

Collaborative Filtering Auto-Encoders for Technical Patent Recommending

Wenlei BAI^{†a)}, Student Member, Jun GUO^{†*b)}, Xueqing ZHANG^{†c)}, Baoying LIU^{†d)},
and Daguang GAN^{††e)}, Nonmembers

SUMMARY To find the exact items from the massive patent resources for users is a matter of great urgency. Although the recommender systems have shot this problem to a certain extent, there are still some challenging problems, such as tracking user interests and improving the recommendation quality when the rating matrix is extremely sparse. In this paper, we propose a novel method called Collaborative Filtering Auto-Encoder for the top-N recommendation. This method employs Auto-Encoders to extract the item's features, converts a high-dimensional sparse vector into a low-dimensional dense vector, and then uses the dense vector for similarity calculation. At the same time, to make the recommendation list closer to the user's recent interests, we divide the recommendation weight into time-based and recent similarity-based weights. In fact, the proposed method is an improved, item-based collaborative filtering model with more flexible components. Experimental results show that the method consistently outperforms state-of-the-art top-N recommendation methods by a significant margin on standard evaluation metrics.

key words: recommender systems, collaborative filtering, Auto-Encoders, item similarity, patent recommendation

1. Introduction

There has been an explosive growth of various scientific and technological resources in research and industry areas. As a crucial scientific and technological resource, patents play an essential role in the modern information society, which contains a large amount of technical knowledge and development information. The rapid increment in patent applications and grants in China has aroused widespread concern in the community. How to efficiently mine the patents to assist researchers in researching and writing patents has become a matter of great urgency. It is also a hot research topic widely concerned by academia and industry.

At present, the Patent Information Service mainly faces the following problems: the number of patents increases exponentially every year, which causes the problem of patent data information overload. At the same time, it is also difficult for researchers to find out the exciting patents from

the massive patent data. Patent Information Service platforms (such as Cnki and Wanfang data in China) employ an efficient search algorithm to discover the information that researchers need and show excellent information retrieval performance when users specify their precise needs. However, when users cannot provide correct keywords, requirements, and directions, search engines cannot work and give satisfactory search results. The emergence of the recommendation system has greatly alleviated this difficulty. The recommendation system estimates the user's research interests and hobbies through the learning model according to users' historical behavior record and then predicts their rating or preference for the given information. The key to the recommendation system is the recommendation algorithm. In the context of patent data matching, the recommendation algorithm can mine user interests and user collaboration relationships, helping researchers to get more accurate results from a massive amount of patent data.

The collaborative filtering (CF) algorithm is the earliest and classical recommendation algorithm [1], [2]. Although there is unprecedented popularity of deep learning nowadays, traditional CF recommendation algorithm models have many applicable application scenarios because of their strong interpretability, low hardware environment requirements, and ease of rapid training and deployment [3].

In practical applications, the existing CF algorithms still face some problems, such as rating sparsity. As the number of users and items has dramatically increased in recent years, the rating data has become very sparse. The similarity measurement used by the traditional collaborative filtering algorithm has become unstable in this case. Besides, the CF algorithm cannot track the change of users' interests, neglecting the fact that the recently viewed papers or patents can reflect users' recent interests and hold valuable recommendations clues.

To solve these problems, some improved CF algorithms have been proposed in the past decade [4]–[6]. These improved collaborative filtering recommendation algorithms rarely take into account the research interest changing of users. Jun Chen and Chaokun Wang proposed a Markov recommendation model based on user interest forgetting [7]; Feng Yong and Zhang Bei proposed a hybrid dynamic recommendation model with long and short interests and multiple neural networks [8]. These methods have made significant progress, but the computational cost is also enormous due to increased model complexity.

Manuscript received November 11, 2020.

Manuscript revised January 18, 2021.

Manuscript publicized April 26, 2021.

[†]The authors are with the School of Information Science and Technology, Northwest University, China.

^{††}The author is with Wanfang Data, China.

*Joint first author.

a) E-mail: 201932108@stumail.nwu.edu.cn

b) E-mail: guojun@nwu.edu.cn

c) E-mail: zhangxueqing@stumail.nwu.edu.cn

d) E-mail: paola.liu@nwu.edu.cn (Corresponding author)

e) E-mail: gandg@wanfangdata.com.cn

DOI: 10.1587/transinf.2020BDP0014

According to the above analysis, this paper presents a novel method called Collaborative Filtering Auto-Encoder (CFAE) to satisfy the tradeoff of recommendation quality, model complexity, and computational cost recommendation. The algorithm's main idea is to divide the recommendation weight into time-based weight and recent similarity-based weight. Besides, the algorithm uses the Auto-Encoder to extract the features of the item. This way, it can reflect the change of users' interest and improve the precision and recall rate of algorithm recommendation. The experimental results on four available data sets show that the proposed algorithm outperforms the traditional algorithms and deep learning models.

The main contributions of this paper are as follows:

- We successfully introduced the Auto-Encoder to compress the high-dimensional and sparse item rating vector into a low-dimensional feature vector.
- According to the human memory forgetting law, this paper introduces the user interest change function to the traditional collaborative filtering recommendation algorithm so that the recommendation list can more accurately reflect the user's recent interest.
- This paper has carried out sufficient experiments on four available data sets. The experimental results show that the proposed method is superior to the traditional CF recommendation algorithms in recommendation quality. Compared with algorithms based on deep learning, the algorithm has lower time complexity and higher precision and recall rate.

The rest of the paper is organized as follows:

Section 2 presents related work. Section 3 describes our method's core idea, including the human memory forgetting law and the Auto-Encoder model. The experiments and evaluation results are presented in Sect. 4. Section 5 concludes the paper and provides potential future research directions.

2. Related Work

Patent recommendation methods mainly include the following: keyword-based methods, topic models methods, and neural network-based recommendation methods.

Keyword-based methods. These methods require the user to input clear keyword information and return items with the highest information matching degree through the search engine. The existing patent search and analysis systems, such as Google Patent[†], Cnki^{††}, Wanfang data^{†††} and so on, are based on these methods. In the academic field, query keyword extraction techniques have been introduced for matching words or phrases to find relevant patents [9], [10]. These methods have effectively improved the quality and efficiency of retrieval. For the problem of

words that have the same meaning, using a thesaurus to include similar words for keywords automatically has been proposed. Still, this method required manual management and expansion of the thesaurus. To deal with this problem, Wang and Lin proposed a new patent query expansion approach by exploiting the semantic knowledge base, enriching the query with semantically related concepts [11]. Patent recommendation based on keywords without considering semantics has great limitations. One of the most important points is that there may be a few identical keywords in queried patents, but the two patent documents' main ideas can be quite similar.

Topic models methods. Topic models methods can automatically extract the keywords and main idea of a patent. Choi et al. proposed to convert text into word lists or digital vector lists based on a package of words (BOW) and recommend patents by mining hidden topics in the full-text [12]. Krestel et al. investigated the application of language and topic modeling to the problem of patent retrieval. Experiments showed that the combination of topic and language modeling provides further significant performance improvements over either alone [13]. The biggest challenge facing topic models is that they are greatly affected by word frequency. High-frequency words will affect the results of topic word distribution. At the same time, the method may ignore the semantics of low-frequency words. The topic model also ignores the co-occurrence information of words. Therefore, the semantic information obtained is not accurate.

Neural network-based methods. Although deep learning-based recommendation systems have exhibited outstanding performance in various domains (such as movies, products), their validity in patent recommendations has not been investigated, owing to the lack of a freely available high-quality dataset. Jaewoong Choi et al. address the challenges in developing a deep learning-based automatic patent citation recommendation system and propose strong benchmark models considering the similarity of textual information and metadata (such as cooperative patent classification code) [14]. Helmers et al. studied how to extract feature vectors using Word2vec, Doc2vec, or other models to obtain the semantic information of texts in patents and finally used feature vectors to search or recommend patents [15].

As a neural network for unsupervised learning, Auto-Encoder can project high dimensional data to low-dimensional representations and has achieved good results in text processing [16] and object detection tasks [17]. These achievements inspired us to employ Auto-Encoder to extract item property features in the patent recommendation system. By the way, to our best knowledge, few efforts were spent on combining the Auto Encoder model with the patent recommendation system.

[†]<https://patents.google.com/>

^{††}<https://www.cnki.net/>

^{†††}<http://www.wanfangdata.com.cn/index.html>

3. Collaborative Filtering with Auto-Encoders

3.1 User Interest Change Model

Usually, the users' interest in items may change with time passing. The user's items have recently been paid attention to reflect the user's recent interest in the recommendation system. Generally speaking, the items users recently pay attention to should have a higher recommendation weight than the items users paid attention to a long time ago. However, traditional collaborative filtering algorithms ignore this factor.

It is generally believed that there are many resemblances between user interest changes and user forgetting law. Thus the recommendation model can introduce the user forgetting law into the collaborative filtering algorithm [18]. The German psychologist H. Ebbinghaus found that the human forgetting curve does not change linearly with time but presents a fast-low trend. This is the famous Ebbinghaus forgetting curve, as shown in Fig. 1.

By fitting the forgetting curve, we can simulate the user interest variation with time passing. This paper adopts the following exponential decay function to fit the forgetting curve:

$$Y(x) = e^{-\alpha x}, (x > 0) \quad (1)$$

Where x indicates the time passing, α is the attenuation adjustment coefficient used to adjust the change degree of user interest in different recommendation systems. For systems with a noticeable change of user interest, α should be larger. Otherwise, it should be smaller.

In the actual data report, the user evaluation time is expressed in the form of a timestamp. However, the timestamp data cannot be directly used in formula 1. Therefore, this paper proposes a mapping relationship between the time parameter x and timestamp.

$$x = \frac{d_{ui}}{L_u} \quad (2)$$

Where d_{ui} represents the time passing from the last time when user u access item i ; L_u represents the difference between the user's first and last access to the system.

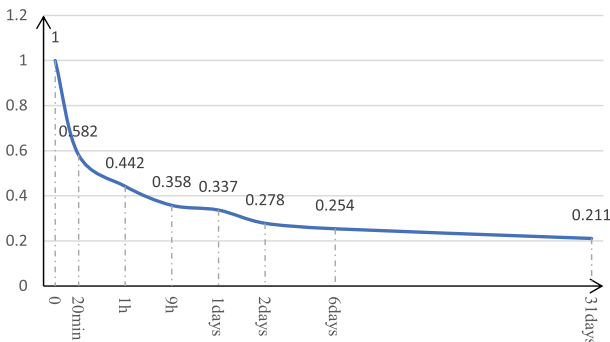


Fig. 1 Ebbinghaus forgetting curve

Finally, the calculation formula of data weight based on time is as follows:

$$WT(u, i) = e^{-\alpha \cdot \frac{d_{ui}}{L_u}} \quad (3)$$

3.2 Auto-Encoder Model

Auto-Encoder model is actually a multi-layer neural network that consists of an input layer, a hidden layer, and an output layer. Whatever image, audio, or text data, Auto-Encoder can convert them into a vector expression. An example of the neural network structure of the Auto-Encoder is shown in Fig. 2.

Assuming that the data vector is r , the Auto-Encoder takes the vector r (such as the column vector in the rating matrix) as input. After passing through the Auto-Encoder, the output vector is as close to itself as possible. The Auto-Encoder, as an unsupervised learning algorithm, does not need to label the training data manually.

The Auto-Encoder learns a function through the encoding and decoding process. If the number of neurons in the hidden layer is limited, the Auto-Encoder model can mine some special input data structures. For example, supposing that the Auto-Encoder input layer contains 784 neurons. If the hidden layer has only 128 neurons, Auto-Encoder's neural network structure will learn the compressed representation of the input data.

In rating-based collaborative filtering, we have m users, n items, and a rating matrix with only partial information R of $m \times n$. Assuming that the user-item rating matrix is shown in Table 1.

So each item $i \in I = \{1 \dots n\}$ can be represented by a partially observed vector $r^{(i)} = (R_{1i}, \dots, R_{mi})$. We aim to use the Auto-Encoder to extract the item's features from the hidden layers.

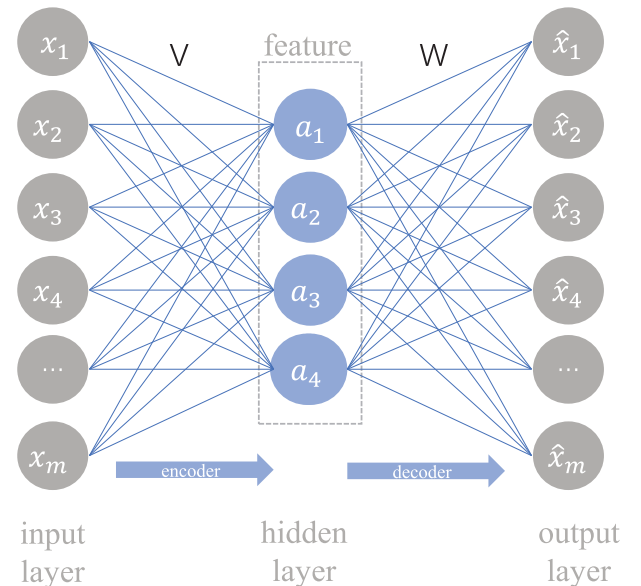


Fig. 2 The neural network structure of an Auto-Encoder

Assuming that the reconstruction function of the Auto-Encoder is $h(\mathbf{r}; \theta)$, then the objective function of the Auto-Encoder is as follows:

$$\min_{\theta} \sum_{\mathbf{r} \in S} \|\mathbf{r} - h(\mathbf{r}; \theta)\|_2^2 \quad (4)$$

Among them, S is the set of all data vectors and $h(\mathbf{r}; \theta)$ is the reconstruction of input $\mathbf{r} \in R^m$.

$$h(\mathbf{r}; \theta) = \mathbf{W} \cdot \sigma(\mathbf{V}\mathbf{r} + \boldsymbol{\mu}) + \mathbf{b} \quad (5)$$

Here, $\theta = \{\mathbf{W}, \mathbf{V}, \boldsymbol{\mu}, \mathbf{b}\}$, $\mathbf{W} \in R^{m \times k}$, $\mathbf{V} \in R^{k \times m}$, and bias $\boldsymbol{\mu} \in R^k$, $\mathbf{b} \in R^m$, σ is the sigmoid activate function. In addition, k is the number of hidden layer neurons and the parameters θ are learned using backpropagation. To avoid overfitting, we introduce the following the objective function:

$$\min_{\theta} \sum_{i=1}^n \|\mathbf{r}^{(i)} - h(\mathbf{r}^{(i)}; \theta)\|_O^2 + \frac{\lambda}{2} \cdot (\|\mathbf{W}\|_F^2 + \|\mathbf{V}\|_F^2) \quad (6)$$

where $\|\cdot\|_O^2$ means that we only consider the contribution of observed ratings [19].

After training the Auto-Encoder model, the vector features of all data is stored in the function $\sigma(\mathbf{V}\mathbf{r} + \boldsymbol{\mu})$. Generally speaking, the number of parameters of the reconstruction function is much smaller than the number of dimensions of the input vector, so the Auto-Encoder completes the work of data compression and dimensionality reduction.

After obtaining the item attributes' features, the similarity between the items can be calculated by the similarity calculation methods include cosine similarity, the Pearson correlation coefficient, and the adjusted cosine similarity [1]. The specific calculation method is shown in Table 2.

In Table 2, i and j represent two different items in the item space; $\text{sim}(i, j)$ represents the similarity between items i and j ; \bar{r}_i represents the average rating of item i ; \bar{r}_u represents the average rating of user u .

3.3 Recent Item Similarity

In formula 3, the value of $WT(u, i)$ decreases exponentially

Table 1 User-item rating matrix.

	<i>item</i> ₁	<i>item</i> ₂	<i>item</i> ₃	...	<i>item</i> _{<i>n</i>}
<i>u</i> ₁	<i>r</i> ₁₁	<i>r</i> ₁₂	<i>r</i> ₁₃	...	<i>r</i> _{1<i>n</i>}
<i>u</i> ₂	<i>r</i> ₂₁	<i>r</i> ₂₂	<i>r</i> ₂₃	...	<i>r</i> _{2<i>n</i>}
...
<i>u</i> _{<i>m</i>}	<i>r</i> _{<i>m</i>1}	<i>r</i> _{<i>m</i>2}	<i>r</i> _{<i>m</i>3}	...	<i>r</i> _{<i>m</i><i>n</i>}

Table 2 Similarity measurement methods.

<i>Measuer</i>	<i>Description</i>
cosine similarity	$\text{sim}(i, j) = \frac{\sum_{u \in U} r_{ui} \cdot r_{uj}}{\sqrt{\sum_{u \in U} r_{ui}^2} \sqrt{\sum_{u \in U} r_{uj}^2}}$
pearson correlation	$\text{sim}(i, j) = \frac{\sum_{u \in U} (r_{ui} - \bar{r}_i)(r_{uj} - \bar{r}_j)}{\sqrt{\sum_{u \in U} (r_{ui} - \bar{r}_i)^2} \sqrt{\sum_{u \in U} (r_{uj} - \bar{r}_j)^2}}$
adjusted cosine similarity	$\text{sim}(i, j) = \frac{\sum_{u \in U} (r_{ui} - \bar{r}_u)(r_{uj} - \bar{r}_u)}{\sqrt{\sum_{u \in U} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{u \in U} (r_{uj} - \bar{r}_u)^2}}$

with the increase of d_{ui} . That is to say, the recommended weight of items that users have not paid attention to for a long time will be smaller, highlighting users' latest interest changes. However, the user's interest may have temporal stability. Although an item has not been browsed for a long time, it may be similar to the user's new interest items. If the recommendation model uses the data weight based on the time to recommend, it will cause the user's interest in the early item recommendation weight to be too low. Therefore, this paper proposes a recommendation weight adjustment strategy based on the recent similarity of items.

We assume that the set of items visited by user u is N_u . The set of items accessed by the user in the most recent T time period is N_{ut} . For item $i \in N_u$, the average similarity between item i and item in N_{ut} shows the new interest correlation between item i and user u . Then, we can get the recommendation weight based on the recent similarity of the item.

The calculation formula is as follows:

$$WS(u, i) = \frac{\sum_{j \in N_{ut}} \text{sim}(i, j)}{|N_{ut}|} \quad (7)$$

This paper discusses the calculation method of recommendation weight from two aspects: time-based and recent similarity-based weights. Both of them have an essential impact on the final recommendation weight, and their contributions need to be considered comprehensively. Therefore, we use the weighted fusion method to find a suitable super parameter b and calculate the final recommended weight. The calculation formula is as follows:

$$W(u, i) = bWS(u, i) + (1 - b)WT(u, i) \quad (8)$$

Finally, user u 's interest in item j is calculated using the following formula:

$$P(u, j) = \sum_{i \in N(u) \cap S(j, k)} W(u, i) \cdot \text{sim}(i, j) \quad (9)$$

4. Experimental Results and Analysis

This section will describe the data set used in the experiment, the determination of formula parameters, algorithm performance evaluation indicators, experimental results, and analysis.

4.1 Data Sets

In the spirit of rigorous experimentation, we evaluate various data sets.

- **MovieTweatings** [20] - This dataset was presented at the CrowdRec 2013 workshop by the University of Ghent Belgium.
- **hetrec2011** [21] - This data set is an extension of the MovieLens10M dataset, published by the GroupLens research organization.
- **learning-from-sets-2019** [22] - This data set is from

Table 3 Data set information statistics table.

Dataset	#ratings	#Users	#Items	#Density
Movietweetings	150K	5.2K	13K	0.22%
Hetrec2011	160K	1.8K	8.4K	1.007%
LearnSet2019	110K	829	10K	1.351%
CiaoDVD	41K	1.2K	13K	0.24%

Table 4 Confusion matrix for calculating precision and recall metrics

	recommended	not recommended
relevant	TP	FN
irrelevant	FP	TN

Sets of Items in Recommender Systems.

- **CiaoDVD** [23] - CiaoDVD is a data set of the entire DVD category of UK websites in December 2013, crawled from dvd.ciao.co.

The statistics in the data set are recorded in Table 3.

4.2 Baselines

We compare CFAE's recommendation result to two traditional collaborative filtering baselines, including:

- **ICF** - item-based collaborative filtering recommendation algorithm. The paper published by Amazon in 2003 [24] makes collaborative filtering a research hotspot for a long time in the future and a mainstream recommendation model in the industry.
- **UCF** - user-based collaborative filtering recommendation algorithm. Similar to ICF, the key to UCF is calculating the similarity between users.

We also compared CFAE with to three state-of-the-art deep learning recommendation algorithms, including:

- **BPR** - Bayesian personalized ranking from implicit feedback [25]. BPR is the state-of-the-art method for recommendation based on implicit feedback.
- **GMF** - Generalized Matrix Factorization [26]. NeuralCF replaces the simple inner product operation in the matrix factorization model with a multi-layer neural network and output layer structure.
- **CML** - Collaborative Metric Learning [27] learns a joint metric space to encode not only users' preferences but also the user-user and item-item similarity.

4.3 Evaluation Standard

Precision and recall are among the most frequently used metrics of the information retrieval field introduced by Cleverdon and Kean [28], [29]. They have been among the first series of the metrics used to evaluate recommendation algorithms. These metrics use a confusion matrix that divides the items into 4 different groups, as shown in Table 4.

In this matrix, relevant items that are recommended by the system are placed in the true positive (TP) group, and

Table 5 Precision and recall rate with α change table.

α	0.1	0.3	0.5	0.8	1
precision%	5.9	5.89	5.88	5.85	5.82
recall%	17.3	17.25	17.22	17.16	17.06

Table 6 Precision and recall rate with T change table.

T	10	30	50	100	200
precision%	5.91	5.9	5.9	5.91	5.92
recall%	17.32	17.29	17.29	17.32	17.36

those relevant items that the system failed to detect as relevant for the user go to the false negative (FN) group. Irrelevant items incorrectly recommended by the system are placed in the false positive (FP) group. Finally, the irrelevant items that are correctly not recommended to the user are considered in the true negative (TN) group.

Precision: precision is calculated as the ratio of the relevant items which are recommended to the number of all recommended items, as:

$$Precision = \frac{TP}{TP + FP} \quad (10)$$

Recall: recall is calculated as the ratio of the relevant items which are recommended to the number of all relevant items, as:

$$Recall = \frac{TP}{TP + FN} \quad (11)$$

4.4 Parameter Setting

In order to make the algorithm perform optimally, it is necessary to find suitable parameter values for Eqs. (3), (7), and (8) through a series of experiments. Namely: α is the attenuation adjustment coefficient; T is the user recent time window; b is the recommended weighting factor.

It should also be noted that the parameter tuning experiment uses the Movietweetings data set and the cosine similarity calculation method. In order to reduce the time complexity, we chose the controlled variable method to determine the parameters.

For α in formula 3, take 0.1, 0.3, ..., 1, and keep the other parameters unchanged ($T=30$, $b=0.5$), and calculate the precision and the recall rate of algorithm under each α value. As shown in Table 5. When α is 0.1, the precision and recall rate are the highest, so α is 0.1.

For T in formula 7, take 10, 30, 50, 100, 200, and keep the other parameters unchanged ($\alpha=0.1$, $b=0.5$), and calculate the precision and the recall rate of algorithm under each T value. As shown in Table 6. When $T=200$, although the recommendation effect is higher than $T=10$, the selection of T is too large to express the user's recent interests and preferences. Therefore, the value of T is 10.

For b in formula 8, take 0, 0.1, 0.3, 0.5, 1, and keep the other parameters unchanged ($T=10$, $\alpha=0.1$), and calculate the precision and the recall rate of algorithm under each b value. As shown in Table 7. When b is 0.1, the precision

Table 7 Precision and recall rate with b change table.

b	0	0.1	0.3	0.5	1
precision%	5.89	5.93	5.89	5.91	5.9
recall%	17.27	17.38	17.27	17.32	17.3

Table 8 Precision rate (%) comparison table (N=10, K=10)

Dataset	ICF	UCF	BPR	GMF	CML	CFAE
Movietweeting	5.02	4.4	3.97	3.7	5.14	5.93
Hetrec2011	1.11	1.31	2	1.6	1.58	2.72
LearnSet2019	1.13	1.72	2.04	2.09	2.13	7.65
CiaoDVD	0.99	1.75	1.4	0.91	1.73	7.61

Table 9 Recall rate (%) comparison table (N=10, K=10)

Dataset	ICF	UCF	BPR	GMF	CML	CFAE
Movietweeting	14.7	12.9	12.82	12.38	16.88	17.38
Hetrec2011	1.24	1.46	3.7	2.9	3	3.04
LearnSet2019	0.79	1.21	2.68	2.53	2.87	5.35
CiaoDVD	2.68	4.79	5.13	3.8	6.46	20.97

and recall rate are the highest, so b is 0.1.

It should be noted that when $b=0$, the algorithm only considers recommendations based on time weights. When $b=1$, the algorithm only considers recommendations based on similarity weights. The experiment shows that the combination of time-based and recent similarity-based weights provides further significant performance improvements over either alone.

4.5 Comparison Experiment

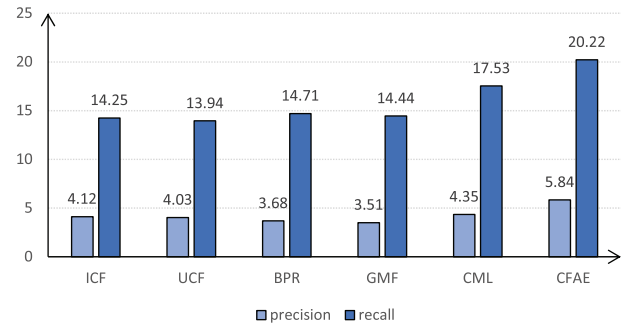
To verify the effectiveness of the algorithm proposed in this paper, we compare the algorithm with the algorithm described in the previous part of the baselines 4.2. The comparison experiments are carried out in the data sets in Table 3. The results are shown in Table 8 and Table 9.

4.6 Supplementary Experiment

We take the Wanfang patent data set as an instance to verify the practicability of our approaches. The data set contains 69326 anonymous ratings of approximately 9207 items presented by 2837 users, the algorithm is compared with the algorithm proposed in baselines 4.2. The experimental results are shown in Fig. 3. As can be seen from Fig. 3, the algorithm can also show good performance in real data.

5. Conclusions

In this paper, we successfully incorporate an Auto-encoder into the collaborative filtering recommendation algorithm. The Auto-Encoder efficiently encodes the high-dimensional sparse item rating vectors into low-dimensional dense feature vectors. The cosine similarity is used to measure item similarities between the encoded item feature vectors. We also introduce a time factor to reflect the user interest changing. Experimental results on four public data sets show that compared with the traditional CF algorithm and the latest

**Fig. 3** Comparison of precision and recall rate (%)

deep learning method, the algorithm proposed in this paper can greatly improve the precision and recall rate and track the user's interest, making the recommendation results closer to the user's current research direction. Although this paper mostly concerns the patent recommendation, we believe that the CFAE could afford many applications in scientific and technological resource services, such as article searching.

6. Acknowledgments

This work is financially supported in part by the National Key Research and Development Program of China under Grant No.2017YFB1400301, the National Natural Science Foundation of China Project under Grant No.62072372, 61973250, 61902318 and the Key Research and Development Program of Shaanxi Province under Grant 2019GY-012.

References

- [1] M.B. Sarwar, G. Karypis, A.J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," WWW, pp.285–295, 2001.
- [2] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," Proc. 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp.165–174, 2019.
- [3] Q. Yao, X. Chen, T.J. Kwok, Y. Li, and C.-J. Hsieh, "Efficient neural interaction function search for collaborative filtering," WWW '20: The Web Conference 2020 Taipei Taiwan April, 2020, pp.1660–1670, 2020.
- [4] L.J. Herlocker, A.J. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," Research and Development in Information Retrieval, pp.227–234, 2017.
- [5] R. Chen, Q. Hua, Y.-S. Chang, B. Wang, L. Zhang, and X. Kong, "A survey of collaborative filtering-based recommender systems: From traditional methods to hybrid methods based on social networks," IEEE Access, vol.6, pp.64301–64320, 2018.
- [6] Y. Zhu, Y. Gong, Q. Liu, Y. Ma, W. Ou, J. Zhu, B. Wang, Z. Guan, and D. Cai, "Query-based interactive recommendation by meta-path and adapted attention-gru," Proc. 28th ACM International Conference on Information and Knowledge Management, pp.2585–2593, 2019.
- [7] J. Chen, C. Wang, and W.J. 0001, "A personalized interest-forgetting markov model for recommendations," AAAI, pp.16–22, 2015.
- [8] Y. Feng, B. Zhang, B.H. Qiang, Y.Y. Zhang, and J.X. Shang, "Mn-hdrn: A novel hybrid dynamic recommendation model based on

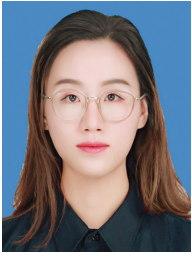
- long-short-term interests multiple neural networks,” *Chinese Journal of Computers*, 2019.
- [9] D. Ganguly, J. Leveling, W. Magdy, and G.J.F. Jones, “Patent query reduction using pseudo relevance feedback,” *CIKM*, pp.1953–1956, 2011.
 - [10] S. Bashir and A. Rauber, “Improving retrievability of patents in prior-art search,” *ECIR*, vol.5993, pp.457–470, 2010.
 - [11] F. Wang and L. Lin, “Exploiting semantic knowledge base for patent retrieval,” *ICNC-FSKD*, pp.2195–2200, 2017.
 - [12] H. Choi, S. Oh, S. Choi, and J. Yoon, “Innovation topic analysis of technology: The case of augmented reality patents,” *IEEE Access*, vol.6, pp.16119–16137, 2018.
 - [13] R. Krestel and P. Smyth, “Recommending patents based on latent topics,” *RecSys*, pp.395–398, 2013.
 - [14] J. Choi, S. Jang, J. Kim, J. Lee, J. Yoona, and S. Choi, “Deep learning-based citation recommendation system for patents,” *arXiv e-prints*, pp.1–11, 2020.
 - [15] L. Helmers, F. Horn, F. Biegler, T. Oppermann, K.-R. Müller, and B. McInnes, “Automating the search for a patent’s prior art with a full text similarity search,” *PloS one*, vol.14, no.3, p.e0212103, 2019.
 - [16] T. Young, D. Hazarika, S. Poria, and E. Cambria, “Recent trends in deep learning based natural language processing,” *arXiv: Computation and Language*, 2017.
 - [17] S. Ren, K. He, B.R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.39, no.6, pp.1137–1149, 2017.
 - [18] Y. Hong and L. Zhuan-Yun, “A collaborative filtering recommendation algorithm based on forgetting curve,” *Journal of Nanjing University (Natural Sciences)*, pp.51–58, 2010.
 - [19] S. Sedhain, K.A. Menon, S. Sanner, and L. Xie, “Autorec: Autoencoders meet collaborative filtering,” *WWW (Companion Volume)*, pp.111–112, 2015.
 - [20] S. Dooms, T. De Pessemier, and L. Martens, “Movietweetings: a movie rating dataset collected from twitter,” *Workshop on Crowdsourcing and Human Computation for Recommender Systems, CrowdRec at RecSys 2013*, 2013.
 - [21] I. Cantador, P. Brusilovsky, and T. Kuflik, “2nd workshop on information heterogeneity and fusion in recommender systems (hetrec 2011),” *Proc. 5th ACM conference on Recommender systems, RecSys 2011*, New York, NY, USA, ACM, pp.395–398, 2011.
 - [22] M. Sharma, F.M. Harper, and G. Karypis, “Learning from sets of items in recommender systems,” *TiS*, vol.9, no.4, pp.1–26, 2019.
 - [23] G. Guo, J. Zhang, D. Thalmann, and N. Yorke-Smith, “Etaf: An extended trust antecedents framework for trust prediction,” *Proc. 2014 International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp.540–547, 2014.
 - [24] G. Linden, B. Smith, and J. York, “Amazon.com recommendations: Item-to-item collaborative filtering,” *IEEE Internet Comput.*, vol.7, no.1, pp.76–80, 2003.
 - [25] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, “Bpr: Bayesian personalized ranking from implicit feedback,” *UAI ’09 Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pp.452–461, 2009.
 - [26] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, “Neural collaborative filtering,” *WWW*, pp.173–182, 2017.
 - [27] C.-K. Hsieh, L. Yang, Y. Cui, T.-Y. Lin, J.S. Belongie, and D. Estrin, “Collaborative metric learning,” *WWW*, pp.193–201, 2017.
 - [28] C.W. Cleverdon, J. Mills, and E.M. Keen, “Factors determining the performance of indexing systems, (volume 1: Design),” *Cranfield: College of Aeronautics*, vol.28, 1966.
 - [29] M. Jalili, S. Ahmadian, M. Izadi, P. Moradi, and M. Salehi, “Evaluating collaborative filtering recommender algorithms: A survey,” *IEEE Access*, vol.6, pp.74003–74024, 2018.
 - [30] A. Mongia, N. Jhamb, E. Chouzenoux, and A. Majumdar, “Deep latent factor model for collaborative filtering,” *vol.169*, p.107366, 2020.
 - [31] J. Chen, C. Wang, and W.J. 0001, “A personalized interest-forgetting markov model for recommendations,” *AAAI*, pp.16–22, 2015.
 - [32] F. Xue, X. He, X. Wang, J. Xu, K. Liu, and R. Hong, “Deep item-based collaborative filtering for top-n recommendation,” *ACM Transactions on Information Systems (TOIS)*, vol.37, no.3, pp.1–25, 2019.
 - [33] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” *IEEE Computer*, vol.42, no.8, pp.30–37, 2009.
 - [34] H.-J. Xue, X. Dai, J. Zhang, S. Huang, and J. Chen, “Deep matrix factorization models for recommender systems,” *IJCAI*, pp.3203–3209, 2017.
 - [35] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol.521, no.7553, pp.436–444, 2015.
 - [36] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.A. Manzagol, “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion,” *Journal of Machine Learning Research*, pp.3371–3408, 2010.
 - [37] S. Oh, Z. Lei, W.-C. Lee, P. Mitra, and J. Yen, “Cv-pcr: a context-guided value-driven framework for patent citation recommendation,” *CIKM*, pp.2291–2296, 2013.
 - [38] S. Arts, J. Hou, and C.J. Gomez, “Text mining to measure novelty and diffusion of technological innovation,” *ISSI*, pp.1798–1800, 2019.
 - [39] D. Liang, G.R. Krishnan, D.M. Hoffman, and T. Jebara, “Variational autoencoders for collaborative filtering,” *WWW ’18: The Web Conference 2018 Lyon France April, 2018*, pp.689–698, 2018.
 - [40] J. Guo, X. Yuan, P. Xu, H. Bai, and B. Liu, “Improved image clustering with deep semantic embedding,” *Pattern Recognition Letters*, vol.130, pp.225–233, 2020.
 - [41] C.B. Clement, M. Bierbaum, K.P. O’Keeffe, and A.A. Alemi, “On the use of arxiv as a dataset,” *arXiv e-prints*, pp.1–7, 2019.



Wenlei Bai received the B.E. degree in mathematics and Computer Science from Shanxi Normal University, Linfen, China, in 2019. He is currently a Master’s degree candidate in Software engineering at Northwest University, China. His research interests include machine learning and recommendation systems.



Jun Guo is a professor in the Department of Computer Science at Northwest University, China. He got his Ph.D. degree in computer science at Northwest University in 2007 and worked as a post-doctoral fellow at Tsinghua University during 2008–2010. His research interests are embedded computing, natural language processing, and AI algorithm.



Xueqing Zhang postgraduate in School of Information Science and Technology, Northwest University, Xi'an, China. She received a B.E. with a major in computer science and technology at Northwest University, in 2019. Her research interests are recommendation systems and machine learning.



Baoying Liu is a professor in Information Science and Technology School, Northwest University of China. She received an M.A. degree in western culture and history from Northwest University, China, in 2004. Then she received the M.A. in cultural heritage preservation and tourism environment and the Ph.D. degree in Cultural Heritage from the University of Salento, Italy, in 2005 and 2012. Her research interests include heritage preservation and social networks.



Daguang Gan received the M.S. degree in computer science from the institute of science and technology, P.R. China, in 2008. He is an assistant to the general manager of Beijing Wan Fang software co., LTD., His research interests include knowledge organization and information recommendation.