Targeting the Right Students Using Data Mining

Yiming Ma, Bing Liu, Ching Kian Wong
School of Computing
National University of Singapore
3 Science Drive 2, Singapore 117543
{liub, maym, wongck}@comp.nus.edu.sg

Philip S. Yu
IBM T. J. Watson Research Center
Yorktown Heights,
NY 10598, USA
psyu@watson.ibm.com

Shuik Ming Lee
Gifted Education Branch
Ministry of Education
51 Grange Road, Singapore 249564
lee_shuik_ming@moe.gov.sg

ABSTRACT
The education domain offers a fertile ground for many interesting and challenging data mining applications. These applications can help both educators and students, and improve the quality of education. In this paper, we present a real-life application for the Gifted Education Programme (GEP) of the Ministry of Education (MOE) in Singapore. The application involves many data mining tasks. This paper focuses only on one task, namely, selecting students for remedial classes. Traditionally, a cut-off mark for each subject is used to select the weak students. That is, those students whose scores in a subject fall below the cut-off mark for the subject are advised to take further classes in the subject. In this paper, we show that this traditional method requires too many students to take part in the remedial classes. This not only increases the teaching load of the teachers, but also gives unnecessary burdens to students, which is particularly undesirable in our case because the GEP students are generally taking more subjects than non-GEP students, and the GEP students are encouraged to have more time to explore advanced topics. With the help of data mining, we are able to select the targeted students much more precisely.

Keywords
Target selection, scoring, data mining application in education.

1. INTRODUCTION
The education domain offers many interesting and challenging applications for data mining. First, an educational institution often has many diverse and varied sources of information. There are the traditional databases (e.g. students' information, teachers' information, class and schedule information, alumni information), online information (online web pages and course content pages) and more recently, multimedia databases. Second, there are many diverse interest groups in the educational domain that give rise to many interesting mining requirements. For example, the administrators may wish to find out information such as admission requirements and to predict the class enrollment size for timetabling. The students may wish to know how best to select courses based on prediction of how well they will perform in the courses selected. The alumni office may need to know how best to perform target mailing so as to achieve the best effort in reaching out to those alumni that are likely to respond. All these applications not only contribute towards the education institute delivering a better quality education experience, but also aid the institution in running its administrative tasks. With so much information and so many diverse needs, it is foreseeable that an integrated data mining system that is able to cater for the special needs of an education institution will be in great demand particularly in the 21st century.

In this paper, we present a real-life application for the Gifted Education Programme (GEP) of the Ministry of Education (MOE), Singapore. Singapore is a small city-state, where human resources are the only source of natural resources. Education is viewed as a critical factor in contributing to the long-term economic well-being of the country. The government of Singapore believes in the importance of maximizing the potential of each and every individual student. As such, the MOE treats the daunting task with great importance as the responsibilities of educating the country's future leaders fall heavily on their shoulders. GEP is a programme initiated by MOE, and is aimed at discovering talented individuals as early as possible so as to further nurture them with a set of specially designed courses.

In Singapore, the mainstream students generally start their education with 6 years of primary school education. At the end of the 6 years, students sit for their first nationwide streaming exams, Primary School Leaving Exams (PSLE) before being promoted to the next level. Following the primary school comes 4 years of secondary school education before the students sit for the next nationwide streaming exams, Ordinary Level Exams (O-level). From here each student has a choice on how to purse his/her further education. The student could go straight into tertiary education by joining a polytechnic or could further study for another 2 years in a junior college (JC) and sit the Advanced Level Exams (A-level), which will eventually land him/her a place in a University. The GEP students are selected when they are in their primary schools. The selection is done with a series of tests. All students interested in joining the GEP program are invited to take the tests. Exceptional students identified through the tests are offered places in the GEP programme.

KDD 2000, Boston, MA USA

In this project, we were given the GEP students' demographic data and school performance data over the past few years. We were requested to do a data mining application, which has a number of separate tasks. The main objective of this application is to validate both the selection process and the effectiveness of the program. Some of the tasks that the MOE is interested in include profiling of the GEP students based on different grouping criteria and multi-angled comparison of students who were admitted into the GEP and those who did not make it and also a performance analysis of students in the GEP with special interest in how further improvements can be achieved.

In this paper, we focus only on one specific task in the application, selecting students for remedial classes. The GEP is are interested in improving the A-level exam results of their students by selecting the weak students to attend remedial classes. There are two main considerations. On one hand, it is undesirable to select too many students to take part in the remedial classes as this increases the load on both the students and the teachers. Asking these students to take more classes is particularly undesirable because the GEP students are already taking more lessons than those students not in the GEP program, and the GEP students are encouraged to have more time to explore advanced topics in science and technology. On the other hand, if we miss the right students for remedial classes, they may end up doing poorly in their A-level exams (the expectations from the MOE on these students are very high). Our task is to evaluate the traditional selection method and to suggest improvements based on findings from data mining.

In this paper, we will describe a data mining based method for selecting the right students for remedial classes. The key component of this method is a new scoring function (called SBA [21]) that is based on association rules [1]. This method has yielded some exceptionally promising results. It outperforms the traditional method significantly.

The paper is organized as follows: In the next section, we introduce the traditional selection method to select the weak students for remedial classes. In Section 3, we present our data mining technique used for the task, which is a new scoring method based on association rules. In Section 4, we apply this technique to our task. We will see that simply applying this technique is far from sufficient. Our data mining results must be combined with some problem specific methods and knowledge in order to produce the required results. Section 5 evaluates our new method and compares with various other techniques. Section 6 concludes the paper.

2. EXISTING METHOD FOR SELECTING STUDENTS FOR REMEDIAL CLASSES

Gifted (also called GEP) students in Singapore are identified in their 3rd year of primary school education. The gifted education programme focuses on both the primary and secondary school levels. In the Ministry of Education, a Gifted Education branch keeps track of all the GEP students over the years. All the exam records related to GEP students are kept in a database.

From past experiences and previous studies, the Gifted Education branch knows that GEP students generally perform well in their O-level exams. Only 14% of the students have not done as well as expected. However, the percentage of students not meeting the expectations in their A-level exams rises to 31% (the student's performance at A-level is measured by an aggregated score computed from individual subject scores using a complex formula). Over the years, these percentages are reasonably consistent. The immediate tension is to reduce the discrepancies between these two statistics. MOE hopes to provide JCs with information on how to help the identified weak students so that remedial classes can be provided to these students.

Out of all subjects, 8 O-level subjects have been identified that are related to the A-level exams that they would like to provide remedial classes. Out of these 8 subjects, 3 of them are language subjects, and the rest are science and mathematics subjects. The O-level examination result of each subject has been studied individually, and a subject cutoff mark for attending the remedial class of the subject is imposed. If a GEP student does not do better than a subject cutoff mark in his/her O-level exam, he/she will be recommended to attend the remedial class of the subject.

In our application, we show that this cutoff method selects too many students for remedial classes. In the rest of the paper, we also refer this method as *traditional method*, as it is commonly used in most educational institutions.

3. SELECTING THE WEAK STUDENTS USING A NEW SCORING TECHNIQUE

After careful study and consultation with our domain experts, we designed the following two steps to solve the problem:

1. Identify the potential weak students.
2. Select the courses that each weak student is recommended to take.

The first step is clearly a data-mining task, while the second step is performed with the help of our domain experts [1]. Domain experts' advice is needed because they do not want our method to depart too far from the normal practice. Otherwise, both the students and teachers will complain that they do not understand what is going on in the computer.

Step 1 can be seen as a classification problem. However, it is not suitable to use a classifier to predict who will definitely perform poorly in his/her A-level exams because the accuracy is too low. For example, using the O-level results, the classification system C4.5 [29] can only identify half of the weak students, and the CBA system [19] can only identify two-third of the weak students. From our experiences and experiments, we found that a scoring method is more appropriate. Instead of assigning each student a definite class ('weak' or 'non-weak' student), a scoring model assigns a probability estimate to each student to express the

[1] It is also possible that we use each A-level subject as the class attribute and learn rules to find out those courses in O-level that affect the performance of this A-level course. We then recommend those students who did not do well in these O-level courses to go for remedial classes. However, this method does not work well because it often generates too many rules. It is very hard to select the right rules to identify the O-level courses.

likelihood that he/she will do poorly in the A-level exams (note that a student's performance at A-level is measured by an aggregated score). Then, the subsequent processing has the flexibility to choose a certain subset of the students for remedial classes. This will be clear later.

Below, we present our new scoring technique. We call it the SBA technique (Scoring Based on Associations). Although there already exist scoring methods based on decision trees and also the Bayesian model, it is shown in [21] (our IBM Research Report) that in general our new method SBA outperforms these traditional techniques substantially. Later in the evaluation section, we will also see that SBA performs better than C4.5 and Naïve Bayesian based scoring methods in our application. We present the SBA method in this section.

Since the SBA method is based on association rules, we begin the presentation by introducing the concept of association rule mining. We then discuss the problems and solutions in using the association rules for scoring. Finally, we give the scoring function adopted by SBA.

3.1. Association Rule Mining

Association rules are an important class of regularities that exist in databases. Since it was first introduced in [1], the problem of mining association rules has received a great deal of attention [e.g., 1, 2, 5, 11, 13, 25, 30, 33, 34]. The classic application of association rules is the market basket analysis [1, 2]. It analyzes how the items purchased by customers are associated. An example association rule is as follows,

$$cheese \rightarrow beer \text{ [sup = 10\%, conf = 80\%]}$$

This rule says that 10% of customers buy *cheese* and *beer* together, and those who buy *cheese* also buy *beer* 80% of time.

The association rule mining model can be stated as follows: Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of items. Let D be a set of transactions (the database), where each transaction d is a set of items such that $d \subseteq I$. An association rule is an implication of the form, $X \rightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \emptyset$. The rule $X \rightarrow Y$ holds in the transaction set D with confidence c if c% of transactions in D that support X also support Y. The rule has support s in D if s% of the transactions in D contains $X \cup Y$.

Given a set of transactions D (the database), the problem of mining association rules is to discover all association rules that have support and confidence greater than or equal to the user-specified minimum support (called minsup) and minimum confidence (called minconf).

3.2. Issues Faced

To use association rules for scoring purposes, we need to solve a number of problems, which are discussed below. Solutions to these problems are also proposed. See also [21].

Mining association rules from a relational table: In our applications, we need to mine association rules from a relational table (rather than a set of transactions) as our task uses this form of data. For association rule mining to work on this type of data, we need to discretize each numeric attribute into intervals since association rule mining only takes categorical values or items. After discretization, we can treat each data case (record) in the dataset as a set of (*attribute*, *value*) pairs and a class label. An (*attribute*, *value*) pair is an *item*. With this transformation, each data case becomes a transaction. An existing association rule-mining algorithm can be applied to the dataset. Discretization of continuous attributes will not be discussed in this paper (see [10]).

In traditional association rule mining, any item can appear on the left-hand-side or the right-hand-side of a rule. For scoring, we have a fixed class attribute with two classes. Thus, we are only interested in rules that use a single class on their right-hand-sides. That is, we only mine association rules of the form:

$$X \rightarrow C_i$$

where C_i is a class of the class attribute, and X is a set of items from the rest of the attributes. The class that we are interested in is often called the *positive class* (e.g., the 'weak' student class). The other class is called the *negative class* (e.g., 'non-weak' student class). We say a rule is *large* if it meets the minimum support.

Using an existing association rule-mining algorithm (e.g., [2]) to mine this type of rules is straightforward. We simply find all large *itemsets* (or rules) of the form:

$$<item_1, \ldots, item_k, C_i>, \text{ where } C_i \text{ is fixed beforehand.}$$

In the mining process, each iteration adds a new item to every itemset. That is, in the first iteration we find all itemsets of the form $< item_1, C_i>$. In the second iteration, we find all itemsets of the form $< item_1, item_2, C_i>$, and so on.

Problems with minsup and minconf: Traditional association rule mining uses a single minsup and a single minconf in the mining process. This is not appropriate for our task because the class distribution of our data can be quite imbalanced. Let us discuss the problems with minsup first. Using a single minsup causes the following problems:

- If the minsup is set too high, we may not find those rules that involve the minority class, which is often the positive class (the class that we are interested in).
- In order to find rules that involve the minority class, we have to set the minsup very low. This may cause combinatorial explosion because the majority class may have too many rules and most of them are over fitted with many conditions and covering very few data cases. These rules have little predictive value. They also cause increased execution time.

While a single minsup is inadequate for our application, a single minconf also causes problems. For example, in a database, it is known that only 5% of the students are weak students and 95% are non-weak students. If we set the minconf at 96%, we may not be able to find any rule of the 'weak' class because it is unlikely that the database contains reliable rules of the 'weak' class with such a high confidence. If we set a lower confidence, say 50%, we will find many rules that have the confidence between 50-95% for the 'non-weak' class and such rules are meaningless (see also [3]).

We solve these problems by using different minsups and minconfs for rules of different classes. For minimum supports, we only

require the user to specify one total or overall minimum support (called *t_minsup*), which is then distributed to each class according to the class distribution in the data as follows:

$$minsup(C_i) = t\_minsup \times \frac{f(C_i)}{|D|}$$

where $f(C_i)$ is the number of C_i class cases in the training data. $|D|$ is the total number of cases in the training data. The reason for using this formula is to give rules with the frequent (negative) class a higher minsup and rules with the infrequent (positive) class a lower minsup. This ensures that we will generate enough rules with the positive class and will not produce too many meaningless rules for the negative class.

For minimum confidence, we use the following formula to automatically assign minimum confidence to each class:

$$minconf(C_i) = \frac{f(C_i)}{|D|}$$

The reason for using this formula is that we should not produce rules of class C_i whose confidence is less than $f(C_i)/|D|$ because such rules make no sense.

Although we have different minsups and minconfs for rules of different classes, no change needs to be made to the original association rule mining algorithm [2] as the *downward closure property* [2] still holds for mining $<item_1, \ldots, item_k, C_i>$. The reason is that item sets of different classes do not interact.

Pruning of association rules (optional): It is well known that many association rules are redundant and minor variations of others. Those insignificant rules should be pruned. Pruning can remove a huge number of rules with no loss of accuracy (see [21] for more details). It also improves the efficiency of scoring.

Our pruning function uses the pessimistic error rate based method in C4.5 [28]. Note that the error rate of a rule is 1 – 'the confidence of the rule'. The technique prunes a rule as follows: If rule *r*'s estimated error rate is higher than the estimated error rate of rule r^- (obtained by deleting one condition from the conditions of *r*), then rule *r* is pruned. Note that when *r* is a 1-condition rule of the form, $x \rightarrow y$, then r^- is, $\rightarrow y$, which has no condition. See [28] for the detailed computation of the pruning method. Pruning can be done very efficiently because if *r* is a rule then r^- must also be a rule.

An important point to note is that when attempting to prune a rule *r*, the r^- rule used (for a *k*-condition rule *r*, there are *k* r^-'s) may have been pruned previously. Then, the procedure needs to go back to the rule that prunes r^-, and uses that rule to prune *r*.

Using association rules for scoring: The key feature of association rule mining is its completeness, i.e., it aims to find all rules in data. This presents a great opportunity to design good scoring functions by making use of the rules, i.e., we have abundant of information. However, it also represents a challenge because when we want to score a data case, there are often many rules that can be applied. Different rules may give different information, and many of them even give conflicting information. For example, one rule may say that the data case should belong to the positive class with a probability of 0.9, while another rule may say that it should belong to the negative class also with a probability of 0.9. The question is which rule we should trust. This is not a problem for traditional classification systems because they typically have only one answer. In the case of association rules, we have many answers. In the next sub-section, we focus on this issue and present a score function that makes use of all rules.

3.3. Scoring Using Association Rules

After the rules are generated (from training data), we can use them to score the new (or test) data. Since each rule is attached with a support and a confidence, it is thus easy to design a method to score the data. To design the best scoring method based on association rules, however, is very difficult, because there are an infinite number of possible methods. Below, we describe a heuristic technique that is both effective and efficient (see [21] for detailed evaluation results).

In SBA, we aims to achieve the following effects:

1. When there are many confident positive class rules that can cover the data case, the data case should be assigned a high score.
2. When the positive class rules that cover the data case are not confident, but the negative class rules are very confident, the data case should be given a low score.

Basically, we try to push data cases toward both ends. If this is done reliably, we will achieve good ranking results. We now present the proposed scoring function.

In the context of association rules, we have the following types of useful information:

- Confidence: Rule confidence is an essential piece of information because confidence is basically a probability estimate. Thus, confidence in the positive class (we are only interested in the positive class) should play an important role in the score function.
- Support: Rule support, to certain extent, reflects the reliability of the rule. A rule that covers too few data cases is often over fitted and unreliable.
- Two types of rules: There are two types of rules that we may use, positive class rules and negative class rules. For a data case, there may be both types of rules that can cover it. Thus, we need to resolve this situation.

We postulate the following general scoring function, which is a weighted average taking into account of the above information (the support information will appear in the weights). Given a data case, *S* is the score of the data case. The value of *S* is between 0 and 1 inclusively.

$$S = \frac{\sum_{i \in POS} W^{i}_{positive} \times conf^{i} + \sum_{j \in NEG} W^{j}_{negative} \times conf^{j}_{positive}}{\sum_{i \in POS} W^{i}_{positive} + \sum_{j \in NEG} W^{j}_{negative}}$$

where:

- *POS* is the set of positive class rules that can cover the data case.
- *NEG* is the set of negative class rules that can cover the data case.
- $conf^{i}$ is the original confidence of the positive class rule.
- $W^{i}_{positive}$ is the weight for the positive class rule *i*.
- $W^{j}_{negative}$ is the weight for the negative class rule *j*.
- $conf^{j}_{positive}$ is the confidence after converting the negative class rule *j* to a positive class rule, i.e., $conf^{j}_{positive}$ = 1- 'the confidence of rule *j*'.

Now the problem is what should be the weights. Since we want to achieve the two desirable effects discussed above, the weights should reflect their needs. We have two pieces of information about each rule to use: support and confidence. We performed a large number of experiments, and found that for both the negative weight and the positive weight, the combination of both confidence and support performs the best. That is:

$$W^{i}_{positive} = conf^{i} \times sup^{i}$$

where $conf^{i}$ and sup^{i} are the original confidence and support of the positive class rule *i*, and

$$W^{j}_{negative} = \frac{conf^{j} \times sup^{j}}{k}$$

where $conf^{j}$ and sup^{i} are the original confidence and support of the negative class rule *j*, and *k* is a constant to reduce the impact of negative class rules (which often have high supports and high confidences). We performed many experiments to determine *k* and found that when $k = 3$, the system performs the best (see [21]).

It is important to note that to compute the weight for a negative class rule, we do not convert the rule to a positive rule and then use the support and confidence in the positive class. Instead, we still use their original support and confidence in the negative class. This helps us to achieve the two effects discussed above. See [21] for the justification and explanation of the formulas. [21] also gives the comparison results of SBA and other scoring methods.

Finally, in ranking, when more than one data case has the same score, we compute a priority value (*P*) using the following formula:

$$P = \frac{\sum_{i \in POS} sup^{i} - \sum_{j \in NEG} sup^{j}}{|POS| + |NEG|}$$

where |*POS*| and |*NEG*| are the numbers of rules in *POS* and *NEG* respectively. This formula uses supports of the rules to calculate the priority. Basically, we give those data cases with higher positive supports and lower negative supports higher priorities. Note that when a data case does not satisfy any rule (i.e., $POS = NEG = \emptyset$), we assign $S = 0$ and $P = 0$.

4. APPLYING SBA TO SELECTING WEAK STUDENTS

Our scoring method assigns each student a likelihood value to express the chance that he/she will be a weak student (i.e. not meeting a certain pre-defined standard) in the A-level exams. Hence, if a student receives a high score from the scoring system, it means that this student is likely to do badly and could thus be a target for remedial classes. After scoring and ranking, we can identify a list of weak students. We then select the courses that each potentially weak student has to attend. Our new technique for selecting the right students to attend remedial classes consists of the following steps:

1. Train the scoring model (e.g., SBA) using the training data
2. Score each student using the model constructed
3. Allocate groupings to each student based on the scores
4. Recommend remedial classes to the students in each group

The first two tasks above are relatively straightforward. We can simply execute the SBA system on the training data (step 1), which will then assign scores to the unseen test data (step 2). In the following 2 subsections (section 4.1 and 4.2), we will describe the techniques employed in the last two steps.

4.1. Allocating Students to Groups

Our technique for selecting courses for students is based on allocating students to different groups using their scores. This helps us discriminate the students based on how likely their group is going to do badly. The question is "how many groups should we allocate and how to decide the boundary of each group?" Based on our experiments using the training data and the suggestions from the domain expert, we decided to use 3 distinct groups, needy, normal, and fine. Our domain experts do not like to have too many groups because it makes things very complex and difficult to understand. In this domain (or any education domain), the users are particularly concerned with the complexity and the fairness, as they have to explain every action to the students and the teachers.

Next, we map the group boundaries. This is done by inspecting the results of our score models using 5-fold cross validation on the training data. We noticed that SBA can identify weak students very well in the first 20% of top ranking students, and in the next 30% there are a small number of weak students, and in the last 50%, almost all the students are good students. Thus, our three groups have 20%, 30% and 50% of the students respectively.

4.2. Recommending Remedial Classes to Students in Each Group

With the students allocated to the groups, we can now easily implement discriminating policies to these groups. The aim of such policies is to heighten the chance of a weak student attending remedial classes while lessening the chance of capable students attending remedial classes. Students allocated to the needy group are those that carry the highest chance of doing badly in their A-level exams. These students should have the highest chance of attending remedial classes. Students allocated to the normal group should have less chance while students allocated to the fine group should have the least chance. We then use the cutoff mark for

each subject used in the traditional method as the base to decide the cutoff mark for each course subject in each group in our method. For the needy group, we use a lower cutoff mark for each course subject, i.e., we make it more likely for those students in this group to attend remedial classes. For the normal group, we use the same cutoff mark as the traditional method. For the fine group, we use a higher cutoff mark for each subject, i.e., we make it harder for these students to take remedial classes. The exact cutoff mark for each course subject of each group is obtained from experiments and verified by the domain experts. This verification was important because our domain experts do not want our method to depart too far from their proposed method so that they can easily explain to the students and the management.

5. EVALUATION

This section compares the results of our data mining based method with the traditional method. In order to perform this comparison, we need to define some criteria, which are meaningful to our domain experts. Our criteria are similar to the precision and recall measures in information retrieval. Below, we define two precision measures, and one recall measure. Basically, they measure whether the teaching effort will be targeted at the right students.

5.1. Performance Measures

One main goal in this project is to target the right students with the right remedial classes. To measure how well our method performs, we define a concept called *unit effort, e*.

Definition: *a unit effort (e)* is a single class attended by a particular student. Alternatively, it is a single student taught by a teacher in a particular class.

Basically, the unit effort can be seen as a unit workload of a student or a teacher. For example, if a student attends a class, we say 1 *unit effort* is put on this student.

Our users are genuinely interested in putting more effort towards teaching potentially weak students than to the strong students. A good model should thus allow MOE to select the right students for the right remedial classes. We use the following performance measure, *precision_by_effort*, to measure this.

Definition: *precision_by_effort* (P_e) is the ratio of the total unit efforts (E_w) spent on weak students to the total unit efforts (E_t) spent in teaching all the students in the remedial classes, i.e.,

$$P_e = \frac{E_w}{E_t}$$

where E_w and E_t are computed as follows (s is the number of students attending remedial classes, k_i is the number of remedial classes taken by student i, j is a remedial class taken by a particular student, and n is the number of actual weak students).

$$E_w = \sum_{i=1}^{n}\sum_{j=1}^{k_i} e \qquad E_t = \sum_{i=1}^{s}\sum_{j=1}^{k_i} e$$

We can see that a higher P_e means that more teaching efforts are spent on the weak students. However, P_e alone is not sufficient. Our users would also like to teach as many potentially weak students as possible without burdening those good students. In other words, a good model should allow MOE to identify more weak students amongst all the students identified for remedial classes. We define this performance measure as *precision_by_student.*

Definition: *precision_by_student* (P_s) is the ratio of the total number of weak students (T_w) attending remedial classes to the total number of students (T_t) attending remedial classes, i.e.,

$$P_s = \frac{T_w}{T_t}$$

Clearly, a higher P_s means that we are teaching more weak students in our remedial classes.

The above two measures are precision measures. We also need a recall measure since our users are also interested in sending as many potentially weak students as possible for remedial classes. A good technique should simply allow MOE to identify as many weak students as possible. We define this performance measure as *recall* (by students).

Definition: *recall (R)* is the ratio of the total number of weak students (T_w) attending remedial classes to the total number of weak students (T_{wa}) in the data, i.e.,

$$R = \frac{T_w}{T_{wa}}$$

Note that it is not meaningful to define a recall measure by effort because it is not clear what is the maximum effort required by the weak students. Our users agreed that the three measures above are sufficient.

5.2. Results

We now present the comparison results. Note that we also used another two scoring techniques in our experiments to show that SBA gives superior results. One scoring method is based on the decision tree system C4.5 [29] (which we call C4.5-score), and the other is based on the Naïve Bayesian technique (which we call NB-score). MOE gave us 4 years of data for building the model. These 4 years of data are also used in determining the cutoff marks in the traditional method. They hold the latest year's student results in O-level and A-level as the test data. We perform many experiments on the 4 years data to determine the number of groups, group boundaries and also the exact cutoff marks of each course subject for each group (see also Section 4.1 and 4.2). Note that for different scoring systems, these experiments are carried out separately. Thus, their group boundaries and cutoff marks of course subjects for each group may be different. The objective is to maximize their performances in terms of the measurements defined in Section 5.1. Table 1 shows the results of various systems. The holdout set has one year of GEP students (153 of them), and out of these 153, 45 of them did not do well in their A-level exams.

Table 1. Selection results based on different selection criteria

Selection Criteria	Precision by Effort	Precision by Student	Recall
Traditional Method	142/269 = *52.79%*	42/112 = *37.50%*	42/45 = *93.33%*
SBA	162/264 = *61.36%*	41/87 = *47.13%*	41/45 = *91.11%*
NB-score	137/261 = *52.5%*	40/92 = *43.5%*	40/45 = *88.9%*
C4.5-Score	136/234 = *58.1%*	36/77 = *46.8%*	36/45 = *80.0%*

From this table we can make the following observations:

- Examination of the *Precision_by_Effort* measurement reveals that with SBA, the total effort (264) spent on teaching remedial classes is less than the traditional selection method (269), while the total effort spent on teaching the weak students is improved from 142 to 162. This is very significant as more efforts are spent on the actual weak students. Examination of the *Precision_by_Student* measurement reveals that the traditional selection method requires a total of 112 students to go for remedial classes as compared to SBA's 87. This means that by applying the data mining technique, we are able to reduce the number of students attending remedial classes by more than 20%. This is a significant improvement. Yet, our *Recall* is almost the same as the traditional method (only lower by 1 student).
- SBA is also better than the other two scoring methods. Note that although the original C4.5 does produce a score in its classification, we have modified the C4.5 system so that it can produce a score (or confidence factor).

	SBA	C4.5-score	NB-score
1	15	11	12
2	10	11	9
3	8	9	8
4	3	0	4
5	5	2	4
6	0	3	1
7	2	1	1
8	2	2	2
9	0	4	1
10	0	2	3

Table 2: Bin configuration of different scoring methods

To explain why SBA performs better than other systems, let us see the rankings of the weak students produced by various systems. To save space, after scoring and ranking, we divide each ranked list equally into 10 bins. Table 2 shows the weak students in each bin produced by each scoring system. Clearly, we see that the SBA scoring method is superior to others. It does the best in pushing those weak students to the top. For example, SBA captures more weak students in its top 3 bins as compared to the other two scoring techniques. The two bottom bins of SBA also have no weak students, while the other systems all have some weak students in the bottom bins, which is undesirable.

6. CONCLUSION

In this paper, we reported a real-life application in an education domain. It aims to select the potentially weak students for remedial classes. We showed that by using a data mining technique we achieve much better results. This reduces the burden on both students and teachers. In our future work, we plan to build an application specific system that can be used by any educational institution to select the right students for various purposes. These students can be good students, weak students or students with special needs, etc.

ACKNOWLEDGEMENT

We are grateful to the Gifted Education Branch of the Ministry of Education, Singapore, in particular Director Dr. Bee Geok Tan for initiating this project and for her constant involvement and help. We thank Yiyuan Xia for modifying the C4.5 program to produce the C4.5-score system. We also thank Chew Lim Tan, Wynne Hsu, and Huan Liu for useful discussions. The project is funded by a research grant from National Science and Technology Board of Singapore and National University of Singapore under RP3981678.

REFERENCES

[1] Agrawal, R., Imielinski, T., Swami, A. "Mining association rules between sets of items in large databases." *SIGMOD-1993*, 1993.

[2] Agrawal, R. and Srikant, R. "Fast algorithms for mining association rules." *VLDB-94*, 1994.

[3] Bayardo, R., Agrawal, R, and Gunopulos, D. "Constraint-based rule mining in large, dense databases." *ICDE-99*, 1999.

[4] Bayardo, R., Agrawal, R. "Mining the most interesting rules." *KDD-99*.

[5] Brin, S. Motwani, R. Ullman, J. and Tsur, S. "Dynamic Itemset counting and implication rules for market basket data." *SIGMOD-97*, 1997.

[6] Chan, P. K., and Stolfo, S. J. "Towards scaleable learning with non-uniform class and cost distributions: a case study in credit card fraud detection", *KDD-98*.

[7] Dietterich, T and Baskiri, G. “Solving multiclass learning problems via error-correcting output code.” *Journal or AI research*, vol 2. 263-286.

[8] Dong. G. Zhang, X. Wong, L. Li, J. 1999. “CAEP: classification by aggregating emerging patterns.” *Discovery-Science-99*.

[9] Fawcett, T., and Provost, F. “Combining data mining and machine learning for effective user profile.” *KDD-96*.

[10] Fayyad, U. M. and Irani, K. B. "Multi-interval discretization of continuous-valued attributes for classification learning." *IJCAI-93*, 1993.

[11] Fukuda, T. Morimoto, Y. Morishita, S and Tokuyama, T. “Data mining using two-dimensional optimized association rules: scheme, algorithms and visualization. *SIGMOD-96*.

[12] Gehrke, J., Ganti, V., Ramakrishnan, R. and Loh, W. “BOAT-optimistic decision tree construction.” *SIGMOD-99*.

[13] Han, J. and Fu, Y. “Discovery of multiple-level association rules from large databases.” *VLDB-95*, 1995.

[14] Hughes, A. M. *The complete database marketer: second-generation strategies and techniques for tapping the power of your customer database*. Chicago, Ill.: Irwin Professional, 1996.

[15] Kohavi, R., John, G., Long, R., Manley, D., and Pfleger, K. MLC++: a machine learning library in C++. *Tools with artificial intelligence*, 740-743, 1994.

[16] Kubat, M. and Matwin, S. “Addressing the curse of imbalanced training sets.” *ICML-1997*.

[17] Lee, W., Stolfo, S. J., and Mok, K. W. “Mining audit data to build intrusion detection models.” *KDD-98*.

[18] Ling, C. and Li C. "Data mining for direct marketing: problems and solutions," *KDD-98*.

[19] Liu, B., Hsu, W. and Ma, Y. “Integrating classification and association rule mining.” *KDD-98*.

[20] Liu, B., Hsu, W. and Ma, Y. “Mining association rules with multiple minimum supports." *KDD-99*.

[21] Liu, B., Ma, Y., Wong, C K. and Yu, P. “Target selection via scoring using association rules”. IBM Research Report 21697, March 2000, Yorktown Heights, NY.

[22] Mannila, H., Pavlov, D and Smyth, P “Prediction with local patterns using cross-entropy.” *KDD-99*.

[23] Meretkis, D. & Wuthrich, B. “Extending naïve bayes classifiers using long itemsets.” *KDD-99*.

[24] Merz, C. J, and Murphy, P. UCI repository of machine learning databases [http://www.cs.uci.edu/~mlearn/MLRepository.html], 1996.

[25] Ng. R. T. Lakshmanan, L. Han, J. “Exploratory mining and pruning optimisation of constrained association rules.” *SIGMOD-98*, 1998.

[26] Pazzani, M., Merz, C., Murphy, P., Ali, K., Hume, T., and Brunk, C. “Reducing misclassification costs.” *ICML-97*, 1997.

[27] Piatetsky-Shapiro, G. and Massand, B. “Estimating campaign benefits and modelling lift.” *KDD-99*, 1999.

[28] Provost, F., and Fawcettt, T. “Analysis and visualization of classifier performance: comparison under imprecise class and cost distributions.” *KDD-97*, 1997.

[29] Quinlan, R. *C4.5*: *program for machine learning*. Morgan Kaufmann, 1992.

[30] Rastogi, R. and Shim, K. 1998. “Mining optimized association rules with categorical and numeric attributes.” *ICDE–98*, 1998.

[31] Shafer, J., Agrawal, R. & Mehta, M. "SPRINT: A scalable parallel classifier for data mining." *VLDB-96*.

[32] Sanjeev, A. P. and Zytkow, J. "Discovering Enrollment Knoweldege in University Database." *KDD-95*. 1995.

[33] Toivonen, H. “Sampling large databases for association rules.” *VLDB-96*, 1996.

[34] Tong, A. Lu, H., Han, J. and Feng, L. “Break the barrier of transactions: mining inter-transaction association rules” *KDD-99*.