Computers, Materials & Continua CMC, vol.64, no.3, pp.1959-1975, 2020

A Recommendation Method for Highly Sparse Dataset Based on Teaching Recommendation Factorization Machines

Dunhong Yao[1, 2, 3], Shijun Li[4, *], Ang Li[5] and Yu Chen[6]

Abstract: There is no reasonable scientific basis for selecting the excellent teachers of the school's courses. To solve the practical problem, we firstly give a series of normalization models for defining the key attributes of teachers' professional foundation, course difficulty coefficient, and comprehensive evaluation of teaching. Then, we define a partial weight function to calculate the key attributes, and obtain the partial recommendation values. Next, we construct a highly sparse Teaching Recommendation Factorization Machines (TRFMs) model, which takes the 5-tuples relation including teacher, course, teachers' professional foundation, course difficulty, teaching evaluation as the feature vector, and take partial recommendation value as the recommendation label. Finally, we design a novel Top-N excellent teacher recommendation algorithm based on TRFMs by course classification on the highly sparse dataset. Experimental results show that the proposed TRFMs and recommendation algorithm can accurately realize the recommendation of excellent teachers on a highly sparse historical teaching dataset. The recommendation accuracy is superior to that of the three-dimensional tensor decomposition model algorithm which also solves sparse datasets. The proposed method can be used as a new recommendation method applied to the teaching arrangements in all kinds of schools, which can effectively improve the teaching quality.

Keywords: Highly sparse dataset, normalized models, teaching recommendation factorization machines, excellent teacher recommendation.

1 Introduction

The personalized recommendation system predicts the potential interests of users based

[1] School of Computer Science and Engineering, Huaihua University, Huaihua, 418000, China.
[2] Key Laboratory of Wuling-Mountain Health Big Data Intelligent Processing and Application in Hunan Province Universities, Huaihua, 418000, China.
[3] Key Laboratory of Intelligent Control Technology for Wuling-Mountain Ecological Agriculture in Hunan Province, Huaihua, 418000, China.
[4] School of Computer, Wuhan University, Wuhan, 430072, China.
[5] Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708, USA.
[6] School of Computer Science and Engineering, Yulin Normal University, Yulin, 537000, China.
[*] Corresponding Author: Shijun Li. Email: shjli@whu.edu.cn.
Received: 15 February 2020; Accepted: 03 May 2020.

on the analysis of users' historical behavior data. It makes a current recommendation to meet the personalized needs [Resnick and Varian (1997)]. In recent years, personalized recommendation systems based on different recommendation algorithms have been rapidly developed in e-commerce, social networks, and search engines.

The techniques used in personalized recommendation systems are categorized into five groups [Xu, Wu, Li et al. (2009); Jiang, Chen, Jiang et al. (2019); Cacheda, Carneiro, Fernández et al. (2011); Chen, Xiong, Xu et al. (2019)]: 1) the nearest neighbor heuristic collaborative filtering recommendation algorithms based on the Euclidean distance, Pearson correlation coefficient, and cosine similarity; 2) the collaborative filtering recommendation algorithms based on the context-aware model, latent factor model, Bayesian model, trust-aware model, clustering model, and maximum entropy model; 3) the content-based recommendation algorithms based on decision trees, neural networks, vectors, term frequency-inverse document frequency (TF-IDF), adaptive filtering, and threshold setting; 4) the other algorithms based on association rules recommendation, utility recommendation, and knowledge reasoning; 5) the combination recommendation algorithms based on tag graph, tag FolkRank, cascade, feature combination, weighting, transformation, hierarchical tag clustering, variable support vector machine (VSVM), and tensor decomposition.

When the dataset is highly sparse, most of the recommendation techniques mentioned above show a significant decrease in the recommendation accuracy and increase the time complexity [Meng, Ji and Zhang (2015)]. Also, traditional factorization models (such as matrix factorization [Guan, Li and Guan (2017)], parallel factor analysis [Yu, Hsieh, Si et al. (2014)], SVD++ [Kumar, Verma and Rastogi (2014)], and pairwise interaction tensor factorization (PITF) [Rendle and Schmidt-Thieme (2010)]) are restricted to a specific data type. To address these problems, Rendle [Rendle (2012)] proposed Factorization Machines (FMs). FMs overcomes the low generality and the data type restriction of the traditional factorization models. FMs can deal with multi-class variables and highly sparse numbers. According to the advantages of model parameter estimation, linear time complexity, and high-quality prediction on any real eigenvector, the algorithm includes many successful collaborative filtering methods. FMs has been successfully applied to many recommendation fields in the existing Chinese and English documents. For example, Rendle et al. [Rendle, Gantner, Freudenthaler et al. (2011)] used the FMs model to realize fast context-aware recommendations. Nguyen et al. [Nguyen, Karatzoglou and Baltrunas (2014)] used an improved Gaussian Process Factor decomposition Machine (GPFM) to realize the most advanced context-aware recommendation. Geuens [Geuens (2015)] used the FMs model to realize customer, product and implicit recommendation. Chen et al. [Chen, Hou, Xiao et al. (2016); He, Fang, Liu et al. (2019); Alemu, Olsen, Vedel et al. (2017)] used the FMs to improve the accuracy of product recommendation and better recommend the mobile app. FMs model is used to predict students' achievement, such as Thai-Nghe et al. [Thai-Nghe, Drumond, Horváth et al. (2012)].

To best our knowledge, no research conducted on the application of recommendation systems to teaching. The limited research conducted on data pre-processing techniques for teachers and course teaching [Qin (2015)]. The local second-tier universities have relatively inadequate teachers and schooling conditions. In this case, the teaching arrangements are mostly based on the teacher's wishes. When encountering some new courses, there is no reasonable scientific basis for teaching recommendation. Such a lack of scientific basis leads to a non-optimized and low quality of teaching.

In this paper, a series of normalized models are proposed to pre-process a large amount of teaching data. Based on the processed data, the key attributes of teacher professional foundation, course difficulty coefficient, teaching comprehensive evaluation, and overall recommendation value are defined. We use these key attributes as the feature vector X and the recommended observation value as the target vector Y to construct the Teaching Recommended Factorization Machines (TRFMs) model. Then, the teaching recommendation algorithm is designed based on the TRFMs model. Finally, we compare it with the three-dimensional tensor recommendation algorithm in accuracy and time complexity. Experimental results show that the proposed TRFMs model and recommendation algorithm provide advantages in the recommendation of a sparse teaching dataset. The proposed methods are promising to address the lack of a scientific recommendation basis.

2 Related works

Referring to Adomavicius' definition of recommendation system [Adomavicius and Tuzhilin (2005)], we define the teaching recommendation system as follows. The course and teacher sets are defined as C and T, respectively. Utility function $f()$ is used to calculate the recommendation degree of teacher object t to course c; that is, $f: C \times T \rightarrow R$ where R is a set of totally ordered non-negative real numbers in a specific range. The problem is to find those T^* whose recommendation degree R is the largest, that is, $\forall c \in C,\ T^* = \arg \max_{t \in T} f(c,t)$.

Steffen proposed the FMs. In FMs, the implicit factor model and the matrix decomposition idea for reference are used to remove the autocorrelation item from polynomial regression, and only the interaction between the categorical variables is used as factor decomposition [Rendle, Gantner, Freudenthaler et al. (2011)]. FMs can be used to deal with three prediction problems: a regression, a binary classification, and a ranking.

The second-order factorization model is commonly used, which is defined as follows:

$$\hat{y}(x) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j>i}^{n} \langle v_i, v_j \rangle x_i x_j \tag{1}$$

The parameter $w_0 \in \mathbb{R}$, $w \in \mathbb{R}^n$, $v \in \mathbb{R}^{n \times k}$, $<v_i,v_j>$ denote the dot product of two vectors v_i and v_j whose size is k, i.e. $\langle v_i, v_j \rangle = \sum_{f=1}^{k} v_{i,f} \bullet v_{j,f}$, where $k \in \mathbb{N}^+$ ($k \ll n$) is called factor decomposition dimension hyper-parameter. In Eq. (1), by defining $j>i$, autocorrelation term is removed; thus, only the interaction between two mutually-distinct feature components is considered. The dot product of two low-rank matrices is used in FMs to approximate the interaction of categorical variables. That is, $w_{i,j} \approx \langle v_i, v_j \rangle = \sum_{f=1}^{k} v_{i,f} \bullet v_{j,f}$

$w_{i,j^*} \approx \langle v_i, v_{j^*} \rangle = \sum_{f=1}^{k} v_{i,f} \bullet v_{j^*,f}$ so that some interactions are shared between $w_{i,j}$ and $w_{i,j}$*. Accordingly, FMs can contain multiple categorical variables and be suitable for the case where the data is very sparse [Rendle (2012)].

To learn model parameter $\Theta=\{w_0, w_1, \dots w_n, v_{1,1}, \dots v_{n,k}\}$ from the training set, different loss functions need to be defined according to different issues.

The time complexity of Eq. (1) is $O(kn^2)$ because the interaction $<v_i,v_j>$ between variables of different types is calculated. Therefore, the second-order model can be obtained by factorization and optimization into the form shown in Eq. (2):

$$\hat{y}(x) = w_0 + \sum_{i=1}^{n} w_i x_i + \frac{1}{2}\sum_{f=1}^{k}((\sum_{i=1}^{n} v_{i,f} x_i)^2 - \sum_{i=1}^{n} v_{i,f}^2 x_i^2) \tag{2}$$

The time complexity of Eq. (2) is reduced to $O(kn)$. Further, the time complexity is only $O(kp)$ in applications where the dataset is extremely sparse (the number of non-zero elements is set to p).

For each pair (x, y) in the observation dataset (S), Eq. (3) can be used to find the minimum of the error sum of all observations y and predicted values $\hat{y}(x \mid \Theta)$ to obtain the ideal parameter set Θ:

$$OPT(\mathrm{S}) = \operatorname{argmin} \Theta \sum_{(x,y)\in S} loss(\hat{y}(x \mid \Theta), y) \tag{3}$$

When the factorization dimension k is large, the L2-norm regularization term prevents the model from overfitting. Thus, employing the L2-norm regularization term, Eq. (3) becomes as follows:

$$OPT(S,\lambda) = \operatorname{argmin} \Theta(\sum_{(x,y)\in S} loss(\hat{y}(x \mid \Theta), y) + \sum_{\theta\in\Theta} \lambda_\theta \theta^2) \tag{4}$$

where λ_θ denotes the regularization coefficient and $\sum_{\theta\in\Theta}\theta^2$ denotes the L2-norm of a parameter set Θ. Moreover, according to Rendle et al. [Rendle, Gantner, Freudenthaler et al. (2011)], for $\forall\theta\in\Theta$, the factorization machine can be expressed as a linear combination of two functions:

$$\hat{y}(x) = \mathrm{g}_\theta(x) + \theta \mathrm{h}_\theta(x) \tag{5}$$

where g_θ and h_θ are independent of the value of parameter θ.

Then, according to Rendle et al. [Rendle, Gantner, Freudenthaler et al. (2011)], the optimization learning method is used to find the optimal parameters. The main idea is that each parameter is learned iteratively while other parameters are fixed until all parameters converge to the optimal solution.

3 Proposed model and algorithm

3.1 Data pre-processing methods

The collected data is pre-processed to construct the requirements of the factorization model and decomposition algorithm for teaching recommendation. Firstly, a teaching information data warehouse is constructed based on the fact constellation model by extract-transform-load (ETL) from several database tables, such as teacher information table, course information table, and teaching evaluation table. The structure of the data warehouse is shown in Fig. 1.

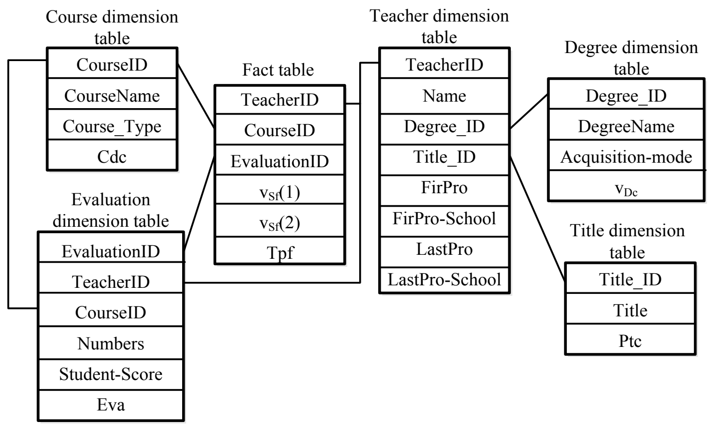

Figure 1: Pattern structure of the fact constellation

Then, we construct the following definitions to normalize the related attributes in the data warehouse.

Definition 1: School factor vector (v_{sf}, v_{sf}=[0.4 0.3 0.2 0.1]^T) is used to quantify the teacher's graduation schools. Vector elements represent the values of graduates from "985 Project" universities, "211 Project" universities, other first-tier universities, second-tier universities and below.

Definition 2: Degree coefficient (v_{Dc}, v_{Dc}=[0.4 0.3 0.2 0.1]^T) is used to quantify the degree earned by teachers. Each element represents the value of Ph.D., master, bachelor, and non-degree, respectively.

Definition 3: Graduation years (Gy, Gy≥0) represents the difference between the current year and the year of graduation (this paper takes the year of graduation of the final academic degree as the teacher's graduation year). That is, Gy=the current year-the year of graduation.

Definition 4: Attenuation function Af(Gy) (Gy$\in\mathbb{N}^+$, $0\le Af(\mathrm{Gy})\le 0.1$) means that with the growth of the teacher's graduation years (Gy), the professional foundation of teachers has a small decline. The function is defined as follows:

$$Af(\mathrm{Gy}) = \frac{1}{5(1+a^{-\mathrm{Gy}})} - s \tag{6}$$

where a is the constant coefficient of convergence rate. With a larger value of a, the function quickly converges to s. In this paper, a=1.3 and s=0.1 are used (the maximum attenuation value converges to 0.1).

Definition 5: Teacher professional foundation (Tpf) is used to quantify the professional foundation of a teacher defined as:

$$\mathrm{Tpf}=\sqrt{n}\sum_{\mathrm{i}=1}^{2}(\lambda_i\times \mathrm{Cv}_i)+\frac{1}{2}\sum_{i=1}^{2}v_{\mathrm{sf},i}+\mathrm{Dw}\times v_{\mathrm{Dc}}-\mathrm{Af(Gy)} \tag{7}$$

where $\lambda_i(i=1,2)$ represents the proportion of the i-th graduation major (generally considered the first and last graduation major) of a teacher, it is defined as follows:

$$\lambda_i=\frac{\mathrm{Cv}_i}{2\sum_{p=1}^{n}\mathrm{Cv}_p},\sum_{p=1}^{n}\sum_{i=1}^{2}\lambda_i=1 \tag{8}$$

where Cv_i represents the correlation coefficient between the teacher's i-th graduation major and the major he is engaged in, the total correlation is 1, and the correlation is r, the concrete definition is as follows:

$$Cv_{\mathrm{i}}=\begin{cases}1 \\ r\end{cases} i\text{=}1,2\ \ 0<r<1 \tag{9}$$

In Eq. (7), Dw(0<Dw≤1) is the teacher's degree factor. If the degree is obtained in full-time, Dw is 1, while if the degree is obtained part-time, Dw is a number between (0, 1) according to the total time.

To verify the validity of the definition, we select 40 teachers as sample data from a school in a second-tier university. The sample data are normalized and sorted by their professional foundation. As shown in Fig. 2, the obtained Tpf is differentiated. The result shows that the definition is reasonable.

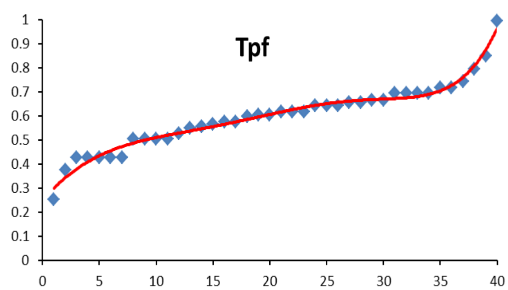

Figure 2: Tpf example

Definition 6: Course difficulty coefficient (Cdc, 0.1≤Cdc≤1) indicates the difficulty of the course. A higher value means a more difficult course. We adopt an online survey system of course difficulty coefficient classified by major, where a difficulty coefficient is scaled from 1 to 10. The respondents are graduates and expert teachers of relevant majors in second-tier or equivalent universities. The total recovered questionnaires for each major shall not be less than the minimum threshold of m (including k teachers' questionnaires and l students' questionnaires, $m=k+l$). Finally, we use min-max normalization to normalize the difficulty coefficient to the interval [$E_{\min}$, $E_{\max}$] ($E_{\min}$ and $E_{\max}$ represent the lowest and highest difficulty coefficients of the course respectively. $E_{\min}$=0.1 and $E_{\max}$=1.0 are used in this paper), which is given by:

$$\mathrm{Cdc} = \frac{\mathrm{Qs} - \mathrm{Qs}_{\min}}{\mathrm{Qs}_{\max} - \mathrm{Qs}_{\min}} \times (E_{\max} - E_{\min}) + E_{\min} \quad (10)$$

where $\mathrm{Qs}_{\min}$ and $\mathrm{Qs}_{\max}$ are the minimum and maximum difficulty values of the courses investigated in a particular major. Qs is defined as:

$$\mathrm{Qs} = \frac{w}{k}\sum_{i=1}^{k}\mathrm{Cd}_i + \frac{(1-w)}{l}\sum_{i=1}^{l}\mathrm{Cd}_i \quad (11)$$

where w ($0<w<1$) is the weight of the expert teacher questionnaire survey and Cd_i is the difficulty coefficient value given in the questionnaire for the i-th course.

Definition 7: Teaching Comprehensive Evaluation (Eva, $0.1\leq \mathrm{Eva}\leq 1$) is used to normalize the score of the course taught by the teacher. We adopt the min-max normalization to normalize Eva to the interval [$E_{\min}$, $E_{\max}$] ($E_{\min}$ and $E_{\max}$ represent the minimum and maximum evaluation values respectively. $E_{\min}=0.1$ and $E_{\max}=1.0$ are used in this paper). A comprehensive evaluation value is defined as follows:

$$\mathrm{Eva} = \frac{\mathrm{Aver} - \mathrm{Aver}_{\min}}{\mathrm{Aver}_{\max} - \mathrm{Aver}_{\min}} \times (E_{\max} - E_{\min}) + E_{\min} \quad (12)$$

where $\mathrm{Aver}_{\min}$ and $\mathrm{Aver}_{\max}$ are the lowest and highest evaluation scores of all courses in the major. Aver represents the average score of student evaluations for the same course taught by a teacher over M semesters defined as follows:

$$\mathrm{Aver} = \sum_{m=1}^{M}(\mathrm{Qty}_m \times \mathrm{Aver}_m) \Big/ \sum_{m=1}^{M}\mathrm{Qty}_m \quad (13)$$

where Qty_m represents the number of students participating in the evaluation of the course in semester m ($1\leq m\leq M$), and Aver_m ($0<\mathrm{Aver}_m\leq 100$) represents the average score of the course in semester m.

3.2 TRFMs model and recommendation algorithm

Five variables are used to construct $x^{(i)}$ in FMs model feature vector $X=\{x^{(1)}, x^{(2)},\ldots, x^{(n)}\}$: Teacher, Course, Teacher Professional Foundation, Course Difficulty Coefficient, and Teaching Comprehensive Evaluation, n is the total number of feature vectors. We use Tpf, Cdc, and Eva to represent the corresponding eigenvalues of teachers' professional foundation, course difficulty coefficient, and teaching comprehensive evaluation, respectively. Then, we use the partial recommendation value $y^{(i)}$ calculated by Eq. (14) to construct the recommendation label in the target vector $Y=\{y^{(1)}, y^{(2)}, \ldots, y^{(n)}\}$ of FMs:

$$y^{(i)} = \rho_1\mathrm{Tpf} + \rho_2\mathrm{Cdc} + \rho_3\mathrm{Eva},\ \sum_{i=1}^{3}\rho_i = 1 \quad (14)$$

where ρ_1, ρ_2, and ρ_3 are partial weights. If $\rho_1>\rho_2$ and $\rho_1>\rho_3$, the recommendation basis lays particular stress on teachers' professional foundation. We name such a Factorization Machine model as Teaching Recommendation Factorization Machines (TRFMs).

Then, we constitute the TRFMs dataset by the feature vector X and the target vector Y. An example of this dataset is shown in Fig. 3.

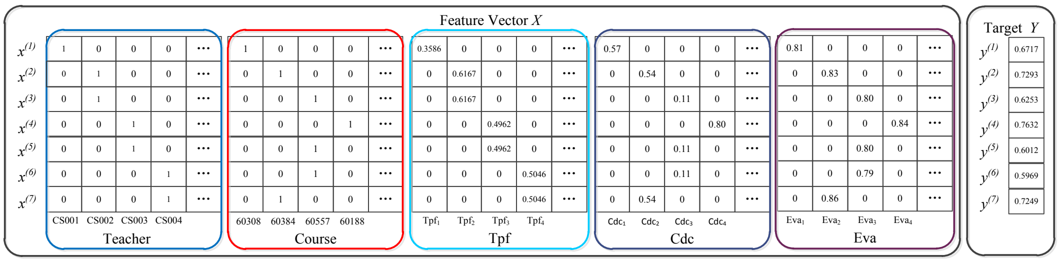

Figure 3: TRFMs model dataset example

In practical application, a course set is a large number of sets, and so is the teacher set. However, the courses taught by each teacher only account for a limited number of elements in the course set, which is bound to cause that the majority of elements in the TRFMs data set are 0. Thus, the TRFMs data set is highly sparse.

We propose an improved recommendation algorithm based on the alternative least square method [Rendle, Gantner, Freudenthaler et al. (2011)]. After learning the optimal parameters of TRFMs from the training set, we calculate the recommendation accuracy of the model under different test sets and the TOP_N teacher recommendation accuracy classified by the specified course. The TRFMs recommendation algorithm is described in Algorithm 1.

Algorithm 1: TRFMs recommendation algorithm

Input: historical data in teaching information data warehouse O, degree obtaining factor Dw, professional correlation coefficient r, expert teacher questionnaire weight w, partial weight ρ_1, ρ_2 and ρ_3, regularization parameter λ, initialize the standard deviation parameter σ.

Output: Root Mean Squared Error (RMSE) of different test sets under the optimized parameter $\Theta=\{w_0, w_1,\ldots w_n, v_{1,1},\ldots v_{n,k}\}$ obtained from the training of the algorithm, as well as TOP_N recommendation and P@N sorting accuracy by course classification.

1: The historical teaching record data O in the teaching information data warehouse is normalized by the given Dw, r, w according to the definitions 1-7, and the recommended label $y^{(i)}$ is calculated according to Eq. (14) according to the given $\rho 1$, $\rho 2$, and $\rho 3$.

2: The pre-processed data is constructed into the experimental data set E required by TRFMs, and divided into training set $T1$ and test set $T2$ according to 80% and 20% of each course.

3: Initialization parameters $w_0=0$, $(w_1, \ldots, w_n)=(0, \ldots, 0)$, $(v_{1,1}\ldots, v_{n,k})\sim N(0, \sigma)$

4: for $(x,y)\in T1$ do // Pre-calculate e and q

5: $\quad e(x,y\,|\,\Theta)\leftarrow \hat{y}(x,y)-y$

6: $\quad$ for $f\in\{1,\ldots,k\}$ do // $k\in\mathbb{N}^+$($k \ll n$)is called hyper-parameter

7: $\qquad q(x,f\,|\,\Theta)\leftarrow \sum_{i=1}^{n} v_{i,f}x_i$

8: $\quad$ end for

---

end for

repeat//loop optimization model parameters

$w_0^* \leftarrow -\frac{\sum_{(x,y)\in T1}(e(x,y|\Theta)-w_0)}{|T1|+\lambda_{(w_0)}}$

$e(x,y|\Theta^*) \leftarrow e(x,y|\Theta)+(w_0^*-w_0)$

$w_0 \leftarrow w_0^*$ //Update the value of w_0

for $l\in\{1, \ldots, n\}$ do

$\quad w_l \leftarrow -\frac{\sum_{(x,y)\in T1}(e(x,y|\Theta)-w_l x_l)x_l}{\sum_{(x,y)\in T1}x_l^2+\lambda_{(w_l)}}$

$\quad e(x,y|\Theta^*) \leftarrow e(x,y|\Theta)+(w_l^*-w_l)x_l$

$\quad w_i \leftarrow w_i^*$ //Update the value of w_l

end for

for $f\in\{1, \ldots,k\}$ do

$\quad$ for $l\in\{1, \ldots, n\}$ do

$\quad\quad w_{l,f}^* \leftarrow -\frac{\sum_{(x,y)\in T1}(e(x,y|\Theta)-v_{l,f}h_{(v_{l,f})}(x))h_{(v_{l,f})}(x)}{\sum_{(x,y)\in T1}h^2_{(v_{l,f})}(x)+\lambda_{(v_{l,f})}}$

$\quad\quad e(x,y|\Theta^*) \leftarrow e(x,y|\Theta)+(v_{l,f}^*-v_{l,f})x_l$

$\quad\quad q(x,f|\Theta^*) \leftarrow q(x,f|\Theta)+(v_{l,f}^*-v_{l,f})x_l$

$\quad\quad v_{l,f} \leftarrow v_{l,f}^*$;

$\quad$ end for

end for

until(all parameters converge)

Use the test set *T2* to verify the RMSE of the model after optimized parameters.

Top_N recommendation is generated according to course classification and calculation of P@N sorting accuracy.

The time overhead in the algorithm is mainly used to update the model parameters according to the training set iteratively. Because the algorithm pre-calculates the intermediate variables $e\in\mathbb{R}^n$ and $q\in\mathbb{R}^{n\times k}$, the time complexity is $O(k|T1|\overline{M}_{|T1|})$ where $|T1|$ is the training set size and $\overline{M}_{|T1|}$ is the average number of non-zero elements in the input feature vector *X*.

4 Experimental evaluations

4.1 Experimental data

The experimental data is collected from the teaching data of a second-tier university in recent years. The data includes 928 teachers after desensitization, 2983 courses, and 1,956,632 valid evaluation scores obtained by averaging the student evaluation scores of courses taught by each teacher. During the normalization process of these data, we set different specific weight coefficients r, Dw, w, ρ_1, ρ_2, and ρ_3 to obtain different experimental data. In this paper, according to the practical situation of the second-tier university, we select weight coefficients r=0.7, Dw=0.4, w=0.4, ρ_1=0.2, ρ_2=0.2, and ρ_3=0.6 for normalization processing to obtain the experimental data set. The data are summarized in Tab. 1.

Table 1: Samples of experimental data

TeacherID	Tpf	CourseID	Cdc	Evaluation	$y^{(i)}$
CS001	0.3586	60308	0.57	0.81	0.6717
CS002	0.6167	60384	0.54	0.83	0.7293
CS002	0.6167	60557	0.11	0.80	0.6253
CS003	0.4962	60188	0.80	0.84	0.7632
CS003	0.4962	60557	0.11	0.80	0.6012
CS004	0.5046	60557	0.11	0.79	0.5969
CS004	0.5046	60384	0.54	0.86	0.7249

4.2 Evaluation index

Root Mean Squared Error (RMSE) evaluation index proposed in Zhu et al. [Zhu and Lu (2012)] is adapted to measure the accuracy of recommended experiments. The RMSE is defined as follows:

$$\text{RMSE} = \sqrt{\sum_{\text{tc}\in T_E} (y_{\text{tc}} - y^{*}_{tc})^2 \Big/ |T_E|} \tag{15}$$

where $|T_E|$ indicates the size of the test set T_E. y_{tc} and y^{*}_{tc} indicate the ground-truth and predicted recommended label values of course c taught by teacher t in the test set.

Besides, we use P@N [Wang, Meng, Zhang et al. (2010)] to evaluate the relevance of the first N recommended teachers for each course. Because the experimental dataset is very sparse, the N values in this experiment only consider the first three, the first four and the first five values. P@N is suitable to evaluate TOP_N recommendation defined as follows:

$$\text{P@N} = \frac{\#\textit{relevant items in the}\ \text{TOP\_N}\ \textit{items}}{\text{N}} \tag{16}$$

4.3 Experiment results and analysis

To ensure that each course has data in both the training set and test set, in the TRFMs model experiment dataset (E), we choose 20% of the data of each course as the test set (T_E), and the remaining 80% of the data (E- T_E) as the training set.

4.3.1 Algorithm accuracy and sorting accuracy

In the experiment, firstly, the TRFMs optimization parameter Θ is trained according to the Algorithm 1. Then, teachers are recommended according to the classification of courses. Here, the initialization standard deviation parameter σ=0.01 is used following Rendle [Rendle (2012)], and the regularization parameter λ is obtained by the adaptive selection method proposed in Rendle [Rendle (2012)].

The accuracy of the teaching recommendation algorithm is compared with the factor decomposition dimension hyper-parameter k=8, 12, 16, 20, 24, 28, 32, 36. Theoretically, the parameter k is required to be large enough. A small value of k is used in the experiment since the experimental data is highly sparse so that not enough sample is given to estimate the complex interaction matrix. We conduct ten experiments for different k values, and then their mean value is used as the result value to obtain the precision comparison chart as shown in Fig. 4:

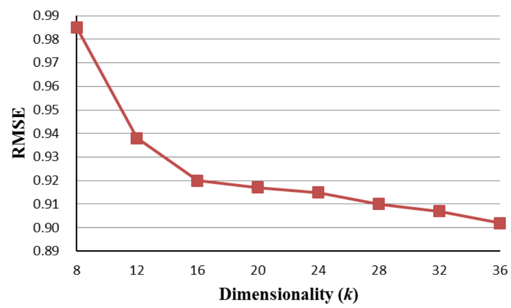

Figure 4: RMSE under different k

Fig. 4 shows that the algorithm can usually recommend from a highly sparse experimental dataset E, and k profoundly affects the recommendation accuracy. If k is small, RMSE is high (i.e., low accuracy). As k increases, RMSE gradually increases correspondingly.

We further sort out the experimental results and obtain the comparison of the sorting accuracy of the algorithm at P@N when the dimension hyper-parameter k is 8, 16 and 32, and N is 3, 4, and 5. The results are shown in Fig. 5.

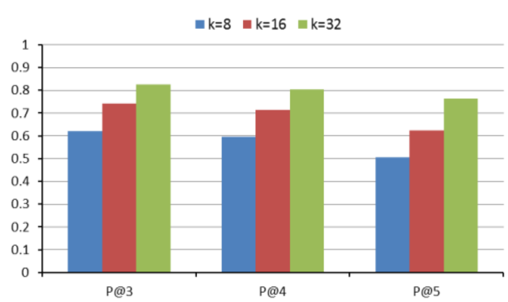

Figure 5: Comparison of P@N under different k and N

As shown in Fig. 5, the larger the value of k is, the higher the ranking accuracy is, while the smaller the value of N is, the higher the sorting accuracy is under a reasonable range of the dimension super-parameter k.

Through the above experiment, we verified that the TRFMS algorithm provides a reasonable range of RMSE and P@N values for highly sparse teaching recommendation data sets. It indicates that the TRFMS algorithm correctly predicts the recommended label y^*_{tc} of each teacher for a given course, and recommends the teachers with the highest TOP_N predictive value y^*_{tc} for a given course. Consequently, the teaching effect will be improved.

4.3.2 Recommended accuracy comparisons between TRFMs and HOSVD

Tensor decomposition is also known as high order singular value decomposition (HOSVD). The Tucker decomposition model [Tucker (1966)] decomposes the three-dimensional tensor X into the product of a low-rank eigenvalue matrix on three dimensions and a core tensor: $X \approx \widehat{X} = C \times_i V^{(i)} \times_j V^{(j)} \times_k V^{(k)}$. C is the compressed tensor of tensor X which is much smaller than the original tensor and has a significant effect on a sparse dataset.

The three-dimensional tensor recommended by teachers is constructed with the experimental data described in Tab. 1. The three-dimensional tensor $X \in R^{I_t \times I_c \times I_e}$ is constructed with dimensions T, C, and E according to the four-tuples relationship (t_i, c_j, e_k, $y^{(i)}$) of "Teacher(T)-Course(C)-Evaluation(E)-Recommendation Label (Y)". The corresponding element index is (t_i, c_j, e_k), and the corresponding element value is $y^{(i)}$ with a partial recommendation value calculated by Eq. (14). If there is a teacher (t_i) with a certain teacher professional foundation of Tpf_i who teaches a course (c_j) with a difficulty coefficient of Cdc_j and gets an evaluation score (e_k) of Eva_k, then the element value of tensor corresponding to the subscript (t_i, c_j, e_k) is the weighted $y^{(i)}$ value. Otherwise, the corresponding element value is 0.

Then, we adopt the Tucker tensor decomposition method [Tucker (1966)] to obtain the approximate tensor after the dimension reduction and generate the Top-N recommendation list classified by the course according to the size of the approximate tensor element values. The time complexity of the algorithm consists of the complexity $\text{O}(\sum_{i=1, i \neq n}^{3} (I_n R_n \prod_{j=1}^{n-1} R_j \prod_{j=n+1}^{3} I_j))$ of calculating the core tensor C in each iteration, the complexity $\text{O}\left(I_n \prod_{j=1, j \neq n}^{3} R_j R_n^2 \right)$ of performing SVD calculation on $\hat{X}$, and the modular multiplication complexity of solving the approximate tensor $\hat{X}$ (the same as solving the core tensor). In the algorithm, the dimension I_n of tensor X is much larger than the dimension R_n of decomposition factor; thus, the complexity of the algorithm can be reduced to $\text{O}\left(\prod_{i=1}^{3} I_i \right)$.

In the experiment, we used the same parameters as in the previous experiment for the TRFMs algorithm but fixed the factorization dimension hyper-parameter k=24. For the HOSVD, we fixed the iteration threshold of $\mathcal{E}$=0.0005 [Kolda and Bader (2009)]. Then we took 60%, 70%, 80%, 90%, and 100% of E-T_E as training sets (the sparsity of training sets varies with the size of training sets) to compare their recommendation accuracy. Through the experiment, we obtained a comparison between their recommended accuracy and the running time of each iteration, as shown in Figs. 6 and 7. When N is 3, 4 and 5, we obtained the comparison of P@N sorting accuracy, as shown in Fig. 8:

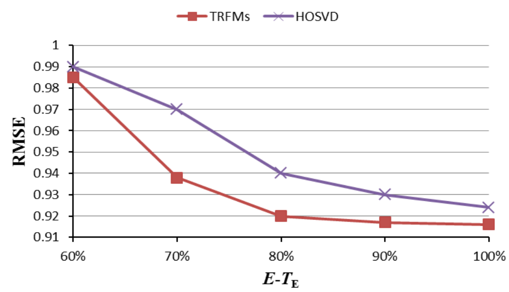

Figure 6: Recommended accuracy for different size training sets

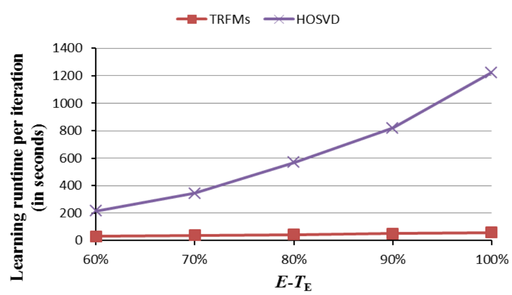

Figure 7: Run-time of each iteration under different size training sets

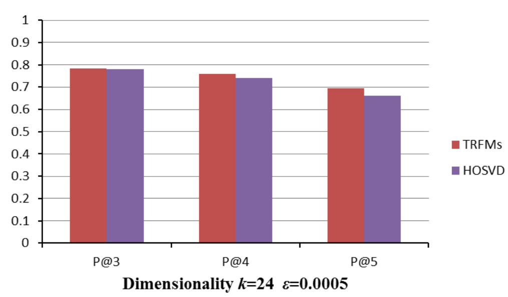

Figure 8: Recommended sorting accuracy for different P@N

The results show that the recommendation accuracy of both models is improved with the increase in the training set. However, the recommendation accuracy obtained by the TRFMs model algorithm is slightly higher than that of the HOSVD model algorithm. The recommended sorting accuracy of the P@N is also slightly higher than the HOSVD model algorithm. The running time is also much lower than that of the HOSVD model algorithm. Those indicate that the recommended performance of TRMFs is superior to HOSVD for a highly sparse dataset.

4.3.3 Recommended differences comparisons under different specific weight factors

To compare the recommended differences of experimental datasets under different specific weight coefficients, we designed a series of experiments. In each experiment, we fixed the dimensional hyper-parameter k=24 of the TRMFs algorithm and compared their recommendation differences of Top_5 in the same course. The specific design is as follows:

(1) Comparison of the effects of different r and Dw on the recommended results: We fixed w=0.4, optional r=0.7 and Dw=0.4 to construct the experimental dataset *E1* while optional r=0.5 and Dw=0.2 to construct the experimental dataset *E2*. We completed TRFMs-based teaching recommendation experiments for *E1* and *E2* respectively, and the comparison results are shown in Tab. 2.

Table 2: Comparison of different r and Dw recommended (CourseID=60264)

E1		***E2***	
TeacherID	y^{*}_{tc}	**TeacherID**	y^{*}_{tc}
CS022	0.6004	CS011	0.5862
CS009	0.5855	CS009	0.5752
CS035	0.5803	CS022	0.5726
CS012	0.5781	CS012	0.5542
CS011	0.5781	CS035	0.5524

Tab. 2 shows that the same course under different proportion coefficients r and DW (i.e., in different Tpf) provide different predicted recommended label values y^{*}_{tc}; that is, the recommended order of teachers is different. Through the analysis, it can be seen from Eqs. (6)-(8) that there are inconsistencies in each teacher's professional relevancy and degree obtaining method. That is, different values of Cv_i and Dw lead to different values of Tpf. In this way, the order of teachers recommended for the same course will also be different. Therefore, different values of r and DW have a significant impact on the recommendation results.

(2) Comparison of the effects of different specific weight coefficients w on the recommended results: We fixed r=0.7, Dw=0.4, optional w=0.3 to construct the experimental dataset *E3* while optional w=0.7 to construct the experimental dataset *E4*. we completed TRFMs-based teaching recommendation experiments for *E3* and *E4* respectively, and the comparison results are shown in Tab. 3.

Table 3: Recommended comparison of different w-values (CourseID=60264)

E3		***E4***	
TeacherID	y^{*}_{tc}	**TeacherID**	y^{*}_{tc}
CS012	0.4451	CS012	0.5220
CS022	0.4405	CS022	0.5174
CS035	0.4386	CS035	0.5155
CS004	0.4373	CS004	0.5142
CS011	0.4283	CS011	0.5052

Tab. 3 shows that the predicted value of recommendation label y^*_{tc} can be changed with the change of *w* (that is, the change of Cdc value) while other data and the specific weight coefficient remain unchanged. However, there is no change in the recommendation results of Top_5 teachers in the same course. As can be seen from Eqs. (9) and (10), only a change of *w* will change the Cdc value of the course and the predicted value. However, when recommending teachers for the same course, a change of the Cdc value does not affect the order of recommendation. All the experimental results show that the TRFMs and the recommendation algorithm can accurately implement the course teaching teacher recommendation according to different focuses.

5 Conclusion

This paper proposes TRFMs and the recommendation algorithm to address the lack of scientific basis for teaching arrangement. Several normalized factors are defined: teachers' professional foundation, course difficulty coefficient and teaching comprehensive evaluation value. Based on the factors, a comprehensive recommendation value is computed using partial weights with historical teaching data. A highly sparse teaching recommendation dataset is constructed where Teacher, Course, Tpf, Cdc, and Eva are used as the attributes of feature vector *X*, and the comprehensive recommendation value is used as target vector *Y*. Then, TRFMs model and the recommendation algorithm are proposed for accurate teacher recommendation. The experimental results show that the proposed methods can be a new solution for the school course to realize intelligent and accurate recommendations of teaching teachers. Also, the proportion coefficients of the proposed methods can be adjusted to fit the situation of the target school, which leads to ideal recommendation results and effectively improve teaching quality.

Acknowledgement: We would like to thank the Huaihua University Double First-Class initiative Applied Characteristic Discipline of Control Science and Engineering. It provides experimental equipment and environmental support for this work. I would also like to thank the reviewers and editors. We successfully completed the format requirements of the manuscript with their helpful advice and comments.

Funding Statement: This work was supported by the Planning Subject for the 13th Five-Year Plan of Hunan Provincial Educational Sciences under Grant XJK17BXX006, author D. Y, http://ghkt.hntky.com/.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Adomavicius, G.; Tuzhilin, A. (2005): Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734-749.

Alemu, M. H.; Olsen, S. B.; Vedel, S. E.; Pambo, K. O.; Owino, V. O. (2017): Combining product attributes with recommendation and shopping location attributes to assess consumer preferences for insect-based food products. *Food Quality and Preference*, vol. 55, pp. 45-57.

Cacheda, F.; Carneiro, V.; Fernández, D.; Formoso, V. (2011): Comparison of collaborative filtering algorithms: limitations of current techniques and proposals for scalable, high-performance recommender systems. *ACM Transactions on the Web*, vol. 5, no. 1, pp. 1-33.

Chen, C.; Hou, C.; Xiao, J.; Yuan, X. (2016): Purchase behavior prediction in E-commerce with factorization machines. *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 1, pp. 270-274.

Chen, Y.; Xiong, J.; Xu, W.; Zuo, J. (2019): A novel online incremental and decremental learning algorithm based on variable support vector machine. Cluster Computing, vol. 22, no. 8, pp. 7435-7445.

Geuens, S. (2015): Factorization machines for hybrid recommendation systems based on behavioral, product, and customer data. *Proceedings of the 9th ACM Conference on Recommender Systems*, pp. 379-382.

Guan, X.; Li, C. T.; Guan, Y. (2017): Matrix factorization with rating completion: An enhanced SVD model for collaborative filtering recommender systems. *IEEE Access*, vol. 5, pp. 27668-27678.

He, J.; Fang, X.; Liu, H.; Li, X. (2019): Mobile app recommendation: An involvement-enhanced approach. *MIS Quarterly*, vol. 43, no. 3, pp. 827-849.

Jiang, W; Chen, J; Jiang, Y.; Xu, Y.; Wang, Y. et al. (2019): A new time-aware collaborative filtering intelligent recommendation system. *Computers, Materials & Continua*, vol. 61, no. 2, pp. 849-859.

Kolda, T. G.; Bader, B. W. (2009): Tensor decompositions and applications. *SIAM Review*, vol. 51, no. 3, pp. 455-500.

Kumar, R.; Verma, B. K.; Rastogi, S. S. (2014): Social popularity based SVD++ recommender system. *International Journal of Computer Applications*, vol. 87, no. 14, pp. 33-37.

Meng, X. W.; Ji, W. Y.; Zhang, Y. J. (2015): A survey of recommendation systems in big data. *Journal of Beijing University of Posts and Telecommunications*, vol. 38, no. 2, pp. 1-15.

Nguyen, T. V.; Karatzoglou, A.; Baltrunas, L. (2014): Gaussian process factorization machines for context-aware recommendations. *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 63-72.

Qin, Z. (2015): Research progress on educational data mining: a survey. *Journal of Software*, vol. 26, no. 11, pp. 3026-3042.

Rendle, S. (2012): Factorization machines with libFM. *ACM Transactions on Intelligent Systems and Technology*, vol. 3, no. 3, pp. 1-22.

Rendle, S. (2012): Learning recommender systems with adaptive regularization. *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, pp. 133-142.

Rendle, S.; Freudenthaler, C.; Schmidt-Thieme, L. (2010): Factorizing personalized Markov chains for next-basket recommendation. *Proceedings of the 19th International Conference on World Wide Web*, pp. 811-820.

Rendle, S.; Gantner, Z.; Freudenthaler, C.; Schmidt-Thieme, L. (2011): Fast context-aware recommendations with factorization machines. *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 635-644.

Rendle, S.; Schmidt-Thieme, L. (2010): Pairwise interaction tensor factorization for personalized tag recommendation. *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, pp. 81-90.

Resnick, P.; Varian, H. R. (1997): Recommender systems. *Communications of the ACM*, vol. 40, no. 3, pp. 56-58.

Thai-Nghe, N.; Drumond, L.; Horváth, T.; Schmidt-Thieme, L. (2012): Using factorization machines for student modeling. *Proceedings of FactMod at the 20th Conference on User Modeling, Adaptation and Personalization.*

Tucker, L. R. (1966): Some mathematical notes on three-mode factor analysis. *Psychometrika*, vol. 31, no. 3, pp. 279-311.

Wang, L.; Meng, X.; Zhang, Y.; Shi, Y. (2010): New approaches to mood-based hybrid collaborative filtering. *Proceedings of the Workshop on Context-Aware Movie Recommendation*, pp. 28-33.

Yu, H. F.; Hsieh, C. J.; Si, S.; Dhillon, I. (2014): Parallel matrix factorization for recommender systems. *Knowledge and Information Systems*, vol. 41, no. 3, pp. 793-819.

Xu, H. L.; Wu, X.; Li, X. D.; Yan, B. P. (2009): Comparison study of Internet recommendation system. *Journal of Software*, vol. 20, no. 2, pp. 350-362.

Zhu, Y. X.; Lu, L. Y. (2012): Evaluation metrics for recommender systems. *Journal of University of Electronic Science and Technology of China*, vol. 41, no. 2, pp. 163-175.