Grouping Students for Maximizing Learning from Peers

Rakesh Agrawal
Data Insights Laboratories
ragrawal@acm.org

Sharad Nandanwar
Indian Institute of Science
sharadnandanwar@csa.iisc.ernet.in

M. N. Murty
Indian Institute of Science
mnm@csa.iisc.ernet.in

ABSTRACT

We study the problem of partitioning a class of N students into k groups of n students each ($N = k \times n$), such that their learning from peer interactions is maximized. In our formalization of the problem, any student is able to increase his score in the subject the class is studying up to the score of the student who is at p-percentile among his higher ability peers. In contrast, the past work presumed that only students with score below the group mean may increase their score. We give a partitioning algorithm that maximizes total gain summed over all the students for any value of p such that $100/(100-p)$ is integer valued. The time complexity of the proposed algorithm is only $\mathcal{O}(N \log N)$. We also present experimental results using real-life data that show the superiority of the proposed algorithm over current strategies.

1. INTRODUCTION

A basic problem that has challenged educators for a long time is how to group students in a class in order to supplement their learning from the teacher with the learning from peers [6, 11]. Two popular strategies currently in vogue are: i) heterogeneous (also called diversity-based) grouping, and ii) homogeneous (also referred to as stratified or ability-based) grouping [5]. Both have their ardent proponents. The results from the empirical studies on the relative effectiveness of the two are inconclusive and the public opinion has also been mixed [3, 9].

In a major departure from the conventional thinking, a computational perspective was taken to address this problem in [1]. However, the learning model underlying the proposed algorithmic approach postulated that only the below average students are able to increase their ability score [4]. This paper removes this limitation, recognizing that every student can benefit from peer interactions [6, 8].

1.1 Contributions

- We admit a general learning model that specifies that any student is able to increase his ability score up to the level of the student who is at p-percentile amongst his higher ability peers. The value of p is an input parameter, selected by the educator. The model in [8] can be viewed as a special case, with p set to 100.
- For the above learning model, we provide an algorithm for partitioning N students into k groups of n students each ($N = k \times n$) with the goal of maximizing learning gain summed over all the students. We show that the algorithm is optimal for the values taken by p such that $100/(100-p)$ is integer-valued. Thus, it is optimal for $p \in \{99, 98, 95, 90, 80, 75, 66\frac{2}{3}, 50\}$. The time complexity of the algorithm is $\mathcal{O}(N \log N)$.
- We present experimental results using real datasets, showing the superiority of our approach over current strategies.

1.2 Limitations

- Although our learning model has been abstracted from the findings in the education literature, a rigorous empirical validation of the model is future work. The insights gained are nonetheless instructive.
- Teaching others and giving help has been shown to be positively correlated to increase in learning [2]. Incorporating such learning gains for high ability students is future work.

2. RELATED WORK

The question of how to group students to maximize their gain from peer interactions was first addressed from a computational perspective in [1]. The authors proposed two functions to model learning gains. The first maximizes the number of students who improve their ability score [4], while the second incorporates the extent of these improvements. In both the cases, however, only the below average students benefit and the higher ability students have zero gain. The authors showed that the partitioning problem with the goal of maximizing the number of benefiting students is NP-complete, while they left open the question of the complexity class of the problem with the second gain function.

The viewpoint that every student can learn from the higher ability peers is also present in [8]. In their model, every student may increase his ability to a fixed level, which is the ability of the highest ability student, i.e. $p = 100$. This assumption is too rigid and optimistic. In contrast, we admit various levels of gain for different students.

Our problem bears resemblance with the expert-team formation problem, in which the experts are multi-dimensional vectors of skills and the goal is to find a team that can collectively perform a given task requiring certain skills [10]. However, our students are described by 1-dimensional scores, and our objective is not to locate a single team, but to partition the students such that their learning gain is maximized.

Our problem also superficially resembles the classical clustering problem [7]. However, unlike the classical clustering, which aims to maximize the similarity of all the points in a cluster to a cluster center, our problem has no one point in a partition with respect to which the distance of all other points needs to be optimized (see Fig. 1).

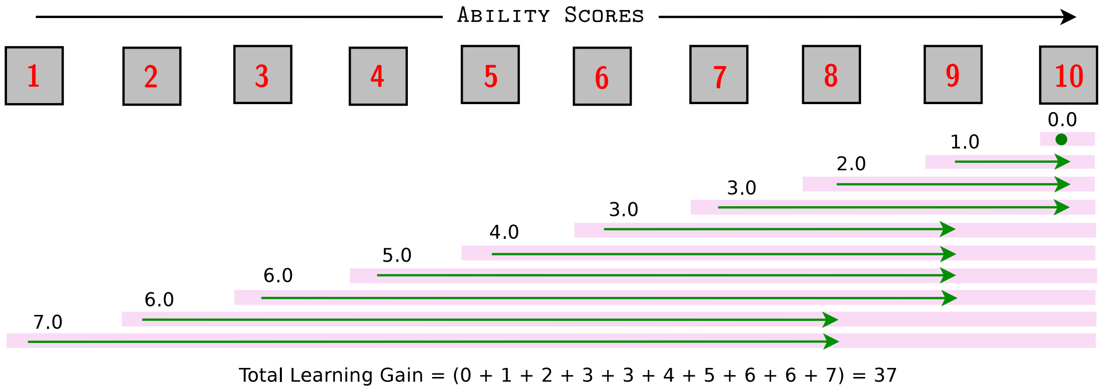

Figure 1: Computation of the potential learning gain for a group of ten students with 75-percentile chosen as the reference point. The i^{th} box contains the score of the i^{th} student. The learning gain for each student is the difference between his score and the score of student at p-percentile amongst his peers having higher score than him. For the first student, the index of the student at 75-percentile amongst his higher ability peers is $(1 + \lceil(10 - 1) * 75/100\rceil) = 8$. Since the score of the latter is eight, the gain for the first student is $(8 - 1) = 7$. For the second student, the index of the student at 75-percentile amongst his higher ability peers is also 8 $(2 + \lceil(10 - 2) * 75/100\rceil)$, thus giving him a gain of $(8 - 2) = 6$, and so on. The gain for the last student is zero, as there is no one above to learn from.

3. PROBLEM STATEMENT

We have a class of N students. Each student i is associated with score $\theta_i \in \mathbb{R}_{\geq 0}$, representing student's ability in the subject the class is studying [4]. For simplicity, scores are assumed to be distinct, so there is a one to one correspondence between the student i and the score θ_i. Students are ordered in the increasing order of scores.

Students are able to increase their score through interactions with peers in the group in accordance with a gain function [12, 13]. The gain from peer learning for a group G is given by a function $\mathcal{L}$. Our objective is to find k groups of n students each ($N = k \times n$), such that the overall gain for students is maximized. That is, our objective is

$$\max_{\mathcal{G}} \sum_{G \in \mathcal{G}} \mathcal{L}(G). \tag{1}$$

The learning function is of the form

$$\mathcal{L}(G) = \sum_{i=1}^{|G|} \left(R_i^G - \theta_i \right), \tag{2}$$

where R_i^G is the reference score for the G's i^{th} ranked student. The intuition is that each student can increase his score up to the reference score.

3.1 Learning up to p-Percentile

PROBLEM 1 (P-PERCENTILE PARTITIONING PROBLEM). *The gain function in Eq. 2 is given by*

$$\mathcal{L}^p(G) = \sum_{i=1}^{|G|} \left(p_i^G - \theta_i^G \right), \tag{3}$$

where p_i is the score of the student whose score is at the p-percentile position of the scores of the students having higher score than the i^{th} student in G.

For a given set of scores, the p-percentile score is the score below which p% of scores fall. To find the p-percentile score, the corresponding index is calculated first, which is $\lceil np/100 \rceil$. The value at this index then is the p-percentile score. Thus,

$$p\text{-percentile}(\theta_1, \theta_2, \ldots, \theta_n) = \theta_{\lceil n.p/100 \rceil}. \tag{4}$$

Fig. 1 graphically illustrates the percentile gain function.

4. SOLUTION

THEOREM 1. *For values of p such that $p/(100 - p)$ is integer-valued, the p-Percentile Partitioning problem can be solved optimally in $\mathcal{O}(N \log N)$ time.*

We shall prove the theorem constructively by providing an optimal algorithm whose time complexity is $\mathcal{O}(N \log N)$. It is named Percentile_Partitions and its pseudo-code is shown in Algorithm 1. The algorithm exploits the special structure of our problem that we elicit next.

We first expand the equation for learning gain w.r.t. p-percentile as given in Eq. 3 into

$$\begin{aligned} \mathcal{L}^P(G) = & \left(\text{p-percentile}(\theta_2^G, \theta_3^G, \ldots, \theta_n^G) - \theta_1^G \right) + \\ & \left(\text{p-percentile}(\theta_3^G, \theta_4^G, \ldots, \theta_n^G) - \theta_2^G \right) + \\ & \ldots + \left(\text{p-percentile}(\theta_n^G) - \theta_{n-1}^G \right). \end{aligned}$$

Using the definition of p-percentile from Eq. 4, the above can be written as

$$\begin{aligned} \mathcal{L}^P(G) = & (\theta_{1+\lceil (n-1)p/100 \rceil}^G - \theta_1^G) + (\theta_{2+\lceil (n-2)p/100 \rceil}^G - \theta_2^G) + \\ & \ldots + (\theta_n^G - \theta_{n-1}^G). \end{aligned}$$

To this we add the term $(\theta_n^G - \theta_n^G)$ corresponding to zero gain of the n^{th} student. Thus, we have

$$\begin{aligned} \mathcal{L}^P(G) = & (\theta_{1+\lceil (n-1)p/100 \rceil}^G - \theta_1^G) + (\theta_{2+\lceil (n-2)p/100 \rceil}^G - \theta_2^G) + \\ & \ldots + (\theta_{(n-1)+\lceil p/100 \rceil}^G - \theta_{n-1}^G) + (\theta_n^G - \theta_n^G). \end{aligned}$$

Collecting the positive and negative terms together, we get

$$\begin{aligned} \mathcal{L}^P(G) = & \left(\theta_{1+\lceil (n-1)p/100 \rceil}^G + \theta_{2+\lceil (n-2)p/100 \rceil}^G + \ldots \right. \\ & \left. + \theta_{(n-1)+\lceil p/100 \rceil}^G + \theta_n \right) \\ & - \left(\theta_1^G + \theta_1^G + \ldots + \theta_{n-1}^G + \theta_n^G \right), \end{aligned}$$

which can be written succinctly as

$$\mathcal{L}^P(G) \;=\; \sum_{i=1}^{n} \theta^G_{i+\lceil (n-i)p/100 \rceil} - \sum_{i=1}^{n} \theta^G_i. \tag{5}$$

Using this equation, our objective becomes

$$\max_{\mathcal{G}} \sum_{G\in\mathcal{G}} \left(\sum_{i=1}^{n} \theta^G_{i+\lceil (n-i)p/100 \rceil} - \sum_{i=1}^{n} \theta^G_i \right).$$

The second component in the above sum is constant for any given set of ability scores. Therefore, our objective can be simplified to

$$\max_{\mathcal{G}} \sum_{G\in\mathcal{G}} \sum_{i=1}^{n} \theta^G_{i+\lceil (n-i)p/100 \rceil}. \tag{6}$$

Lemma 1. *Given $p \in [0, 100]$ and an ascending sequence of $\theta_i \in \mathbb{R}_{\geq 0}$, for $(100-p)|100$, $\sum_{i=1}^{n} \theta_{i+\lceil (n-i)p/100 \rceil}$ is equivalent to $\sum_{i=1}^{n} \gamma_i \cdot \theta_i$, where*

$$\gamma_i = \begin{cases} \frac{100}{100-p}, & \text{if } \lceil \frac{np}{100} \rceil < i \leq n \\ mod(n, \frac{100}{100-p}), & \text{if } \frac{100}{100-p} \nmid n \text{ and } i = \lceil \frac{np}{100} \rceil \\ 0, & \text{otherwise.} \end{cases} .$$

Proof. It is to be noted that a student at index i improves up to the score of student at index $i + \lceil (n-i)p/100 \rceil$. As the student indexes are traversed from the higher-score end to the lower end, with unit decrease in value of i, the quantity $\lceil (n-i)p/100 \rceil$ increments by unity, except for the values of i for which $(n-i)p$ is a multiple of hundred. In the latter case, although there is a decrement in the value of i by one, the value of $\lceil (n-i)p/100 \rceil$ stays the same as that of $\lceil (n-i-1)p/100 \rceil$, causing the index up to which students are improving to decrement by one. It is easy to derive that this process repeats itself after a period of $100/(100-p)$. Further, when n is not a multiple of the above period, there will be $\mathrm{mod}(n, 100/(100-p))$ students who will be improving up to the smallest index value. For the remaining students, as no other student improves up to their score, a γ value of zero is straightforward. □

Example 1. *In Fig. 1, we have $n = 10$ and $p = 75$. Thus, in accordance with Lemma 1, we have*

$$\gamma_i = \begin{cases} 4, & \text{if } 8 < i \leq 10 \\ 2, & \text{if } i = 8 \\ 0, & \text{otherwise.} \end{cases}$$

.

The above may also be verified visually from Fig. 1. It is easy to note that the students at 7^{th}, 8^{th}, and 9^{th} index improve up to the score of the 10^{th} student, while the 10^{th} student with zero gain remains at the same score. This makes the score of the 10^{th} student visible four times in the updated scores, leading to the γ value of four. Similarly, the score of the student at 9^{th} index is also visible four times because of students at 3^{rd}, 4^{th}, 5^{th}, and 6^{th} indexes improving up to his score. On the other hand, only students at 1^{st} and 2^{nd} indexes improve up to the score of 8^{th} student. Hence, a γ value of two for the 8^{th} student. No one is improving his score up to the score of any of the students at index below eight. So, the γ values corresponding to them are zero.

Unfortunately, when $(100-p) \nmid 100$, the coefficients γ_i's have complex structure and we defer their study to future work.

Algorithm 1 (Percentile_Partitions) Optimal Partitioning for maximizing Learning Gain - learning up to p-percentile

Input: Distinct descending scores $\{\theta_1, \theta_2, \ldots, \theta_N\}$, Percentile p, Number of groups k, Size of each partition n, $k \times n = N$.

$G_1 = G_2 = \ldots = G_k = \phi$
$m \leftarrow 100/(100-p)$
$q \leftarrow \lfloor n/m \rfloor$
$\hat{q} \leftarrow \lceil n/m \rceil$
if $\mathrm{mod}(n, m) \neq 0$
    $M \leftarrow \{\theta_{kq+1}, \ldots, \theta_{kq+k)}\}$
    for $i \in \{1, 2, \ldots, k\}$
        $G_i \leftarrow G_i \bigcup M_i$
    end for
end if
$H1_{global} \leftarrow \{\theta_1, \theta_2, \ldots, \theta_{kq}\}$
$H2_{global} \leftarrow \{\theta_{k\hat{q}+1}, \ldots, \theta_{N-1}, \theta_N\}$
for $i \in \{1, 2, \ldots, k\}$
    $H1_{part} \leftarrow$ randomly sample q scores from $H1_{global}$ without replacement.
    $H2_{part} \leftarrow$ randomly sample $(n - \hat{q})$ scores from $H2_{global}$ without replacement.
    $G_i \leftarrow G_i \bigcup H1_{part} \bigcup H2_{part}$
end for
return $\{G_1, G_2, \ldots, G_k\}$

4.1 Percentile_Partitions

Lemma 1 leads to our optimal partitioning algorithm, which is shown in Algorithm 1. The algorithm first divides the input ability scores into two or three sets depending on whether $\mathrm{mod}(n, 100/(100-p))$ is zero or not respectively. The first set $H1_{global}$ consists of scores that contribute by a factor of $100/(100-p)$ to the learning gain. The second set M if present, consists of scores that contribute by a factor of $\mathrm{mod}(n, 100/(100-p))$. Finally, the third set $H2_{global}$ consists of scores that have zero contribution. These sets correspond to the three different values of the γ coefficients. They are such that $H1_{global} \succcurlyeq M \succcurlyeq H2_{global}$, where $A \succcurlyeq B$ means all elements of set A are greater or equal compared to any element of set B. For each of these sets then, the algorithm creates k equal random partitions. These partitions are then merged to create the final k partitions. The example below illustrates the algorithm.

Example 2. *Consider a set of 20 students with ability scores $\{\theta_1, \theta_2, \ldots, \theta_{20}\}$, sorted in the descending order. The set is to be partitioned into four groups, each containing five students. Each student can learn up to the score of the student who is at $66\frac{2}{3}$-percentile of students above.*

For $p = 66\frac{2}{3}$ and $n = 5$, we have $m = 3$, $q = 1$, and $\hat{q} = 2$. The algorithm breaks the scores into three sets:

$$\begin{aligned} H1_{global} &= \{\theta_1, \theta_2, \theta_3, \theta_4\} \\ M &= \{\theta_5, \theta_6, \theta_7, \theta_8\} \\ H2_{global} &= \{\theta_9, \theta_{10}, \theta_{11}, \theta_{12}, \theta_{13}, \theta_{14}, \theta_{15}, \theta_{16}, \theta_{17}, \theta_{18}, \theta_{19}, \theta_{20}\} \end{aligned}$$

For each set, four equal-sized random partitions are created, which are then merged to create four groups:

$$\begin{aligned} G_1 &= \{\theta_3\} \bigcup \{\theta_6\} \bigcup \{\theta_{17}, \theta_{10}, \theta_{15}\} \\ G_2 &= \{\theta_1\} \bigcup \{\theta_7\} \bigcup \{\theta_{19}, \theta_{16}, \theta_9\} \\ G_3 &= \{\theta_2\} \bigcup \{\theta_5\} \bigcup \{\theta_{13}, \theta_{18}, \theta_{12}\} \\ G_4 &= \{\theta_4\} \bigcup \{\theta_8\} \bigcup \{\theta_{14}, \theta_{20}, \theta_{11}\} \end{aligned}$$

Note: *There are many equally good ways of partitioning $H1_{global}$, M, and $H2_{global}$. The above is just one of them.*

4.2 Proof of Theorem 1

Clearly, if the input scores were already in the descending order, the time complexity of the Algorithm 1 is $\mathcal{O}(N)$. If the input scores were unsorted, then the extra sorting step would make the complexity $\mathcal{O}(N \log N)$.

The optimality of the algorithm follows from the structure in the values taken by the coefficient γ's. Before proceeding further, we state the following lemma:

LEMMA 2. *For given ordered sets of real numbers, $A = \{a_1, a_2, \ldots, a_n\}$ and $B = \{b_1, b_2, \ldots, b_n\}$, the quantity $\sum_{a \in A, b \in B} ab$, s.t. each $a \in A$ and $b \in B$ is used exactly once, is maximized if the elements are chosen in a manner such that the product of elements at the same index from A and B is taken.*

Now, according to Lemma 1, γ_i can take only one of the three values and they have ordering amongst them given by $100/(100-p) > mod(n, 100/(100-p) \geq 0$. The partitions created by the algorithm satisfy, $H1_{global} \succcurlyeq M \succcurlyeq H2_{global}$. Thus, in light of Lemma 2, it is easy to observe that our objective is maximized as the set of students with higher(lower) scores get mapped to highest(lowest) coefficient. Moreover, the random perturbations within $H1_{global}$, M, or $H2_{global}$ do not affect the gain value as all the scores from a set are involved in product with the same γ value.

5. EXPERIMENTS

5.1 Datasets

1. SSC Scores (Normal distribution): Staff Selection Commission - Combined Graduate Level Examination (SSC-CGL) is conducted all across India to recruit employees for various departments of Government of India. The scores of candidates for the 2016 examination, categorized into different regions of the country, are available at ssc.nic.in. The distribution of scores in every region is close to normal. We took the scores from the North Western (SSC-NWR) region that exhibits the largest variance.

2. GATE Scores (Log-Normal distribution): In India, Graduate Aptitude Test in Engineering (GATE) is conducted every year to test the competency of undergraduate students in various engineering disciplines. We took the available scores from year 2016. We experimented with scores from Mech. (GATE-ME), with largest variance.

3. StkXchg UpVotes (Pareto distribution): On the Stack Exchange platform, users can ask and answer questions on various topics. Additionally, they can up-vote or down-vote a question. The number of up-votes a user receives is an indicative measure of his level of expertise. Pareto distribution fitted the data for the active users having at least one up-vote. The Stack Exchange data dump is available from archive.org/details/stackexchange. We take data for Stack Overflow that ehibits lowest skew in distribution.

5.2 Algorithms

In addition to Percentile_Partitions, we consider two algorithms that correspond to the strategies currently prevalent in practice: Stratified and Random.

1. Stratified: This algorithm puts in each group those students who exhibit similar ability. This grouping represents the practice of homogeneous or ability-based grouping.

2. Random: Students are assigned to groups randomly. This method corresponds to the practice of heterogeneous or diversity-based grouping.

5.3 Set Up

We conducted our experiments setting the number of students, N, to 1024. We varied the number of groups, k, over {2,4,8,..., 512}, and the reference percentile point p over {50, $66\frac{2}{3}$, 75, 80, 90, 95, 98, 99}. Thus, for each dataset, we randomly sample 1024 scores and generate the groups for different combinations of k and p values. In order to have tight confidence intervals, we repeat this exercise 30 times each and report average learning gain.

For the groups generated by Percentile_Partitions, we compute learning gain using Eq. 3. When applying Stratified or Random to a dataset, we generate groups only once but compute gain using the appropriate parameter value for p.

We also study the group structures generated by different algorithms. By the structure of a group, we mean the distribution of scores in the group. Although we run each algorithm 30 times, we only show the structure of the group generated by the first run.

5.4 Results

Fig. 2 shows the learning gain as the reference percentile value, p, is varied for different algorithms on various datasets. We show the plots for three values for the number of groups, $k \in \{128, 32, 8\}$ (and the corresponding group sizes, $n \in \{8, 32, 128\}$). Fig. 3 shows the learning gain as the number of groups, k, is varied. We show the plots for two percentile values, $p \in \{75, 90\}$. Fig. 4 shows the group structures generated by different algorithms. We show the structures for groups of size, $n = 8$, and for the reference percentile, $p = 75$. We alert the reader that different scales have been used for Y-axis in Figs. 2-3 and a logarithmic scale has been employed for X-axis in Fig. 3 for the sake of clarity.

We see that the overall behavior of different algorithms remains similar across different group sizes and reference percentile values. Clearly, Percentile_Partitions consistently outperforms the other algorithms that corroborates its theoretical optimality. The following additional observations are noteworthy:

- With increasing value of p, total learning gain increases super linearly (Fig. 2). It is because the extent of learning gain for each student increases. The gain plateaus for small groups because beyond some percentile value, all students improve up to the same highest ability student. Then, it does not matter whether the reference percentile is at 90 or 95.
- The advantage of Percentile_Partitions over Random is more pronounced when the number of students in a group is in a more realistic range of 32 or less (Fig. 3). When the number of groups is small and each group is large, Percentile_Partitions assigns very many students randomly

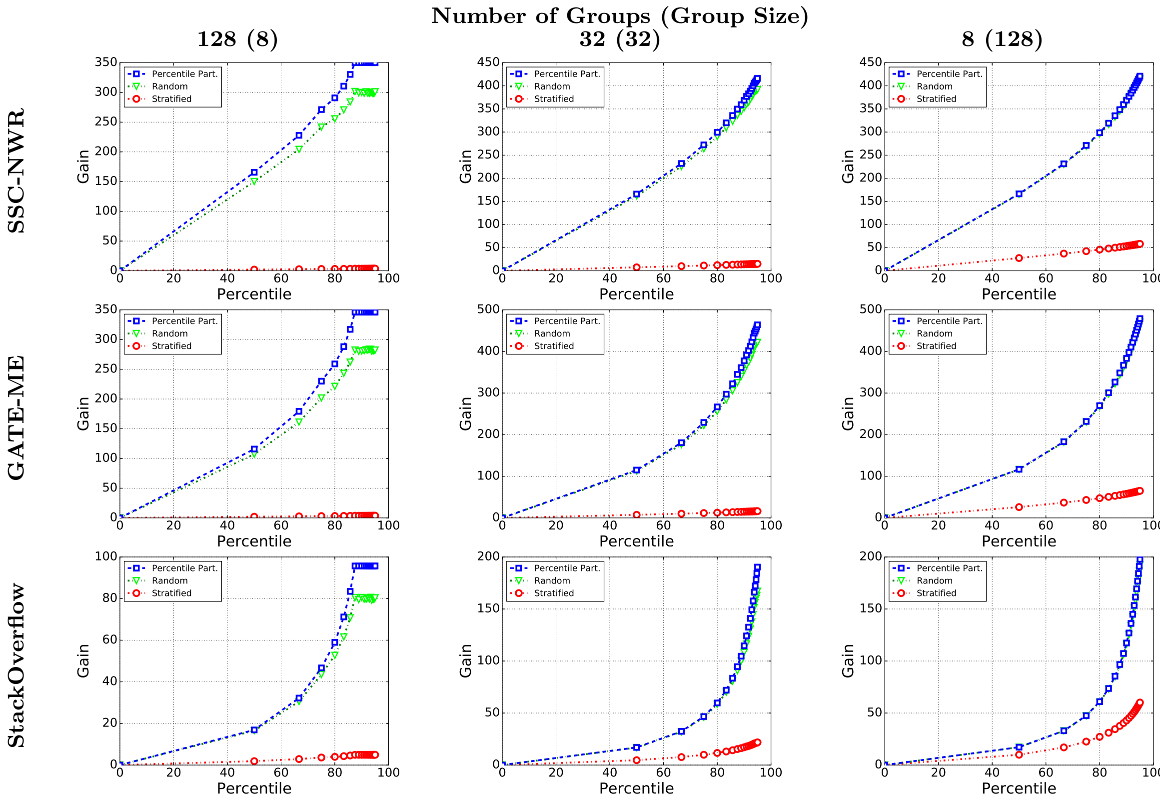

Figure 2: Learning Gain as the reference percentile point, p, is varied.

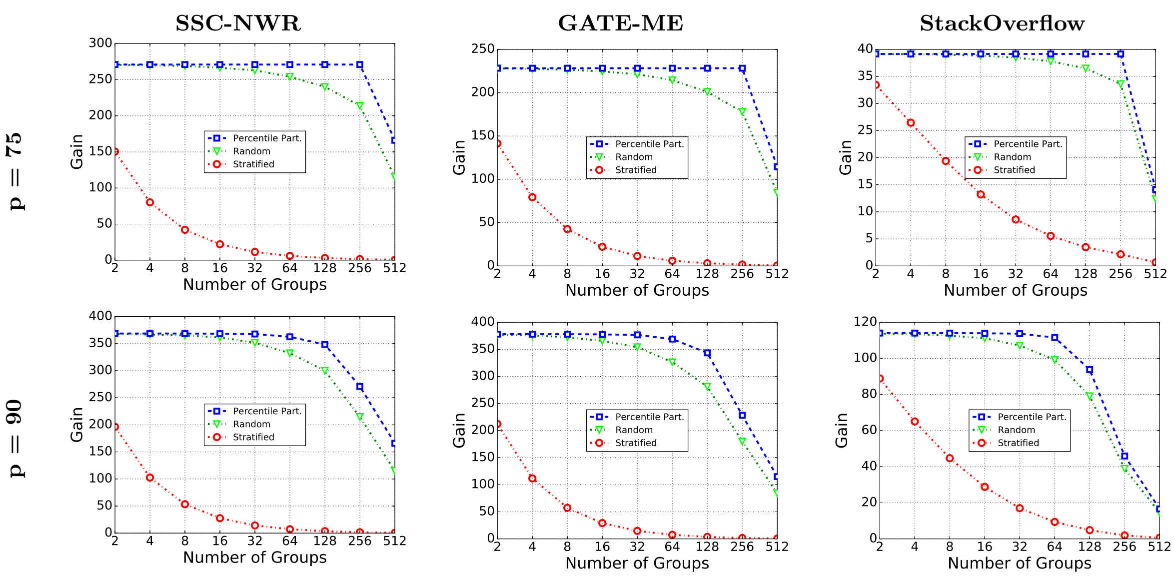

Figure 3: Learning Gain as the number of groups, k, is varied.

and therefore the group structure and gain produced by it become similar to that of Random.

- The learning gain is worst with the stratified strategy. Fig. 4 shows that this strategy produces groups in which the students have similar scores. Therefore, the improvements from peer interactions are small. Fig. 4 also shows that the p-percentile value of every group produced by Percentile_Partitions is higher than the global p-percentile value of all the undivided scores. However, this pattern is not true for Random. Some groups generated by Random have p-percentile to the extreme right of global p-percentile. The scores in between the two p-percentiles in such groups do not contribute to the total gain. But then some other groups end up having smaller scores above p-percentile that leads to smaller additions to the total gain. Hence, the superior performance of Percentile_Partitions.

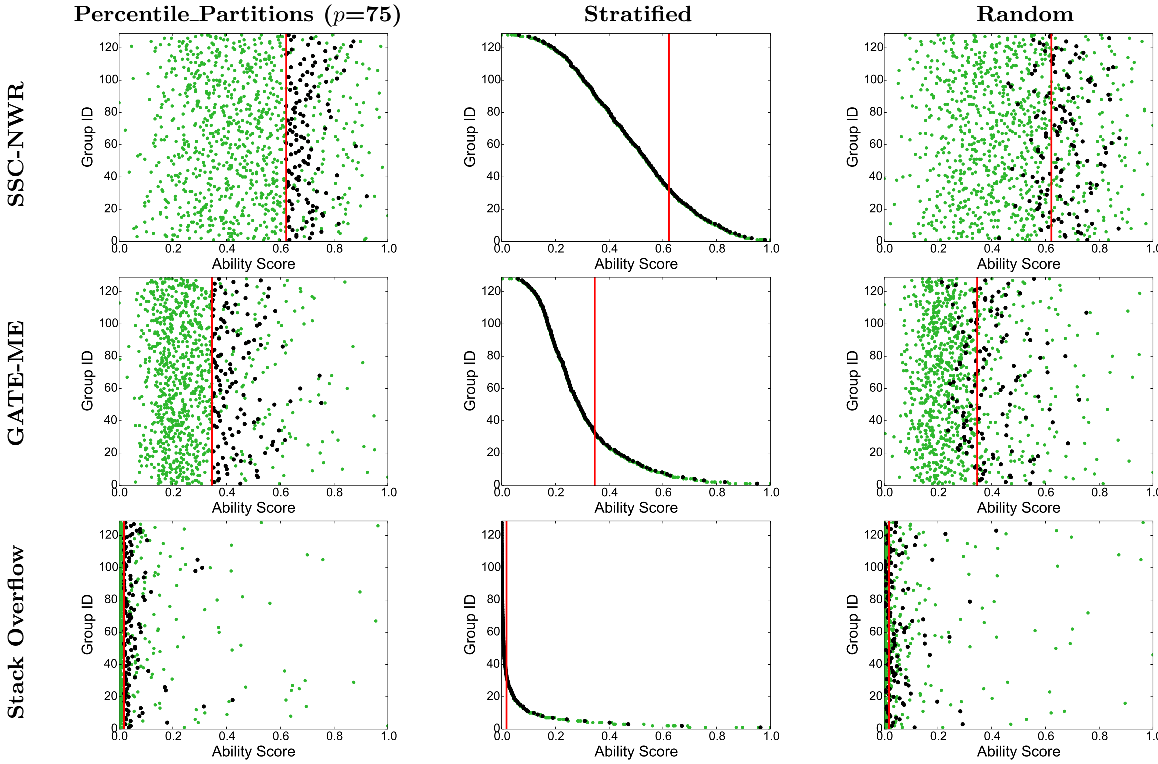

Figure 4: Group structure generated by different algorithms for groups of size 8. Each row in the plots corresponds to a particular group and there is a dot for each ability score in that group. The p-percentile score for each group is plotted in black. The vertical red line shows the global p-percentile score. The groups are numbered according to the order in which they are generated. Only for Percentile_Partitions, the p-percentile score for every group is higher than the global p-percentile value.

6. SUMMARY

We investigated the important educational data mining problem of how to group students in a class to maximize their learning gains from peer interactions. We worked with a general learning gain function in which every student is able to increase his ability score up to the score of the student who is at p-percentile amongst his higher ability peers. We gave an algorithm which is provably optimal for maximizing learning gain, the value of p is such that $100/(100-p)$ is integer valued. We also studied the performance characteristics of the proposed algorithm using real-life datasets that corroborated the theoretical analysis and showed its superiority over the current approaches. Surprisingly, the time complexity of optimally grouping N students using our algorithm is only $\mathcal{O}(N \log N)$.

Acknowledgment Rakesh Agrawal's visit to the Indian Institute of Science, where bulk of the present work was done, was funded through the Rukmini Gopalakrishnachar Visiting Chair Professorship. His work was also partially funded by EPFL - School of Computer and Communication Sciences, Data Intensive Applications and Systems Laboratory.

7. REFERENCES

[1] Agrawal, R., Golshan, B., and Terzi, E. Grouping students in educational settings. In *20th ACM SIGKDD international conference on Knowledge discovery and data mining* (2014), pp. 1017–1026.

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

[3] 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.

[4] Crocker, L., and Algina, J. *Introduction to classical and modern test theory.* ERIC, 1986.

[5] 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.

[6] Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., and Wenderoth, M. P. Active learning increases student performance in science, engineering, and mathematics. *Proceedings National Academy of Sciences 111*, 23 (2014), 8410–8415.

[7] Jain, A. K. Data clustering: 50 years beyond k-means. *Pattern recognition letters 31*, 8 (2010), 651–666.

[8] Kosheleva, O. Diversity is the optimal education strategy: A mathematical proof. *International Journal of Innovative Management, Information & Production 4*, 1 (2013), 1–8.

[9] Kulik, J. A. An analysis of the research on ability grouping: Historical and contemporary perspectives. Tech. Rep. NRC-G/T-9204, National Research Center on the Gifted and Talented, Storrs, CT, February 1992.

[10] Lappas, T., Liu, K., and Terzi, E. Finding a team of experts in social networks. In *15th ACM SIGKDD international conference on Knowledge discovery and data mining* (2009), pp. 467–476.

[11] Slavin, R. Cooperative learning. *Learning and Cognition in Education* (2011), 160–166.

[12] Vygotsky, L. S. *Mind in society: The development of higher psychological processes.* Harvard Press, 1980.

[13] Webb, N. M. Peer interaction and learning in small groups. *International journal of Educational research 13*, 1 (1989).