

A cognitive diagnostic framework for computer science education based on probability graph model

HU Xinying, HE Yu, SUN Guangzhong*

School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China

* Corresponding author. E-mail: gzsun@ustc.edu.cn

Abstract: A new cognitive diagnostic framework was proposed to evaluate students' theoretical and practical abilities in computer science education. Based on the probability graph model, students' coding ability was introduced, then the students' theoretical and practical abilities was modeled. And a parallel optimization algorithm was proposed to train the model efficiently. Experimental results on multiple data sets show that the proposed model has a significant improvement in MAE and RMSE compared with the competing methods. The proposed model provides more accurate and comprehensive analysis results for computer science education.

Keywords: cognitive diagnosis; probability graphic model; educational data mining

CLC number: TU459 **Document code:** A

1 Introduction

Students acquire knowledge in schools from diverse courses, and teachers give students assignments or tests to practice the skills taught in courses. Giving accurate and rapid feedback to students during their daily practice plays an important role in teaching process. It has been proved that rapid feedbacks can improve students' performance; in a controlled experiment, students' final grades had been improved when feedback was delivered quickly, but not if delayed by 24 hours^[1]. In the traditional teaching process, scores or grades are provided as feedback for students. However, students with the same score may have different cognitive processes. A single score can not distinguish cognitive differences between students. With the rapid development of information technology in education, we hope to analyze students' various abilities in courses and learning characteristics of students.

Recent decades have witnessed the development of educational data mining (EDM), which refers to the mining of valuable information from the data collected during the education process. Cognitive diagnosis is one of the key applications of EDM. It refers to analyzing students' answers on a set of questions to infer students' mastery of knowledge concepts. Nowadays, people are dissatisfied with giving each student a simple test score or a grade to indicate their ability. They prefer the ways that can provide diagnostic information and

report the cognitive structure of students. Cognitive diagnostic models (CDMs) can be used to model students, estimate their abilities and predict their performances on each question. The existence of CDMs allows us to know the cognitive structure of students precisely, as well as provides a basis for teachers' personalized guidance. At present, cognitive diagnosis has achieved good performance in the evaluation of students in traditional subjects such as mathematics and English^[2,3].

Although cognitive diagnosis has performed well in students' evaluations of traditional subjects, it still has some shortcomings in the field of the computer science education. The reason is that computer science is different from traditional subjects. In addition to theoretical knowledge concepts, the training of programming is also essential, which means cultivating students' ability to turn theoretical knowledge concepts into codes. In the field of the computer science education, the ability to write codes is the bridge that applies knowledge concepts to real life and solves practical problems with coding. Therefore, it is indispensable in cognitive diagnosis of computer science education to master the students' practiceabilities and help students improve their coding skills. However, existing cognitive diagnosis methods only consider the way to model students' theoretical knowledge concepts with ignoring the ability to use knowledge concepts in practice. In order to solve this problem, we propose a

new cognitive diagnostic framework for the computer science education (CDF-CSE), which can model students' theoretical knowledge concepts and programming ability at the same time. We can evaluate students' programming ability according to the practices of students by using our framework, thereby assisting students to learn and improve in coding. The proposed method can be applied in the computer science education, in which we can diagnose students comprehensively and explore the potential factors and characteristics of students in various aspects.

To the best of our knowledge, this is the first attempt to combine theoretical learning with their practical abilities. The proposed method models students' programming abilities to bring cognitive diagnostics to the field of the computer science education. And it models theoretical and practical abilities at the same time to predict students' performance and analyze students comprehensively. We design an effective algorithm for parameter estimation and conduct extensive experiments on multiple datasets (including two data sets collected from computer science courses in the University of Science and Technology of China) to demonstrate the effectiveness of our framework.

2 Related work

2.1 Cognitive diagnosis

In educational psychology, many cognitive diagnostic models^[3] have been developed to mine students' skill proficiency (mainly related to the mastery of theoretical knowledge concepts). Study of CDMs includes two aspects, discrete and continuous. The fundamental discrete CDMs is deterministic inputs, noisy "and" gate model (DINA)^[4-6]. DINA describes a student by a binary vector, each value of which indicates whether the student has mastered a certain skill. In addition, DINA also introduced a skill matrix Q to represent skills required for each problem. The Q -matrix can guarantee the interpretation of diagnosis results. Based on DINA, higher-order DINA (HO-DINA) that contains a higher-order cognitive parameter to represent overall abilities of students was proposed^[7]. Besides, in order to meet the needs of processing large-scale data, a generalization of DINA, also called G-DINA, has appeared^[6]. Though discrete CDMs are interpretable, their diagnosis results are usually not very accurate.

For continuous CDMs, the basic method is item response theory (IRT)^[8], which characterizes a student by a continuous variable that corresponds to the latent trait of student, and use a logistic function to model the probability that a student correctly solves a problem. A single latent trait only shows the general cognitive status of students. Therefore, multidimensional IRT has been

proposed to describe students' skill proficiency comprehensively. Multidimensional IRT are divided into compensatory (MIRT-C) and non-compensatory (MIRTNC)^[3]. It is supposed that skills that students do not know can be made up by other related skills in MIRT-C, while opposition in MIRT-NC. Continuous models describe students more accurately than discrete models, but its assumptions may not be suitable for the computer science education. Furthermore, neither model is suitable for those subjective questions.

Based on cognitive diagnosis results, it emerges predictions of the students' performance on questions that need specific skills^[9]. Besides, some researchers analyzed the impact of objective and subjective factors on students' question answering process^[10]. And some efforts were tried to visualize the results of cognitive diagnosis for further analyses in a more convenient way^[11].

2.2 Data mining

Some studies attempted to use the matrix factorization (MF) in recommendation systems for cognitive diagnosis. The basic idea is to treat students as users, questions as items, and test scores as user's scores on items. In this case, we can factorize the score matrix to get the student vector and question vector, and predict the student's score on new questions. Related work includes using the singular value decomposition (SVD) and other factor models to model students^[12]. Some researchers compared MF techniques with regression methods to predict students' performance^[13]. In MOOC, a MF-based approach was proposed to model learning preferences^[14]. In addition, there are some work applied non-negative matrix factorization to infer the Q matrix^[15, 16]. And some scholars used relational MF to model students in an intelligent tutoring system^[17]. Even if there are many attempts on MF, the parameters obtained by MF are unexplained compared with the serious diagnosis. We don't know what kind of user information the user vector represents, nor do we know what characteristics the problem vector corresponds to. Although the matrix factorization method can achieve good performance in predicting students' scores, it still can not give us sufficient information.

3 Cognitive diagnostic modeling

In this section, we will introduce our cognitive diagnostic framework for computer science education (CDF-CSE). In existing cognitive diagnostic models, students' proficiency in skills, or knowledge concepts, refers to their ability to use these skills to solve theoretical problems. In this paper, we call it theoretical abilities. In the computer science education, we also need to consider the ability of students to turn

knowledge concepts into codes. Based on what was mentioned above, we will model students' theoretical as well as their experimental abilities (abilities to write codes) in our model. In addition, we have added a parameter that indicates the student's overall programming abilities rather than a specific knowledge concept. We will introduce our model in the details in the following subsections.

3.1 Problem definition

It is necessary to formalize our problem first. We assume that we have M students in a course, then, the teacher teaches K skills and assigns N_i theoretical questions, N_e experiments. For homework or exam questions in the course, the score matrix R is a matrix of M rows and N_i columns. R_{ji} represents score of student j on question i , where $j=1,2,\dots,M$, $i=1,2,\dots,N_i$, and $R_{ji} \in [0,1]$. For the programming experiments in the course, let R'_{je} indicates score of student j on experiment e , where $e=1,2,\dots,N_e$, and $R'_{je} \in [0,1]$. The higher values of R_{ji} and R'_{je} , the better the student's performance. Let matrix Q be the indicator matrix that indicates knowledge concepts investigated in each theoretical question. Q includes N_i rows and K columns. And q_{ik} indicates whether question i investigate knowledge concept k , where $k=1,2,\dots,K$. Let Q' be the matrix that indicates skills investigated in each experiment. There are N_e rows and k columns in Q' , q'_{ek} indicates whether experiment e investigate knowledge concept k . In general, $q_{ik}=1$ when question i requires knowledge concept k , and $q_{ik}=0$ is the opposite. Similarly, $q'_{ek}=1$ means that knowledge concept k is needed for solving the experiment e , and $q'_{ek}=0$ when it is not needed. Then we normalize the matrix Q and Q' by making

$$q_{ik} = \frac{q_{ik}}{\sum_{l=1}^K q_{il}},$$

$$q'_{ek} = \frac{q'_{ek}}{\sum_{l=1}^K q'_{el}}.$$

Given R, R', Q, Q' , our goal is to make cognitive diagnosis for students in the computer science education, which is divided into three parts:

(I) Diagnose the student's overall programming ability c_j in the proposed model.

(II) Find out the theoretical mastery α_{jk} and experimental mastery β_{jk} of student j for a certain skill k .

(III) Predict the performance of students for a new theoretical problem i or experimental problem e that requires some skills to solve. The predicted performance of student j on the theoretical problem i is recorded as

η_{ji} . Similarly, the predicted performance of student j on the experiment e is recorded as η'_{je} .

The above diagnostic targets are all valued in $[0,1]$, where 1 means that the student has completely mastered the skill or question, 0 means the opposite.

3.2 Model description

Programming Ability: In order to evaluate students' coding abilities, we refer to the research results of the educational psychology: each person has a high-order latent trait, which represents the person's general ability to learn something^[7]. We model this high-order latent trait as a parameter c_j that can describe the programming ability of student j . That is, c_j does not involve any skill, it involves the ability of programming itself. Generally speaking, each student j has an independent parameter c_j to indicate the student's ability to write programs.

Mastery and application of skills According to the problem definition, α_{jk} is ability of student j using skill k to do theoretical problems, and β_{jk} is the ability of student j using skill k to do experiments (e.g. writing code). In the proposed model, we assume that there is no direct correlation between different abilities. Therefore, we assume that abilities of a student in different skills are independent of each other, and abilities of different students are also independent of each other. With common sense, two abilities α_{jk} and β_{jk} of the same skill should be related. According to the experience gained during the education process, we believe that they can apply skills in experiments after they have mastered them in theory. In summary, we propose an assumption:

Assumption 3.1 The programming ability β_{jk} in skill k of student j is directly proportional to student's theoretical knowledge concept α_{jk} of the skill and basic coding ability c_j .

In other words, we believe that a person's experimental ability depends on his/her theoretical mastery on the corresponding knowledge concept, and is limited by his/her basic programming ability. We write this hypothesis as:

$$\beta_{jk} = c_j \alpha_{jk}$$

Problem mastery As mentioned above, η_{ji} is student's mastery of a theoretical question and η'_{je} is student's mastery of an experiment. The traditional cognitive diagnostic models believe that the questions are independent of each other. Even if the same knowledge concept may be examined between them, we do not think that there is a direct relationship between questions (the relationship of knowledge concepts between questions is provided by Q matrix). Traditional CDMs assumes that a student's mastery of a problem is related to the knowledge concepts the student has

learned and the knowledge concepts required to answer the question^[18]. In actual courses, each question needs one or more knowledge concepts. In general, a student solves a question completely or solves a part of it, which indicates that the student has successfully used specific knowledge concepts required by the question. In this case, we think that the student has mastered or partially mastered the corresponding knowledge concepts. Based on the above analysis, we define students' mastery of problems (and experiments mastery) as the following assumptions:

Assumption 3.2 Mastery of student j of theoretical problem i is related to the mastery of knowledge concepts examined in the problem, and mastery of the experimental problem is calculated from the theoretical ability of knowledge concepts needed for the problem.

That is, we thought that the performance of a student on a problem is directly proportional to the mastery of the corresponding knowledge concepts in the problem. Student can perform well when he/she is proficient in knowledge concepts. Mathematically, student's mastery of the theoretical problem is:

$$\eta_{ji} = \sum_{k=1}^K \alpha_{jk} q_{ki}$$

and the mastery of student on the experimental problem is written as:

$$\eta'_{je} = \sum_{k=1}^K \beta_{jk} q'_{ke}$$

Actual score In actual situations, students may give correct answers without mastering knowledge concepts because they guess out or write the wrong answer due to carelessness and other reasons. This results in the actual score often deviating from real mastery^[19]. According to the practice of probability matrix decomposition in the recommendation system^[20], we use the Gaussian distribution to simulate the actual situation from students' mastery of the problem score.

$$\begin{aligned} R_{ji} &\sim N(\eta_{ji}, \sigma_R^{-1}I), \\ R'_{je} &\sim N(\eta'_{je}, \sigma_{R'}^{-1}I); \end{aligned}$$

where σ_R and $\sigma_{R'}$ are hyper-parameters and I is identity matrix.

In our model, actual score obeys a Gaussian distribution with the mastery as the mean. It is reasonable that the actual score is related to the student's mastery of the problem with a little bias due to some uncontrollable factors.

Summary We summarize the proposed model into the probability graph shown in Figure 1, where the gray circles in the graph are known quantities and the white are unknown quantities. It could be seen from the probability graph that there are four observable values, the scores R of M students on the N_i theoretical problems, the performance R' of M students on N_e

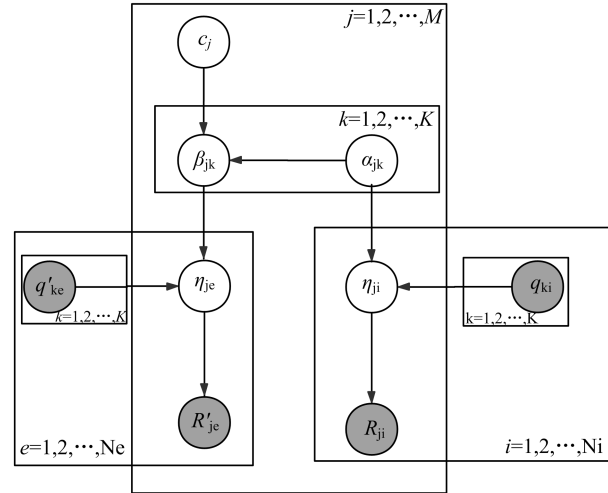


Figure 1. The probability graph model of CDF-CSE

experimental questions, the matrix Q of the correspondence between each theoretical question and each knowledge concept, and matrix Q' indicates the knowledge concepts required for each experimental problem. In order to meet the computer science education situation, we introduce the parameter c_j to model the student j 's basic programming abilities. Besides, each student has a set of theoretical mastery of knowledge concepts $\{\alpha_{jk}\}_{k=1}^K$ and a set of experimental mastery of knowledge concepts $\{\beta_{jk}\}_{k=1}^K$, where β_{jk} is calculated by c_j and α_{jk} . Student j 's theoretical problem mastery η_{ji} is determined by the student's theoretical mastery $\{\alpha_{jk}\}_{k=1}^K$ and the knowledge concepts of the problem $\{q_{ki}\}_{k=1}^K$. Student j 's experimental problem mastery η'_{je} is determined by the student's experimental mastery $\{\beta_{jk}\}_{k=1}^K$ and the knowledge concepts of the problem $\{q'_{ke}\}_{k=1}^K$. Finally, student j 's actual scores R_{ji} and R'_{je} on theoretical problem i and experimental problem e are affected by mastery η_{ji} and η'_{je} respectively. Therefore, it can be seen that after solving the parameters c_j and α_{jk} , we could find the remaining unknown parameters. Following the settings of HO-DINA model^[7], our parameters obey following prior distribution:

$$\begin{aligned} c_j &\sim N(\mu_c, \sigma_c^{-1}I), \\ \alpha_{jk} &\sim N(\mu_\alpha, \sigma_\alpha^{-1}I); \end{aligned}$$

where σ_c and σ_α are hyper-parameters.

3.3 Parameter optimization

According to the above probability graph model and assumptions of parameters, given the observable data, the posterior distribution of c and α can be written as:

$$P(c, \alpha | R, R') \propto P(R | \alpha) P(R' | c, \alpha) P(c) P(\alpha).$$

The probability distribution of these parameters are:

$$P(R_{ji} | \alpha_j) = N\left(\sum_{k=1}^K \alpha_{jk} q_{ki}, \sigma_R^{-1}I\right),$$

$$P(R'_{je} | c_j, \alpha_j) = N\left(\sum_{k=1}^K c_j \alpha_{jk} q'_{ke}, \sigma_R^{-1} I\right),$$

$$P(c_j) = N(\mu_c, \sigma_c^{-1} I),$$

$$P(\alpha_{jk}) = N(\mu_\alpha, \sigma_\alpha^{-1} I).$$

Let $F(c, \alpha)$ be the negative log-posterior distribution of c and α for entire data omitting the constants, which is written as:

$$F(c, \alpha) = \sum_{j=1}^M \sum_{i=1}^{N_i} \frac{\sigma_R}{2} (R_{ji} - \eta_{ji})^2 + \sum_{j=1}^M \sum_{e=1}^{N_e} \frac{\sigma_{R'}}{2} (R'_{je} - \eta'_{je})^2 + \frac{\sigma_\alpha}{2} \sum_{j=1}^M \sum_{k=1}^K (\alpha_{jk} - \mu_\alpha)^2 + \frac{\sigma_c}{2} \sum_{j=1}^M (c_j - \mu_c)^2.$$

Our goal is to minimize the objective function $F(c, \alpha)$.

Noticing the conditional independence relationships among model parameters, we can devise the following alternating optimization algorithm. In this algorithm, we repeat two optimization steps, one with respect to c and the other with respect to α until convergence.

Step 1 Optimization w. r. t c

Given α fixed, the parameters c_j for each student j is independent of each other. Therefore, we could work on the independent optimization problem for a particular j . This implies that a large problem could be decomposed into relatively small problems, which leads to an efficient algorithm. To solve each optimization problem, we can use any numerical optimization method. In our implementation, we employ the gradient descent method.

$$c^{\text{new}} = c^{\text{old}} - r_1 g(c),$$

where $r_1 > 0$ is the step length, and the gradient $g(c) =$

$\frac{\partial}{\partial c} F(c, \alpha)$ is given as:

$$g(c) = -\sigma_{R'} \sum_{e=1}^{N_e} (R'_{je} - \eta'_{je}) \left(\sum_{k=1}^M \alpha_{jk} q'_{ke} \right) + \sigma_c (c_j - \mu_c)$$

Step 2 Optimization w. r. t α

Similarly, given c fixed, parameter α_{jk} for each student j and each question k is independent. Therefore, we can optimize each α_{jk} in parallel.

$$\alpha^{\text{new}} = \alpha^{\text{old}} - r_2 \frac{\partial}{\partial \alpha} g(\alpha),$$

where $r_2 > 0$ is the step length too, and $g(\alpha) = \frac{\partial}{\partial \alpha} F(c, \alpha)$ is

$$g(\alpha) = -\sigma_R \sum_{i=1}^{N_i} q_{ki} (R_{ji} - \eta_{ji}) - \sigma_{R'} \sum_{e=1}^{N_e} c_j q'_{ke} (R'_{je} - \eta'_{je}) + \sigma_\alpha (\alpha_{jk} - \mu_\alpha).$$

4 Experiment

4.1 Datasets

We collected data from the computerscience courses of the University of Science and Technology of China to verify our model. We train our model in three kinds of data sets, a real data set from course “data structure”, a data set from course “network security”, and a synthetic data set. All of three data sets contain students’ score R and R' on theoretical questions and experiments, Q and Q' that indicate the knowledge concepts examined by questions. In real data sets, the scores of students and the knowledge concepts required by questions or experiments are given by teachers. A brief summary of these data sets is shown in table 1.

Table 1. Overview of datasets

| Data Set | # Student | # Skill | # Problem | |
|--------------------|-----------|---------|-------------|--------------|
| | | | Theoretical | Experimental |
| “Data Structure” | 96 | 19 | 58 | 10 |
| “Network Security” | 194 | 7 | 10 | 8 |
| Synthetic | 1000 | 20 | 200 | 50 |

4.2 Competing methods

We will compare three methods to demonstrate the effectiveness and interpretation of our proposed cognitive modelling framework.

(I) The item response theory (IRT)^[8] is a typical method of cognitive diagnosis, which assumes that students’ abilities are unidimensional and items (the items in IRT are the questions in cognitive diagnostic scenarios) are locally independent;

(II) Probabilistic matrix factorization (PMF)^[20] is one of the popular methods in matrix factorization. The

main idea of this method is to factorize the score matrix into two matrices, one of which represents potential characteristics of users and the other represents potential characteristics of items. Then it uses these two matrices to predict the new scores;

(III) Fuzzy CDF^[21] introduces the concept of the fuzzy system to CDM so that cognitive diagnosis can be used for objective problems. Therefore, the model can predict students’ scores as continuous values. It combines logistic regression used in IRT and Q matrix used in DINA to perfect itself.

4.3 Experimental setup

As mentioned above, our data sets include theoretical and experimental questions, but none of the above three competing methods can train such data sets. In order to solve this problem, we train in two situations; ① Treat both kinds of questions as the same kind of question; ② Divide two kinds of problems into two data sets and train them separately.

Parameters α and β are the theoretical and practical abilities of students given by the model. We use these two parameters to predict students' scores. The reliability of the model is judged by the error of the predicted score. Since the evaluation metrics of the proposed model and other methods is the error between predicted student's scores and real student's scores. We use two metrics MAE and RMSE to measure the value of errors. Both our CDF-CSE and other baseline approaches are implemented by using python on a Core i5 3.2 Ghz machine with Windows 7 and 8 GB memory.

4.4 Results

To observe how these methods behave at different sparsity levels, we construct different sizes of training sets, with 10% to 80% of score data of each data set, and the rest for testing. For comparison, we tuned parameters to record the best performance of each algorithm. In experiments, we consider three implementations of matrix factorization method PMF. That is, PMF-5D, PMF-10D and PMF-KD represent the PMF with 5, 10 and K (the number of knowledge concepts) latent factors, respectively. Thus, there are totally six results in each split.

Figures 2–4 show the predicting scores performance results of our CDF-CSE and baseline methods on different data sets. From Figures 2–4, we observe that, our CDF-CSE performs the best over all data sets. Specifically, by combining educational hypotheses it beats PMF, by quantitatively analysing examinees from a fuzzy viewpoint, it beats IRT, and by combining the theory and the experiment it beats all other methods. More importantly, with the increasing of sparsity of training data (training data ratio declines from 80% to 20%), the superiority of our CDF-CSE method becomes more and more significant. For instance, when the training data is 20% and under the metric of MAE, the improvement of CDF-CSE compared to the best baseline method can reach 47.8%, 65.8%, and 49.8% on each data set that treat both kinds of questions as the same kind of question.

It is obvious that the proposed model is more accurate than other methods. The reason is that our CDF-CSE can be trained from both theoretical, and experimental questions on a data set that separates two kinds of questions. That is, compared with other models that only consider one kind of problem, our model can obtain more information in the training process. On data sets that consider two kinds of questions as one, our model will provide different probability hypotheses for two kinds, which is in line with the real experience. Even if in the special situation that students have the same probability distribution of scores on both questions, our model can work well. However, only one probability distribution can be considered in other

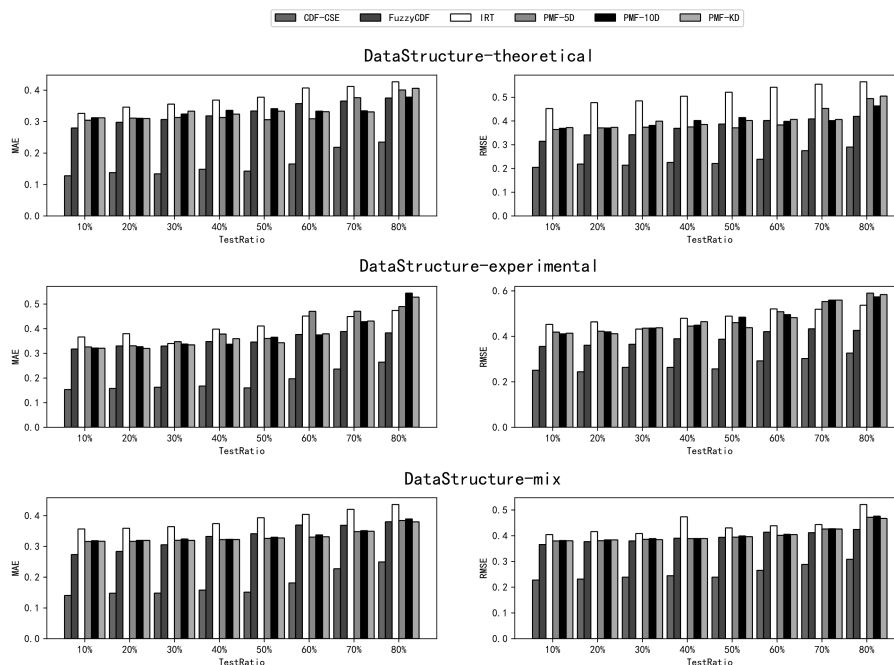


Figure 2. The performance of each model in data set “data structure”

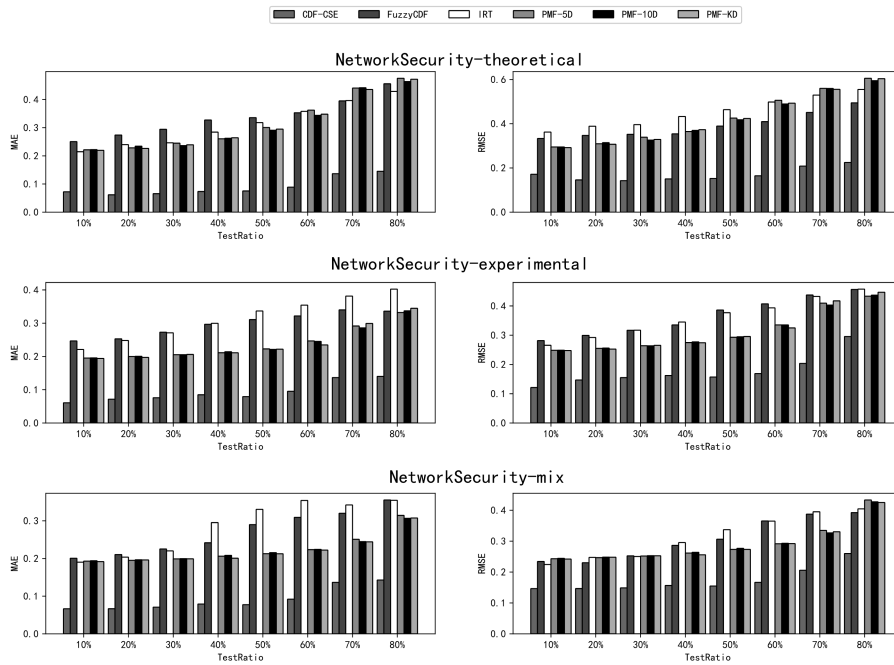


Figure 3. The performance of each model in data set “network security”

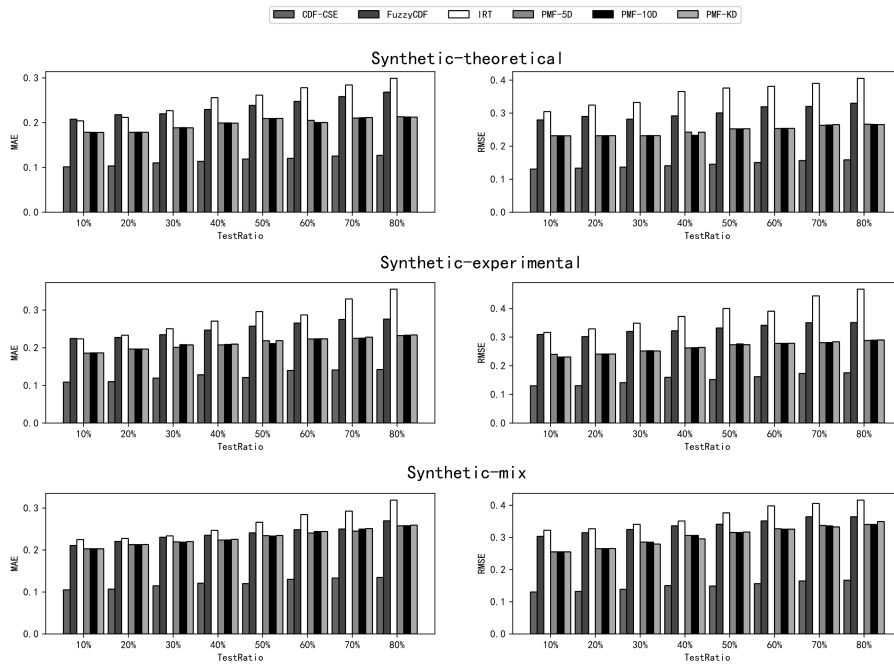


Figure 4. The performance of each model in synthetic data set

models, which would produce errors inevitably. In other words, a student’s performance is identical on the theoretical and experimental problems that examine the same knowledge concepts. Our model makes good use of this characteristic. When the model observes that students have a good grasp of theory in a knowledge concept, it would predict that the students have a good grasp of the experiment in this knowledge concept. At the same time, we can also use the experimental

performance of students to deduce his/her theoretical ability. This method is in line with the teaching experience. And we also see from the experimental results that the method is feasible. Therefore, we can see from the experimental results that the competitive method performs poorly on a data set sometimes. In summary, CDF-CSE captures the characteristics of students more precisely and it is also more suitable for real-world and synthetic scenarios, where the data is

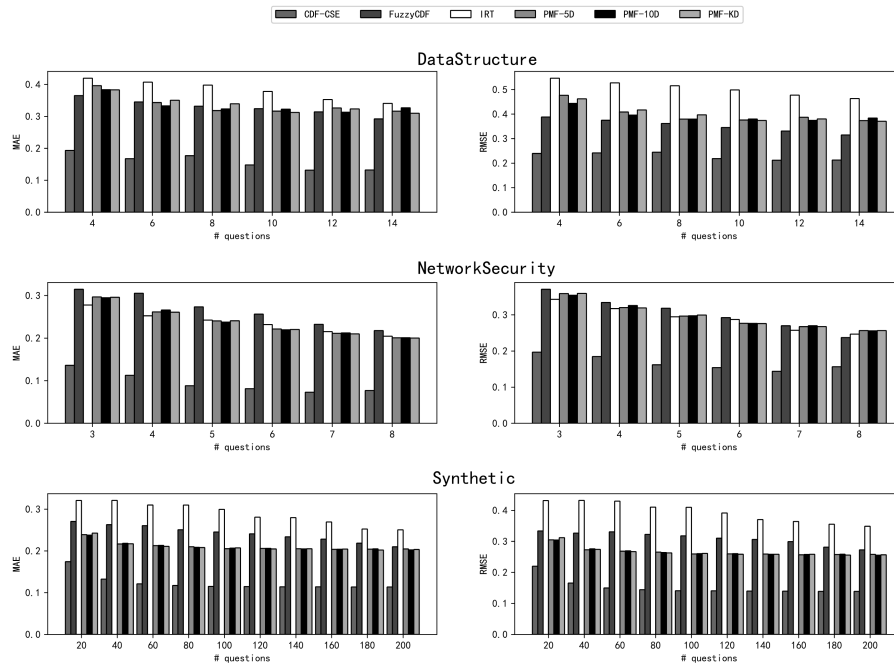


Figure 5. The performance of each model in teaching process

sparse.

Besides, we hope that cognitive diagnostic models will not only have an evaluation of students after course is completed, but also can give students feedbacks during the course. In this way, cognitive diagnostic models can help students find their shortcomings and adjust their learning plans in a timely manner while they are studying. Therefore, we conducted an experiment based on the process of the course, that is, training in the chronological the order of theoretical and experimental arrangement. In this experiment, we fixed the data amount of training set to 80%, and the rest was used as the test set. At the same time, in the chronological order, only a few questions are used for training at the beginning, and then the amount of questions is gradually increased.

Figure 5 shows the results in teaching process of our CDFCSE and baseline methods on data sets. We can see from the pictures that, our CDF-CSE still performs best on all data sets. From the perspective of the following course, our model can perform better at an early stage (when there are fewer knowledge concepts and questions). As the amount of data increases, the advantages of our model become gradually obvious. For example, when the number of questions is small and under the MAE metric, compared with the best competing method, the improvement of our CDF-CSE can reach 37.8%, 42.5% and 27.7% on each data set. And under the same circumstances with more questions, the improvement of our model can reach 32.3%, 36.5% and 45.6% on each data set. This proves that it is feasible to combine both theoretical and experimental

performances to analyze students when there are few data. At the early stage of teaching, our model can also analyze the characteristics of students well. With the development of courses, the analysis results will be more and more accurate. In summary, the proposed model can follow up a complete computer education course as well.

4.5 Discussion

It can be seen from experimental results that CDF-CSE outperforms other competing methods in predicting student performance. This is because our model can extract the common characteristics from the two kinds of questions and distinguish their differences at the same time, so as to diagnose students' theoretical and practical cognition. Compared with other models, our results are more adequate and accurate. The experimental results also confirm that our model can be applied to different situations. Therefore, we can use different data of students to analyze more comprehensive cognitive information. We can make conclusion that our model can solve the problem of inaccurate feedbacks in the traditional teaching. In future applications, CDF-CSE can obtain interpretative cognitive analysis results for students, which can be used for composing a detailed and human readable diagnosis report. At the same time, its prediction of the student performance can help teachers know the teaching situation and conduct their personalized teaching. In courses of the computer education, it can help students improve themselves, as well as assist teachers to adjust their teaching plans for students.

5 Conclusion

In this paper, we propose a cognitive diagnostic framework (CDF-CSE) for the computer science education, so that we can explore students' theoretical and practical abilities in the computer science education at the same time. Specifically, our model defines students' programming ability and combine students' theoretical ability with their experimental ability. We propose an algorithm to optimize the parameters of the model. The experimental results on the data sets of the computer science courses of the University of Science and Technology of China demonstrated that CDF-CSE can diagnose characteristics for each student quantitatively and interpretatively, thus performing better in predicting students' performance. In particular, experiments on real computer education data sets have proved that our model can be applied in real courses to help students understand their programming level in the future. And our model can get accurate results in the teaching process, which facilitates teachers to know students' learning status and adjust their teaching plan.

However, there is still some room for improvement. First, CDF-CSE confronts the problem of high computational complexity currently, it is important for us to design an efficient parameter optimization algorithm. Second, the prerequisite relationship of knowledge concepts should be considered for cognitive modelling. Last but not least, there are many code-related features that should be considered in the cognitive diagnosis model for the computer science education. Besides, we plan to apply our improved model in actual courses to prove the practicability of our model, and perfect our model according to the feedback.

Acknowledgments

The work is supported by The Key research project for Teaching of Anhui Province (2019jyxm0001); Research project for Teaching of Anhui Province (2020jyxm2304).

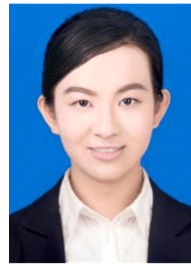
Conflict of interest

The authors declare no conflict of interest.

Author information



HU Xinying is currently a PhD student in the Department of Computer Software and Theory under the supervision of Prof. Sun Guangzhong at University of Science and Technology of China. Her research focuses on educational data mining.



HE Yu is currently a PhD student under the supervision of Prof. Sun Guangzhong at University of Science and Technology of China. Her research mainly focuses on educational data mining.



SUN Guangzhong (corresponding author) received his PhD degree in Computer Software and Theory from University of Science and Technology of China. He is currently a professor at University of Science and Technology of China. His research interests include high performance computing, algorithm optimization, and big data processing.

References

- [1] Kulkarni C E, Bernstein M S, Klemmer S R. Peerstudio: Rapid peer feedback emphasizes revision and improves performance. Proceedings of the Second ACM Conference on Learning@ Scale. Vancouver, Canada; ACM, 2015: 75–84.
- [2] Leighton J, Gierl M. Cognitive Diagnostic Assessment for Education: Theory and Applications. Cambridge University Press, 2007.
- [3] Dibello L V, Roussos L A, Stout W. 31a review of cognitively diagnostic assessment and a summary of psychometric models. Handbook of statistics, 2006, 26: 979–1030.
- [4] Haertel E. An application of latent class models to assessment data. Applied Psychological Measurement, 1984, 8(3): 333–346.
- [5] Junker B W, Sijtsma K. Cognitive assessment models with few assumptions, and connections with nonparametric item response theory. Applied Psychological Measurement, 2001, 25(3): 258–272.
- [6] De La Torre J. The generalized Dina model framework. Psychometrika, 2011, 76(2): 179–199.
- [7] De La Torre J, Douglas J A. Higher-order latent trait models for cognitive diagnosis. Psychometrika, 2004, 69(3): 333–353.
- [8] Embretson S E, Reise S P. Item Response Theory. New York; Psychology Press, 2013.
- [9] Wu R, Liu Q, Liu Y, et al. Cognitive modelling for predicting examinee performance. Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence. Buenos Aires, Argentina; ACM, 2015: 1017–1024.
- [10] Gu J, Wang Y, Heffernan N T. Personalizing knowledge tracing: Should we individualize slip, guess, prior or learn rate? International Conference on Intelligent Tutoring Systems. Springer, 2014: 647–648.
- [11] Leony D, Pardo A, De La Fuente Valentín L, et al. Glass: A learning analytics visualization tool. Proceedings of the 2nd International conference on Learning Analytics and

- Knowledge. Vancouver, Canada: ACM, 2012: 162–163.
- [12] Toscher A, Jahrer M. Collaborative filtering applied to educational data mining. KDDCup, 2010.
- [13] Thainghe N, Drumond L, Krohngrimberghe A, et al. Recommender system for predicting student performance. *Procedia Computer Science*, 2010, 1(2): 2811–2819.
- [14] Díez J, Luaces Ó, Alonso-Betanzos A, et al. Peer assessment in MOOCs using preference learning via matrix factorization. *NIPS Workshop on Data Driven Education*. 2013.
- [15] Desmarais M C. Mapping question items to skills with nonnegative matrix factorization. *ACM SIGKDD Explorations Newsletter*, 2012, 13(2): 30–36.
- [16] Sun Y, Ye S, Inoue S, et al. Alternating recursive method for q-matrix learning. *Proceedings of the 7th International Conference on Educational Data Mining*. London: ACM, 2014: 14–19.
- [17] Tha-Nnghe N, Schmidt-Thieme L. Multi-relational factorization models for student modeling in intelligent tutoring systems. *Seventh International Conference on Knowledge and Systems Engineering*. Ho Chi Minh City, Vietnam: IEEE, 2015: 61–66.
- [18] Cen H, Koedinger K, Junker B. Learning factors analysis – a general method for cognitive model evaluation and improvement. *International Conference on Intelligent Tutoring Systems*. Springer, 2006: 164–175.
- [19] Baker R S J D, Corbett A T, Alevan V. More accurate student modeling through contextual estimation of slip and guess probabilities in Bayesian knowledge tracing. *International Conference On Intelligent Tutoring Systems*. Springer, 2008: 406–415.
- [20] Mnih A, Salakhutdinov R R. Probabilistic matrix factorization. *Advances in Neural Information Processing Systems*. 2008: 1257–1264.
- [21] Liu Q, Wu R, Chen E, et al. Fuzzy cognitive diagnosis for modelling examinee performance. *ACM Transactions on Intelligent Systems and Technology*, 2018, 9(4): 1–26.

基于概率图模型的计算机课程教学认知诊断框架

胡心颖, 何钰, 孙广中*

中国科学技术大学计算机科学与技术学院, 安徽合肥 230026

摘要: 提出一种新的认知诊断框架, 用于在计算机课程教学中评估学生的理论学习能力与代码实践能力. 基于概率图模型, 引入学生代码能力, 同时对学生的理论能力以及应用能力进行建模. 进而提出一个并行优化算法以快速对模型进行训练. 在多个数据集上进行的实验结果表明, 与基准模型相比, 该模型在 MAE、RMSE 指标上都有较大幅度的提升. 所提出的模型可为计算机课程教学提供更准确全面的分析结果.

关键词: 认知诊断, 概率图模型, 教育数据挖掘