Grouping Students in Educational Settings*

Rakesh Agrawal
Microsoft Research
rakesha@microsoft.com

Behzad Golshan
Boston University
behzad@cs.bu.edu

Evimaria Terzi
Boston University
evimaria@cs.bu.edu

ABSTRACT

Given a class of large number of students, each exhibiting a different ability level, how can we group them into sections so that the overall gain for students is maximized? This question has been a topic of central concern and debate amongst social scientists and policy makers for a long time. We propose a framework for rigorously studying this question, taking a computational perspective. We present a formal definition of the grouping problem and investigate some of its variants. Such variants are determined by the desired number of groups as well as the definition of the gain for each student in the group.

We focus on two natural instantiations of the gain function and we show that for both of them the problem of identifying a single group of students that maximizes the gain among its members can be solved in polynomial time. The corresponding partitioning problem, where the goal is to partition the students into non-overlapping groups appear to be much harder. However, the algorithms for the single-group version can be leveraged for solving the more complex partitioning problem. Our experiments with generated data coming from different distributions demonstrate that our algorithm is significantly better than the current strategies in vogue for dividing students in a class into sections.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications-Data Mining

Keywords

Clustering, Partitioning, Teams, Groups, Large Classes, Massive Courses, MOOC

---

1. INTRODUCTION

That good education is critical for enhancing societal growth and individual prosperity is a widely accepted tenet [18, 33]. While the problem of providing high quality education is multi-faceted and complex [8, 14], one particular problem that has vexed social scientists and policy makers for a long time is how to create groupings of students so that they can augment their learning from the teacher with cooperative learning from each other [3, 30, 32]. Two popular strategies are to group students either homogeneously (stratified by ability level, so low-ability students are in one section, high-ability students in another) or heterogeneously (students of all ability levels in one section) [11, 27]. The verdict from the empirical studies on which of these is more effective is inconclusive [15, 20, 29]. The public sentiment toward these strategies has also swayed back and forth over the years [7, 24].

We endeavour to rigorously study the problem of grouping students in a class, by taking a computational perspective. We consider a setting in which groupings are decided on per subject basis for students within a class. Each student i is associated with *ability* $\theta_i \in [0, 1]$ in the corresponding subject. The ability θ_i can be estimated by administering a test designed using modern practices such as those based on item response theory [9, 16]. Students are able to increase their abilities through interactions and collaborations with more capable peers [19, 31]. The higher ability members also gain as teaching others and giving help has been shown to be positively correlated to increase in ability [5, 32]. We thus strive to form groups such that the overall gain for students is maximized.

We present a formal definition of the grouping problem and describe several of its variants. More specifically, we consider two high-level problems: the 1-GROUP and the ℓ-GROUPS problem. In the former, the goal is to identify a single group from a large set of students such that the gains of the students constituting the group is maximized. In the latter, the goal is to partition a large set of students into smaller groups of approximately the same size such that the sum of the gains of all groups is maximized. Clearly, different definitions of "gain" lead to different instantiations of the 1-GROUP and the ℓ-GROUPS problems.

We consider two types of gain functions – each formalizing different viewpoint. The first maximizes the number of students who improve their abilities by interacting with the higher ability students, while the second incorporates the extent of these improvements. Thus, the former maximizes the number of students that benefit, while the latter

is geared towards pulling up the weakest students as much as possible. Depending upon the social goal the school is pursuing, it can accordingly choose the gain function.

For both these functions, we show that the 1-GROUP problem can be solved optimally in polynomial time. For lack of optimum algorithms for the ℓ-GROUPS version, we provide effective heuristics. Our experimental evaluation with generated data demonstrates that our design yields groupings that beat the current strategies. The experiments with different distributions of abilities of students demonstrate that this superior performance is independent of the underlying distribution of the abilities of the students.

Roadmap: The rest of the paper is organized as follows. After reviewing the related work in Section 2, we formally define our framework in Section 3. In Sections 4 and 5, we describe the two problem variants we study and their corresponding algorithms. In Section 6, we describe our experiments. Finally, we conclude with a summary and directions for future work in Section 7.

2. RELATED WORK

Although their importance in the educational process has been identified by educators and social scientists alike, to the best of our knowledge we are the first to introduce the formulations for computationally addressing the problem of grouping students. Naturally, the nature of our problem is related to a large body of work both in the domain of education as well as computer science. We review some of this work below.

Cooperative learning: The origins of the cooperative learning theory can be traced back to at least as early as the 1937 work of May and Doob who found that people who worked together were more successful in attaining the same goals than those who worked independently [26]. Modern cooperative learning theory is heavily influenced by the Vygotsky's idea of co-construction, central to which is the notion of the zone of proximal development (ZPD) – "... the distance between the actual developmental level as determined by independent problem solving and the level of potential development as determined through problem solving under adult guidance or in collaboration with more capable peers" [31]. In our formulation, ZPD for a student can be thought of as the distance between the student's ability and that of the group. One of our objective is striving to maximize the number of students who are able to increase their ability to the group level, while the other maximizes the total increase in the ability of the group members.

Forming teams of experts: Recently, there has been a lot of work on the formation of teams of experts whose goal is to complete a given project task [1, 2, 12, 21, 22, 25]. In these settings, experts are viewed as multi-dimensional vectors describing the expertise of individuals across different aspects. Then the goal is to select a team that best fits the task at hand. In contrast, our students are described by 1-dimensional values – namely their ability in the subject. Moreover, our goal is not to find a single team that can complete a specific task, but to partition the given set of students into teams from which their performance can benefit. These differences in the respective settings lead to different computational problems as well, and therefore the algorithms used for existing team-formation formulations cannot be used for our setting.

Clustering: At a high level, the ℓ-GROUPS version of our problem can be also viewed as a classical clustering problem [17]; afterall, in both cases the goal is to partition an input set of objects. Although in many of these clustering problems each cluster is represented by the *average* of the points in the cluster (exactly as in our setting) their goal is to find a partitioning that minimizes the sum of the distances of each points to its corresponding average. On the other hand, in our ℓ-GROUPS problem the goal is to maximize the number of students that are *below* the corresponding average. This difference in the objective functions makes our problem distinct from existing work on clustering. Specifically, many of the existing clustering problems are solvable in polynomial time for dimensionality 1, while some version of our ℓ-GROUPS problems appear to be NP-hard even though our input consists of 1-dimensional points.

3. FRAMEWORK

Assume there is a population of n students $S = \{1, \ldots, n\}$, studying a particular subject. Each student i is associated with *ability* $\theta_i \in [0, 1]$ (determined using techniques such as item response theory [9, 16]). For simplicity, we will assume that the students have distinct abilities and they are ordered in decreasing order of their abilities, i.e., $\theta_1 > \theta_2 > \ldots > \theta_n$.

We will be interested in forming groups of students $T \subseteq S$, such that students on average benefit by their participation in T. In order to quantify the benefit that a group provides to its members, we need to define the *collective ability* $\widehat{\Theta}_T$ of a group T. Following [28], we assume that the collective ability of a group $T \subseteq S$ is the average ability level of its members. That is, $\widehat{\Theta}_T = 1/|T| \sum_{i \in T} \theta_i$.

We consider the collective ability $\widehat{\Theta}_T$ of a group as being an important characteristic of the group. More specifically, the collective ability $\widehat{\Theta}_T$ of a group T partitions the members of T into two sets: the *leaders* and the *followers*. The set of leaders L_T of group T are all the members of T with ability above $\widehat{\Theta}_T$. That is,

$$L_T = \{i \in T \mid \theta_i \geq \widehat{\Theta}_T\}.$$

Similarly, the set of followers F_T of group T are all the members of T with ability below $\widehat{\Theta}_T$. That is,

$$F_T = \{i \in T \mid \theta_i < \widehat{\Theta}_T\}.$$

When the group T is clear from the context, we drop the subscript T when referring to the set of leaders and followers of T.

The participation in a group amplifies both the leaders and the followers. If we use function $A_f(i, T)$ (resp. $A_\ell(i, T)$) to denote the gain of a follower (resp. leader) i by participating in group T, then we define the gain of a group T as the sum of the gains of its members. That is,

$$\mathcal{A}(T) = \sum_{i \in F_T} A_f(i, T) + \sum_{i \in L_T} A_\ell(i, T). \qquad (1)$$

The gain of the followers may be a result of the fact that they are able to increase their abilities through interactions and collaborations with more capable peers [19, 31]. The leaders also gain as teaching others and giving help has been shown to be positively correlated to increase in ability [5, 32].

In this work, we will only focus on the gain of the followers and leave incorporating the gain of leaders for future work.

That is, the gain of a group T

$$A(T) = \sum_{i \in F_T} A_f(i, T). \qquad (2)$$

We consider two instances of the $A(T)$ function, namely the *count-based* gain Ac and the *value-based* gain Av.

The count-based gain function: Given a group T with leaders L_T and followers F_T, we define the count-based gain $\text{Ac}(T)$ as:

$$\text{Ac}(T) = \sum_{i \in F_T} \text{Ac}_f(i, T), \qquad (3)$$

where $\text{Ac}_f(i, T) = 1$ for every $i \in F_T$. In other words, the count-based gain of group T is simply the number of followers in the class. These are the students who are able to increase their ability through collaboration with more capable peers [19, 31].

The value-based gain function: A limitation of the count-based gain function is that it considers all followers gaining the same by their participation in T. The value-based gain function takes into consideration the actual values of the followers' abilities as well as the collective ability of the group. More specifically, given a group T with leaders L_T and followers F_T, the value-based gain $\text{Av}(T)$ is defined as:

$$\text{Av}(T) = \sum_{i \in F_T} \text{Av}_f(i, T), \qquad (4)$$

where $\text{Av}_f(i, T) = (\widehat{\Theta}_T - \theta_i)$ for every $i \in F_T$. In other words, students with abilities that are further from $\widehat{\Theta}_T$ gain more by their participation in T.

The problems: In this paper we are interested in two problems: the 1-GROUP and the ℓ-GROUPS problems. The former takes as input the set of students and their abilities and aims to identify a single group T that maximizes the value of $A(T)$. The latter focuses on partitioning the input set of students into groups such that the total gain across groups is maximized. The formal definitions of these two problems are given below. We give these definitions in terms of the generic gain functions, but we also discuss their instantiations for the count and value-based functions.

PROBLEM 1 (1-GROUP). *Given a set of n students $S = \{1, \ldots, n\}$, identify a group $T \subseteq S$ of at most k students such that $A(T)$ is maximized.*

When we use the count-based (resp. value-based) gain function for computing the gain of the group T, then we refer to the 1-GROUP problem as the COUNT1G (resp. VALUE1G) problem. When the form of the gain function is not important we use the generic 1-GROUP term.

PROBLEM 2 (ℓ-GROUPS). *Given an integer k and a set of n students $S = \{1, \ldots, n\}$ (with $n = k\ell$) find a partition of S into groups $T_1, \ldots T_\ell$, where each group is of size k and $\sum_{i=1}^{\ell} A(T_i)$ is maximized.*

As before, when we use the count-based (resp. value-based) gain function for computing the gain of the group T, then we refer to the ℓ-GROUPS problem as the COUNTℓG (resp. VALUEℓG) problem. When the form of the gain function is not important we use the generic ℓ-GROUPS term.

4. COUNT-BASED GROUP FORMATION

In this section we study the computational complexity of the COUNT1G and the COUNTℓG problems and we give algorithms for solving them.

4.1 The COUNT1G problem

Clearly, the desired group will consist of a set of *leaders L* of highest ability who will pull up the group's overall ability and a set of *followers F* whose abilities will be below the group's ability yet their abilities will not be as low so as to decrease the overall ability of the group.

EXAMPLE 1. *Consider a set of five students with the following ability scores: $\theta_1 = 0.9$, $\theta_2 = 0.8$, $\theta_3 = 0.6$, $\theta_4 = 0.5$, and $\theta_5 = 0$. Imagine that we need to create a group of three students with maximum possible gain. If students 1 and 2 are picked together (with any other student as the third member), only the third member of this group will be a follower. On the other hand, if we pick students 1, 4 and 5 then the collective group ability would be $\frac{0.9+0.5+0}{3} = 0.466$, which makes only student 5 a follower. Alternatively, grouping student 2, 3 and 4 leads to a collective group ability of $\frac{0.8+0.6+0.5}{3} = 0.633$, which makes students 3 and 4 to be the followers. This example shows that grouping the strongest and weakest students together does not necessarily leads to the maximum possible gain.*

Our algorithm for the COUNT1G problem, which we call the L&F algorithm, finds an optimal group composition by efficiently identifying the best sets L and F of leaders and followers that will compose the group $T = L \cup F$. The pseudocode of the L&F algorithm is shown in Algorithm 1.

Algorithm 1 L&F

   Input: Set of students $S = \{1, \ldots, n\}$ with sorted abilities $\theta_1 > \theta_2 > \ldots > \theta_n$, Group size k.
   Output: Group T with the maximum $\text{Ac}(T)$.
1: $T = \emptyset$
2: **for** $i = 1 \ldots k$ **do**
3:    $L = \text{top-}i$ ability students
4:    **for** $j = i + 1 \ldots n - (k - i) + 1$ **do**
5:       $F = $ students with abilities θ_j, θ_{j+1}, $\ldots$, $\theta_{j+k-i-1}$
6:       **if** F and L satisfy feasibility constraint **then return** $T = L \cup F$

In order to see the intuition behind L&F and also understand why it is optimal, we start with the following lemma:

LEMMA 1 (FEASIBILITY CONSTRAINT). $\sum_{i \in F} \theta_i > (k \times \max_{i \in F} \theta_i) - \sum_{i \in L} \theta_i$.

PROOF. Observe that $\widehat{\Theta}_T$ should be larger than the highest ability score of a student in the set F. Thus, if θ_F represents the highest ability in set F, the following condition should be satisfied:

$$\widehat{\Theta}_T = \frac{\sum_{i \in L} \theta_i + \sum_{i \in F} \theta_i}{k} > \theta_F.$$

A simple rewrite of the above inequality yields the lemma. $\square$

We do not know a priori the number of leaders in set L. However, since the gain for a group T is defined to

be $\text{Ac}(T) = k - |L(T)|$, the optimal solution will have the minimal number of leaders. Our algorithm therefore tries increasing number of leaders, starting with the student of highest ability. The reason for preferring higher ability leaders is that the sum of their abilities appears as a negative term in the right hand side of the inequality in the feasibility constraint. Note that for any fixed set L, all possible sets of followers that satisfy the feasibility constraint lead to the same value of Ac. A computationally efficient way to pick the followers is to look for them in $k - |L|$ consecutive positions starting with the first student who is not in L. And that is what our algorithm does.

EXAMPLE 2. *To clarify the details of Algorithm 1, we revisit the students of Example 1. The algorithm starts by placing a single student (i.e., student 1) in the set of leaders. Then, the inner loop (lines 4-5) searches for the consecutive group of students that satisfy the feasibility constraint. First, students 2 and 3 are considered for the followers set, but they do not pass the feasibility test. Next, the algorithm tries students 3 and 4, who pass the test. Thus, the algorithm returns the group consisting of students $\{1, 3, 4\}$, having $\text{Ac} = 2$.*

Complexity: With a preprocessing step of complexity $O(n)$ for computing the cumulative sums of all the ability levels of all the students sorted in decreasing ability level, `L&F` algorithm has complexity $O(nk)$. If we take a closer look at Algorithm 1, we can observe that the outer loop (line 2) searches for the smallest possible number of students that should be placed in set L to lift students in set F. This linear search, however, can easily be replaced by a binary search. Based on this observation, a more efficient implementation of `L&F` runs in time $O(n \log k)$.

Remark 1: Problem 1 is defined so that the formed group T is of specific size k. A natural question is whether we can form a group T of *any* size. We can observe that the `L&F` algorithm can be used for this problem as well. For example, one can simply run the `L&F` algorithm for all values of $k \in \{1, \ldots, n\}$ and output the group with the best Ac score among all those reported for the different values of k.

Remark 2: As stated in Algorithm 1, `L&F` selects the follower set in such a way that the relatively high ability followers are grouped with the highest ability leaders. We will sometime refer to this version of `L&F` as `max-L&F` to emphasize that amongst all the feasible follower sets, the group formed by this algorithm has the highest-ability follower. One can easily modify Algorithm 1 to select the follower set to contain the lowest highest-ability follower or for that matter select a random set amongst all the feasible follower sets. We will refer to these versions as `min-L&F` and `any-L&F` respectively.

4.2 The COUNTℓG problem

In this section, we first show that the COUNTℓG problem is NP-complete and then discuss heuristics for solving it.

LEMMA 2. *When $k = n/2$ then the COUNTℓG is NP-complete.*

PROOF. We prove that COUNTℓG is NP-complete by reducing the SUBSET SUM problem [13] to this problem. The SUBSET SUM problem asks whether a given set of positive integers $W = \{w_1, w_2, \cdots, w_N\}$ can be partitioned into two groups W_1, W_2 such that the sum of numbers in W_1 equals the sum of numbers in W_2. In other words, if we use $z = \sum_{i=1}^{N} w_i$ to denote the sum of all numbers in W, we want the sum of the elements in W_1 to be equal to the sum of the elements in W_2 and thus equal to $\frac{z}{2}$.

Now, given an instance of the SUBSET SUM problem we create an instance of the COUNTℓG problem with $n = 2N+6$ students which we want to partition into 2 groups each of size $k = (N + 3)$. We create the instance of the COUNTℓG problem as follows: for each number w_i in W, we create a student with ability $\theta_i = -w_i$. Beside these low ability students, we create $n + 4$ students with abilities equal to 0, and two strong students with abilities $\frac{z}{2} + \epsilon$ where ϵ is a small value. We claim that (under this construction) the answer to the decision problem of SUBSET SUM is "yes" *iff* the optimal solution to the COUNTℓG would have a total Ac of $(2N + 4)$.

First, assume that there is a perfect partitioning of W into W_1 and W_2. Now, we can create two groups T_1 and T_2 in the COUNTℓG problem as follows. First, we place a strong student with ability $\frac{z}{2} + \epsilon$ in both T_1 and T_2. Then, we place the low ability students that correspond to the numbers in set W_1 (resp. W_2) in T_1 (resp. T_2). Finally, we distribute the normal students (i.e., those with ability equal to zero) over T_1 and T_2 to make both groups of the same size. Note that the size of each group must be $N + 3$ and

$$\widehat{\Theta}_{T_1} = \widehat{\Theta}_{T_2} = \frac{-\frac{z}{2} + \frac{z}{2} + \epsilon}{N + 3} = \frac{\epsilon}{N + 3} > 0.$$

Since the collective ability levels of both groups are positive, all students except the two strongest students are below the mean. Thus $\text{Ac}(T_1) + \text{Ac}(T_2) = N + 2 + N + 2 = 2N + 4$.

Now, assume that there is an optimal grouping of students into groups T_1 and T_2 each of size $(N+3)$ and total Ac equal to $(2N + 4)$ for COUNTℓG problem. Let z_1 and z_2 denote the sum of the negative abilities in T_1 and T_2 respectively. We would like to show that $z_1 = z_2$ which further implies a perfect partitioning of the set W. First, observe that due to the pigeon-hole principle, both groups should have at least one student with ability equal to zero. Moreover, one can observe that the two strong students (ability $= \frac{z}{2} + \epsilon$) should be in different groups. This is simply because these are the highest abilities and they can never be below the mean of any group. Thus, if they are placed together there would be at least 3 student above the group average (i.e., the two strong students and the strongest student of the other group).

Given that each group has exactly one student with ability $\frac{z}{2} + \epsilon$ and at least one student with ability equal to 0, the collective ability of both groups must be positive in order to guarantee an overall Ac equal to $2N + 4$. Formally,

$$\begin{cases} \widehat{\Theta}_{T_1} > 0 \\ \widehat{\Theta}_{T_2} > 0 \\ z_1 + z_2 = z \end{cases} \Rightarrow \begin{cases} \frac{\frac{z}{2} + \epsilon - z_1}{N+3} > 0 \\ \frac{\frac{z}{2} + \epsilon - z_2}{N+3} > 0 \\ z_1 + z_2 = z \end{cases}$$

This implies that

$$\begin{cases} \frac{z_2}{2} - \frac{z_1}{2} + \epsilon > 0 \\ \frac{z_1}{2} - \frac{z_2}{2} + \epsilon > 0 \end{cases} \Rightarrow |z_1 - z_2| < 2\epsilon.$$

Now, it is easy to see that by picking a small value of ϵ one can guarantee z_1 and z_2 to be equal. As mentioned earlier, this further implies that the students with negative abilities in group T_1 and T_2 define a perfect partitioning

of the elements in set W. Finally, to guarantee that our reduction take polynomial time, we would like to point that ϵ does not require to be arbitrary small, it can simply be smaller than the precision in which the ability scores are reported.

Given that the ability scores in an instance of COUNTℓG problem are bound to be between zero and one, it might seem that our reduction is incomplete as it creates students with negative ability scores. However, one can see that the optimal solution to an instance of COUNTℓG problem would not change if all the ability scores are shifted (i.e., incremented by a constant) or scaled (i.e., multiplied by a constant). Thus, one can simply renormalize the ability scores to be between zero and one without affecting the value of the optimal solution. $\square$

`IterL&F`: Given there is no polynomial-time algorithm for solving COUNTℓG, we propose `IterL&F`, which is an iterative heuristic. In every iteration of `IterL&F`, one group of size k is formed. The selection of the group is done using the `L&F` algorithm and using as input only those students that have not yet been assigned to any group. Clearly, there are n/k iterations of `IterL&F`, each taking time $O(n \log k)$. Therefore, the overall running time is $O(\frac{n^2 \log k}{k})$.

Depending upon the version of `L&F` algorithm employed (see Remark 2 in Section 4.1), we get three versions of `IterL&F`, namely `max-IterL&F`, `min-IterL&F` and `any-IterL&F`. Our experiments indicate that they obtain similar overall value of the AC objective, but the structure of the groups built by them exhibit differences. Intuitively, `max-IterL&F` groups the highest ability leaders with relatively high ability followers, whereas `min-IterL&F` groups them with the lowest ability followers, while `any-IterL&F` groups them with mixed ability followers.

5. VALUE-BASED GROUP FORMATION

We next study the VALUE1G and VALUEℓG problems.

5.1 The VALUE1G problem

We first show that the VALUE1G problem can be solved in polynomial time with an algorithm, which we call `Endpoints`. Algorithm 2 provides the pseudo-code of `Endpoints`.

Algorithm 2 `Endpoints`

Input: Set of students $S = \{1, \ldots, n\}$ with sorted abilities $\theta_1 > \theta_2 > \ldots > \theta_n$, Group size k.
Output: Group T with the maximum $\mathrm{Av}(T)$.
1: **for** $i = 1 \ldots k - 1$ **do**
2: $\quad L_i = \{1, ..., i\}$
3: $\quad F_i = \{(n - i + 1), ..., n\}$
4: $\quad \widehat{\Theta}_i = ((\theta_1 + ... + \theta_i) + (\theta_{n-i+1} + ... + \theta_n))/k$
5: $\quad$ **if** $\widehat{\Theta}_i > \theta_{n-i+1}$ **then**
6: $\quad\quad T_i = L_i \cup F_i$
7: $\quad$ **else**
8: $\quad\quad T_i = \phi$
9: **return** $\arg\max_{T_i} \mathrm{Av}(T_i)$

We start with the following lemma:

LEMMA 3. *Given a group T that maximizes $\mathrm{Av}(T)$ and consists of x leaders and $y = k - x$ followers, it must be the case that $L = \{1, ..., x\}$ and $F = \{(n - y + 1), ..., n\}$.*

PROOF. We definitely need to pick as followers the y students with the lowest ability since $\mathrm{Av}(T)$ will be strictly worse for any other choice of y students. We also need to select the top-x ability students as the leaders; this is because the top-x ability students maximize the value of $\widehat{\Theta}_T$ and as a result also the value of our objective $\mathrm{Av}(T)$. $\square$

However, since the values of x and y are not known a priori, `Endpoints` iterates over all possible values of x and y that satisfy $x + y = k$ and reports the optimal solution in time $O(k)$. The IF condition ensures that the set F under consideration satisfies the constraint imposed by the definition of a follower set.

EXAMPLE 3. *For the students of Example 1,* `Endpoints` *returns $L = \{1, 2\} \cup F = \{5\}$. Contrast this group with the group formed by* `L&F` *in Example 2 for the same set of students. The latter uses the top student 1 to lift up the students 3 and 4, whereas* `Endpoints` *uses the two top students to lift up the very weak student 5.*

5.2 The VALUEℓG problem

Determining the exact complexity class of the VALUEℓG is still unresolved. Meanwhile, we propose two iterative heuristics for solving the problem: `IterEndpoints` and `RoundRobin`. Their performance characteristics is empirically studied later in the paper.

`IterEndpoints`: This heuristic builds upon the `Endpoints` algorithm. In each iteration, it identifies a group of size k by applying `Endpoints` to the remaining body of students. The running time of `IterEndpoints` is $O(n)$, since there are n/k iterations, each requiring time $O(k)$.

`RoundRobin`: Clearly, `IterEndpoints` does a perfect job of maximizing the gain for the first group. However, the gains for the subsequent groups can decrease considerably. `RoundRobin` tries to balance the gain across all the groups. It works off the sorted list of student abilities and creates the first group consisting of k students at positions 1, $k + 1$, $(2k + 1)$, etc. The ith group is formed by students at positions i, $k + i$, $(2k + i)$, etc. The running time of `RoundRobin` for a sorted input consisting of n students is $O(n)$. Intuitively, this heuristic mimics how groups are often formed (particularly in recreational team sports) by first selecting the leaders and then letting the leaders take turn in adding members to their respective groups.

Observe that `RoundRobin` can also be used for solving the COUNTℓG problem. In fact, we use it as a baseline algorithm in our experiments with the COUNTℓG problem as well.

6. EXPERIMENTS

Our experimental evaluation focuses on studying the performance of our algorithms for the COUNTℓG and the VALUEℓG problems. Moreover, we present some illustrative partitions created by different algorithms and highlight their characteristics.

Datasets: For all the experiments we report here, we experiment with a set of $n = 1024$ students with ability values randomly sampled from (a) normal, (b) uniform and (c) pareto distributions. We refer to the resulting datasets as
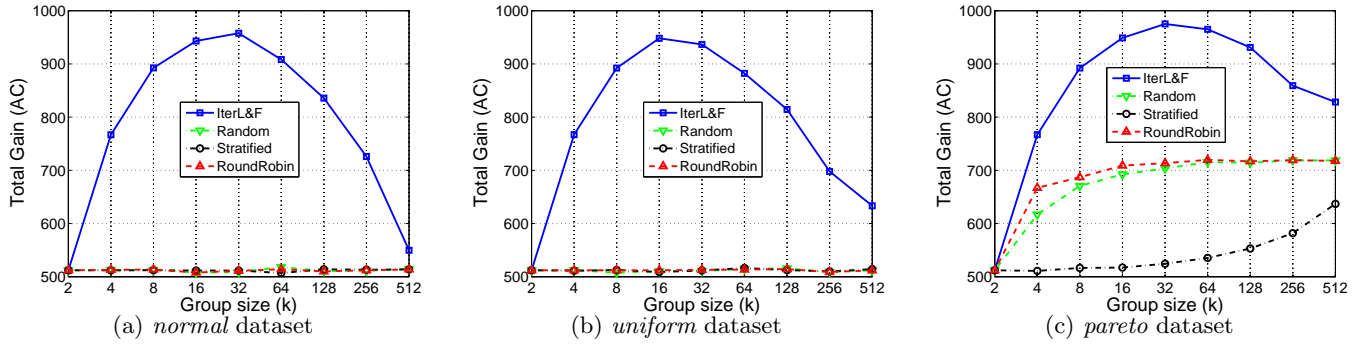
Figure 1: **Performance of** `IterL&F`, `Random`, `Stratified` **and** `RoundRobin` **for the** COUNTℓG **problem;** x**-axis (log-scale): number of students per group** (k)**;** y**-axis: sum of the Ac values of the groups.**
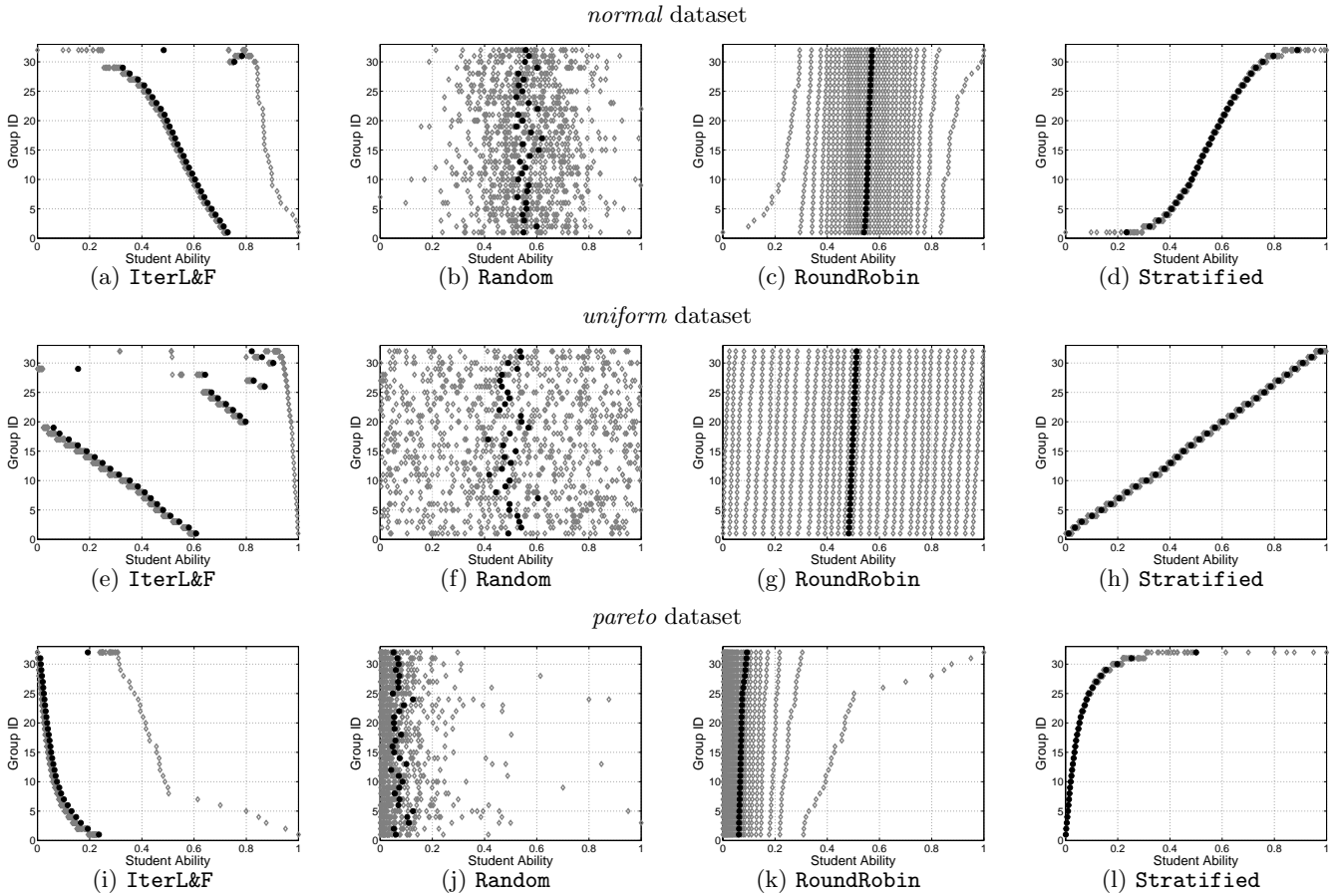

Figure 2: COUNTℓG: **Structure of the groups of size** $k = 32$ **formed by different algorithms for different datasets. Every row of a plot corresponds to a group and the points within a row represent the ability levels of the members of the group; dark points are group abilities.**
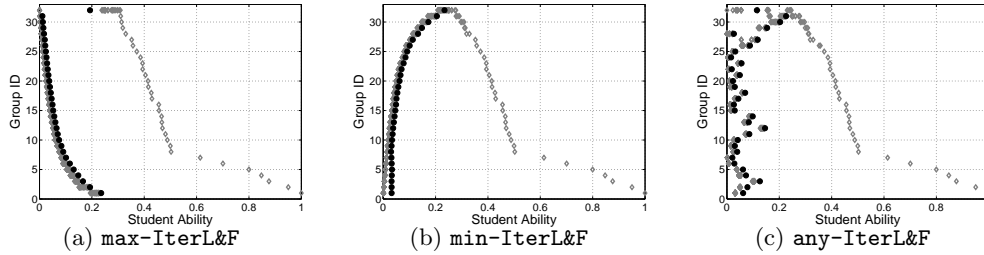
(a) max-IterL&F   (b) min-IterL&F   (c) any-IterL&F

Figure 3: CountℓG: **Structure of the groups of size $k = 32$ formed by different versions of the IterL&F algorithm for the** *pareto* **dataset.**

normal, *uniform* and the *pareto* datasets respectively. The abilities of students in the *normal* dataset are sampled from a normal distribution with mean 0 and standard deviation 1. For the *uniform* dataset the abilities are sampled uniformly from the interval $[0, 1]$. Finally, the abilities of the students in the *pareto* dataset are samples from a pareto distribution, having the shape parameter set to 3. After the datasets are sampled from their respective distributions, we normalize all values to be in $[0, 1]$. This normalization is only done for consistency and plays no role in the conclusions we draw.

Baseline algorithms: In addition to the algorithms we discussed in Sections 4 and 5, we also experiment with two baseline algorithms: Stratified and Random.

The Stratified algorithm sorts the students in decreasing order of their abilities. Then, the first group is created by considering the first k students with the highest abilities and putting them in a group by themselves. The second group is created with the subsequent k students, and so on. The running time of this algorithms for a sorted input consisting of n students is $O(n)$. This algorithm can be thought of as an idealized version of the oft-used, ability-based homogeneous grouping of students [20, 24].

The Random algorithm creates ℓ groups of size k by randomly assigning students to groups. The running time of the Random algorithm is $O(n)$, since it is adequate to create a random permutation of the students and then create the ℓ groups by considering consecutive members of this permutation. Note that some approximation of Random is often used for partitioning a class into heterogeneous sections [27].

6.1 Evaluating algorithms for CountℓG

We study the performance of different algorithms for the CountℓG problem by computing the total gain Ac due to the groups they form. We use the max-IterL&F version of IterL&F as well as Random, Stratified and RoundRobin in this evaluation. Figure 1 shows their Ac values as a function of the group size $k \in \{2, 4, 8, \dots, 512\}$ for each one of the three datasets (*normal*, *uniform* and *pareto*).

The results (which are averages over 20 random datasets drawn from the respective distribution) demonstrate that for all values of k (except for $k = 2$), IterL&F is significantly better than any other algorithm. In fact, there are values of k (e.g., $k = 32$) for which the IterL&F achieves total Ac of more than 950, while the maximum possible value is less than 1024. This means that more than 90% of the students are assigned into groups in which the group ability is higher than their ability.

For $k = 2$, all algorithms have the same total Ac, which is equal to $n/2 = 1024/2 = 512$. The reason is that in groups of size 2 inevitably there is one student who is above and one who is below the group average and therefore the group is beneficial for exactly half of the students, independently of how the group assignment is performed. As k increases, the ability of IterL&F to make conscious and judicious use of the high ability students to pull up as many lower ability students as possible starts to shine and the performance gap between IterL&F and other algorithms starts to increase. Beyond some value of k (around 32), the law of diminishing return sets in and the performance gap starts to reduce. Note, however, that the x-axis in Figure 1 is in logarithmic scale and the decrease in performance gap is more gradual than it might appear from a cursory glance at the plots.

Turning our attention to the baseline algorithms (i.e., Random, Stratified, and RoundRobin), we observe that for *normal* and *uniform* datasets the performance of these algorithms is almost constant allowing about half of the students to benefit from group participation. The landscape is different for the *pareto* dataset. In this case, Random and RoundRobin are better than Stratified – yet again significantly worse than IterL&F. In order to understand that one has to remember that the *pareto* dataset has a small number of exceptionally high-ability students. The Stratified algorithm by definition puts these students together in one group and therefore their high abilities cannot be leveraged to lift up the average abilities of other groups. This phenomenon is not observed in the groups formed by Random and RoundRobin since these algorithms distribute the high-ability individuals into different groups allowing more groups to benefit from them.

Characteristics of groups in CountℓG**:** Although the plots in Figure 1 demonstrate the superiority of IterL&F with respect to the objective function (i.e., total Ac), they do not provide intuition on the types of students that are brought together into groups by different algorithms. We do that in Figure 2 for groups of size 32 using the *normal*, the *uniform* and the *pareto* datasets. The x-axis of all the plots in this figure corresponds to ability values and the y-axis takes values $\{1, 2, \dots, 32\}$; these y-values correspond to the group IDs. Thus the points on the same horizontal line in a plot represent the abilities of the students that are assigned to the group with ID equal to the value of the y-axis to which this horizontal line corresponds to. The group ID is the same as the order in which the group was output by the corresponding algorithm. For example, the group with ID equal to 1 is the the first group that was built by IterL&F,

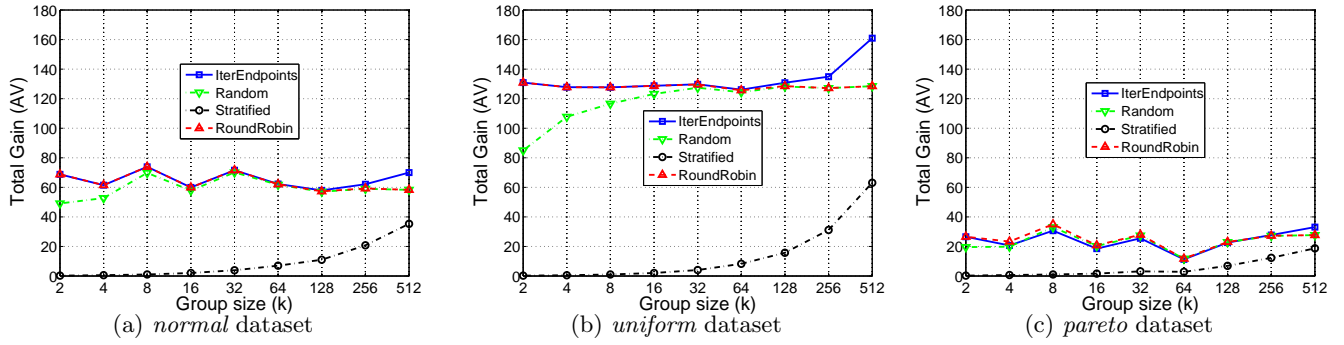Figure 4: **Performance of IterEndpoints, Random, Stratified and RoundRobin for the** VALUEℓG **problem;** x-**axis (log-scale): number of students per group** (k); y-**axis: sum of the Av values of the groups.**
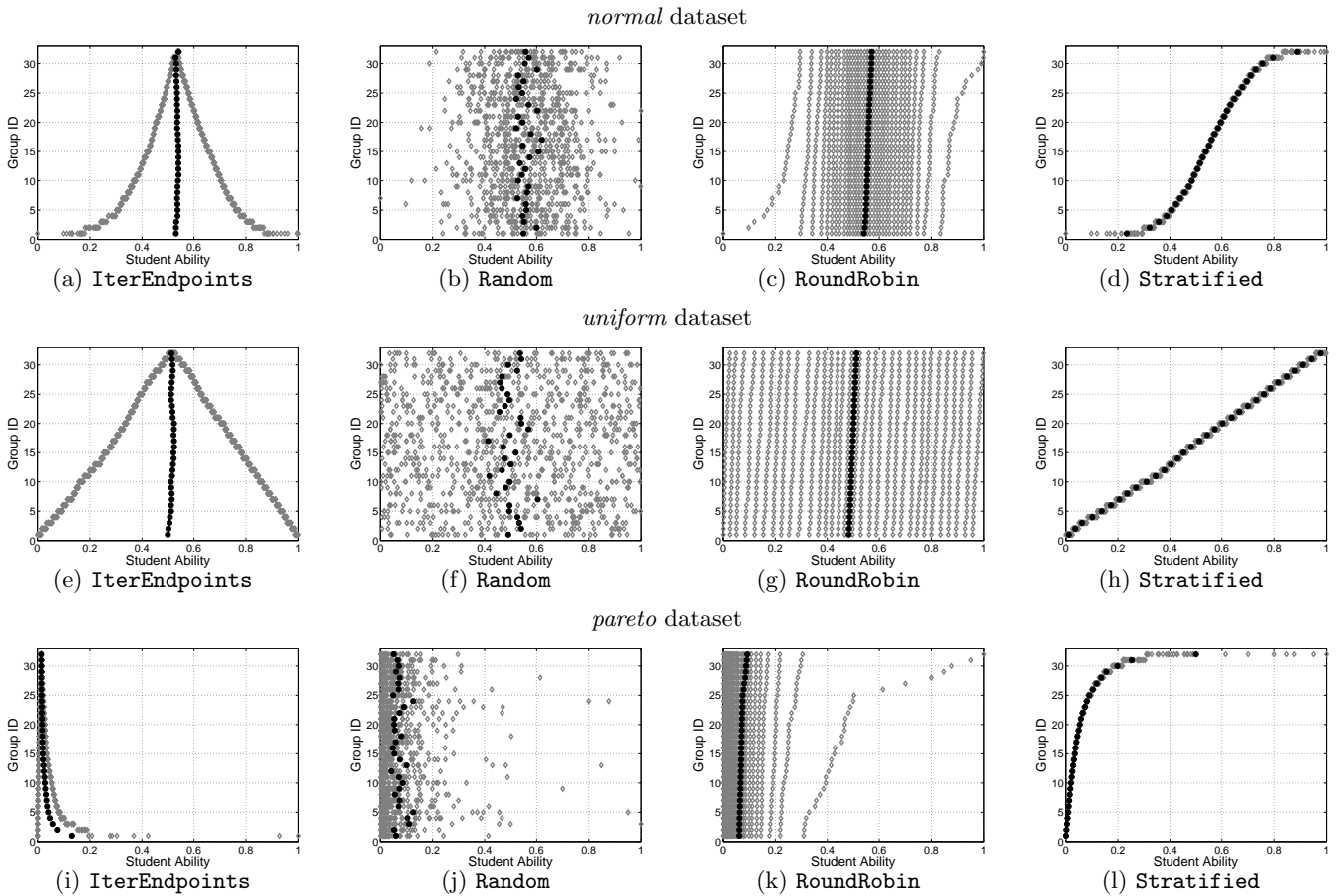

Figure 5: VALUEℓG: **Structure of the groups formed for** $k = 32$ **by different algorithms for different datasets.**

and group with ID equal to i was built by `IterL&F` at its i-th iteration. The gray triangles correspond to the members of each group while black dots represent the mean of the abilities within each group. Keep in mind that we are using the `max-IterL&F` version of `IterL&F` in this experiment.

Several interesting observations emerge from this figure. First of all, there is a clear separation between the "leaders" and the "followers" in the groups built by `IterL&F`. We see that `IterL&F` judiciously puts together in each group few strong leaders with followers that are clearly weaker than them, yet they are not as weak as the followers grouped with relatively less strong leaders. The groups formed by `Random` have a mix of students from all ability levels, with no clear separation between leaders and followers. The groups formed by `RoundRobin` are a more deterministic version of the groups built by `Random`. Finally, the groups formed by `Stratified` clearly exhibit ability grouping, where the students with the same ability levels are grouped together. These differences in group composition underlie the superiority of `IterL&F` we see in Figure 1.

Finally, we remark that we found that all three versions of `IterL&F`, namely `max-IterL&F`, `min-IterL&F` and `any-IterL&F`, had similar performance but the structure of the groups they formed differed somewhat. We illustrate in Figure 3 the composition of the groups built by them for $k = 32$ for the *pareto* dataset. As in `max-IterL&F`, there continues be a separation between the "leaders" and the "followers" in the groups constituted by the other two. However, while `max-IterL&F` groups strongest leaders with strongest followers, `min-IterL&F` initially groups the weakest followers with the strongest leaders. Subsequently, it groups decreasingly strong leaders with relatively stronger weak followers until in the final groups the abilities of the leaders and the followers converge. The groups built by `any-IterL&F` are similar to those yielded by `min-IterL&F`, but the followers now have relatively more mixed abilities.

6.2 Evaluating the algorithms for $\textsc{Value}\ell\textsc{G}$

Here, we describe our experiments for the $\textsc{Value}\ell\textsc{G}$ problem. Our experimental methodology is identical to the one we followed for the $\textsc{Count}\ell\textsc{G}$ problem. We study the performance of `IterEndpoints`, `Random`, `RoundRobin` and `Stratified` with respect to the total Av of the groups they form. We again use *normal*, *uniform* and *pareto* datasets and group sizes $k \in \{2, 4, 8, \ldots, 512\}$.

Figure 4 shows the results. We see that `IterEndpoints`, `Random` and `RoundRobin` consistently outperform `Stratified`. To understand this phenomenon, recall that the goal of $\textsc{Value}\ell\textsc{G}$ is to maximize the gain for the students in the follower sets. The gain for followers in a particular group depends on the gap between the their abilities and the group ability; the larger the gap the larger will be the value of Av. Thus, an algorithm that forms groups in such a way that the ability of leaders is relatively high compared to the abilities of followers will perform better. Finally, since $\textsc{Value}\ell\textsc{G}$ aims to maximize the *sum* of the Av values of all the groups, the above property needs to hold only in aggregate.

Examine now Figure 5 that illustrates the abilities of students in each group formed by the different algorithms for the *normal*, the *uniform* and the *pareto* datasets for $k = 32$. As before, the x-axis of all the plots in this figure corresponds to ability values and the y-axis takes values $\{1, 2, \ldots, n/k\}$, which correspond to the group IDs. Clearly, `Stratified`

forms homogeneous groups such that in every group there is very little difference in the abilities of the group members and the individual ability of each of the member is quite close to the collective ability of the group. Thus, the total gain for this algorithm is small.

In contrast, the groups formed by `Random` have greater diversity of abilities. It brings together students of varying abilities that has the effect of creating groups having leaders and followers at quite different levels of ability. `RoundRobin` can be viewed as a deterministic version of `Random` and hence its performance characteristics is similar. However, it is interesting to contrast the composition of groups these two heuristics form with those built by `IterEndpoints`. Though they all achieve roughly similar total Av values, particularly for groups of reasonable size, they form very different types of groups. In some of the groups formed by `IterEndpoints`, there is significant difference in the abilities of the leaders and the followers, while in others the difference is small. There is some empirical evidence in support of such structures [23].

7. CONCLUDING REMARKS

We proposed a formal framework for studying the socially important problem of grouping students in a large class into sections so that the gain aggregated over all the students is maximized. Using two different definitions of gain for students, we studied two distinct optimization problems: $\textsc{Count}\ell\textsc{G}$ and $\textsc{Value}\ell\textsc{G}$. Intuitively, $\textsc{Count}\ell\textsc{G}$ aims to maximize the number of students who can do better by interacting with the higher ability students, while $\textsc{Value}\ell\textsc{G}$ incorporates the extent of these improvements. From the computational viewpoint, we studied the computational complexity of these grouping problems and provided novel polynomial time heuristics for them.

Our experiments indicate that by appropriately choosing the definition of the gain function, it is possible to achieve different social goals. For example, the groups reported as solutions to $\textsc{Count}\ell\textsc{G}$, put strong students together with students who are weaker than them, yet they are not as weak as the students grouped with less strong students. Thus, the group members can learn in their zones of proximal development [19, 31]. On the other hand, groups reported as solutions to $\textsc{Value}\ell\textsc{G}$ put together the weakest with the strongest students aiming to maximize diversity of the groups. Such groups may provide larger motivation/benefit to the weak students while giving room to the strongest students to learn by teaching and become role models [5, 23, 32]. It is really up to the school to decide its social goals and philosophy and as a result its choice of objective.

The gain functions we studied in this paper are just two of many functions admitted by our framework. In the future, we would like to study other functions particularly those incorporating the gains of strong students and non-linear gains. We would also like to enrich our problem formulation with constraints due to socio-emotional factors such as interpersonal relations [4, 10]. The group dynamics that sets in once a group is formed can have major impact on the outcomes [6]. We would like to explore how to apply computational perspective to studying the group interactions in an educational setting. Finally, we would like to partner with some educational institutions to study the performance characteristics of our work in real-life.

8. ACKNOWLEDGMENTS

9. REFERENCES

[1] Anagnostopoulos, A., Becchetti, L., Castillo, C., Gionis, A., and Leonardi, S. Power in unity: forming teams in large-scale community systems. In *CIKM* (2010), 599–608.

[2] Anagnostopoulos, A., Becchetti, L., Castillo, C., Gionis, A., and Leonardi, S. Online team formation in social networks. In *WWW* (2012), 839–848.

[3] Ashman, A., and Gillies, R. *Cooperative learning: The social and intellectual outcomes of learning in groups.* Routledge, 2003.

[4] Azmitia, M., and Montgomery, R. Friendship, transactive dialogues, and the development of scientific reasoning. *Social Development 2*, 3 (1993), 202–221.

[5] Bargh, J., and Schul, Y. On the cognitive benefits of teaching. *Journal of Educational Psychology 72*, 5 (1980), 593.

[6] Barron, B. When smart groups fail. *Journal of the Learning Sciences 12*, 3 (2003), 307–359.

[7] Boaler, J., Wiliam, D., and Brown, M. Students' experiences of ability grouping-disaffection, polarisation and the construction of failure. *British Educational Research Journal 26*, 5 (2000), 631–648.

[8] Chimombo, J. P. G. Issues in basic education in developing countries: An exploration of policy options for improved delivery. *Journal of International Cooperation in Education 8*, 1 (2005), 129–152.

[9] Crocker, L., and Algina, J. *Introduction to classical and modern test theory.* Wadsworth, 2006.

[10] Da Silva, F. E. D. O., Motta, C. L., Santoro, F. M., and De Oliveira, C. E. T. A social matching approach to support team configuration. In *Groupware: Design, Implementation, and Use.* Springer, 2009, 49–64.

[11] Esposito, D. Homogeneous and heterogeneous ability grouping: Principal findings and implications for evaluating and designing more effective educational environments. *Review of Educational Research 43*, 2 (1973), 163–179.

[12] Gajewar, A., and Sarma, A. D. Multi-skill collaborative teams based on densest subgraphs. In *SDM* (2012), 165–176.

[13] Garey, M., and Johnson, D. S. *Computers and intractability: A guide to the theory of NP-completeness.* W.H. Freeman, 1979.

[14] Gillies, J., and Quijada, J. Opportunity to learn: A high impact strategy for improving educational outcomes in developing countries. *USAID Educational Quality Improvement Program (EQUIP2)* (2008).

[15] Grossen, B. *How should we group to achive excellence with equity.* University of Oregon, July 1996.

[16] Hambleton, R. K., and Jones, R. W. Comparison of classical test theory and item response theory and their applications to test development. *Educational Measurement: Issues and Practice 12*, 3 (1993), 38–47.

[17] Hand, D., Mannila, H., and Smyth, P. *Principles of Data Mining.* MIT Press, 2001.

[18] Hanushek, E. A., and Woessmann, L. The role of education quality for economic growth. *Policy Research Department Working Paper 4122, World Bank* (2007).

[19] Hertz-Lazarowitz, R., and Miller, N. *Interaction in cooperative groups: The theoretical anatomy of group learning.* Cambridge University Press, 1995.

[20] Kulik, J. A. An analysis of the research on ability grouping: Historical and contemporary perspectives. Tech. Rep. 9204, NRC/GT, University of Connecticut, February 1992.

[21] Lappas, T., Liu, K., and Terzi, E. Finding a team of experts in social networks. In *KDD* (2009), 467–476.

[22] Li, C.-T., and Shan, M.-K. Team formation for generalized tasks in expertise social networks. In *SocialCom/PASSAT* (2010), 9–16.

[23] Lou, Y., Abrami, P. C., Spence, J. C., Poulsen, C., Chambers, B., and d'Apollonia, S. Within-class grouping: A meta-analysis. *Review of Educational Research 66*, 4 (1996), 423–458.

[24] Loveless, T. The resurgence of ability grouping and persistence of tracking. Tech. rep., Part II of the 2013 Brown Center Report on American Education, Brookings Institution, March 2013.

[25] Majumder, A., Datta, S., and Naidu, K. V. M. Capacitated team formation problem on social networks. In *KDD* (2012), 1005–1013.

[26] May, M. A., and Doob, L. W. Competition and cooperation. *Social Science Research Council Bulletin* (1937).

[27] McPartland, J. M., Coldiron, J. R., and Braddock, J. H. School structures and classroom practices in elementary, middle, and secondary schools. Tech. Rep. 14, Johns Hopkins University, February 1987.

[28] Mislevy, R. J. Item response models for grouped data. *Journal of Educational Statistics 8*, 4 (1983), 271–288.

[29] Richer, S. Reference-group theory and ability grouping: A convergence of sociological theory and educational research. *Sociology of Education 49*, 1 (1976), 65–71.

[30] Slavin, R. E. *Learning to cooperate.* Springer, 1985.

[31] Vygotsky, L. S. *Mind in society: Development of higher psychological processes.* Harvard University Press, 1978.

[32] Webb, N. M. Peer interaction and learning in small groups. *International Journal of Educational Research 13*, 1 (1989), 21–39.

[33] World-Bank. *Knowledge for Development: World Development Report: 1998/99.* Oxford University Press, 1999.