Expert Systems With Applications 186 (2021) 115775

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Biased autoencoder for collaborative filtering with temporal signals

Runliang Dou, Oguzhan Arslan, Ce Zhang *,[1]

College of Management and Economics, Tianjin University, Tianjin, China

ARTICLE INFO

Keywords:
Collaborative filtering
AutoRec
Temporal dynamics
Bias
Autoencoder

ABSTRACT

Recommendation systems are used in various types of online platforms and in e-commerce. Collaborative filtering (CF) is one of the most popular approaches for recommendation systems and has been widely studied in academia. In recent years, several models based on neural networks that can discover nonlinear relationships have been proposed and compared to traditional CF models. The results showed that they performed better in terms of their prediction accuracy. However, these models do not consider user bias and item bias together, and they do not include temporal signals. This paper proposes a biased autoencoder model (Biased AutoRec) for CF, which is built on the well-known AutoRec CF approach. Several approaches are also proposed to integrate temporal signals into the Biased AutoRec model to merge the power of nonlinearity and temporal signals. Experiments on several public datasets showed that the new models outperformed the AutoRec model, which outperformed the prediction accuracy of previous state-of-the-art CF models (i.e., biased matrix factorization, RBM-CF, LLORMA).

1. Introduction

Social platforms such as YouTube and Netflix, and online marketplaces such as Amazon, use recommendation systems to improve user engagement and increase conversation rates. Because of their practical benefits, recommendation systems have been extensively studied and have gained significant popularity in recent years. Recommendation systems can be content-based, collaborative filtering (CF)-based, or hybrid models. Along with other techniques, CF has been widely studied because of its outstanding performance. The underlying principle of CF is that if two users showed similar preferences in past circumstances, their preferences will be more similar to those of random people in future circumstances. Therefore, CF aims to exploit past information about user preferences to suggest better personalized recommendations. In particular, CF is independent of context; therefore, there is no domain knowledge required (Koren, 2010).

Popular choices among CF models are matrix factorization, neighborhood models, and their extensions. Although these CF models provide successful results, they are inadequate for exploring the nonlinear similarity of users and items. On the contrary, deep learning algorithms utilize nonlinear functions and provide more detailed representations of the similarity of users and items. The outstanding performance of deep learning has been proven empirically in different domains such as natural language processing (NLP) (Tsubaki et al., 2016) and image classification (Szegedy et al., 2015). Moreover, deep learning methods can also be used for all-purpose parameter fitting through advanced stochastic gradient descent algorithms, while achieving speed and scalability using graphical processing units (GPUs) (R et al., 2020). Therefore, in recent years, deep learning-based CF models, including recurrent neural networks and AutoRec, have gained considerable popularity. However, although such models have provided better results than traditional CF methods, they are still in their infancy, with room for more advanced models to be developed.

An autoencoder is a type of artificial neural network in which the input layer is reconstructed after the encoding and decoding phases. Autoencoders are mainly used for image recognition, computer vision, clearing noise from images, and text processing. AutoRec is a CF model based on an autoencoder, and its efficiency and effectiveness have already been proven in empirical tests. Because of the activation wrappers around the encoder and decoder matrices, AutoRec can explore the nonlinear similarity between users and items. This paper proposes an extension of the AutoRec model, called Biased AutoRec. Empirical tests conducted on several public datasets proved that Biased AutoRec outperformed AutoRec.

On the contrary, temporal dynamics are extremely important for determining users' attitudes toward rating and reviewing items. Users'

* Corresponding author at: College of Management and Economics, Tianjin University, Tianjin 300072, China.
E-mail addresses: drl@tju.edu.cn (R. Dou), 6219000040@tju.edu.cn (O. Arslan), tjuzc@tju.edu.cn (C. Zhang).
[1] https://orcid.org/0000-0001-7190-1058

https://doi.org/10.1016/j.eswa.2021.115775
Received 4 June 2021; Received in revised form 11 August 2021; Accepted 13 August 2021
Available online 21 August 2021
0957-4174/

preferences may change over time. While a user enjoys a genre for movies during a certain period, they may not enjoy the same genre in another period. Similarly, their attraction to and preference for items may also change over time (Wu et al., 2017). A product may gain popularity after information related to the product emerges. For example, a movie's overall rating may increase after the movie wins movie awards or a food product that is not preferred because of its taste may start to attract people and its overall rating may start to increase after promotions mentioning the benefits of the product. Moreover, users' emotions play an important role in reviewing and rating items. A person has a greater tendency to leave positive ratings when happy and has a greater tendency to leave negative ratings when unhappy. Therefore, it is necessary to integrate time information into CF to provide effective personalized recommendations (Zheng & Li, 2011).

To discover such temporal effects in CF, three extensions were proposed. These were built on top of the Biased AutoRec model and contained different temporal biases. Experiments showed that the accuracy of the proposed models was better than those of AutoRec and other state-of-the-art methods. The contributions of this study can be summarized as follows.

- The proposed Biased AutoRec algorithm strengthens the prediction accuracy of AutoRec, which is a popular CF model based on the autoencoder. Similar to AutoRec, the model can discover the nonlinear similarity between users and items thanks to the activation functions wrapping decoder and encoder matrices. On the other hand, the model includes user bias and item bias together first time among autoencoder based CF models. Therefore, the proposed model not only increases accuracy but also provides a better representational advantage.
- The three proposed models built on Biased AutoRec contained different temporal biases, including periodical biases, the periodical biases of items, and the daily biases of users. In these models, the goal was to learn the user and item biases corresponding to certain time periods.
- The efficient and effective models proposed in this paper could be implemented in practical use to provide better item recommendations to users. Moreover, the experiments demonstrated empirically that the three proposed models with temporal biases outperformed the Biased AutoRec and AutoRec models in most scenarios.

The remainder of this paper is organized as follows. Section 2 discusses the existing research and describes the traditional matrix factorization models and autoencoder-based collaborative filtering models. Then, section 3 describes the improved autoencoder-based recommendation model, Biased AutoRec, along with its temporal variations. The methodology used to stack these variations is also mentioned in this section. Section 4 reports and compares the results of tests of the existing and proposed models using several public datasets. The results of the various stacked models are also mentioned in this section. Section 5 discusses the results of the research, as well as future work.

2. Related works

This study was related to collaborative filtering and temporal dynamics. The following sections briefly explain the works related to this research.

2.1. Collaborative filtering

CF is one of the most commonly employed techniques for rating predictions. It has been extensively studied in academia and is widely used in real-world applications. The nearest neighborhood algorithm (kNN) and matrix factorization (MF) are the two most popular methods for CF. In empirical tests, MF models are superior to neighborhood models (Koren et al., 2009). MF takes the user-item rating table into low-dimension user and item latent spaces. These latent spaces are learned by utilizing the observed ratings obtained by the stochastic gradient descent (SGD) or alternating least squares (ALS) method (Koren et al., 2009). The unknown ratings can be calculated by multiplying the latent vectors of the user and item. There is a large variety of MF implementations, including probabilistic matrix factorization (PMF) (Salakhutdinov & Mnih, 2009), SVD++ (Koren, 2008), timeSVD++ (Koren, 2010), trustSVD (Guo et al., 2016), and timetrustSVD (Tong et al., 2019). To alleviate the data sparsity problem, several models utilize rich user and item factors, i.e., the review text content is taken into consideration (Wang et al., 2016).

Deep learning has been proven to be efficient and effective in several domains, including computer vision, image recognition, natural language processing, and reinforcement learning. Its performance has attracted researchers, and it has been used in several studies on CF. One of the earliest models to use deep learning for CF was the restricted Boltzmann machine (RBM) (Salakhutdinov et al., 2007). Subsequently, several neural network models have been proposed, including AutoRec (Sedhain et al., 2015), recurrent recommender networks (RRN) (Wu et al., 2017), and neural collaborative filtering (NCF) (He et al., 2017).

Autoencoders are based on neural networks and are relatively new to CF, providing competitive results in rating prediction. The main advantage of autoencoders is their ability to discover the nonlinear relationships between users and items. The first framework proposed to use autoencoders for CF was AutoRec (Sedhain et al., 2015). Later, several other CF models based on autoencoders were proposed, including those of Yi et al. (2017) and Zhang et al. (2017). An AutoRec model can be item-based or user-based. Empirical tests have shown that an item-based AutoRec model provides more promising results than a user-based AutoRec model (Sedhain et al., 2015).

2.1.1. AutoRec model

In an item-based autoencoder model, each user $u \in U = \{1 \cdots m\}$ is represented by a partially observed vector, $\boldsymbol{r}^{(u)} = (R_{u1}, \cdots R_{un}) \in \mathbb{R}^n$, and each item is represented by a partially observed vector, $\boldsymbol{r}^{(i)} = (R_{1i}, \cdots R_{mi}) \in \mathbb{R}^m$. The model iterates over each user and tries to learn a common latent space (hidden layer) that contains dense representation knowledge of the input layer. The number of elements in the hidden layer is represented by k, and is determined by cross-validation. According to the empirical tests, a larger k yields better prediction accuracy, but also increases the computational effort. $\boldsymbol{W} \in \mathbb{R}^{d \times k}$ is a matrix that converts the input vector into a hidden layer, and $\boldsymbol{V} \in \mathbb{R}^{k \times d}$ is a matrix (in item-based AutoRec, d is equal to m; in user-based AutoRec, d is equal to n) that converts the hidden layer back to the output layer, which has the same number of elements as the input layer. In the model, the transformations of these two steps are surrounded by activation functions such as sigmoid and ReLu, which play the main role in empowering the model to discover nonlinear relationships between the users and items. In addition, $\theta = \{\boldsymbol{W}, \boldsymbol{V}, \boldsymbol{\mu}, \boldsymbol{b}\}$ is a cluster of parameter sets that represents all the parameters needed to train the model. Here, $\boldsymbol{\mu}$ is the set of parameters that represent the bias for the hidden layer, and $\boldsymbol{b}$ is the set of parameters that represent the bias for the output layer. The item-based AutoRec (Fig. 1) objective function can be written as follows:

$$\min_{\theta} \sum_{i=1}^{n} \left\| r^{(i)} - h\left(r^{(i)}; \theta\right) \right\|_O^2 + \frac{\lambda}{2} \cdot \left(\|W\|_F^2 + \|V\|_F^2 \right)$$

which is subject to

$$h(\boldsymbol{r}; \theta) = f(\boldsymbol{W} \cdot g(\boldsymbol{V}\boldsymbol{r} + \boldsymbol{\mu}) + \boldsymbol{b})$$

In the formula above, $h(\boldsymbol{r}; \theta)$ denotes the reconstruction of input $\boldsymbol{r}$ by considering θ values; and f and g are the activation wrappers such as sigmoid and ReLu. In addition, $\|.\|_{\mathscr{O}}^2$ denotes that only the observed ratings should be considered during training. The same set of θ is trained for $i = 1,2,\ldots,n$ and attempts to minimize the difference between the

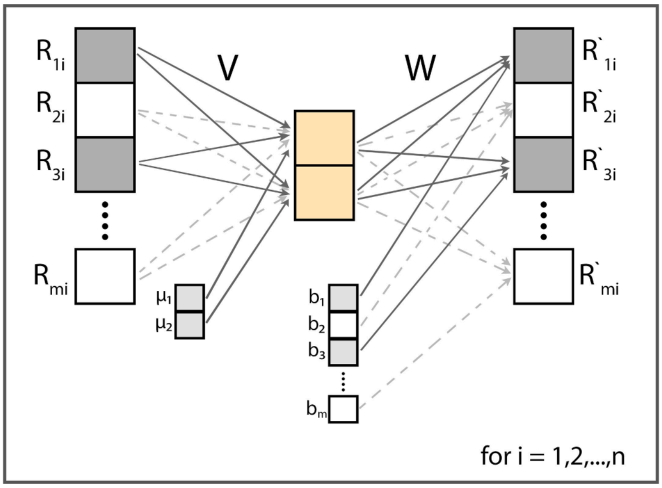

Fig. 1. AutoRec model.

observed ratings in the input layer and their predictions in the output layer. Notably, there are n copies of the neural network, and parameter set θ is tied across each neural network, where θ is learned by back-propagation (Sedhain et al., 2015). After the training is completed, the determined set of θ is used to predict unknown user-item rating pairs.

2.2. Temporal dynamics

Researchers have tried to implement temporal effects in CF models since the beginning of CF. One of the first studies to emphasize the significance of temporal signals in recommendation systems involved collaborative filtering with temporal dynamics (Koren, 2010). This research proposed the algorithm timeSVD++, and temporal aspects were discussed at great length (Koren, 2010). In another study based on matrix factorization, time effects were combined with trust factors (Tong et al., 2019). Xu et al. (2017) proposed a model that considered time information and exploited the temporal interactions among review texts and co-clusters of user communities and item groups. The TimeMF model was proposed by Li et al. (2019). Lo et al. (2018) proposed a CF model that was specifically designed to track the concept drift in individual user preferences. Moreover, some studies have been based on the time sensitivity that than on time itself (Cheng & Wang, 2020). Wu et al. (2018) proposed a novel method based on matrix factorization that combined the dynamic user factor of TimeSVD++ with the hidden topic of each review text mined by the topic model of TopicMF.

A few attempts have been made to implement temporal dynamics in neural network-based CF models. Wu et al. (2017) used long short-term memory (LSTM) recurrent neural networks to capture the temporal dependencies for both users and movies. Saurav et al. (2018) used a temporal model based on recurrent neural networks (RNNs) to detect anomalies in time-series data.

2.2.1. timeSVD++: Biased MF with temporal dynamics

Matrix factorization maps each item and user with a low-dimension vector of latent space, in which the ratings are the inner products of the latent vectors of users and items.

$$\widehat{r}_{ui} = q_i^T p_u$$

To learn the latent space, p_u and q_i, it is necessary to optimize the object function:

$$\min_{q_*, p_*} \sum_{(u,i,t)\in K} \left(r_{ui} - q_i^T p_u\right)^2 + \lambda\left(\| q_i\|^2 + \| p_u\|^2\right)$$

where λ denotes the regularization parameter. In the biased matrix factorization model, the prediction can be expressed as follows:

$$\widehat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u$$

In the formula above, μ denotes the average value of all the ratings, b_u denotes the bias of user u, and b_i denotes the bias of item i. In each iteration phase, the algorithm arranges these biases to minimize the objective function of the model. Therefore, user and item biases help the latent space to learn a better relationship between users and items. Thus, including user and item biases improves the performance of matrix factorization models in most scenarios.

User preferences and item attractiveness change over time. While a user may like a movie or a product at a certain time, they may dislike it at another time. Similarly, an item may be liked by most of the population during a specific period, but users' perceptions may change, and the item may become a disliked item by most of the users. Moreover, a user's emotions during a review and rating are crucial for the rating value. For example, a person might be promoted on their job, which may make them feel extremely positive. This could cause them to rate every movie with 5 stars within that period. In contrast, they might lose their job and rate similar movies with 1 star. Empirical tests have shown that the difference between the values of ratings on the same day by a user is lower than the difference between the values of the user's overall ratings. Based on the timeSVD++ algorithm, the user and item biases with temporal dynamics are written as follows:

$$b_{ui}(t) = \mu + b_u + \alpha_u \cdot \mathrm{dev}_u(t) + b_{u,t} + \left(b_i + b_{i,\mathrm{Bin}(t)}\right) \cdot c_u(t)$$

In the above formula, $\alpha_u \cdot \mathrm{dev}_u(t)$ denotes the linear change in a user's preferences over time; $b_{u,t}$ indicates a user's bias for a certain product on a certain day; and $b_{i,\mathrm{Bin}(t)}$ denotes an item's bias in a certain period bin. In the formula above, another parameter, $c_u(t)$, is assigned to personalize the item biases for users.

The studies mentioned above involved either 1) advanced models developed on traditional CF models that did not involve any nonlinearity or 2) neural network-based approaches that involved nonlinearity but did not completely or partially consider temporal signals, as detailed in the previous approaches. However, this study considered not only the Biased AutoRec algorithm but also several approaches and a methodology to integrate temporal signals into neural network-based models, similar to the previous algorithms, to empower predictions by combining the power of nonlinear similarity and temporal signals in CF.

3. Proposed models

The following section proposes an algorithm called Biased AutoRec, which is a bias-added version of the AutoRec model. Then, three models are also proposed, which were built based on Biased AutoRec in order to experiment with temporal biases and autoencoders for CF.

3.1. Biased autoencoder model

As mentioned earlier, the parameter component of the AutoRec model is $\theta = \{\boldsymbol{W}, \boldsymbol{V}, \boldsymbol{\mu}, \boldsymbol{b}\}$, which consists of $\boldsymbol{W}$ (decoder matrix), $\boldsymbol{V}$ (encoder matrix), $\boldsymbol{\mu}$ (bias set for hidden layer), and $\boldsymbol{b}$ (bias set for output layer). Here, $\boldsymbol{\mu}$ has the same length as the number of hidden layer units, and $\boldsymbol{b}$ has the same length as vector $\boldsymbol{r}^{(i)}$. Each parameter in $\boldsymbol{b}$ represents the bias for the corresponding parameter in $\boldsymbol{r}^{(i)}$, which contains ratings from m users on item i. Similarly, each parameter in $\boldsymbol{\mu}$ represents the bias for the corresponding parameter in the hidden layer, which is a dense representation of $\boldsymbol{r}^{(i)}$. In particular, bias set $\boldsymbol{b}$ can be considered as a bias set for users, and $\boldsymbol{\mu}$ can be considered as a bias set for the hidden layer and dense representation of the users. Similarly, in the user-based AutoRec model, $\boldsymbol{b}$ represents the bias set for items, and $\boldsymbol{\mu}$ represents the bias set for the dense representation of the items. Therefore, by continuing with item-based AutoRec, it can safely be claimed that the model only considers user biases and misses the implementation of item biases.

Here, the Biased AutoRec model includes user biases and item biases together in any version of the autoencoder, either I-AutoRec or U-AutoRec. The new model adds $\boldsymbol{q}$ and $\boldsymbol{c}$ sets to the model parameters, which become $\theta = \{\boldsymbol{W}, \boldsymbol{V}, \boldsymbol{\mu}, \boldsymbol{b}, \boldsymbol{q}, \boldsymbol{c}\}$. Here, $\boldsymbol{q}$ and $\boldsymbol{c}$ represent the item biases that are added to the hidden and output layers, respectively. To clarify this, the methodology is shown in the figure above (Fig. 2).

Although μ and $\boldsymbol{b}$ are parameter sets with the lengths of the hidden and output layers, respectively, $\boldsymbol{q}$ and $\boldsymbol{c}$ both have the same length, $\boldsymbol{n}$. In the Biased AutoRec model, in addition to AutoRec, the corresponding parameters from the $\boldsymbol{q}$ and $\boldsymbol{c}$ parameter sets were added to the model. The objective function of Biased AutoRec can be written as follows:

$$\min_{\theta} \sum_{i=1}^{n} \left\| r^{(i)} - h\left(r^{(i)};\theta\right) \right\|_{O}^{2} + \frac{\lambda}{2}\cdot\left(\|W\|_F^2 + \|V\|_F^2\right)$$

which is subject to

$$h(r;\theta) = f(W\cdot g(Vr + \mu \ + \ q^{*}) + b + c^{*})$$

In the above formula, $\boldsymbol{q}^{*}$ represents the $\boldsymbol{q}_i$ value. Similarly, $\boldsymbol{c}^{*}$ represents the $\boldsymbol{c}_i$ value for the corresponding $\boldsymbol{r}^{(i)}$ vector for $i = 1, 2, \ldots, n$. In the above formulas, $\boldsymbol{q}_i$ and $\boldsymbol{c}_i$ are added to each element of the hidden layer and the observed elements of $\boldsymbol{r}^{(i)}$, respectively. The new parameters $\boldsymbol{q}$ and $\boldsymbol{c}$ help the model to discover item biases and increase the prediction performance over the normal AutoRec model.

The model parameter set θ is learned through backpropagation similar to AutoRec, and the learned θ is used to predict unknown user-item rating pairs.

3.2. Biased AutoRec model with temporal dynamics

The core principle behind temporal dynamics is that a user's preferences, popularity, and attractiveness may change over time. For example, users' attitudes toward wired earphones may change after the wireless earphones gain popularity and may start to be disliked more often by customers because of inconvenience. Similarly, an animated film may initially be liked because of its great effects, but then, after users have watched the film, they may find that the effects were not good enough and may dislike it. These examples show the importance of considering temporal dynamics when predicting user-item ratings.

The effectiveness of temporal dynamics has been proven in matrix factorization models for CF (Koren, 2010). One of the main objectives of this research was to test the temporal dynamics of the AutoRec model, which already outperforms matrix factorization models without containing any temporal factors. This research implemented several parts of temporal dynamics in Biased AutoRec, which is an autoencoder-based collaborative filtering model.

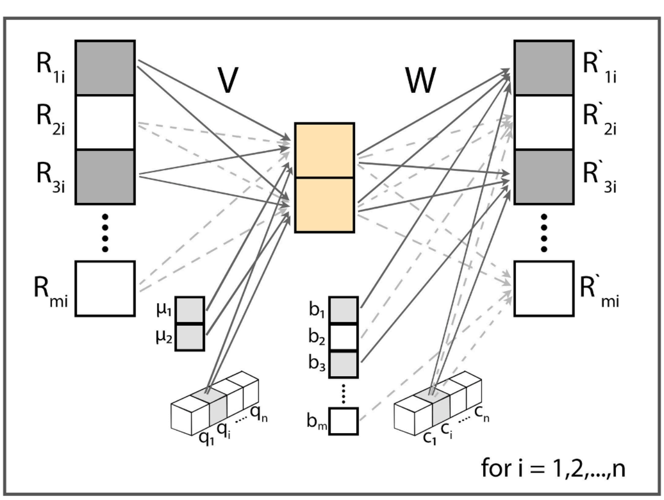

Fig. 2. Biased AutoRec model.

The core structures of the neural network CF models and matrix factorization models are different. While matrix factorization models are linear, AutoRec is a nonlinear model because of its activation functions (i.e., sigmoid and ReLu). Although adding many temporal signals to MF jointly provides outstanding performance in terms of accuracy, it was observed that implementing many temporal signals simultaneously in the Biased AutoRec model did not provide promising results in empirical tests. This may have stemmed from the nonlinearity of the Biased AutoRec model. Subsequently, several temporal signals were implemented independently of the Biased AutoRec model, and an increase in the accuracy of the predictions was observed over the plain Biased AutoRec model in most of the empirical tests.

The following section proposes three models that were built on Biased AutoRec and contained independent temporal signals, including the period biases, period biases of items, and daily biases of users.

3.2.1. Biased autoencoder model with periodical bias

Including item biases with user biases in Biased AutoRec model resulted in an increase in the accuracy of the predictions. Therefore, it was decided to test several temporal biases in the Biased AutoRec model. The first temporal signal considered for the model was period bias, which is not dependent on items or users. Period biases were added to the model with the goal of capturing the overall biases for each period and enabling the model to discover the relationships between users and items more sensitively. To achieve this task, the timestamps included in the datasets were utilized.

The timestamps were first converted to dates in the training and test data. Next, the dates in the test data that were earlier or later than the earliest and latest dates in the training set were clipped. Then, a new parameter set, $\boldsymbol{p}$, was added to the existing model parameters. The new model parameters were $\theta = \{\boldsymbol{W}, \boldsymbol{V}, \boldsymbol{\mu}, \boldsymbol{b}, \boldsymbol{q}, \boldsymbol{c}, \boldsymbol{p}\}$. The parameter set $\boldsymbol{p}$ had a length of k, and each parameter in $\boldsymbol{p}$ represented the bias for the corresponding period bin. For example, if there were 1000 days between the earliest and latest dates in the rating records of the data, and if a value of 40 was selected for k, the period bias set contained 40 parameters, each of which represented the bias for the corresponding 25-day period. The period bias was added to the observed user-item ratings in the output layer for $\boldsymbol{i} = 1,2,\ldots,\boldsymbol{n}$, by utilizing the rating's corresponding period bin. For example, R_{14} denoted that the rating belonged to the fourth user on the first item andR_{25} denoted that the rating belonged to the fifth user on the second item, both for the seventh-period bin. In this case, the same parameter from period bias set $\boldsymbol{p}$ was added to these two ratings, which belonged to different users and were in different output layers.

In this model, the new objective function becomes as follows:

$$\min_{\theta} \sum_{i=1}^{n} \left\| r^{(i)} - h\left(r^{(i)};\theta\right) \right\|_{O}^{2} + \frac{\lambda}{2}\cdot\left(\|W\|_F^2 + \|V\|_F^2\right)$$

which is subject to

$$h(\boldsymbol{r};\theta) = f(\boldsymbol{W}\cdot g(\boldsymbol{V}\boldsymbol{r} + \boldsymbol{\mu} + \boldsymbol{q}^{*}) + \boldsymbol{b} + \boldsymbol{c}^{*} + \boldsymbol{p}^{*})$$

In the above formula, $\boldsymbol{p}^*$ represents a dynamic vector, which has a length of m and is composed of the parameters from period bias set $\boldsymbol{p}$, according to the transaction date of the elements in the $\boldsymbol{r}^{(i)}$ vector. The $\boldsymbol{p}^*$ vector is dynamically reconstructed for $i = 1,2,\ldots,\boldsymbol{n}$, and is added to the observed elements of $\boldsymbol{r}^{(i)}$. Similar to the previous models, the parameter set θ is learned through backpropagation, and the learned θ is used to predict unknown user-item rating pairs. The methodology for implementing period bias set $\boldsymbol{p}$ in the model is shown in the figure above (Fig. 3). Notably, Biased AutoRec with period bias provided better accuracy in the majority of the empirical tests that were conducted compared to Biased AutoRec.

Attempts were also made to implement period bias in the hidden layer in several ways. Implementing period bias in the hidden layer is a

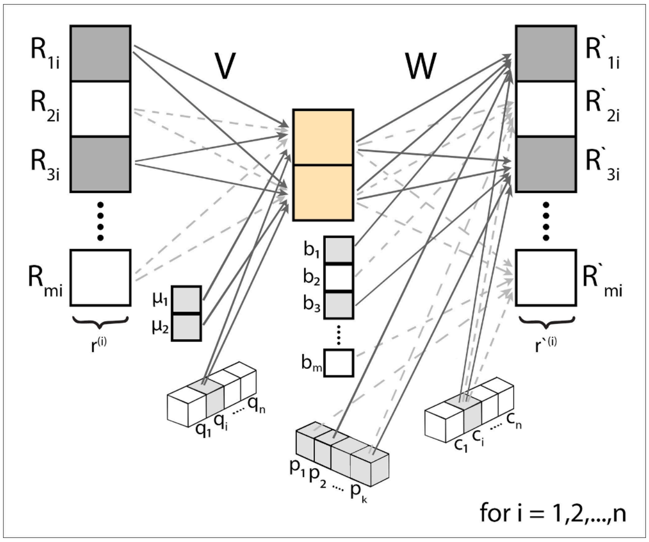

Fig. 3. Biased Autoencoder model with periodical bias.

relatively difficult task because it is a dense representation of the output layer, and the condition in which several input nodes are tied to each node in the hidden layer makes it impossible to assign a single bias parameter to each node in the hidden layer. In the first method, an attempt was made to add multiple period biases to the nodes in the hidden layer by utilizing the input layer element's period bins; however, adding several period bins jointly to the dense hidden layer nodes did not improve the prediction accuracy as expected.

Another method for implementing temporal biases in the hidden layer was tested. In the new model, in addition to the hidden layer, which was created by the multiplication of $\boldsymbol{r}^{(i)}$ and encoder matrix $\boldsymbol{V}$, a second hidden layer was created, which consisted of k nodes. While the first hidden layer acted the same as in previous models, the second hidden layer was created conditionally and had a different structure. In the second hidden layer, a unique node was assigned to each period bin. Then, input nodes with the same period bins were tied to the corresponding nodes. Finally, while the first hidden layer was tied to all of the observed ratings in the input and output layers, the new hidden layer was conditionally tied to the corresponding nodes in the input and output layers by considering their period bins. In this model, the goal was to capture the temporal representation of the ratings in each node in the second hidden layer, while capturing the overall relationship through the first hidden layer. However, creating two hidden layers, devoting one of them to period bins, and tying it to the observed nodes in the input and output layers did not yield the expected results.

3.2.2. *Biased autoencoder model with periodical biases for items*

In various studies, users have been considered dynamic, whereas the items were considered static because of their nature. Therefore, item popularity and attractiveness change over relatively long periods rather than quickly. This principle was used to add period biases for items to the model. The new bias parameter set is called $\boldsymbol{L}$, and after adding it to the model, the new model parameters become $\theta = \{\boldsymbol{W}, \boldsymbol{V}, \boldsymbol{\mu}, \boldsymbol{b}, \boldsymbol{q}, \boldsymbol{c}, \boldsymbol{L}\}$. Item period bias set $\boldsymbol{L}$ has the dimensions of $k \times n$, and contains a separate period bias set for each item. For period binning, the same methodology was used as described in the previous section.

The objective function of this model can be written as follows:

$$\min_{\theta} \sum_{i=1}^{n} \left\| r^{(i)} - h\left(r^{(i)}; \theta\right) \right\|_O^2 + \frac{\lambda}{2} \cdot \left(\|W\|_F^2 + \|V\|_F^2 \right)$$

which is subject to

$$h(\boldsymbol{r}; \theta) = f(\boldsymbol{W} \cdot g(\boldsymbol{V}\boldsymbol{r} + \boldsymbol{\mu} + \boldsymbol{q}^*) + \boldsymbol{b} + \boldsymbol{c}^* + \boldsymbol{l}^*)$$

In the above formula, $\boldsymbol{l}^*$ represents a dynamic vector, which has a length of m and is composed of the parameters from item period bias set $\boldsymbol{L}$, according to item i and the corresponding period bins of the nodes. Then, dynamic vector $\boldsymbol{l}^*$ is added to the $\boldsymbol{r}^{(i)}$ vector for $i = 1, 2, \ldots, n$. For example, if R_{14} and R_{25} both have the seventh-period bin, different parameters from item period bias set $\boldsymbol{L}$ are added to these two ratings because, although they are in the same period bin, they belong to two different items.

Similar to the previous models, parameter set θ is learned through backpropagation, and the learned θ is used to predict unknown user-item rating pairs. The methodology for implementing item period bias set $\boldsymbol{L}$ can be seen in the figure above (Fig. 4). Notably, similar to Biased AutoRec with period bias, Biased AutoRec with period biases for items also provided better accuracy in most of the empirical tests.

3.2.3. *Biased autoencoder model with daily biases for users*

When users leave reviews of items, their rating values are affected by their emotions. Based on empirical tests, the variance of a user's ratings left on the same day is lower than the variance of the user's overall ratings. Therefore, the daily biases of users were added to Biased AutoRec with the goal of capturing the changes in user biases on a certain day. For example, when a user is happy, they may enjoy all the movies they watch and give them relatively higher ratings. However, if a user is depressed or experiencing problems, this may result in bad ratings for all the movies they rate. To capture the daily biases of users, a new bias set, called $\boldsymbol{D}$, was added to the model. Then, the new model parameters became $\theta = \{\boldsymbol{W}, \boldsymbol{V}, \boldsymbol{\mu}, \boldsymbol{b}, \boldsymbol{q}, \boldsymbol{c}, \boldsymbol{D}\}$. User daily bias set $\boldsymbol{D}$ had the dimensions of $t \times m$, where t denoted the number of days between the earliest and latest ratings in the training set. $\boldsymbol{D}$ contained a separate daily bias set for each user.

The objective function of the model can be written as follows:

$$\min_{\theta} \sum_{i=1}^{n} \left\| r^{(i)} - h\left(r^{(i)}; \theta\right) \right\|_O^2 + \frac{\lambda}{2} \cdot \left(\|W\|_F^2 + \|V\|_F^2 \right)$$

which is subject to

$$h(\boldsymbol{r}; \theta) = f(\boldsymbol{W} \cdot g(\boldsymbol{V}\boldsymbol{r} + \boldsymbol{\mu} + \boldsymbol{q}^*) + \boldsymbol{b} + \boldsymbol{c}^* + \boldsymbol{d}^*)$$

In the formula above, $\boldsymbol{d}^*$ denotes a dynamic vector, which has a length of m and is composed of the parameters from user daily bias set $\boldsymbol{D}$, according to item i and the corresponding day of the nodes. Then, the dynamic $\boldsymbol{d}^*$ vector is added to the $\boldsymbol{r}^{(i)}$ vector for $i = 1, 2, \ldots, n$. Similarly, parameter set θ is learned through backpropagation, and the learned θ is

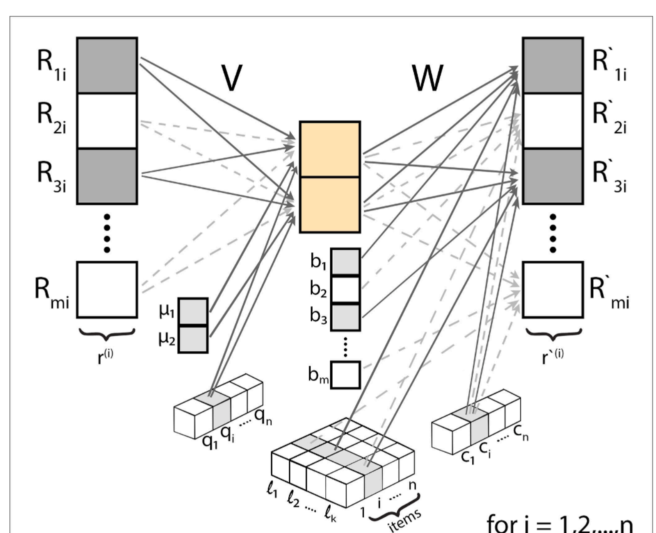

Fig. 4. Biased Autoencoder model with periodical biases for items.

used to predict unknown user-item rating pairs. The methodology for the implementation of user daily bias set $\boldsymbol{D}$ is shown in the figure above (Fig. 5). Notably, similar to the other Biased AutoRec models with temporal information discussed above, Biased AutoRec with the daily biases of users also provided outstanding accuracy in the empirical tests.

3.3. Model stacking

Empirical tests conducted with several datasets from different domains proved that Biased AutoRec had better prediction accuracy than AutoRec. In particular, adding temporal biases resulted in even better performances in most of the empirical tests, as listed in Table 2 in section 4. In the next step, the three temporal biases mentioned in the previous section were simultaneously implemented in the Biased AutoRec model. Although this was expected to increase the model accuracy, the empirical test results showed a slight decrease in the prediction accuracy, unlike the matrix factorization models. Therefore, stacking was applied to jointly employ several temporal signals in Biased AutoRec with the purpose of increasing the accuracy.

Stacking is a commonly used technique that utilizes the predictions of multiple models as inputs and creates the final set of predictions through a second-level learning algorithm (Jahrer et al., 2010). One remarkable example of model stacking was seen in the Netflix Prize, where the winning team won the big prize by stacking multiple models. The principle behind model stacking is that different models may perform better for a specific cluster of users or items. Therefore, even though the two models have similar performances over a particular dataset, the combination of their predictions often performs better. Stacking also prevents models from overfitting the data. This section shows how model stacking was implemented for the three Biased AutoRec models with temporal biases.

- Stacking AutoRec with the period bias and AutoRec with the daily biases of users
- Stacking AutoRec with the period biases of items and AutoRec with the daily biases of users
- Stacking AutoRec with the period bias, AutoRec with the period biases of items, and AutoRec with the daily biases of users

Linear stacking is one of the most frequently used methods for stacking because it is simpler than several other approaches, including rigid regression, bagging, and xgboost, which are more successful. The main goal of this study was not to compete with other models, but to

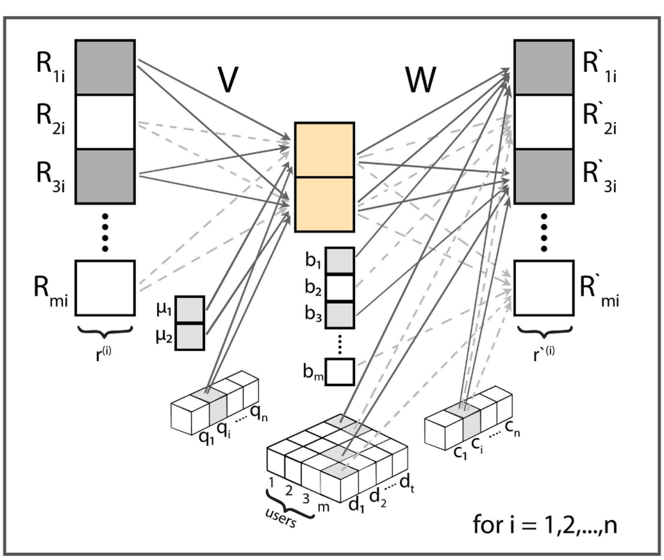

Fig. 5. Biased Autoencoder model with daily biases for users.

Table 1
Dataset details.

Datasets	Users	Items	Ratings	Density
MovieLens 100 K	943	1682	100,000	6.30%
MovieLens 1 M	6040	3706	1,000,209	4.47%
MovieLens 10 M	69,878	10,677	10,000,054	1.34%

Table 2
Test results (RMSE).

	MovieLens 100 K	**MovieLens 1 M**	**MovieLens 10 M**
Biased MF	0.908	0.846	0.805
timeSVD++	0.903	0.842	0.796
U-AutoRec	0.934	0.878	0.868
Biased U-AutoRec	0.904	0.862	0.857
Biased U-AutoRec w Items' Period Biases	0.898	0.866	0.860
Biased U-AutoRec w Users' Daily Biases	0.900	0.858	0.855
I-AutoRec	0.888	0.832	0.784
Biased I-AutoRec	0.886	0.828	0.781
Biased I-AutoRec w Period Bias	**0.877**	0.829	**0.780**
Biased I-AutoRec w Items' Period Biases	0.895	0.830	0.782
Biased I-AutoRec w Users' Daily Biases	0.884	**0.826**	0.782

prove the power of combining temporal dynamics and AutoRec, which is a nonlinear CF technique. Therefore, linear stacking was used instead of more complex stacking approaches. The model stacking was implemented using the following steps. First, a portion of the training set was designated as the holdout data. Then, each of the three models was trained in two versions, where the training set without holdout data was fit and the complete training set was fit in versions 1 and 2, respectively. Then, the rating prediction sets were extracted from each model and its two versions. Here, let H_1, H_2, and H_3 be model prediction sets in version 1, and let R_1, R_2, and R_3 denote the model prediction sets in version 2 for the three models. For two-input stacking, the following method is implemented:

$$V(H_1, H_2) = (1 - \alpha)H_1 + \alpha H_2$$

For the sake of simplicity, a grid search is used for $\alpha = \{0, 0.1, 0.2, 03, \ldots, 1\}$, and the value of α that provides the best results is selected. Then, the same α is used to stack the final rating predictions.

$$P(R_1, R_2) = (1 - \alpha)R_1 + \alpha R_2$$

For three-input stacking, a grid search is implemented for $\alpha = \{0, 0.1, 0.2, 03, \ldots, 1\}$ and $b = \{0, 0.1, 0.2, 03, \ldots, 1\}$, and the best α and b pairs are selected.

$$V(H_1, H_2, H_3) = \alpha H_1 + bH_2 + (1 - \alpha - b)H_3$$

Similarly, the sameα and b values are used to calculate the final set of rating predictions.

$$P(R_1, R_2, R_3) = \alpha R_1 + bR_2 + (1 - \alpha - b)R_3$$

The following section shows how the accuracies of the three stacked models were calculated and compared them with the accuracies of the previous models.

4. Experiments

4.1. Datasets and Experiments

This section shows how the models described in previous sections were evaluated using MovieLens 100 K, 1 M and 10 M, which are very common movie rating datasets (Harper & Konstan, 2016). Each user has rated a minimum of 20 movies in each MovieLens dataset and rated a minimum of five products in each product review dataset that was used. These datasets are commonly used to evaluate the performances of CF models. The details of the datasets used in this study are listed in Table 1.

For the empirical tests, each dataset was split into 90% and 10% portions for the training and test sets, respectively. In addition, 10% of the training set was held out for hyperparameter tuning and model stacking. The models were built based on the AutoRec model and used the original code of the author as a base. In the original AutoRec paper, the author stated that choosing the identity function for $f()$ and sigmoid function for $g()$ resulted in the best accuracy among the combinations in empirical tests. Moreover, they also stated that increasing the number of hidden layer units kept increasing the model accuracy, but the marginal benefit started to decline after 500. For convenience, the same parameters were used in Biased AutoRec and in its temporal versions. L-BFGS was used as the learning algorithm for all the proposed models and nonlinear baseline models. Moreover, stochastic gradient descent (SGD) was implemented to train the biased matrix factorization and timeSVD++, which were used in a comparison. Each test was repeated five times, and the average root mean squared error (RMSE) scores are reported in Table 2.

The empirical test results for the three stacked models are presented in Table 3. The test results indicate that stacking the three temporal models provided the lowest RMSE scores compared to the other models for each dataset.

4.2. Performance evaluation

4.2.1. Evaluation metrics

During the empirical tests, the results of the different models were checked using the RMSE scores:

$$\sqrt{\sum_{(u,i)\in T}\left(r_{ui}-\widehat{r}_{ui}\right)^2 \Big/ |T|}$$

where $|T|$ denotes the number of ratings in the test set. The RMSE was selected instead of other popular measures because the RMSE penalizes larger errors more, which provides more sensitive results. Notably, a lower RMSE score indicates a better accuracy for the model.

4.2.2. Baseline models

To prove the effectiveness of the proposed models, their performances were compared with those of several baseline models in terms of the RMSE scores.

Empirical tests were also conducted, and the hyperparameters for all the compared baseline models were tuned as well as possible. The following baseline models were considered.

BMF: Biased matrix factorization is one of the most well-known CF models and has been proven to be effective in empirical tests.

timeSVD++: It includes temporal dynamics that are different from BMF and provides more impressive results in empirical tests.

U-AutoRec: It is an innovative CF model based on the autoencoder paradigm and utilizes nonlinear similarity.

I-AutoRec: Similar to U-AutoRec, it is also based on autoencoders. It performs better than U-AutoRec because items generally have more reviews than users and high variance in the number of user ratings leads to less reliable predictions for user-based methods (Sedhain et al., 2015).

4.2.3. Performance analysis

The empirical tests proved the effectiveness of the Biased AutoRec model, which had lower RMSE scores than BMF, timeSVD++, and I-AutoRec in the empirical tests conducted on the separate datasets. As expected, adding temporal signals to the Biased AutoRec model resulted in successfully lowering its RMSE score in most of the empirical tests. Moreover, although adding several temporal signals jointly to the same model did not result in success, stacking these models and combining their predictions after training the models independently provided the best results. The detailed results and improvements in Biased AutoRec, Biased AutoRec with period bias, Biased AutoRec with the period biases of items, Biased AutoRec with the daily biases of users, and their stacked versions over the baselines for all the datasets are listed in Table 2.

Moreover, even though the U-AutoRec model does not perform well compared to I-AutoRec, a biased version of U-AutoRec was also implemented, which was called Biased U-AutoRec. Surprisingly, empirical tests showed that the marginal improvements achieved by the proposed models were larger in U-AutoRec than in I-AutoRec. The test details and improvement information are presented in Table 2.

Table 3
Model stacking (RMSE).

	MovieLens 100 K	MovieLens 1 M	MovieLens 10 M
I-AutoRec	0.888	0.832	0.784
Biased I-AutoRec	0.885	0.827	0.781
Biased I-AutoRec w PB + Biased I-AutoRec w UDB	0.882	0.824	0.776
Biased I-AutoRec w IPB + Biased I-AutoRec w UDB	**0.875**	0.823	0.774
Biased I-AutoRec w PB + Biased I-AutoRec w IPB + Biased I-AutoRec w UDB	**0.875**	**0.822**	**0.772**

5. Conclusions and future work

Implementing temporal signals in models has generally resulted in better prediction accuracies in previous studies. On the contrary, neural network CF models have been proven to be effective over traditional CF models because of their structure, which can take nonlinear similarity into consideration. This work combined these two approaches and proposed Biased AutoRec, which could be considered to be a biased version of the AutoRec model. In addition, the use of temporal biases with Biased AutoRec was also investigated.

Empirical tests proved the success of nonlinear models over traditional models and the effectiveness of adding temporal signals to nonlinear CF models.

On the contrary, adding more biases results in the requirement of more resources and a longer training duration, with decreasing marginal benefits. Therefore, although the models proposed in this paper provided better results, they may not be preferable for each scenario because of the resource requirements.

Future research could study the implementation of temporal dynamics in AutoRec or other neural network CF models in more complex ways. Moreover, other rich data features, including trust and occupation, can be implemented in the Biased AutoRec model, similar to the implementation of temporal signals in this study.

CRediT authorship contribution statement

Runliang Dou: Conceptualization, Supervision, Validation. **Oguzhan Arslan:** Conceptualization, Methodology, Software, Data curation, Visualization, Writing – original draft. **Ce Zhang:** Conceptualization, Investigation, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

Our acknowledgement to the Grouplens Research Group.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.eswa.2021.115775.

References

Cheng, S., & Wang, W. (2020). Rating prediction algorithm based on user time-sensitivity. *Information (Switzerland), 11*(1). https://doi.org/10.3390/info11010004

Guo, G., Zhang, J., & Yorke-Smith, N. (2016). A Novel Recommendation Model Regularized with User Trust and Item Ratings. *IEEE Transactions on Knowledge and Data Engineering, 28*(7), 1607–1620. https://doi.org/10.1109/TKDE.2016.2528249

Harper, F. M., & Konstan, J. A. (2016). The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems, 5*(4), 1–19. https://doi.org/10.1145/2827872

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. S. (2017). Neural collaborative filtering. 26th International World Wide Web Conference, WWW 2017. 10.1145/3038912.3052569.

Jahrer, M., Töscher, A., & Legenstein, R. (2010). Combining predictions for accurate recommender systems. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. https://doi.org/10.1145/1835804.1835893

Koren, Y. (2008). Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. https://doi.org/10.1145/1401890.1401944

Koren, Y. (2010). Collaborative filtering with temporal dynamics. *Communications of the ACM, 53*(4), 89–97. https://doi.org/10.1145/1721654.1721677

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer, 42*(8), 30–37. https://doi.org/10.1109/MC.2009.263

Li, M., Wu, H., & Zhang, H. (2019). Matrix Factorization for Personalized Recommendation with Implicit Feedback and Temporal Information in Social Ecommerce Networks. *IEEE Access, 7*, 141268–141276. https://doi.org/10.1109/Access.628763910.1109/ACCESS.2019.2943959

Lo, Y.-Y., Liao, W., Chang, C.-S., & Lee, Y.-C. (2018). Temporal Matrix Factorization for Tracking Concept Drift in Individual User Preferences. *IEEE Transactions on Computational Social Systems, 5*(1), 156–168. https://doi.org/10.1109/TCSS.2017.2772295

R, K., Kumar, P., & Bhasker, B. (2020). DNNRec: A novel deep learning based hybrid recommender system. *Expert Systems with Applications, 144*, 113054. https://doi.org/10.1016/j.eswa.2019.113054

Salakhutdinov, R., & Mnih, A. (2009). Probabilistic matrix factorization. *Advances in Neural Information Processing Systems 20 - Proceedings of the 2007 Conference*.

Salakhutdinov, R., Mnih, A., & Hinton, G. (2007). Restricted Boltzmann machines for collaborative filtering. *ACM International Conference Proceeding Series, 227*. https://doi.org/10.1145/1273496.1273596

Saurav, S., Malhotra, P., Vishnu, T. V., Gugulothu, N., Vig, L., Agarwal, P., & Shroff, G. (2018). Online anomaly detection with concept drift adaptation using recurrent neural networks. *ACM International Conference Proceeding Series, 10*. https://doi.org/10.1145/3152494.3152501

Sedhain, S., Menony, A. K., Sannery, S., & Xie, L. (2015). AutoRec: Autoencoders meet collaborative filtering. In *WWW 2015 Companion - Proceedings of the 24th International Conference on World Wide Web*. https://doi.org/10.1145/2740908.2742726

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 07-12-June-2015. 10.1109/CVPR.2015.7298594.

Tong, C., Qi, J.i., Lian, Y.u., Niu, J., & Rodrigues, J. J. P. C. (2019). TimeTrustSVD: A collaborative filtering model integrating time, trust and rating information. *Future Generation Computer Systems, 93*, 933–941. https://doi.org/10.1016/j.future.2017.07.037

Tsubaki, M., Duh, K., Shimbo, M., & Matsumoto, Y. (2016). Non-Linear similarity learning for compositionality. 30th AAAI Conference on Artificial Intelligence, AAAI 2016.

Wang, B., Huang, Y., & Li, X. (2016). Combining Review Text Content and Reviewer-Item Rating Matrix to Predict Review Rating. *Computational Intelligence and Neuroscience, 2016*, 1–11. https://doi.org/10.1155/2016/5968705

Wu, C. Y., Ahmed, A., Beutel, A., Smola, A. J., & Jing, H. (2017). Recurrent recommender networks. WSDM 2017 - Proceedings of the 10th ACM International Conference on Web Search and Data Mining. 10.1145/3018661.3018689.

Wu, T., Feng, Y., Sang, J. X., Qiang, B. H., & Wang, Y. N. (2018). A novel recommendation algorithm incorporating temporal dynamics, reviews and item correlation. IEICE Transactions on Information and Systems, E101D(8). 10.1587/transinf.2017EDP7387.

Xu, Y., Yu, Q., Lam, W., & Lin, T. (2017). Exploiting interactions of review text, hidden user communities and item groups, and time for collaborative filtering. *Knowledge and Information Systems, 52*(1), 221–254. https://doi.org/10.1007/s10115-016-1005-1

Yi, B., Shen, X., Zhang, Z., Shu, J., & Liu, H. (2017). In *Expanded autoencoder recommendation framework and its application in movie recommendation*. Information Management and Applications. https://doi.org/10.1109/SKIMA.2016.7916236.

Zhang, S., Yao, L., & Xu, X. (2017). Autosvd++: An efiicient hybrid collaborative filtering model via contractive auto-encoders. In *SIGIR 2017 - Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. https://doi.org/10.1145/3077136.3080689

Zheng, N., & Li, Q. (2011). A recommender system based on tag and time information for social tagging systems. *Expert Systems with Applications, 38*(4), 4575–4587. https://doi.org/10.1016/j.eswa.2010.09.131