsilon}_k \right)
\]

\[
\frac{\partial J}{\partial \hat{\epsilon}_j} = \psi \epsilon_j \left( \tilde{r}_{u,j} \left( \frac{z_{i,j}}{Z} - \frac{z_{i,j}^2}{Z^2} \right) + \sum_k \tilde{r}_{u,k} z_{i,k} \tilde{\epsilon}_k \right)
\]

\[
\frac{\partial J}{\partial \hat{q}_j} = \psi \left( \tilde{r}_{u,j} \left( \frac{z_{i,j}}{Z} - \frac{z_{i,j}^2}{Z^2} \right) + \sum_k \tilde{r}_{u,k} z_{i,k} \tilde{\epsilon}_k \right)
\]

\[
\frac{\partial J}{\partial \hat{q}_j} = \psi \sum_{j \in N^u} \left( \alpha_i \hat{q}_j \right)
\]

\[
\frac{\partial J}{\partial q_i} = \psi \sum_{j \in N^u} \alpha_i \hat{q}_j
\]

\[
\frac{\partial J}{\partial \tilde{q}_j} = \psi \left( \alpha_i (p_u + q_i) \right)
\]

where \( \psi = 2(r_{ui,j} - \tilde{r}_{ui,j}) \), \( z_{i,j} = \exp^\top \tilde{r}_{ui,j} \). \( Z = \sum_{k \in N^u} z_{i,k} \). \( \tilde{r} = r_{ui,j} + \tilde{r}_{ui,j} \).

#### 2.5. Computational and space complexity

The computational complexity of attention based CF per-epoch in training depends on the amount of time required to estimate...
one target rating and the amount required to compute the recommendation using this model.

During predicting target rating, there are two phases, (1) acquire the relationship scores between target item and the items in the user history; (2) obtain corresponding rating biases. The complexity of phase (1) is $O(kt)$, as we need to compute $k$ similarities, each potentially requiring $l$ operations, where $k$ is size of user history and $l$ is the size of embedding. Likewise, the complexity of phase (2) is also $O(kt)$. Thus, the overall computational complexity of predicting one target rating is $O(kt)$.

For each epoch, the upper bound on the times of making recommendation for all samples is $O(mn)$, where $m$ is the number of users and $n$ is the number of items. Consequently, the computational complexity of attention based CF is $O(mnk)$ per-epoch. Note that $k \ll n$ and $l \ll n$ since the typical values of $k$ and $l$ are hundreds.

To make recommendation via attention based CF, the lookup tables with respect to $e_\ast, \hat{e}_\ast, p_\ast, q_\ast, b_\ast, q_\ast$ are required to be stored. Specially, $e_\ast, \hat{e}_\ast, q_\ast, \hat{q}_\ast$ are the item embeddings, and the space complexity of them are $4nl$. Similarly, the space complexity of item bias $b_\ast$ and user embeddings $p_\ast$ are $O(n)$ and $O(ml)$, respectively. Consequently, the space complexity of attention based CF is $O((4n + m)t)$.

3. Experiments

In this section, numerous experiments are conducted to demonstrate the superiority of attention-based CF in terms of rating prediction and the interpretability. In addition, we show the impact of several hyper-params to the accuracy of prediction. Finally, the effectiveness of attention and attention-dropout also are presented.

3.1. Evaluation datasets

We evaluate the performance of our proposed methods on 3 real-world benchmarks MovieLens 1M and MovieLens 10M.

- **MovieLens 1M** contains around 1 million ratings of approximately 3900 movies by 6040 users in which there are 5 grade ratings and each user rated at least 20 times.
- **MovieLens 10M** contains about 10 million ratings of 10,681 movies by 71,567 users in which there are 10 grade ratings and each user also rated at least 20 times.
- **Netflix** contains about 100 million ratings of 17,770 movies by 480,189 users in which there are 5 grade ratings.

3.2. Evaluation metric

Following [6,21,22,35,43], 10% of the ratings are randomly selected as the test set, leaving the remaining 90% of the samples as the training set. Among the items in the test set, few items were not contained in training set. We used the averaged rating in training set to assign to them. To measure the performance of prediction, we have adopted the most popular metric, Root Mean Square Error (RMSE) as shown below.

$$\text{RMSE} = \sqrt{\frac{\sum_{u,i} (r_{u,i} - \hat{r}_{u,i})^2}{|S|}},$$

where $S$ is the tested set and $r_{u,i}$ is the actual rating user $u$ gave to item $i$, and $\hat{r}_{u,i}$ is the predicted rating. Lower RMSE means better accuracy. We reported the average RMSE on test set over 5 different splits.

3.3. Experiment settings

All parameters of our model were determined via cross-validation. Unless otherwise stated, use the values as follows. To construct a set of $N_k(u)$ out of the training set for user $u$, we select the first $k = 500$ items from the user history, and the rest are dropped directly. Note that a rating $r_{u,j}$ is excluded from $N_k(u)$ during minimizing the difference between $r_{u,j}$ and the estimated $\hat{r}_{u,j}$. Moreover, in case the number of recorded ratings for a certain user $u$ is lower than $k$, we will fill the missing rest with default neutral rating of 3 into $N_k(u)$ to keep its size of $k$ throughout. The learning rate is $\eta = 0.1$ which is the default value of AdaGrad. The maximum number of epochs for attention-based CF with attention-dropout in the datasets 1M, 10M and Netflix are [10,28,20]. The maximum number of epochs for attention-based CF without attention-dropout are [5,10,–]. The embedding size of all kinds of embeddings is equal to 100. We initialize all of embeddings by truncated normal distribution with 0 mean and 0.1 standard deviation. The dropout ratio is 0.5. The factor of L2 regularization for the datasets of $[10^{-3}, 10^{-6}, 10^{-7}]$, and the batch-size of $[5,25,50]$.

3.4. Rating prediction

In this section, we compare attention-based CF with other state-of-the-art model-based methods with their prediction performance expressed in term of RMSE.

3.4.1. Baseline methods

We compare our model with the following strong baseline algorithms.

- **KNNBaseline** [18]: traditional neighborhood-based method with baseline estimate. Here, the shrunk-Pearson coefficient was employed to measure the similarity between items.
- **SVD** [19]: the most popular model-based method. It obtain predictive rating from the corresponding user and target item learned embedding. In here, we reported the performance of SVD with regard to user and item biases.
- **LLORMA** [21]: LLORMA (Low-Rank Matrix Approximation) is a matrix approximation-based ensemble method which use multiple local models to describe ratings, and the unknown ratings which can be estimated by the weighted sum of the predictions from multiple local models.
- **RBM-CF** [34]: RBM (Restricted Boltzmann Machine) make recommendation via reconstructing the missing ratings from known ratings.
- **AutoRec** [35]: similar to RBM-CF, but the RBM is replaced to auto-encoder.
- **CF-NADE** [43]: it is also similar to RBM where NADE (Neural Autoregressive Distribution Estimator) is used to alternate RBM. It achieved state-of-the-art performance on Movielens 1M.
- **SMA** [22]: SMA is a low-rank matrix approximation method.
- **GLOMA** [6]: a clustering-based matrix approximation method, which is the state-of-the-art method on Movielens 10M and Netflix.
- **ACF** [7]: ACF is the most similar model to ours. In ACF, the target rating $\hat{r}_{u,j}$ can be obtained by $\hat{r}_{u,j} = (\hat{u}_u + \sum_{i \in N(u)} \hat{q}_i)^T q_j$, and the attention $\hat{a}_{u,j}$ of user $u$ on item $j$ can be obtained by a two layers neural network, $\hat{a}_{u,j} = \text{softmax}(w^T g(W_{ih}^u p_u + W_{ij}^q q_j + W_{ik} x_i + b) + c)$, where the matrices $W_{ih}$ and bias $b$ are the first layer parameters, and the vector $w$ and bias $c$ are the second layer parameters, $g(x) = \max(x, 0)$ is ReLU activation function. Here, $x_j$ is the content representation of item $j$, but these content information is unavailable from our evaluated datasets. Thus, we removed $x_j$ in the calculation of $\hat{a}_{u,j}$
in our experiments. In addition, the BPR (Bayesian Personalized Ranking) pairwise learning objective function using in [7] also was alternated to MSE objective function in our experiments since BPR is not adequate enough to train a rating prediction model effectively.

We show the detailed settings of above methods as follows.

For LLORMA, SMA and GLOMA, all parameters of them are the optimal values in the datasets as shown in their corresponding literatures. Likewise, all parameters in the RBM-like methods, such as RBM, AUTOREC and CF-NADE, are also the optimal values in the datasets. Especially, the hidden sizes of these models are equal to 500. Note that the hidden size of RBM-like method is equivalent to the embedding size of our method. However, the embedding size of our model is set to 100 for better scalability during training.

For SVD, the embedding size is also equal to 100. For neighborhood-based methods, the neighborhood size is also set to 500. For ACF, the embedding size is 100 and neighborhood size is 500, same as our model.

3.4.2. Comparison

Table 1 illustrates experimental results measured by RMSE on three datasets.

As can be seen in Table 1, the RMSE value of basic attention-based CF without attention-dropout are 0.851 and 0.780 on 1M and 10M respectively. In contrast with KNNBaseline, our model can significantly outperform it on both dataset. The results indicate that our method is superior to the traditional neighborhood-based methods measured by pre-defined function. In contrast with the most similar method ACF, the gap of RMSE between our model and ACF is extremely large. We assume the reason is the perspектив of attention. In ACF, the attention is about the interesting of user on item, which is unstable since the user’s preferences may be easy to change. The perspective of our model is on the similarity between items and the similarity is stable.

In contrast with the model-based methods, our basic method can beat some of the baseline model-based CF. When the attention-dropout is added into our model, the performance of attention-based CF can be significantly enhanced to 0.015 and 0.014 on 1M and 10M respectively. By employing attention-dropout, our method can beat the state-of-the-art model-based methods. Attention-based CF with attention-dropout can obtain 0.766 and 0.795 in Movielens 10M and Netflix, which outperforms 0.767 and 0.801 achieved by the previous state-of-the-art method GLOMA. Although the performance of our method is almost equal to GLOMA on Movielens 10M, our method exceeds GLOMA with large margin in terms of RMSE on Netflix. Note that 0.006 RMSE improvement is quite significant, compared to the 0.002 by which GLOMA improves over SMA. The results also suggest that our method has better performance on large scaled dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Movielens 1M</th>
<th>Movielens 10M</th>
<th>Netflix</th>
</tr>
</thead>
<tbody>
<tr>
<td>KNNBaseline</td>
<td>0.862</td>
<td>0.803</td>
<td>–</td>
</tr>
<tr>
<td>ACF</td>
<td>0.912</td>
<td>0.851</td>
<td>–</td>
</tr>
<tr>
<td>U-RBM</td>
<td>0.881</td>
<td>0.823</td>
<td>0.845</td>
</tr>
<tr>
<td>SVD-100</td>
<td>0.845</td>
<td>0.803</td>
<td>0.844</td>
</tr>
<tr>
<td>SVD-500</td>
<td>0.843</td>
<td>0.804</td>
<td>–</td>
</tr>
<tr>
<td>LLORMA-LOCAL</td>
<td>0.833</td>
<td>0.782</td>
<td>0.834</td>
</tr>
<tr>
<td>I-AUTOREC</td>
<td>0.831</td>
<td>0.782</td>
<td>0.823</td>
</tr>
<tr>
<td>U-CF-NADE-S (2 layers)</td>
<td>0.845</td>
<td>0.771</td>
<td>0.803</td>
</tr>
<tr>
<td>SMA</td>
<td>–</td>
<td>0.768</td>
<td>0.803</td>
</tr>
<tr>
<td>GLOMA</td>
<td>–</td>
<td>0.767</td>
<td>0.801</td>
</tr>
<tr>
<td>attention-based CF</td>
<td>0.851</td>
<td>0.780</td>
<td>–</td>
</tr>
<tr>
<td>attention-based CF + attention-dropout</td>
<td>0.836</td>
<td>0.766</td>
<td>0.795</td>
</tr>
</tbody>
</table>

On the other hand, the computational and space complexity of baseline methods are exhibited in Table 2. As shown in the table, our method requires more computation than other latent factor based methods such as SVD and GLOMA. This is due to the additional neighborhood information that is considered in our method. However, the additional computational and space consumption conduct to much better prediction performance. In contrast with other methods explicitly using neighborhood information, such as ACF and CF-NADE, the computational complexity of our method is lower since the models employ multi-layer neural networks to calculate the relationship between items. Nevertheless, our method outperforms them with large margin.

<table>
<thead>
<tr>
<th>Method</th>
<th>Computational</th>
<th>Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>KNNBaseline</td>
<td>O(mnk + n^2)</td>
<td>O(n^2)</td>
</tr>
<tr>
<td>ACF</td>
<td>O(mnkh)</td>
<td>O((2n + m)l)</td>
</tr>
<tr>
<td>U-RBM</td>
<td>O(mnh)</td>
<td>O(nh)</td>
</tr>
<tr>
<td>SVD</td>
<td>O(ml)</td>
<td>O((m + n)l)</td>
</tr>
<tr>
<td>LLORMA-LOCAL</td>
<td>O(mln)</td>
<td>O((m + n)l)</td>
</tr>
<tr>
<td>I-AUTOREC</td>
<td>O(nmh)</td>
<td>O(nh)</td>
</tr>
<tr>
<td>U-CF-NADE-S</td>
<td>O(mlkh)</td>
<td>O((m + h)l)</td>
</tr>
<tr>
<td>SMA</td>
<td>O(ml)</td>
<td>O((m + n)l)</td>
</tr>
<tr>
<td>GLOMA</td>
<td>O(mln)</td>
<td>O((m + n)l)</td>
</tr>
<tr>
<td>attention-based CF</td>
<td>O(mnk)</td>
<td>O((4n + m)l)</td>
</tr>
</tbody>
</table>

3.5. Impacts of different factors

In attention-based CF, the most important factors are the neighborhood size k and the embedding dimension l. To show the effect of these factors, in this section, we compare attention-based CF with other baseline methods under different setting of params. Note that k only relate to the neighborhood-based methods and l for model-based methods.

A relatively small dataset MovieLens 100K is adopted to evaluate the performance. The model settings for MovieLens 100K are same as the ones for Movielens 1M.

In Fig. 2 (a), we compare our methods with other item-oriented neighborhood-based methods across different values of neighborhood size k and the hybrid method SVD++(SVD+++ always uses the full neighborhood size). Here, shrunk Pearson coefficient and cosine similarity are two popular pre-defined methods to measure the correlation between items. Following [18], we also combine them with the baseline estimate which can further enhance their accuracy.
3.6. Analysis of justifiability

Our model not only predicts accurately, but it can also show the reason for the forecast. Several examples are illustrated in Fig. 3 to exhibit the prediction and top 5 highest attention movies to the objective movie.

Evidently, our model can effectively pay attention to the relative historical movies of user to estimate the rating of the objective movie. For example, “Toy Story” can gain high attention on recommending “Toy Story 2” for different users in case (1) and case (2); To recommender the movies belonging to a series such as “star wars” and “star trek”, case(3) to case (6), movies in the series will be mainly considered. In addition, the watched movies of user can be gained different amount attention to estimate different objective movie. in this contrast, case (2) and case (3) have different top-5 highest attention movies. The results show that the assignment of attention can be used to explain the reason of recommending such as “we recommend Toy Story 2 to you because you like Toy Story” in case (1). The justifiability of attention-based CF is the key advantage in contrast with other state-of-the-art model-based methods, which are totally “black-box” for explain their prediction.

3.7. Analysis of attention

In this section, we conduct experiments with the aim of answering the following questions with respect to the attention:

1. **RQ1**: Does attention-based CF can effectively pay attention to corresponding items in user history for recommending target item?
2. **RQ2**: Is our approach still effective in generating attention when the size of user history is large? What is the effect of the size of user history on prediction accuracy?
3. **RQ3**: Can higher concentration lead to more accurate predictions?

3.7.1. Effectiveness of attention (RQ1)

Although previous experiments have demonstrated the high accuracy of attention-based CF, the accuracy of prediction cannot show the effectiveness of the attention. To answer RQ1, we conduct the experiment as follows.

We consider the sum of top-k attention values averaged over the test-set (top-k average attention), these values are supposed to indicate the most significant attention scope, and the results are exhibited in Fig. 4(a).

As can be seen, the average attention for top-1 is around 0.09, and 0.6 for top-50. Note that there are 500 ratings in K500(u) with a natural average attention distribution of 0.002, however the top-1 have much higher value than the natural attention distribution, which show that our model can effectively pay attention to specific items. In addition the attention concentration of our model is not very sharp avoiding overly hight attention on limited items.

3.7.2. Effect of size of user history (RQ2)

Intuitively, it is more difficult to effectively focus on items with large number of recorded ratings.

To investigate the impact of the number of known ratings of user (excluding the “dummy ratings”), we separate all of estimated ratings into 3 groups in accordance of known ratings of user, “small” (0 to 200), “medium” (200 to 400) and “large” (above 400), consequently we show the average top-k attention of different groups in Fig. 4(b) where k is from 1 to 100.

Clearly, the top-k average attention with respect to different k values are large enough on each groups, and there are relatively small margin between them. Especially, the gaps are more narrow before k = 10 and after k = 80. The results show that, regardless of the number of items, our model can effectively generate reasonable attention value.

In addition, we investigate the correlation between the number of known ratings to the accuracy prediction. As before, all of estimated ratings are divided into 5 groups, and we report the average RMSE value on each groups in Fig. 4(c).
Fig. 3. Cases of estimations produced by attention-based CF. The cases are acquired from the test set, and each prediction of a certain user to a target movie is listed on a table, which shows the id of user, the name of target movie, the actual and estimated ratings and the top-5 highest attention movies acquired by our model with their corresponding attention.

Fig. 4. (a) Shows the sum of top-k attention values averaged over the test-set with respect to different k; (b) shows top-k attention of different groups which are divided according to the number of known ratings; (c) shows the correlation between the number of known ratings to the RMSE performance; (d) shows the effect of top-k attention on the RMSE values.
As can be seen the RMSE value decreased with an increase in the number of historical ratings of the user, this is because the correlation of items to the target item can be easier to find with relatively larger rating history.

3.7.3. Correlation between attention and accuracy (RQ3)

Here, we investigate the effect of top-k average attention on the RMSE values. Fig. 4(d) shows the averaged RMSE performances with respect to different attention ranges in top-25 cumulative attention.

As can be seen in the figure, RMSE value of 0.858 resides in relatively low-level attention range (0.0, 0.2), which means our model cannot focus well on all items. In the next 3 med-level attention ranges (0.2, 0.4), (0.4, 0.6) and (0.6, 0.8), The performances become better than the low-level attention range. In these groups the performances are relatively stable with slight change. In the high-level attention (0.8, 1.0), the RMSE drops to a staggering value of 0.7807. This results indicate that the performance is generally enhanced with the increase of top-25 cumulative attention since higher top-25 average attention usually means that the model is much more clear to which items are relevant.

3.8. Analysis of attention-dropout

3.8.1. Effects of attention-dropout

To exhibit the effect of attention-dropout, the learning and attention curves in the different variations are plotted in Fig. 5. Note that the attention-dropout is only applied on the softmax of the intermediate features to generate the attention, on the other hand the traditional dropout operations are applied on the corresponding embeddings in the attention part of our attention-based CF.

As can be seen in Fig. 5(a) and (b) for MovieLens 1M, the RMSE value gradually decreased with the increase of top-50 average attention in attention-based CF without the implementation dropout, however it will quickly start to over-fit after the 4th epoch. At the same time the top-50 average attention starts to get to a relatively high value. The rate of change at this point is also relatively steep compared to other variations of the model. On the other hand with the implementation of the different variation of dropout over-fitting generally can be greatly reduced. Addition of dropout also results in more steady attention increase. This behavior persists for both datasets as can be seen in Fig. 5(c) and (d). From this stand point, we assume the over-fitting is caused due to overly high attentions values produced shortly after few epoch runs.

In contrast the common attention-based CF, using dropout can effectively reduce over-fitting and also reduce the ratio of RMSE accompanied by a more gradual increase in the accumulation of top-k attention values. when it comes to comparison between attention and traditional dropout, relatively large RMSE-gaps are noticed at the end Fig. 5(a) and (b) which are around 0.009 and 0.016 for 1M and 10M datasets respectively. These results show attention-dropout has much lower RMSE value than the traditional-dropout in both dataset. We assume the reason for this is attention-dropout can directly shut down the attention on some items, but traditional dropout cannot do that they can only alter the attention distribution on items.

3.8.2. Effect of attention-dropout ratio

To investigate the effect of attention-dropout ratio, we placed painted the prediction results on MovieLens 1M acquired by our method with different dropout ratio settings in Fig. 6. In the figure, the ratio 0.1 indicates that only 10% historical items are kept for training. On the other hand, 1.0 setting actually leads to the attention-dropout to be turned off, and thus nothing will be discarded. Clearly, this figure shows that lowest dropout ratios cannot result in promising performance which even cannot
exceed the results obtained by the method without the implementation dropout, since the 0.1 setting bring about most of the information to be ignored. However, in the range of [0.2,0.9], corresponding results can surpass the one without dropout. Especially, middle dropout settings in the range of [0.3,0.5], corresponding performances can be optimal.

3.9. Effect of L2

In Fig. 7, the performance curves of different settings with respect to regularization are exhibited. Here, the red curve correlates to the model without L2 and Dropout regularization approaches, the blue one relates to the model with only L2 regularization, the yellow one portrays the model using only Dropout, and the purple one adopts both L2 and Dropout. As shown in the figure, the L2 regularization can consistently improve the performance regardless of whether the dropout is used. We also observed that attention-dropout can bring greater performance improvement than L2 as painted with the blue and yellow curves.

3.10. Effect of optimization approaches

Because there are a large number of parameters needed to be learned in our method, the choice of optimization approaches is a key of generating the model with relatively high prediction accuracy. Fig. 8 shows the performance curves of our method optimized by traditional stochastic gradient descent (SGD) and AdaGrad on the test set of MovieLens 1M. As shown in Fig. 8, the prediction performance obtained via AdaGrad outperforms the SGD with large margin. We assume the superior optimization of AdaGrad due to its adaptive learning rate in relation to the frequency of parameters. More specifically, in our situation, the frequency of items are different, and AdaGrad relieves the embeddings of popular items from being overtrained.

3.11. Time complexity

Our method is implemented by Tensorflow with one GPU (Titan GTX). Table 3 shows the running time of one epoch of attention-based CF with attention dropout. Training time in our model is relatively large due to the relatively small batch size that is used in training. We observe that using small batch size benefits better prediction performance in our experiments. However, during inference, the running time is negligible.

4. Conclusion

In this paper we proposed a new CF method using the mechanism of attention and a special extension of dropout. Empirical evaluations on large real-world benchmarks demonstrate that our model has high prediction accuracy in terms of RMSE, and it can beat all previous state-of-the-art methods on MovieLens 10M and Netflix. In addition, the intermediate values of attention also can be used to reveal the reason behind prediction, which is the significant advantage in contrast with other state-of-the-art methods. Besides, the experiment results also show that using attention-dropout can achieve much lower RMSE value than the traditional dropout, and we believe that attention dropout can also be applied on other instances of attention mechanism.

Acknowledgment

This work was supported in part by the National Science Foundation of China under Grant 61573081, and in part by the foundation for Youth Science and Technology Innovation Research Team of Sichuan Province under Grant 2016TD0018.

References


Mingsheng Fu is a current Ph.D. student in Department of computer science and engineering, University of Electronic Science and Technology of China. His current research interests relate to Neural Networks, Machine Learning, and recommender system.

Hong Qu received the Ph.D. degree in computer science from the University of Electronic Science and Technology of China, Chengdu, in 2006. From 2007 to 2008, he was a Postdoctoral Fellow at the Advanced Robotics and Intelligent Systems Lab, School of Engineering, University of Guelph, Guelph, ON, Canada. From 2014 to 2015, he work as a visiting scholar in the Potsdam Institute for Climate Impact Research (PIK), 14473 Potsdam, Germany, and in Humboldt University of Berlin. Currently, he is a professor in Computational Intelligence Laboratory, School of Computer Science and Engineering, University of Electronic Science and Technology of China. His research interests include Neural Networks, Machine Learning and Artificial General Intelligence.

Dagmawi Moges is a current Ph.D. student in Department of computer science and engineering, University of Electronic Science and Technology of China. From 2014 to 2015, he was working as lead engineer on AI and robotics at iCog in Addis Ababa. And from 2015–2016 at OpenCog, Hong Kong. His current research interests include to Neural Networks, Machine Learning and Artificial General Intelligence.

Li Lu received the Ph.D. degree from the Key Lab of Information Security, Chinese Academy of Science, in 2007. He is a Professor with the School of Computer Science and Engineering, University of Electronic Science and Technology of China. His research interests include RFID technology and system, wireless network and network security.