PAPER Special Section on Data Engineering and Information Management

Improve the Prediction of Student Performance with Hint's Assistance Based on an Efficient Non-Negative Factorization

Ke XU[†,††], *Member*, Rujun LIU[†], Yuan SUN[†††a)], Keju ZOU[††††], Yan HUANG[†], *and* Xinfang ZHANG[†], *Nonmembers*

SUMMARY In tutoring systems, students are more likely to utilize hints to assist their decisions about difficult or confusing problems. In the meanwhile, students with weaker knowledge mastery tend to choose more hints than others with stronger knowledge mastery. Hints are important assistances to help students deal with questions. Students can learn from hints and enhance their knowledge about questions. In this paper we firstly use hints alone to build a model named Hints-Model to predict student performance. In addition, matrix factorization (MF) has been prevalent in educational fields to predict student performance, which is derived from their success in collaborative filtering (CF) for recommender systems (RS). While there is another factorization method named non-negative matrix factorization (NMF) which has been developed over one decade, and has additional non-negative constrains on the factorization matrices. Considering the sparseness of the original matrix and the efficiency, we can utilize an element-based matrix factorization called regularized single-element-based NMF (RSNMF). We compared the results of different factorization methods to their combination with Hints-Model. From the experiment results on two datasets, we can find the combination of RSNMF with Hints-Model has achieved significant improvement and obtains the best result. We have also compared the Hints-Model with the pioneer approach performance factor analysis (PFA), and the outcomes show that the former method exceeds the later one.

key words: *student performance, hints, matrix factorization, non-negative matrix factorization, RSNMF*

1. Introduction

Intelligent tutoring systems (ITS) are based on models which relate knowledge components that students study to different questions [8]. Assistance is one of these platforms that have been introduced for a decade. In recent years, the assistance system has become more popular in teaching and study environment. It is worthwhile to predict student performance after students has interacted with the system. So we can generate questions of those students which could be used to predict to their study performance, and which are also more efficient to improve students' knowledge because of knowing which skills they have not grasped. However, predicting student performance (PSP) has been a complicated work for tutors in the education field. There are so many factors that can influence the students' performance on questions other than students' ability.

There are several student modeling methods which have been widely utilized to analyze study performance, such as Bayesian knowledge tracing (BKT) [3] and performance factor analysis (PFA) [9]. The BKT assumes that student knowledge is represented as a set of binary variables, one per skill, where the skill is either mastered by the student or not. The PFA model provides the adaptive flexibility to create the model overlay that could be used adaptively in a tutor, and it also retains the advantages of data mining technique that could be used in search procedures. Distinct with the above methods, factorization method for PSP is derived from recommender systems (RS). RS has been paid attention to for several years which focuses on recommending items suit for users to achieve user stickiness. One core technique of RS is collaborative filtering (CF). There are two means of achieving the sketch of this technique. One traditional method is based on neighborhood similarity and the other state-of-the-art method is matrix factorization, which is utilized by Koren et al. [1] in the Netflix competition. Both of the two methods tend to predict the rate of user for the item. Student model and performance prediction is similar to this problem. So we use factorization methods in performance modeling and prediction in the paper. However, many previous approaches use direct students' first attempt results of the questions, which are not able to consider the specific procedures between two attempts, for example, the hint attempt times on the same question.

In this paper, we assume firstly that students with more hints per question tend to have poorer performance on the questions, and in the meanwhile problems with more hints per student tend to be more difficult than problems which have fewer hints per student. The last experiment is to combine Hints-Model with the factorization method using logistic regression to achieve improvement. We compare performance results by using different metrics, namely the root mean square error (RMSE) and the mean accuracy of the approach.

2. Related Work

Recently some student modeling methods introduce data mining technique in the solution strategy. The traditional

Manuscript received June 29, 2016.
Manuscript revised November 6, 2016.
Manuscript publicized January 17, 2017.

[†] The authors are with Huazhong University of Science and Technology, Wuhan, 430074 China.
[††] The author is with South-Central University for Nationalities, Wuhan, 430074 China.
[†††] The author is with Hubei University of Traditional Chinese Medicine, Wuhan, 430065 China.
[††††] The author is with Sun Yat-Sen University, Guangzhou, China.
a) E-mail: elyn1202@hbtcm.edu.cn (Corresponding author)

DOI: 10.1587/transinf.2016DAP0026

method BKT for PSP has explicitly set four parameters to model the specific skill: L_0 (the prior probability that the skill has been learned), L (the probability from unlearned state to learned state), S (the probability that student grasps the skill but get wrong answer), G (the probability that student does not grasp the skill but get right answer). We predict students' study performance according to the Bayesian inference theory. In order to fit the parameters, we consider two popular approaches including expectation maximization (EM) [21] and brute force (BF). So we can easily build BKT model with BNT toolkit [19] and regard the BKT skill model as a hidden Markov model (HMM) [22]. The PFA model needs to set three parameters per skill for a single knowledge component refers to one question step. One is d indicated skill difficulty, the second is l denoted learn rate from one success and the third is l_f represented the effect from one failure. They use logistic regression to fit the data. The PFA is sensitive to students' success and failures, which is the most significant difference from learning factor analysis (LFA) [17]. Besides the pioneer models BKT and PFA, techniques of recommend system have also been utilized for students' performance prediction. Comparisons have been made among PFA, BKT, BPMF and BPTF [6] in the parametric questions to discuss the temporal influence [7].

Some traditional approaches have been utilized in education data mining field. Thai-Nghe et al. [2], [24] has utilized the MF to predict student performance on the two datasets from KDD Cup Challenge 2010. They have got good results on the two large datasets. Š. Pero et al. [23] has utilized several approaches from RS to predict student performance on small-scale datasets and obtain the result which deviates from the one got on the large-scale datasets in Thai-Nghe's research. In addition to the MF model, we can see another popular method NMF which is widely used in image processing and text mining proposed in Ref. [4]. The NMF is so popular because of its easiness of results to interpret. In recent researches, Michel C. Desmarais [5] has mapped items to skills like a cluster method, and the result can be described as a Q-matrix [16]. This has encouraged us to solve the study performance prediction problem by using the NMF to PSP. Michel's NMF model does not implicitly encode "slip" and "guess" factor. For there is no negative element in factor matrices, it assumes that the skill can only be learned and the grasp of skill has a non-negative effect on scores. In Michel's later work [20], he takes a specific example of using MF to get Q-matrix and compare it with the expert Q-matrix.

Despite the NMF puts good interpretation on theory, it has obviously low efficiency on large datasets and is inadequate for sparse matrix because of its multiplicative update rules [18] for optimization. So we introduce an efficient RSNMF approach [11] which is derived from the WNMF [10] designed for large and sparse matrices and improves greatly in efficiency than the WNMF algorithm. In order to achieve a better result, many researchers utilize some ensemble methods to take advantage of those approaches. A comprehensive description of ensemble methods [12] has taken into consideration the most well known ensemble approaches.

There is some useful information in assistance that can be a good representation of student performance. For example, the response time has been discussed by some researchers [13], [14] and is combined with other model's results using ensemble methods mentioned above, and it results in improvement in varying degrees. The hint is another good indicator of student performance which will be discussed in later chapter. Unlike Ref. [15] which also takes the hints into account by data driven results, we make a linear model with the hints called Hints-Model and get a good performance and interpretable result. The hint also inspires some researchers to study partial credits [25] which may be closer to our practical condition in study.

3. Method

3.1 Hints-Model

We utilize the number of hints as an indication of student performance. The idea was derived from the user-item biased approach [1] and our reasonable assumption about hint is based on previous work [15]. For all competition datasets in KDD Cup 2010 [26], each record will be a step that contains many attributes including anon student id, problem hierarchy, problem name, problem view, step name, and et al. We consider combined attribute as the characterization of knowledge component referred to [26]. The combined attribute named "problem-view-step", which is composed of problem name, problem view and step name and similar to the data of KDD Cup 2010. We construct a matrix in which each row represents a student and each column represents a "problem-view-step" group which is denoted as an item in the model. For each element in the matrix, the default value is -1. If a student obtains a "HINT" at his or her first attempt on the item, we change the element to be 1. To the contrary, we change it to be 0.

For the original hints number used in model that may result in huge calculating size, overfitting and poor stable, we consider the average number of hints as characteristic representation. We firstly should get the average hints number for each student gets on all the questions it answered. We can denote the average hints number of the u^{th} student obtains as sa_u, and the average hints number of each item can be gained in the same way, and we can denote the hints number of the i^{th} item as ia_i. Secondly, we should normalize these hints number by dividing the maximum minus the minimum of the hints statistics of sa and ia respectively. The coefficients vector of sa, ia can be set as sa_coef, ia_coef respectively. So we can get a linear model:

$$\mathrm{T}'_{u,i} = \mathrm{sa\_coef}_u * \mathrm{sa}_u + \mathrm{ia\_coef}_i * \mathrm{ia}_i + 0.5 \tag{1}$$

In the Eq. (1), $\mathrm{T}'_{u,i}$ represents the prediction value of u^{th} row and i^{th} column element of the data matrix. The constant is set to 0.5 for it is the middle of 0 and 1. So the sign of the sum of the former part in the equation will determine whether it belongs to the positive or negative class. We hope to find an approximate $\mathrm{T}'_{u,i}$ to replace the original value of u^{th} row and i^{th} column element $\mathrm{T}_{u,i}$. The final target of the model is to fit the data by modifying the coefficients. The objective function is:

$$\mathrm{Min}_{\mathrm{sa\_coef,\,ia\_coef}}((\mathrm{T}_{u,i} - \mathrm{T}'_{u,i})^2 + \lambda * (||\mathrm{sa\_coef}||^2 + ||\mathrm{ia\_coef}||^2)) \tag{2}$$

We can utilize the gradient descent method to fit the model, and obtain proper coefficients and utilize the 5-fold cross-validation to select a model which performs better under specific criterions.

We looked at four statistics of coefficients for quantifying how easy these problems for the student to solve:

(1) Positive sa_coef: average number of positive elements of sa_coef.

(2) Total sa_coef: average number of elements of sa_coef.

(3) Positive ia_coef: average number of positive elements of ia_coef.

(4) Total ia_coef: average number of elements of ia_coef.

In order to compare the proposed model to others, we utilize those four common measures including TN, FN, TP and FP. TN is the abbreviate of True Negative, FN denotes False Negative, TP denotes True Positive and FP denotes False Positive.

3.2 Matrix Factorization (MF) and Regularized Single-Element-Based NMF (RSNMF)

Matrix T is the partially observed scoring matrix, $W \in R^{U \times K}$ is a matrix where each row u is a vector containing the K latent factors describing the student u, and $H \in R^{K \times I}$ is a matrix where each column i is a vector containing the K factors describing the item (“problem-view-step” group) i [1], [2]. So the observed matrix T can be approximately replaced by the multiplication of the matrix W and matrix H, the objective function is:

$$\mathrm{Min}_{W,H}((\mathrm{T}_{u,i} - \mathrm{T}'_{u,i})^2 + \lambda * (||W||_F^2 + ||H||_F^2)) \tag{3}$$

Where $\mathrm{T}'_{u,i} = (WH)_{u,i}$ and $||\cdot||_F^2$ denotes Frobenius norm of feature matrices.

Derived from the WNMF [10] which was utilized in the field of recommend system, the RSNMF can also be easily understood literally. It is a simple version of WNMF and is single-element-based like the MF. We denote the scoring matrix as T and its low-rank estimate matrix as WH. T is a sparse matrix and the default value is -1, and other non-negative elements' entries can be contained in a set T_k. We can get the regularized square error on the set T_k refers to [11], which is denoted as follow:

$$\begin{aligned} \mathrm{Error} &= ||T - WH||^2 + \lambda_{\mathrm{w}}||W||_F^2 + \lambda_{\mathrm{H}}||H||_F^2 \\ &\approx \sum_{u,i \in T_k} \Big((T_{u,i} - W_{u,.} * H_{.,i})^2 \\ &\quad + \lambda_{\mathrm{w}}||W_u||_F^2 + \lambda_{\mathrm{H}}||H_i||_F^2\Big) \\ &= \sum_{u,i \in T_k} \left(\left(T_{u,i} - \sum_{k=1}^{f} W_{u,k} * H_{k,i}\right)^2 + \lambda_{\mathrm{w}} \sum_{k=1}^{f} w_{u,k}^2 \right. \\ &\quad \left. + \lambda_{\mathrm{H}} \sum_{k=1}^{f} h_{u,k}^2 \right) \end{aligned} \tag{4}$$

In the Eq. (4), the parameter f denotes the number of latent factors.

It is similar to the WNMF algorithm. We utilize the Lagrange multiplier on this objective function and utilize the non-negativity part of the Kuhn-Tucker condition [10]. Finally, we can generate the update rules as the following:

$$w_{u,k} = w_{u,k} + \frac{\sum_{i \in I_u} h_{k,i} T_{u,i}}{\lambda_w |I_u| w_{u,k} + \sum_{i \in I_u} h_{k,i} T_{u,i}^t} \tag{5}$$

$$h_{k,i} = h_{k,i} + \frac{\sum_{u \in U_i} w_{u,k} T_{u,i}}{\lambda_w |U_i| h_{k,i} + \sum_{u \in U_i} w_{u,k} T_{u,i}^t} \tag{6}$$

In the Eqs. (5) and (6), I_u and U_i denotes the item set rated (scored) by user (student) and the user (student) set rated (scored) item (“problem-view-step” group).

3.3 Performance Factor Analysis (PFA)

Recently one of the most popular approaches adapted for student knowledge modeling is the PFA proposed in [9], which we use in this paper as a comparison with RS methods and Hints-Model. It is a simple regression model which explicitly takes into account the effect of successes and failures, and it can obtain more understandable than learning factor analysis (LFA) [17]. The PFA model used in the paper is as the following, which referred to [9]:

$$Lm(u, i \in KCs, s, f) = \sum_{i \in KCs} (\beta_i + \gamma_u * s_{u,i} + \rho_i * f_{u,i}) \tag{7}$$

In the Eq. (7), Lm is a logarithmic value representing the accumulated learning skills for a student u using one or more KCs (knowledge components) i. The easiness of these KCs is indicated by the β parameters for each KC, and $s_{u,i}$ denotes the successes student u has on KC_i, while $f_{u,i}$ denotes the failures student u has on KC_i, and γ and ρ indicates the effect of these observation counts [9]. We fit those parameters (β, γ and ρ) for the PFA to maximize log likelihood of the proposed model.

3.4 User-Item Biased

According to the investigation of those related works [2], [23], user-item biased can be a good choice for PSP, and it often performs better than MF. Here we compare it with other factorization techniques. The user-item biased model can be easily denoted as a linear model [23]:

$$\mathrm{T}_{u,i} = \mu + b_u + b_i \tag{8}$$

Where μ is the global average of student scores, b_u, and b_i is student and item biases respectively [1], [2], [23]. We can simply use stochastic gradient descent to fit the model.

4. Datasets and the Experiment

Our datasets are collected from the Pittsburgh science of learning center datashop service. There are two datasets that are generated by the assistance of intelligent tutoring system. One records 912 students' 580785 transactions with the system, which contains 524 problems and 1745 unique solving steps. The other records 3136 students' 685615 transactions, which contains 846 problems and 2514 steps. We denote the smaller dataset as Dataset1 and the larger as Dataset2.

At the initial stage of the experiment, we should firstly preprocess the dataset and convert it to a simple matrix that contains students' performance (correct or incorrect) for the specific "problem-view-step" groups. After converting, the Dataset1 has been converted to a matrix with 912 rows and 9609 columns and the Dataset2 has been converted to a matrix with 3136 rows and 8613 columns. Our target is to predict whether a student is correct at first attempt (CFA) at the specific step. There are binary records, for example, 1 denotes the success (correct) and 0 denotes failure (incorrect or hint). Testing records are determined by the algorithm which randomly selects one problem for each student within a unit, and places all student-step rows for that student and problem in the test set. The quantity of the training set and the testing set accounts 80% and 20% respectively of the whole dataset. We use 5-fold cross-validation to avoid overfitting. The measure criterion is RMSE and the accuracy of the model. The model obtains smaller RMSE and larger accuracy will be a good method.

We made a comparison among these models and analyze the outcomes with some useful statistics about the data. In the meanwhile, we made a data-driven experiment and obtain some illustrations which supported our previous results.

4.1 Hints-Model

4.1.1 Data-Driven Method

In order to utilize Hints-Model, we firstly analyze the dataset and acquire some illustrations to support our idea. We scan the whole dataset and make divisions to obtain the average hints for each student asked for on a single step and calculate each student's correct rate on one step. Then we permute the average hints from small to large and draw a line chart for average hints and the corresponding student's correct rate. This is a data-driven method based on the whole dataset.

In the four graphs above-mentioned, the *hint* (*total*) denotes that we consider all the hints on one step and the *hint* (*FA*) means that we only consider the hint at the student's first attempt on one step.

Although it exists some fluctuations in the graph, those figures (Figs. 1, 2, 3, and 4) show a trend that with fewer hints students are more likely to perform better than those with more hints. For the noise of the data shown in the graph is obvious, we want to build a model that can efficiently reduce the noise.

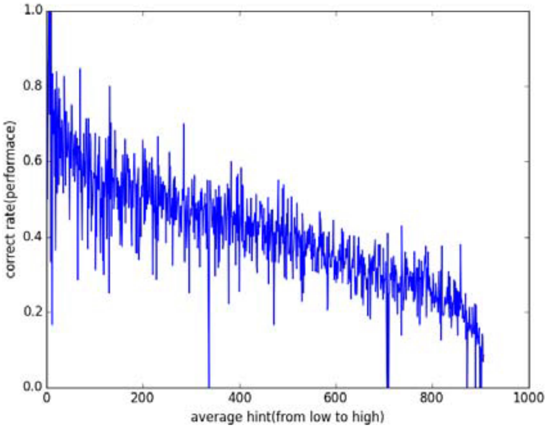

Fig. 1 Average hints (total) and correct rate on Dataset1.

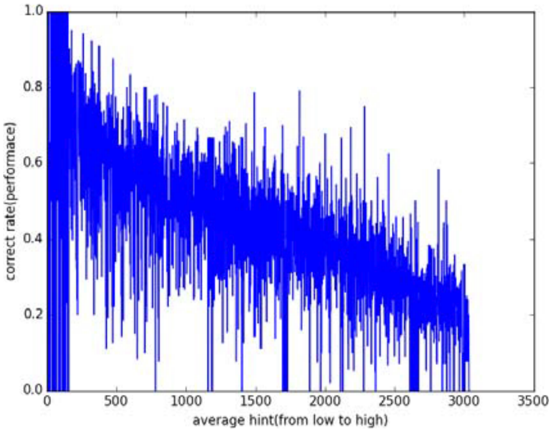

Fig. 2 Average hints (total) and correct rate on Dataset2.

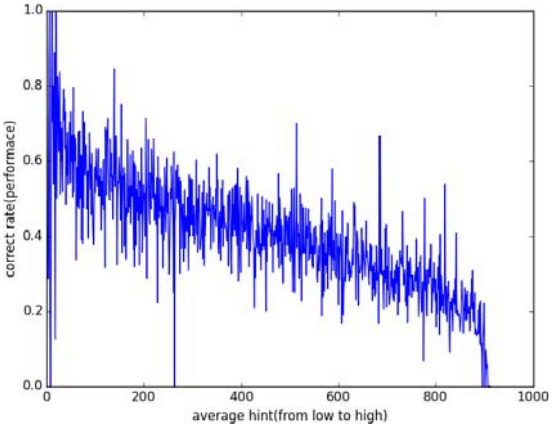

Fig. 3 Average hints (FA) and correct rate on Dataset1.

Table 1 Hints-Model's results on Dataset1.

	RMSE	Accuracy	TN	FN	TP	FP
Hint_Model_FA	0.455333	0.672978	28561.6	13700.2	9681.8	4884.6
Hint_Model_Sum	0.464696	0.652371	28337.6	14646.8	8735.2	5108.6

Table 2 Hints-Model's results on Dataset2.

	RMSE	Accuracy	TN	FN	TP	FP
Hint_Model_FA	0.46609	0.65583	33009.8	19044.6	11484.8	4310
Hint_Model_Sum	0.46848	0.647252	32306.8	18924	11605.4	5013

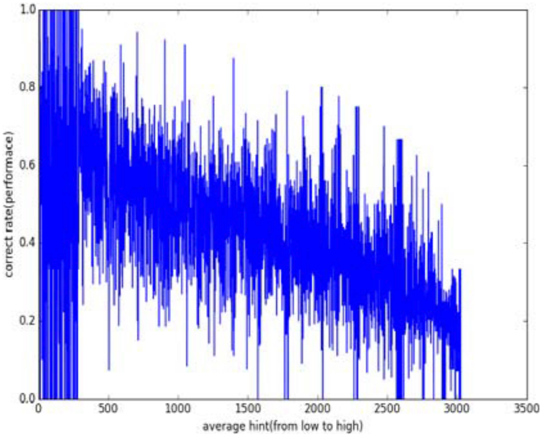

Fig. 4 Average hints (FA) and correct rate on Dataset2.

4.1.2 Choose a Proper Hints-Model

First of all, we can see that there are two choices that we process the hint matrix. One is that if a student obtains a "HINT" at his or her first attempt on the item, we change the element to be 1. To the contrary, we change it to be 0. Another is that if a student obtains a "HINT" on the item, we change the element to plus 1. The element indicates the number of hints required by the student on the specified item. The two approaches to deal with data generated two different outcomes. We denote the former choice as Hint_Model_FA and the latter as Hint_Model_Sum. Both the models are introduced with 5-fold cross-validation, and the RMSE and accuracy result is as the following.

Those prediction performances of the models are shown in Tables 1 and 2. From the result we can easily see that Hint_Model_FA is better than Hint_Model_Sum, so we choose Hint_Model_FA as the model that will be compared with other models.

Here we come to our assumption: students with more hints per question tend to have poorer performance on the questions, in the meanwhile, and problems with more hints per student tend to be more difficult than problems which have fewer hints per student. We want to find some evidence that supports our assumption.

There are some statistics that support our assumption, for example, the coefficients of Hints-Model. In the process of our experiment, we divided the coefficients *sa_coef* and *ia_coef* into groups corresponding to the average hints *sa* and *ia*. For example, we set the *large* average hint's lower bound of *sa* and *ia* as 0.5, and the *middle* average hint's lower bound of *sa* and *ia* as 0.2, average hints under 0.2 are *small* average hints. The following tables show how many coefficients in the corresponding group are positive and the number of total coefficients in the group.

Table 3 The statistics of the coefficients of sa and ia on Dataset1.

	Positive sa_coef	Total sa_coef	Positive ia_coef	Total ia_coef
Large	0.2	64.6	0.8	1346.4
Middle	37	376	14.2	1285.2
Small	258.8	471.4	405.6	6977.4

Table 4 The statistics of the coefficients of sa and ia on Dataset2.

	Positive sa_coef	Total sa_coef	Positive ia_coef	Total ia_coef
Large	0	86.4	0	509.2
Middle	8.6	457.8	2.4	743.2
Small	420	2591.8	537.2	7360.6

Table 5 Accuracy results of PFA and Hints-Model.

	Accuracy 1	Accuracy 2
Hint_Model_FA	0.672978	0.655830
PFA	0.646972	0.628668

Tables 3 and 4 shows that if the average hint gets larger, the corresponding coefficient is more likely to be negative. It means that smaller average hint is more likely to have positive effect on student performance and indicates that the problem is easier for student to answer correctly. This is a good evidence for our assumption.

4.1.3 Comparison of PFA and Hints-Model

The PFA is a popular student knowledge modeling method. We use scikit-learn's logistic regression model to fit the method. After using 5-fold validation with those models, we compare the average accuracies of the Hints-Model and PFA.

As shown in Table 5, Accuracy1 and Accuracy2 denotes the accuracy of Dataset1 and Dataset2 respectively. We can see that the Hint_Model_FA outperforms the PFA in accuracy.

4.2 Comparison of Factorization Techniques

We can simply utilize stochastic gradient descent to fit the model based on the objective function in Eq. (2) by the criterions including root mean squared error (RMSE) and prediction accuracy in testing dataset. The traditional way of performing hyperparameter optimization has been grid search,

Table 6 Comparison of factorization techniques on Dataset1.

	RMSE	Accuracy	TN	FN	TP	FP
MF	0.44120048	0.6999	29661.6	13258.4	10123.6	3784.6
SRNMF	0.44038808	0.710281	29205.8	12224.2	11157.8	4240.4
User-item Biased	0.43581993	0.716013	27136.8	9829	13553	6309.4

Table 7 Comparison of factorization techniques on Dataset2.

	RMSE	Accuracy	TN	FN	TP	FP
MF	0.44277546	0.697768	32018.4	15204.2	15325.2	5301.4
SRNMF	0.440656445	0.710245	30977.6	13318.8	17210.6	6342.2
User-item Biased	0.436767095	0.715511	28526.6	10512.8	20016.6	8793.2

Table 8 Results of ensemble methods on Dataset1.

	RMSE	Accuracy	TN	FN	TP	FP
Hint_SRNMF	0.416600852	0.74155	28385.4	9626.4	13755.6	5060.8
Hint_MF	0.427976407	0.723148	27533.2	9820.8	13561.2	5913
Hint_biased	0.420521118	0.733233	26669	8381.8	15000.2	6777.2

Table 9 Results of ensemble methods on Dataset2.

	RMSE	Accuracy	TN	FN	TP	FP
Hint_SRNMF	0.415812515	0.743867	29729.8	9785.6	20743.8	7590
Hint_MF	0.426914942	0.724769	29333.2	10687.2	19842.2	7986.6
Hint_biased	0.420728157	0.734476	30184	10881.4	19648	7135.8

or a parameter sweep, which is simply an exhaustive searching through a manually specified subset of the hyperparameter space of a learning algorithm. According to the grid search result of hyperparameters, we set the learning rate of stochastic gradient descent as 0.05, and the coefficient of regularization term as 0.07 in Eq. (3). In our experiment, we set the parameter $f = 5$ in Eq. (4) for both the MF and RSNMF methods and set the coefficient of RSNMF's regularization term as 0.05.

We compare MF to user-item biased with RSNMF, and try to find out which factorization method is better on two datasets. The two models are similar, but the significant difference is that RSNMF ensures the non-negative property of matrices decomposition of the original matrix. So we assume that the result will be better with more limits.

We can see from Tables 6 and 7 that the best technique is user-item biased, while the SRNMF performs a little better than MF, that is, our assumption is proper, and SRNMF is more likely to predict accurate positive instances compared with MF. User-item biased method is more likely to predict positive values.

4.3 Combination of Different Methods

It has been proved that the ensemble methods will improve the performance of student knowledge modeling [12]. Inspired by this idea, we utilize a logistic regression model to combine the result of two different models and analyze these results to come up with some useful proposals. Since the Hints-Model has taken into account the hints' influence while the other models have not considered, we simply use logistic regression to combine the results of Hints-Model and other factorization models.

In Tables 8 and 9, the Hint_SRNMF denotes the combination of Hints-Model and the SRNMF model, and the other two names can be understood like Hint_SRNMF.

Table 10 Correlation coefficients of original models' results.

	Corrcoef1	Corrcoef2
Hint_SRNMF	0.756329236	0.762789
Hint_MF	0.770724909	0.765048
Hint_biased	0.765869807	0.76363

From the results shown in Tables 8 and 9, we can easily find that the Hint_SRNMF is better than the other two methods, and the combination of Hints-Model with other factorization models performs better than any one of the original model, which may be a good direction for our further researches. There are some reasons that may contribute to Hint_SRNMF's better performance. We assume that they are more likely to be complementary when the results of two models have weaker correlationship. So we compare the correlation coefficients of Hints-Model's result and other three models' results.

As shown in Table 10, the corrcoef1 denotes the correlation coefficients on Dataset1, and corrcoef2 denotes the correlation coefficients on Dataset2. We can see that large correlation coefficient corresponds with small RMSE and large accuracy, which is better in prediction.

5. Conclusions and Future Work

Hint is very important assistance in intelligent tutoring systems. However, hint is not fully used for student performance prediction. In order to utilize hints, we proposed a Hints-Model which is a linear model and can generate an understandable outcome. It is fast and outperforms the traditional PFA according to our experiment. An efficient non-negative matrix factorization method named RSNMF has been used in our experiment which is proposed in the field of recommend system. In addition, we combine Hints-Model with other factorization models to generate a more accurate prediction. We find that the ensemble method exceeds either of those original models obviously. However, better performance of those original models does not necessarily contribute to the better performance of the ensemble method. At last, we come up with the assumption that the correlation coefficient can be a good represent of the performance of ensemble methods.

There are also some problems that can be further explored. First of all, for the datasets are large, we have not preprocessed the datasets very well. Secondly, the way we utilize hints is simple, so we want to find some more efficient ways to use hints entirely. What's more, other more complicated ensemble methods may outperform our logistic regression model, which is worth paying attention.

Acknowledgments

This research is supported by National Science and Technology Project of the Ministry of Education of China (MCM20121061, MCM20121041). We used the datasets of the assistments of Math 2004–2005 (912 Students) and Math 2005–2006 (3136 Students) and accessed them via datashop (pslcdatashop.org).

References

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

[2] N. Thai-Nghe, L. Drumond, T. Horváth, A. Nanopoulos, and L. Schmidt-Thieme, "Matrix and tensor factorization for predicting student performance," Proc. 3rd International Conference on Computer Supported Education, pp.69–78, Noordwijkerhout, Netherlands, 2011.

[3] A.T. Corbett and J.R. Anderson, "Knowledge tracing: Modeling the acquisition of procedural knowledge," User Modeling and User-Adapted Interaction, vol.4, no.4, pp.253–278, 1995.

[4] D.D. Lee and S.H. Seung, "Learning the parts of objects by non-negative matrix-factorization," Nature, vol.401, pp.788–791, 1999.

[5] M.C. Desmarais, "Mapping question items to skills with non-negative matrix factorization," SIGKDD Explorations, vol.13, no.2, pp.30–36, 2011.

[6] L. Xiong, X. Chen, T.-K. Huang, J. Schneide, and J.G. Carbonell, "Temporal collaborative filtering with Bayesian probabilistic tensor factorization," Proc. SIAM International Conference on Data Mining, pp.211–222, Columbus, Ohio, United States, 2010.

[7] S. Sahebi, Y. Huang, and P. Brusilovsky, "Predicting student performance in solving parameterized exercises," Proc. 12th International Conference on Intelligent Tutoring Systems, pp.496–503, Honolulu, HI, United States, 2014.

[8] Z.A. Pardos, M. Feng, N.T. Heffernan, and C. Linquist-Heffernan, "Analyzing fine-grained skill models using Bayesian and mixed effects methods," Proc. 13th International Conference on Artificial Intelligence in Education, pp.626–628, Los Angeles, California, United States, 2007.

[9] P.I. Pavlik, H. Cen, and K.R. Koedinge, "Performance factors analysis — A new alternative to knowledge tracing," Proc. 14th International Conference on Artificial Intelligence in Education, pp.531–538, Brighton, United Kingdom, 2009.

[10] S. Zhang, W. Wang, J. Ford, and F. Makedon, "Learning from incomplete ratings using non-negative matrix factorization," Proc. Sixth SIAM International Conference on Data Mining, pp.549–553, Bethesda, MD, United States, 2006.

[11] X. Luo, M. Zhou, Y. Xia, and Q. Zhu, "An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems," IEEE Trans. Ind. Informat., vol.10, no.2, pp.1273–1284, 2006.

[12] Z.A. Pardos, S.M. Gowda, R.S.J.D. Baker, and N.T. Heffernan, "The sum is greater than the parts: Ensemble models of student knowledge in educational software," ACM SIGKDD Explorations Newsletter, vol.13, no.2, pp.37–44, 2011.

[13] Y. Wang, and N.T. Heffernan, "Leveraging first response time into the knowledge tracing model," Proc. 5th International Conference on Educational Data Mining, pp.176–179, Chania, Greece, 2012.

[14] X. Xiong, Z.A. Pardos, and N.T. Heffernan, "An analysis of response time data for improving student performance prediction," 2011. [Online]. Available: https://pslcdatashop.web.cmu.edu/KDD2011/papers/M-kddined2011.pdf. [Accessed: 10-Dec-2015].

[15] Y. Wang and N.T. Heffernan, "The 'Assistance' model: Leveraging how many hints and attempts a student needs," Proc. Twenty-Fourth International Florida Artificial Intelligence Research Society Conference, pp.549–554, Palm Beach, Florida, United States, 2011.

[16] T. Winters, C. Shelton, T. Payne, and G. Mei, "Topic extraction from item level grades," AAAI-05 Workshop on Educational Data Mining, 2005.

[17] H. Cen, K. Koedinger, and B. Junker, "Learning factors analysis — A general method for cognitive model evaluation and improvement," Proc. 8th International Conference on Intelligent Tutoring Systems, pp.164–175, Jhongli, Taiwan, 2006.

[18] D.D. Lee and S.H. Seung, "Algorithms for non-negative matrix factorization," Proc. 13th International Conference on Advances in Neural Information Processing Systems, pp.556–562, Denver, CO, United States, 2000.

[19] K.-M. Chang, J. Beck, J. Mostow, and A. Corbett, "A Bayes net toolkit for student modeling in intelligent tutoring systems," Proc. 8th International Conference on Intelligent Tutoring Systems, pp.104–113, Jhongli, Taiwan, 2006.

[20] M.C. Desmarais and R. Naceur, "A matrix factorization method for mapping items to skills and for enhancing expert-based q-matrices," Proc. 16th International Conference on Artificial Intelligence in Education, pp.441–450, Memphis, TN, United States, 2013.

[21] A. Corbett, L. Kauffman, B. Maclaren, A. Wagner, and E. Jones, "A cognitive tutor for genetics problem solving: Learning gains and student modeling," Journal of Educational Computing Research, vol.42, no.2, pp.219–239, 2010.

[22] B.V.D. Sande, "Properties of the Bayesian knowledge tracing model," J. Educational Data Mining, vol.5, no.2, pp.1–10, 2013.

[23] Š. Pero and T. Horváth, "Comparison of collaborative-filtering techniques for small-scale student performance prediction task," Innovations and Advances in Computing, Informatics, Systems Sciences, Networking and Engineering, Lecture Notes in Electrical Engineering, vol.313, pp.111–116, 2015.

[24] N. Thai-Nghe, L. Drumond, T. Horvath, A. Nanopoulos, and L. Schmidt-Thieme, "Factorization techniques for predicting student performance," ed. O.C. Santos and J.G. Boticario, Educational Recommender Systems and Technologies: Practices and Challenges, pp.129–153, IGI Global, 2011.

[25] K. Ostrow, C. Donnelly, and N. Heffernan, "Optimizing partial credit algorithms to predict student performance," Proc. 8th International Conference on Educational Data Mining, pp.404–407, Madrid, Spain, 2015.

[26] J. Stamper, A. Niculescu-Mizil, S. Ritter, G.J. Gordon, and K.R. Koedinger, "Challenge data set from KDD Cup 2010 Educational Data Mining Challenge," 2010. [Online]. Available: http://pslcdatashop.web.cmu.edu/KDDCup/downloads.jsp [Accessed: 22-Oct-2015].

Ke Xu received the Ph.D. degrees from Huazhong University of Science and Technology in 2015. During 2007–2010, he stayed in South-Central University for Nationalities. He is now postdoctoral Candidate in the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China. His research interests include education data mining and social network analyze.

Rujun Liu received his Master degree from The University of Edinburgh for Artificial Intelligence in 2008. He was admitted to Huazhong University of Science and Technology in 2011 and is currently a Ph.D. Candidate. His research interests are distributed computing and storage system.

Yuan Sun received the M.S. degree from South-Central University for Nationalities in 2012. She is now a research assistant in Science and Technology Department, Hubei University of Traditional Chinese Medicine, Wuhan, China. Her research interests include the data mining and analyze of traditional Chinese medicine.

Keju Zou received his B.S. degree from Sichuan University in 2008. He is currently a Ph.D. candidate at School of Engineering of Sun Yat-sen University, Guangzhou, China. His research interests include sleep medicine and biomedical signal processing.

Yan Huang received his B.S. degree from South-Central Univercity for Nationalities in 2012. He is currently a Ph.D. Candidate in School of Computer Science and Technology, Huazhong University of Science and Technology. His research interests are machine learning, pattern recognition and knowledge base.

Xinfang Zhang received the Ph.D. degree from Huazhong University of Science and Technology in 1993. Now he is a Professor in the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China. His research interests include education data mining and cloud computing.