Collaborative Learning Team Formation: A Cognitive Modeling Perspective

Yuping Liu[1], Qi Liu[1], Runze Wu[1], Enhong Chen[1(✉)], Yu Su[2], Zhigang Chen[1], and Guoping Hu[3]

[1] University of Science and Technology of China, Hefei 230026, China
{liuyup,wrz179}@mail.ustc.edu.cn,
{qiliuql,cheneh}@ustc.edu.cn, zgchen@iflytek.com
[2] Anhui University, Hefei, China
yusu@iflytek.com
[3] Anhui USTC IFLYTEK Co., Ltd., Hefei, China
gphu@iflytek.com

Abstract. With a number of students, the purpose of collaborative learning is to assign these students to the right teams so that the promotion of skills of each team member can be facilitated. Although some team formation solutions have been proposed, the problem of extracting more effective features to describe the skill proficiency of students for better collaborative learning is still open. To that end, we provide a focused study on exploiting cognitive diagnosis to model students' skill proficiency for team formation. Specifically, we design a two-stage framework. First, we propose a cognitive diagnosis model SDINA, which can automatically quantify students' skill proficiency in continuous values. Then, given two different objectives, we propose corresponding algorithms to form collaborative learning teams based on the cognitive modeling results of SDINA. Finally, extensive experiments demonstrate that SDINA could model the students' skill proficiency more precisely and the proposed algorithms can help generate collaborative learning teams more effectively.

1 Introduction

Collaborative learning is the instructional use of small heterogeneous group of students who team together (e.g., 5 to 8 students [10]) to work on a structured activity. Over the past decades, many researches have confirmed the effectiveness of this type of learning [23]. By maximizing the promotion of skills of students, collaborative learning can not only help stdd,ents exhibit higher academic achievement, but also can reduce the workload of instructors [22].

Indeed, the success of collaborative learning can be only guaranteed by assigning each student to the right team. Generally speaking, two types of solutions, based on manual decisions or automatical algorithms, have been studied for this team formation problem. In manual approaches, students may select their own teammates [11], or teams are assigned by instructors [20]. Unfortunately, by

S.B. Navathe et al. (Eds.): DASFAA 2016, Part II, LNCS 9643, pp. 383–400, 2016.
DOI: 10.1007/978-3-319-32049-6_24

self-selecting best students tend to cluster and leave weak ones to shift for themselves. By instructors-assigning, it is almost impossible for instructors to assign all students effectively. Thus, it is necessary to form learning teams automatically. Based on students' specific characteristics, some researchers focus on generating heterogenous teams which mix students of different levels [14], while others try to form teams which can quantize and maximize the gain of students [1]. Usually, characteristics of students in these studies are directly extracted from the biographical data, simple performance attributes or personality traits [27]. Meanwhile, without the real-world data, simulated values on these characteristics are used to design the team formation algorithms [2,12,13,18,27]. In spite of the importance of the existing research, features in these studies are too simple to capture students' skill proficiency very well, and thus, the performance of the corresponding team formation solutions could be further improved. Actually, one of the best ways to get the skill proficiency of students is to model the cognitive information of them, e.g., from their performance in the exams [4,8]. However, there are still two challenges to be addressed for exploiting cognitive diagnosis. Firstly, how to precisely quantify the skill proficiency of students? Secondly, how to design the appropriate algorithms to automatically get collaborative learning teams based on this type of feature?

To conquer these two challenges, we propose a two-stage framework to apply cognitive diagnosis for collaborative learning team formation. Specifically, in the first stage we propose a novel cognitive diagnosis model Soft-DINA (SDINA). Compared to the existing diagnosis model DINA [4,21], which quantifies the students' skill proficiency in binary values (either 0 or 1), SDINA is able to model students with continuous values. Then, the output of SDINA is further exploited to generate collaborative learning teams in the second stage. Following the views that students in the same team should be diverse and the improvement of each student should be maximized, we consider two optimization objectives – dissimilarity based objective and gain based objective, and we propose effective algorithms to generate collaborative learning teams for each of these objectives. Finally, the results of extensive experiments demonstrate that (1) SDINA could model the students' skill proficiency by predicting their performance (i.e., exam scores) more precisely, and (2) the proposed algorithms can help generate collaborative learning teams effectively under several evaluation metrics. The main contributions of this paper could be summarized as:

- To the best of our knowledge, this is the first comprehensive attempt for the problem of collaborative learning team formation by introducing cognitive diagnosis to extract features of students' skill proficiency.
- We propose a novel cognitive diagnosis model SDINA, which improves existing model DINA. SDINA automatically quantifies students' skill proficiency in continuous values for more accurate analysis of students.
- Given students' skill proficiency, we propose two objectives following the existing research achievements, and then we design effective algorithms to generate collaborative learning teams for each of these objectives.

2 Related Work

In this section, we will introduce the related studies in two categories.

2.1 Student Modeling

Here we focus on team formation-oriented student modeling methods. In traditional collaborative team formation problems, the students were modeled by features extracted directly from biographical data like age, gender, or some simple performance attributes like grades, self-evaluation, peer-assessment, or personality traits like learning styles [27]. For instance, Hwang *et al.* [15] considered the number of already known concepts and scores of a pre-test to model students.

Though few of existing team formation studies explored examination records, this kind of data has been widely used for other student modeling tasks, e.g., performance prediction. Many data mining efforts have been conducted, for instance, matrix factorization (MF) technique [19] has been adopted by considering student as user, problem as item, and student's score on a problem as rating. E.g., Toscher *et al.* [25] utilized singular value decomposition (SVD) to model and predict students. But, latent factors of students inferred by MF are unexplainable which limits the applicability of MF in scenarios where the explanation of skills need to be specified.

To model students with examination data in an interpretative way, psychometricians in education psychology have developed a series of cognitive diagnosis models (CDMs) [8]. By capturing the students' cognitive characteristics, CDMs can predict students' performance and obtain targeted remedy plan for each student. Generally speaking, there are two main categories of CDMs: continuous ones and discrete ones. The fundamental continuous CDMs are *item response theory* (*IRT*) models [9], which characterize a student by a continuous variable, i.e., latent trait. However, IRT is unable to get the latent cognitive character like students' skill proficiency. For discrete CDMs, the basic method is *deterministic inputs, noisy "and" gate model* (*DINA*) [4,21]. DINA describes a student by a latent binary vector variable which denotes whether she masters the skills required by the exam or not. And the specific relationship between problems and skills is a prior knowledge given by education experts (e.g. the exam designer). E.g., based on fuzzy system, Runze Wu *et al.* [26] proposed a solution for cognitive examinee modeling from both objective and subjective problems.

2.2 Team Formation

Existing studies on collaborative learning team formation can be broadly split into two basic types. The first type focuses on forming heterogenous or homogeneous teams considering multiple student characteristics. E.g., authors in [18] designed an algorithm named SPOS to form heterogeneous teams. Similarly, the fuzzy c-means and random selection algorithm were used in [2] for formulating homogenous and heterogeneous teams. Unfortunately, all these works implemented their algorithms based on simulated student characteristics. The second type focuses on forming teams which can maximize the gain of students. E.g., Rakesh *et al.* [1] proposed a framework for grouping students in order to maximize the overall gain of students. Given specific objectives, there are also researches which formed teams through optimization methods. E.g.,Virginia *et al.* [27] proposed a deterministic crowing evolutionary algorithm to form teams. However, few of the existing team formations exploit cognitive diagnosis results as features to better model students. Therefore, the problem of how to apply this type of feature to design more suitable team formation solutions is still open.

Besides collaborative learning team formation, the traditional team formation problem usually focused on forming a team from a large set of candidates of experts such that the resulting team is best suited to perform the assignment. This kind of team formation entails set-cover and is modeled by experts' skills and their collaboration network [16]. In contrast, the collaborative team formation problem is a partition problem rather than a set-cover one.

3 Collaborative Learning Team Formation

We will introduce details of our two-stage framework in this section. The flowchart is shown in Fig. 1. Given students' examination results, we propose a cognitive diagnosis model SDINA to automatically quantify the skill proficiency of students. Then, we propose two objectives with heuristic algorithms to form collaborative learning teams which can facilitate the promotion of all students.

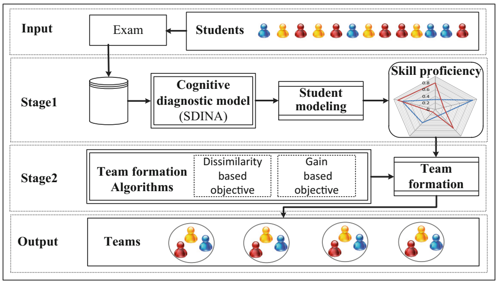

Fig. 1. Our two-stage framework.

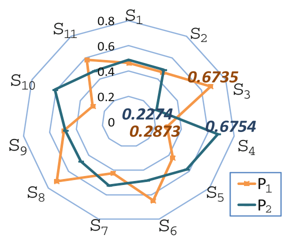

Fig. 2. Example of students' proficiency in skills S_1 to S_{11}.

3.1 Cognitive Modeling of Students' Skill Proficiency

In the following, we assume there are U students $\mathcal{P} = \{P_1, P_2, \ldots, P_U\}$, participating in the same course, e.g., math. We also assume $\mathcal{S} =\{S_1, S_2, \ldots, S_V\}$ to be a universe of V skills in this course, e.g., skills in a math course may include math concepts like set, formulas like computing sphere volume, process skills like calculation or induction. After cognitive modeling, each P_i will be associated with a vector of skill proficiency $\alpha_i = \{\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{iV}\}$, where α_{ij} ranges from 0 to 1 and represents that P_i' proficiency in S_j is α_{ij}. We summarize the proficiency of all students in all skills as $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_U\}$, which need to be evaluated according to an examination. Actually, Fig. 2 shows the proficiency of 2 real-world students in skills S_1 to S_{11}, which is the output by our SDINA. Although the two students have similar average abilities (about 0.5), there are still distinctively differences in their proficiency in certain skills. If these two students are put into a same team, they can learn from others' strong points to offset their own weakness. We will first briefly review the traditional cognitive diagnosis model DINA, and then show the way to get α by our SDINA.

DINA Review. *DINA* [4] (*the deterministic inputs, noisy and gate model*) assumes that each problem in an exam is involved with multiple predefined skills (tagged by education experts in advance), and then it characterizes a student by a binary vector variable, which denotes whether she has mastered the skills required in the exam. Specifically, for an exam designed to assess V skills of students, given a problem l, a student i, we observe a dichotomous response which is a binary variable R_{il} with a value in $\{0, 1\}$. The response indicates the correctness of the answer provided by the student i to the problem l, i.e., 1 represents true and 0 represents false. Then, DINA is defined as:

$$P(R_{il} = 1 | \alpha_i, s_l, g_l) = (1 - s_l)^{\eta_{il}} g_l^{1-\eta_{il}}. \tag{1}$$

Here, student i is characterized by a latent binary vector variable $\alpha_i = (\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{iV})$, i.e., $\alpha_{ij} = 1$ represents that student i has mastered skill j and vice versa, and problem l is characterized by two parameters: s_l represents carelessness or slipping; g_l is a guessing parameter. η_{il} is a latent variable that indicates whether student i is able to solve the problem l, and is defined as $\eta_{il} = \prod_{j=1}^{V} \alpha_{ij}^{q_{lj}}$ where q_{lj} indicates whether problem l requires skill j. It means student i is disable to solve problem l unless all of the skills required for problem l have already been mastered by her. As for parameter estimation, we could maximize the marginalized likelihood of Eq. (1), which can be implemented using *EM* algorithm [3]. Then, with the estimated parameters $\hat{s}_1, \hat{s}_2, \cdots, \hat{s}_Z$ and $\hat{g}_1, \hat{g}_2, \cdots, \hat{g}_Z$ (Z is the number of problems), α_i can be determined via maximizing the posterior probability given student i's response vector:

$$\begin{aligned} \hat{\alpha}_i &= argmax_\alpha P(\alpha | R_i) = argmax_\alpha L(R_i | \alpha, \hat{s}_l, \hat{g}_l) P(\alpha) \\ &= argmax_\alpha L(R_i | \alpha, \hat{s}_l, \hat{g}_l) = argmax_\alpha \prod_{l=1}^{Z} P(R_i | \alpha, \hat{s}_l, \hat{g}_l), \end{aligned} \tag{2}$$

Thus, for an exam with V skills, α has 2^V possible patterns and each pattern is assumed to be with an equal prior distribution without loss of generality.

Soft-DINA. In DINA, binary skill vector α_i of each student i can be found by maximizing the posterior probability. Though it is intuitive, this binary representation of students' skill proficiency is too coarse to characterize the mastery degree (cognitive level) of students. For instance, a student with a mastery degree of 0.9 in a specific skill and a student with 0.6 may have the same binary value of 1 based on DINA, while the significant difference is missing. For the sake of keeping as much information of vectors of proficiency as possible, we propose another cognitive model *Soft-DINA* or *SDINA* for short, which is an improvement of DINA, to get vectors of proficiency with continuous values.

Specifically, considering the 2^V kinds of α from all zeros $(0, 0, \ldots, 0)$ to all ones $(1, 1, \ldots, 1)$ and given the responses of a student i, R_i, each of these α is involved with one posterior probability $P(\alpha|R_i)$, DINA only chooses the specific skill vector α which can maximize the posterior probability. To precisely measure the probability that the student i masters a specific skill j, we propose to consider the posterior probability from all the possible α. Formally, in SDINA, we redefine the estimated skill vector $\tilde{\alpha}_i$ and calculate the posterior probability that student i masters skill j as follows:

$$\begin{aligned}\tilde{\alpha}_{ij} &= P(\alpha_{ij}=1|R_i) = \frac{\sum_{\alpha_{xj}=1} P(\alpha_x|R_i)}{\sum_{x=1}^{2^V} P(\alpha_x|R_i)} \\ &= \frac{\sum_{\alpha_{xj}=1} L(R_i|\alpha_x, \hat{s}_l, \hat{g}_l)P(\alpha_x)}{\sum_{x=1}^{2^V} P(\alpha_x|R_i)} = \frac{\sum_{\alpha_{xj}=1} \prod_{l=1}^{Z} L(R_{il}|\alpha_x, \hat{s}_l, \hat{g}_l)P(\alpha_x)}{\sum_{x=1}^{2^V} P(\alpha_x|R_i)},\end{aligned} \tag{3}$$

where $x = 1, 2, \ldots, 2^V$, represents the 2^V kinds of possible α_i, and the numerator part computes the probability of $\alpha_{ij} = 1$ in these 2^V kinds of possible α_x. To simplify the formulation, we also assume each α_x has an equal prior probability, then the equation above can be rewritten as follows:

$$\tilde{\alpha}_{ij} = \frac{\sum_{\alpha_{xj}=1} \prod_{l=1}^{Z} L(R_{il}|\alpha_x, \hat{s}_l, \hat{g}_l)}{\sum_{x=1}^{2^V} \prod_{l=1}^{Z} L(R_{il}|\alpha_x, \hat{s}_l, \hat{g}_l)}. \tag{4}$$

In this way, we get $\tilde{\alpha}_i = (\tilde{\alpha}_{i1}, \tilde{\alpha}_{i2}, \ldots, \tilde{\alpha}_{iV})$, a vector of continuous values between 0 and 1, where $\tilde{\alpha}_{ij}$ represents student i's proficiency(i.e. probability) in skill j. That is to say, $\tilde{\alpha}_{ij} = 1$ means i has fully mastered skill j, and $\tilde{\alpha}_{ij} = 0$ means i has not mastered skill j at all.

Note that, although the skills accessed by each problem are manually labeled, it is feasible and commonly used in pedagogy [8]. In fact, to construct an examination, designers must clearly delineate the assement purpose, specically describe what skills are measured and develop proper assessment tasks[1].

3.2 Collaborative Learning Team Formation

In this subsection, we show the way to form teams based on α. Assume $\mathcal{G} = \{G_1, G_2, \ldots, G_M\}$ to be a set of M teams. We will put students into teams, with two basic constraints. Firstly, only one team is assigned to a student, making sure that a student only belongs to one team. Secondly, the team size is better to be equal, with a difference of no more than one student, to ensure fairness and

[1] There are also studies about the automatic labeling of skills [7], which is beyond the scope of our research.

team balance. Formally, given the students' skill proficiency, the collaborative learning team formation problem can be formulated as follows.

Problem 1 *(Collaborative Learning Team Formation): Given a set of students* $\mathcal{P}$*, each student* P_i*'s proficiency* $\alpha_i = (\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{iV})$*, and a set of teams* $\mathcal{G} = \{G_1, G_2, \ldots, G_M\}$*. Under the following constraints,*

$$\begin{aligned}&G_{k_1} \bigcap G_{k_2} = \emptyset,\\&-1 \leq |G_{k_1}| - |G_{k_2}| \leq 1,\\&k_1, k_2 = 1, 2, \ldots, M, k_1 \neq k_2\end{aligned} \tag{5}$$

assign every student P_i *to a team* G_k*,* $P_i \subseteq G_k$*,* $k = 1, 2, \ldots, M$*. In order that, the promotion of all students in these* V *skills can be facilitated.*

With this definition, we should first clarify the measurements/objectives of a good team, and then, the specific algorithms to generate effective collaborative learning teams in terms of these objectives could be designed. Generally, there are two types of different objectives in existing studies, i.e., the skill proficiency of students in the same team should be heterogeneous [14] and the improvement of each student should be maximized [1]. In terms of these objectives and our extracted feature, we propose a *dissimilarity based objective* to form teams by maximizing the average dissimilarity of students within a team and a *gain based objective* to form teams by maximizing the average gain of students, respectively.

Dissimilarity Based Objective. According to [13], in a reasonably heterogeneous group student-scores reveal a combination of low, average and high student-scores. However, this measurement is limited to 3 discrete classes of only one attribute value (student-score). Indeed, a better mechanism is to use continuous values of several attributes, e.g., the continuous value of students' proficiency in several skills. Inspired by the heterogeneity definition in [12], we also use the average dissimilarity between team members as the metric of heterogeneous degree. Without loss of generality, the difference between P_{i_1} and P_{i_2} in the proficiency of S_j, $D_j(P_{i_1}, P_{i_2})$, is defined as $|\alpha_{i_1 j} - \alpha_{i_2 j}|$. Thus, the heterogeneity of G_k consisting of N students with respect to S_j, is defined as

$$HG_k(S_j) = \sum\nolimits_{i=2}^{N} D_j(P_i, P_{i-1}). \tag{6}$$

Here, the students in G_k have been sorted by values of skill proficiency in S_j, i.e., $\alpha_{1j} \leq \alpha_{2j} \leq \ldots \leq \alpha_{Nj}$. Then, the heterogeneity of G_k is computed as

$$HG_k = \sum\nolimits_{j=1}^{V} HG_k(S_j). \tag{7}$$

One step further, the *heterogeneity of the solution* $\mathcal{G}$ is the average of the heterogeneity of all teams in the solution, i.e.,

$$HG(\mathcal{G}) = \frac{\sum_{G_k \in \mathcal{G}} HG_k}{|\mathcal{G}|}. \tag{8}$$

Since not only heterogeneity but also the team balance, i.e., the difference of the heterogeneity among the teams, should be considered for the quality of the team formation solutions [18], we also define the *balance of the solution* $\mathcal{G}$ as

$$B(\mathcal{G}) = Variance(\{HG_k | \forall G_k \in \mathcal{G}\}). \tag{9}$$

That is, the teams in each solution should be as balanced as possible to ensure the fairness. Overall, the higher the solution heterogeneity is and the lower the solution balance is, the better the team formation result is.

Given this dissimilarity based objective, consisting of both solution heterogeneity $HG(\mathcal{G})$ and solution balance $B(\mathcal{G})$, we utilize the idea of clustering to solve the team formation problem. Intuitively, students could be first clustered using clustering algorithm, e.g., k-means [17], where features are their skill proficiency vectors. Then students of same cluster will be assigned to different teams. However,clusters under the classical k-means settings are often of different size, and this will have a negative effect on the team's heterogeneity. For instance, if there is a cluster with a very large size, which is bigger than the number of teams, according to the pigeon-hole principle , at least two students in the same cluster will be assigned to the same team. To address this problem, we think of an improved clustering method called *uniform k-means* to get uniform clusters with the same size. More specifically, in the process of k-means, the number of clusters is set to be $\lfloor \frac{U}{M} \rfloor$. Then, for every object(student), after calculating the distance between it and the center of every cluster, this student will be put into cluster which is not merely with shortest distance but also is not full, i.e., size of this cluster is no larger than M. After this, students are equally divided into $\lfloor \frac{U}{M} \rfloor$ clusters. Next, students in same cluster should be assigned into M different candidate teams. In this way, as students with similar skill proficiency will be put into different teams, the dissimilarity based objective can be easily achieved. The Uniform K-means Based algorithm (UKB), is summarized in Algorithm 1.

Algorithm 1. UKB: the uniform k-means based algorithm

Require:
    The set of U students $\mathcal{P} = \{P_1, P_2, \ldots, P_U\}$; The number of teams M;
Ensure:
    The set of teams $\mathcal{G} = \{G_1, G_2, \ldots, G_M\}$;
1: Divide U students into $\lfloor \frac{U}{M} \rfloor$ clusters using uniform k-means;
2: Determine the size of teams, each one with student number of $\lfloor \frac{U}{M} \rfloor$ or $\lfloor \frac{U}{M} \rfloor + 1$;
3: Calculate the number of students that every team still needs;
4: **for** each cluster c **do**
5:     **while** c is not empty **do**
6:         Put one student into every not-full team;
7:         Calculate the number of students that every team still needs;
8:     **end while**
9: **end for**
10: **return** $\mathcal{G}$;

Gain Based Objective. In addition to the dissimilarity based objective, another intuitive approach of measuring the quality of a team is to quantize the promotion of every student. Inspired by [1], we define a gain function to measure the promotion of students in terms of their skill proficiency.

In collaborative learning teams, the students can promote their skills through mutual exchanges and emulations. As a general rule, there are two factors which can influence the students' promotion. The first factor is the proficiency of each student, i.e., a student with higher proficiency is easier to promote than a student with lower proficiency. Another factor is the gap between her and the other students in the team, i.e., a student who collaborates with more capable teammate can get more knowledge. According to these facts, we define a student's promotion in a skill as follows.

Suppose there is a non-empty team G_k with N students $G_k = \{P_1, P_2, \ldots, P_N\}$. G_k has a vector of maximum proficiency in every skill $a_k = \{a_{k1}, a_{k2}, \ldots, a_{kV}\}$, a_{kj} is the *maximum proficiency* in S_j among the students in G_k, i.e., $a_{kj} = MAX(\alpha_{1j}, \alpha_{2j}, \ldots, \alpha_{Nj})$. Then the *leader* of G_k in S_j, namely L_{kj} is the student with maximum proficiency in S_j,

$$L_{kj} = \{P_i | \alpha_{ij} = a_{kj}, i = 1, 2, \ldots, N\}. \tag{10}$$

Now, we put a new student P_i into G_k, if P_i is not the leader in S_j, i.e., $\alpha_{ij} < a_{kj}$, then P_i's *promotion* in S_j, $Q_j(P_i, G_k)$, is defined as:

$$Q_j(P_i, G_k) = (a_{kj} - \alpha_{ij}) \cdot \alpha_{ij}. \tag{11}$$

Here, $(a_{kj} - \alpha_{ij})$ is as the gap between P_i and L_{kj}, which is always positive. The definition is in conformity with the actual situation. For instance, there is a team with a leader who has a proficiency of 0.9 in a certain skill, suppose we put three new students A, B, C with proficiency of 0.2, 0.5, 0.8 respectively in the team. According to our definition, the promotions of them will be 0.14, 0.20, 0.08. A gets a small promotion of 0.14 because A has a low proficiency which brings bad influence for the promotion, C only gets 0.08 because there is only a little gap between C and the leader. Only B gets a big promotion of 0.20 due to B's higher proficiency and the bigger gap between B and the leader. For simplicity, we ignore the leader's promotion in her leading skill. Then, we define $Q(P_i, G_k)$, the *gain* of P_i as the overall promotion in every skill,

$$\begin{aligned} Q(P_i, G_k) &= \textstyle\sum_{j=1}^{V} Q_j(P_i, G_k) \\ &= (a_k - \alpha_i) \cdot \alpha_i^T. \end{aligned} \tag{12}$$

Next,the average gain of G_k will be:

$$Q(G_k) = \frac{\sum_{P_i \in G_k} Q(P_i, G_k)}{|G_k|}. \tag{13}$$

Here, $|G_k|$ is the number of students in G_k. Finally, the *average gain of a solution* $\mathcal{G}$, $Q(\mathcal{G})$, can be defined as the average gain of all the teams,

$$Q(\mathcal{G}) = \frac{\sum_{G_k \in \mathcal{G}} Q(G_k)}{|\mathcal{G}|}. \tag{14}$$

We use the solution gain $Q(\mathcal{G})$ as the evaluative criterion for the quality of team formation solutions. We also propose a team formation algorithm which consists of two steps: First, the leader in each skill is chosen for each team; Then, all the non-leader students are put into teams according to their gains.

Specifically, $Q(\mathcal{G})$ is a monotone-increasing function of a_{kj}, so for maximizing $Q(\mathcal{G})$, the leader in each skill for each team should have proficiency as greater as possible. Given a set of U students and the number of teams M, the leader-choosing process is as follows. For each skill S_j, firstly we pick out M students $\mathcal{P} = \{P_1, P_2, \ldots, P_M\}$ with maximum proficiency. Secondly, for every student $P_i \in \mathcal{P}$, if P_i has been a leader in a team, then P_i will be the leader in S_j of this team; Otherwise, we choose a team G_k which still has no leader in S_j, and if G_k is not full, then P_i will be the leader in S_j of G_k, or else, a student in G_k with the maximum proficiency in S_j will be picked out as the leader.

After the leaders have been chosen, we show the way to put all non-leader students into teams. Assume all teams have the same size of λ, and the λ students in team G_k are presented by $P_{ki}, i = 1, 2, \ldots, \lambda$. Then $Q(\mathcal{G})$ will be

$$\begin{aligned} Q(\mathcal{G}) &= \frac{\sum_{G_k \in \mathcal{G}} Q(G_k)}{|\mathcal{G}|} \\ &= \frac{1}{|\mathcal{G}| \cdot \lambda} \cdot \sum_{G_k \in \mathcal{G}} \sum_{i=1}^{\lambda} Q(P_{ki}, G_k). \end{aligned} \tag{15}$$

Obviously, $Q(\mathcal{G})$ will only be determined by $\sum_{G_k \in \mathcal{G}} \sum_{i=1}^{\lambda} Q(P_{ki}, G_k)$ since $\frac{1}{|\mathcal{G}| \cdot \lambda}$ is a constant. Also, $Q(\mathcal{G})$ increases with $Q(P_{ki}, G_k)$. As one student can only belong to one team, it is naturally to put P_i to G_k which can maximize $Q(P_i, G_k)$. In addition, since size of teams is limited, students with higher proficiency should be put into teams in priority. However, such team formation result violates the principle of heterogeneity because students with higher proficiency tend to be put into teams with higher maximum proficiency. To avoid this unbalanced result, we propose an algorithm which takes both the gain of students and the average level of teams into consideration, to get balanced teams.

To be specific, there are two factors determining the selection of groups for non-leader students. One is the gain of this student, another is the average level of the team. Here, we define LP_i, the level of a student P_i, as the average of P_i's proficiency in every skill, and LG_k, the *average level of a team* G_k, as the average level of all the students in team G_k,

$$LP_i = \frac{\sum_{j=1}^{V} \alpha_{ij}}{V}, \quad LG_k = \frac{\sum_{P_i \in G_k} LP_i}{|G_k|}.$$

Then we define a *balanced gain vector* $BG_i = \{BG_{i1}, BG_{i2}, \ldots, BG_{iM}\}$, where BG_{ik} is called the *balanced gain* of P_i if putting her to G_k,

$$BG_{ik} = \frac{Q(P_i, G_k)}{LG_k}. \tag{16}$$

With the above definition, the entire process of B*alanced* G*ain* B*ased algorithm* (*BGB*), is shown in Algorithm 2. In summary, we first choose the leaders

Algorithm 2. BGB: the balanced gain based algorithm

Require:
The set of U students $\mathcal{P} = \{P_1, P_2, \ldots, P_U\}$; The number of teams M;
Ensure:
The set of teams $\mathcal{G} = \{G_1, G_2, \ldots, G_M\}$;
1: Determine the size of teams, each one with student number of $\lfloor \frac{U}{M} \rfloor$ or $\lfloor \frac{U}{M} \rfloor + 1$;
2: Select the leaders in every skill for each team.
3: Sort all the non-leader students by their levels LP in descending order;
4: **for** each sorted students P_i **do**
5: Calculate the balanced gain vector BG_i;
6: Put student P_i in a not-full team G_k with maximum BG_{ik};
7: **end for**
8: **return** $\mathcal{G}$;

in each skill for each team, and then sort all the non-leader students by their levels LP in descending order. Next, for each sorted student P_i we put her into a not-full team G_k with maximum BG_{ik}. In this way, the higher level students will be put into relatively low level teams and vice versa, getting her a relatively high gain and keeping the heterogeneity of the teams, simultaneously.

4 Experiments

Firstly, we use the prediction of students' scores to evaluate the effectiveness of SDINA. Secondly, we make an expert evaluation to explore the effectiveness of features extracted by SDINA. At last, we evaluate the performance of our proposed team formation algorithms from various aspects.

4.1 Experimental Setup

Our experiments are conducted on three real-world datasets and two simulated ones. The real-world datasets contain two real private datasets and a public online dataset. The public dataset is Tatsuoka's fraction subtraction dataset [5], consisting of scores from middle school students on fraction subtraction problems. The two private datasets[2] are from two final math exams for high school students. Each of these three datasets is represented by a score matrix and also has multiple predefined skills. We denote these datasets as *FrcSub, Math1* and *Math2*. The brief summary is shown in Table 1. Figure 3 gives a brief preview of these datasets, where each column for each subfigure stands for a problem and each row for a student. The black element means the student is wrong in the problem, while white one means right. The two simulated datasets are made up of 500 and 1,000 students with 10 features[3]. The value of each feature is

[2] They will be publicly available after the paper acceptance.
[3] Unlike the real-world datasets, the simulated ones only consist of students with values between 0 and 1 on some features rather than students' test scores.

generated by random sampling from a uniform distribution of 0 to 1. We denote these two datasets as *SiData1* and *SiData2*.

All experiments are implemented by Matlab on a Core i5 3.10 Ghz machine with Window 7 and 4 GB memory.

Table 1. Datasets Summary.

Dataset	#Student	#Skill	#Problem
FrcSub	536	8	20
Math1	4,206	11	12
Math2	3,907	16	12

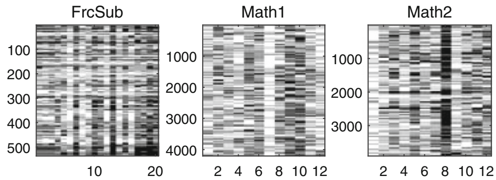

Fig. 3. The preview of the datasets.

4.2 Evaluation on Student Modeling

To demonstrate the effectiveness of *SDINA* on modeling students' skill proficiency, we conduct experiments of predicting students' scores (right as 1 and wrong as 0) on each problem. We perform 5-fold cross validation on the real-world datasets, i.e., 80 % of the students are randomly selected for training while the rest for testing. We consider two baseline approaches:

- *DINA* [4]: a cognitive diagnosis model which is detailed in Sect. 3.1.
- *PMF* [19]: a latent factor model, widely used in recommending system.

We record the best performance of each method by tuning their parameters, e.g., the latent dimension of PMF is set to be 10. As this task is actually a binary classification problem, *Accuracy* and *F1-measure* are used as evaluation metrics.

The experimental results are shown in Fig. 4. SDINA performs better than DINA and PMF in both Accuracy and F1 over all datasets. In Accuracy DINA performs better than PMF but in F1 PMF performs better than DINA. As a cognitive diagnosis model, SDINA is effective in modeling students' skill proficiency.

4.3 Evaluation on Feature Selection

After evaluating SDINA's effectiveness on student modeling, to demonstrate SDINA's effectiveness on team formation, first the performance of three different kinds of features on team formation are evaluated by educational experts, then teams based on these features are visually displayed for better illustration.

Specifically, we compare reasonablity of teams formed with the same algorithm based on three kinds of features, i.e., the continuous skill proficiency inferred by SDINA, the binary skill proficiency inferred by DINA and the raw examination scores[4]. Since conducting large-scale in-classroom experiments are

[4] The latent factor getting by PMF has not been used here since it's unexplainable.

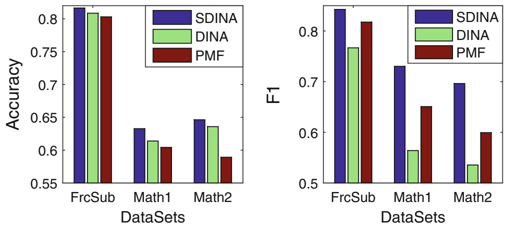

Fig. 4. Performance on score prediction.

Table 2. Gold standard evaluation.

Feature	HR
Raw score	0.066
DINA	0.267
SDINA	**0.667**

currently impractical, we choose to collect a *Gold Standard* [24] to evaluate effectiveness of team formation by various features. As educational experts, teachers are obviously able to give a relatively fair and convincing assessment with decades of educating experience. We will first specify the process of collecting a gold standard and then compare the performance of three kinds of teams.

To construct a gold standard, we simulate the team formation on real dataset and then ask corresponding teachers who are familiar with the chosen students, to evaluate the effectiveness of different methods. Specifically speaking, due to the labour cost of manual assessment, we first randomly draw five classes with 283 students in total, then we form teams with a fixed size (e.g. 5) by three different features based on UKB. And for each class, we randomly draw three formed teams for each method, that is, nine teams will be chosen and randomly ordered. Subsequently, we ask teachers of the five classes to pick three most reasonable teams for each class out of their understanding of students. So, nine of the forty-five formed teams are chosen and regarded as most effective.

Taking the gold standard as ground truth, we compute hit rate (HR) [6] for each method. Here this metric measures how closely the output of a method is to the gold standard and is defined as $HR_i = \frac{|T_i \cap GS|}{|GS|}$. Here, HR_i is the HR of the ith team formation method, T_i represents the teams formed by the ith method and GS means the teams picked by the gold standard. As is shown in Table 2, the effectiveness of team formation by SDINA greatly outperforms DINA and Raw score methods by a quantitative and more accurate cognitive diagnosis, and at the meanwhile DINA obtains more satisfying results than Raw score method.

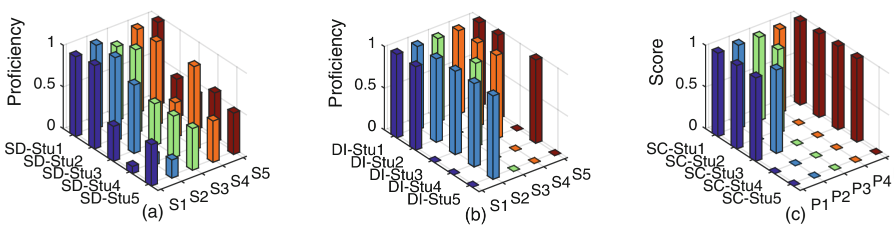

Fig. 5. An example of teams by using three kinds of features.

One step further, for better illustration about the difference between teams formed by three kinds of features, Fig. 5 shows a toy visualized example. We use UKB to form five-students teams on public dataset FrcSub. Figure 5(a) shows a team formed through continuous skill proficiency output by SDINA. Here, SD-Stu1 to SD-Stu5 represent five students and S1 to S5 represent five skills. Figure 5(b) shows a team formed through binary skill proficiency output by DINA. DI-Stu1 to DI-Stu5 represent students and S1 to S5 represent skills which are the same with Fig. 5(a). Figure 5(c) shows a team formed directly using students' score on every problem. P1 to P5 represent five problems respectively. Only five skills or problems are shown for better visualization. We can observe that team in Fig. 5(a) contains different levels of students, SD-Stu1 and SD-Stu2 are of highest ability. Others can promote skills following the lead of them. Since proficiency of students in each skill is also of great difference, students can promote their skills through learning from each other. Compared with other two kinds of features team formation using SDINA has better explanation. SDINA not only can support automatic team formation but also can provide guidance for manually forming teams.

4.4 Evaluation on Team Formation

In this subsection, we first fix features as continuous skill proficiency inferred by SDINA and evaluate performance of UKB and BGB on real-world datasets. Then, to demonstrate that UKB is not just effective with features of SDINA's output, we evaluate its performance on two simulated datasets. Quality of team formation solutions is evaluated with *Heterogeneity*, *Balance* and *Gain*, which are defined in Sect. 3.2. For Heterogeneity and Gain, higher value is better while for Balance, lower value is better. Our algorithms are compared with:

- *SPOS*: short for Semi-Pareto Optimal Set, which is proposed in [18].
- *RANDOM*: a standard random algorithm to form heterogeneity teams [2].

We should note that the algorithm in [1] is not chosen as a baseline, because it can only be applied to form teams for students with 1-dimensional ability.

Firstly, we perform team formation experiments on three real-world datasets and use the output of SDINA as input features for all the algorithms, to make sure grouping algorithms will be comparable. Figure 6 shows the experimental results. The subfigures in row 1 to row 3 represent three datasets and columns represent three measurements. The X-axis in each subfigure represents team size from 5 to 8 since the optimal team for collaborative learning should contain 5 to 8 students [10]. Please note that, if the student number is not evenly divisible by the team size, then actually team size here represents the basic student amount in every team, and there is at least one team which has one more student. The result shows the effectiveness of our proposed algorithms. In terms of the dissimilarity based objective, i.e., Heterogeneity and Balance, UKB outperforms the two baselines among all the team sizes in three datasets, and among the baselines, SPOS has a relatively good performance. Similarly, in terms of the gain based objective, BGB outperforms the two baselines in all cases.

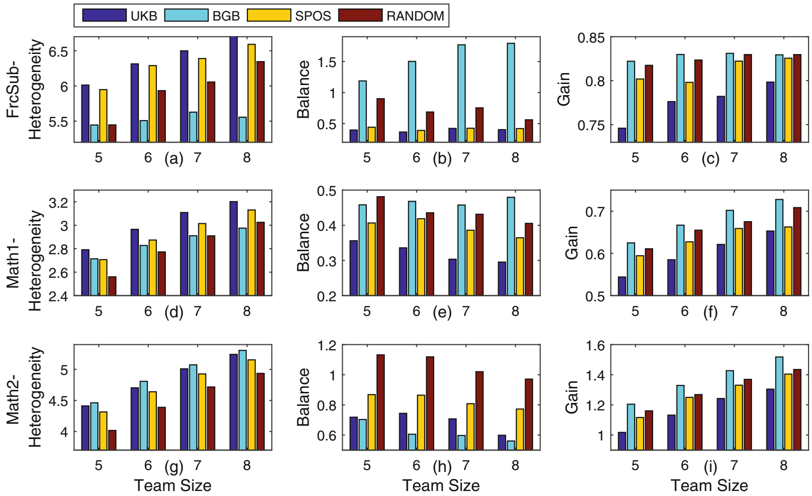

Fig. 6. Collaborative team formation performance on three real-world datasets.

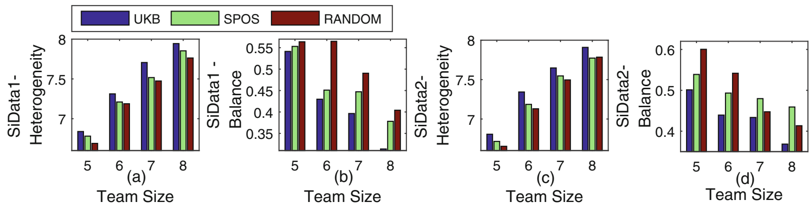

Fig. 7. Collaborative team formation performance on simulated datasets.

Table 3. Runtime(seconds).

Datasets	UKB	BGB	SPOS	RANDOM
FrcSub	1.936	0.174	1.811	**0.122**
Math1	595.2	**16.89**	750.3	22.7
Math2	664.2	**10.73**	620.4	19.5

Secondly, to demonstrate that UKB is not just effective with features of SDINA's output, we perform team formation experiments on datasets with simulated features. We can see from Fig. 7 that UKB has larger Heterogeneity and lower Balance among all team sizes on both datasets. We don't test BGB as it only focuses on features of students' skill proficiency.

Moreover, Table 3 shows runtime of each method to form five-students teams. BGB and RANDOM run much faster in general. For our methods UKB run faster than SPOS on Math1, BGB run faster than RANDOM on Math1 and Math2.

5 Discussion

The experimental results demonstrate that SDINA could better model students with continuous skill proficiency. Through more accurate analysis on students and following the existing research achievements, UKB and BGB could generate more effective collaborative learning teams for dissimilarity based objective and gain based objective. In the meanwhile, the team formation results are explainable, which makes the framework has more practical value.

Both stages of this general framework may be further improved. Firstly, we can employ SDINA on more data (e.g., the homework data) for feature extraction. Secondly, relationship between these two objectives could be studied and maybe the trade-off between them can be researched. Optimization methods should be tried to formulate an optimization problem for maximizing gain and heterogeneity, and minimizing balance. Thirdly, we plan to design more efficient solutions than UKB and we would like to consider more influence factors to get a more reasonable definition of students' promotion in BGB. Finally, we plan to apply this theoretical research in the real-world teaching and learning, e.g., we already served high schools where we collected the data. Indeed, given that modeling students' cognitive skills for collaborative learning has largely been neglected, there are many research directions remain to be explored.

6 Conclusion

In this paper, we designed a two-stage framework to exploit cognitive diagnosis for collaborative learning team formation. Firstly, we proposed a cognitive diagnosis model SDINA, which can automatically quantify students' skill proficiency in continuous values. Secondly, we proposed two objectives, the dissimilarity based objective and the gain based objective with heuristic algorithms to solve the team formation problem. At last, extensive experiments on several datasets demonstrated that our SDINA could model the students' skill proficiency more precisely and the proposed algorithms can help generate collaborative learning teams more effectively. We hope this work could lead to more future studies.

Acknowledgements. This research was partially supported by grants from the National Science Foundation for Distinguished Young Scholars of China (Grant No. 61325010), the Natural Science Foundation of China (Grant No. 61403358) and the Science and Technology Program for Public Wellbeing (Grant No. 2013GS340302). Qi Liu gratefully acknowledges the support of the Youth Innovation Promotion Association of CAS and acknowledges the support of the CCF-Intel Young Faculty Researcher Program (YFRP).

References

1. Agrawal, R., Golshan, B., Terzi, E.: Grouping students in educational settings. In: SIGKDD, pp. 1017–1026. ACM (2014)
2. Christodoulopoulos, C.E., Papanikolaou, K.: Investigation of group formation using low complexity algorithms. In: Proceeding of PING, Workshop, pp. 57–60 (2007)
3. De La Torre, J.: Dina model and parameter estimation: a didactic. J. Educ. Behav. Stat. **34**(1), 115–130 (2009)
4. De La Torre, J.: The generalized dina model framework. Psychometrika **76**(2), 179–199 (2011)
5. DeCarlo, L.T.: On the analysis of fraction subtraction data: the dina model, classification, latent class sizes, and the q-matrix. APM (2010)
6. Deshpande, M., Karypis, G.: Item-based top-N recommendation algorithms. ACM Trans. Inf. Syst. (TOIS) **22**(1), 143–177 (2004)
7. Desmarais, M.C.: Mapping question items to skills with non-negative matrix factorization. ACM SIGKDD Explor. Newsl. **13**(2), 30–36 (2012)
8. DiBello, L.V., Roussos, L.A., Stout, W.: 31a review of cognitively diagnostic assessment and a summary of psychometric models. Handb. Stat. **26**, 979–1030 (2006)
9. Embretson, S.E., Reise, S.P.: Item response theory for psychologists. Psychology Press, New York (2013)
10. Gall, M.D., Gall, J.P.: The discussion method. The psychology of teaching methods, (75 ppt 1), pp. 166–216 (1976)
11. Gibbs, G.: Learning in teams: a tutor guide. Oxford Centre for Staff and Learning Development (1995)
12. Gogoulou, A., Gouli, E., Boas, G., Liakou, E., Grigoriadou, M.: Forming homogeneous, heterogeneous and mixed groups of learners. In: Proceeding ICUM, pp. 33–40 (2007)
13. Graf, S., Bekele, R.: Forming heterogeneous groups for intelligent collaborative learning systems with ant colony optimization. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) ITS 2006. LNCS, vol. 4053, pp. 217–226. Springer, Heidelberg (2006)
14. Hooper, S., Hannafin, M.J.: Cooperative cbi: The effects of heterogeneous versus homogeneous grouping on the learning of progressively complex concepts. J. Educ. Comput. Res. **4**(4), 413–424 (1988)
15. Hwang, G.-J., Yin, P.-Y., Hwang, C.-W., Tsai, C.-C., et al.: An enhanced genetic approach to composing cooperative learning groups for multiple grouping criteria. Educ. Technol. Soc. **11**(1), 148–167 (2008)
16. Lappas, T., Liu, K., Terzi, E.: Finding a team of experts in social networks. In: Proceedings of the 15th SIGKDD, pp. 467–476. ACM (2009)
17. Li, Q., Wang, P., Wang, W., Hu, H., Li, Z., Li, J.: An efficient K-means Clustering Algorithm on MapReduce. In: Bhowmick, S.S., Dyreson, C.E., Jensen, C.S., Lee, M.L., Muliantara, A., Thalheim, B. (eds.) DASFAA 2014, Part I. LNCS, vol. 8421, pp. 357–371. Springer, Heidelberg (2014)
18. Mahdi, B., Fattaneh, T.: A semi-pareto optimal set based algorithm for grouping of students. In: ICELET, pp. 10–13. IEEE (2013)
19. Mnih, A., Salakhutdinov, R.: Probabilistic matrix factorization. In: Advances in neural information processing systems, pp. 1257–1264 (2007)
20. Ounnas, A., Davis, H., Millard, D.: A framework for semantic group formation. In: ICALT, pp. 34–38. IEEE (2008)
21. Ozaki, K.: Dina models for multiple-choice items with few parameters considering incorrect answers. In: APM (2015)

22. Slavin, R.E.: Cooperative learning: theory, research, and practice, vol. 14. Allyn and Bacon, Boston (1990)
23. Smith, K.A., Sheppard, S.D., Johnson, D.W., Johnson, R.T.: Pedagogies of engagement: classroom-based practices. JEE **94**(1), 87–101 (2005)
24. Štajner, T., Thomee, B., Popescu, A.-M., Pennacchiotti, M., Jaimes, A.: Automatic selection of social media responses to news. In: 19th ACM SIGKDD, pp. 50–58. ACM (2013)
25. Toscher, A., Jahrer, M.: Collaborative filtering applied to educational data mining. In: KDD Cupp (2010)
26. Wu, R., Liu, Q., Liu, Y., Chen, E., Su, Y., Chen, Z., Hu, G.: Cognitive modelling for predicting examinee performance. In: Proceedings of the 24th International Conference on Artificial Intelligence, pp. 1017–1024. AAAI Press (2015)
27. Yannibelli, V., Amandi, A.: A deterministic crowding evolutionary algorithm to form learning teams in a collaborative learning context. Expert Syst. Appl. **39**(10), 8584–8592 (2012)