A combinational incremental ensemble of classifiers as a technique for predicting students’ performance in distance education

S. Kotsiantis, K. Patriarcheas, M. Xenos *

Hellenic Open University, School of Sciences and Technology, Computer Science, Greece

ABSTRACT

The ability to predict a student’s performance could be useful in a great number of different ways associated with university-level distance learning. Students’ marks in a few written assignments can constitute the training set for a supervised machine learning algorithm. Along with the explosive increase of data and information, incremental learning ability has become more and more important for machine learning approaches. The online algorithms try to forget irrelevant information instead of synthesizing all available information (as opposed to classic batch learning algorithms). Nowadays, combining classifiers is proposed as a new direction for the improvement of the classification accuracy. However, most ensemble algorithms operate in batch mode. Therefore a better proposal is an online ensemble of classifiers that combines an incremental version of Naive Bayes, the 1-NN and the WINNOW algorithms using the voting methodology. Among other significant conclusions it was found that the proposed algorithm is the most appropriate to be used for the construction of a software support tool.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

Distance education is an educational method whose main characteristic, setting it apart from other educational methods, is that the student is being taught and instructed without the physical presence of a tutor in a teaching classroom, based on a special, tutorially designed learning material and on his/her communication with the tutor [1]. In distance education the student often feels isolated consequently the communication between the student and the teacher as well with the other students is an important parameter for the success of a distance education program [2,3]. The student may become disheartened by difficulties encountered in the learning process and which may lead to a slowdown and/or quitting his/her studies. Student encouragement and support are a primary concern of a good, modern programme of distance study. Consequently, for a web-based open and dynamic learning environment, personalized support for learners becomes more important [4]. Previous research studies [5,6] have shown that students come to distance education courses with variable expectations of the levels of service and support they will receive from their tutors. It is obvious, the tutors in a distance education course have a particular role and it is important to be able to recognize and locate students with high probability of poor performance in order to take precautions and be better prepared to face such cases. Consequently, given that the distance education it addresses to adults with special educational needs and incongruity (age, professional and family obligations, etc.), there is a growing interest in the factors predicting the student’s performance particularly in distance education environments [7–12]. A very promising arena to attain this objective is the use of data mining and machine learning algorithms [13].

Supervised learning algorithms are presented with instances, which have already been pre-classified in some way. That is, each instance has a label, which identifies the class to which it belongs and so this set of instances is sub-divided into classes. Supervised machine learning explores algorithms that reason from the externally supplied instances to produce general hypotheses, which will make predictions about future instances.

To induce a hypothesis from a given dataset, a learning system needs to make assumptions about the hypothesis to be learned. These assumptions are called biases. A learning system without any assumptions cannot generate a useful hypothesis since the number of hypotheses that are consistent with the dataset is usually enormous. Since every learning algorithm uses some biases, it behaves well in some domains where its biases are appropriate while it performs poorly in other domains [14]. Therefore, combining classifiers is proposed as a new direction for the improvement of the classification accuracy.

However, most ensemble algorithms operate in batch mode, i.e., they repeatedly read and process the entire training set. Basically, they require at least one pass through the training set for every...
base model to be included in the ensemble. The base model learning algorithms themselves may require several passes through the training set to create each base model. In situations where data is being generated continuously as in an educational environment, storing data for batch learning is impractical, which makes using these ensemble-learning algorithms impossible.

Incremental learning ability is very important to machine learning approaches designed for solving real-world problems due to two reasons. Firstly, it is almost impossible to collect all helpful training examples before the trained system is put into use. Therefore when new examples are fed, the learning approach should have the ability of doing some revisions on the trained system so that unlearned knowledge encoded in those new examples can be incorporated. Secondly, modifying a trained system may be cheaper in time cost than building a new system from scratch, which is useful especially in real-time applications.

In the present work an ensemble that combines an incremental version of Naive Bayes, the 1-NN and the WINNOW algorithms using the voting methodology is proposed. This paper uses the proposed ensemble in order to predict the students’ performance in a distance learning system.

The application of the proposed technique in predicting students’ performance proved to be useful for identifying poor performers and it can enable tutors to take precautionary measures at an earlier stage, even from the beginning of an academic year, in order to provide additional help to the groups at risk. The probability of more accurate diagnosis of students’ performance is increased as new curriculum data has entered during the academic year, offering the tutors more effective results.

This paper is organised as follows: Section 2 introduces some basic themes about Educational data mining for predicting student performance and in online learning algorithms and incremental ensemble classifiers, while Section 3 discusses the proposed ensemble method. The dataset used for experiments is described in Section 4. Experiment results and comparisons of the proposed combining method with other learning algorithms are presented in Section 5. Finally, Section 6 describes summary and further research topics.

2. Background

At this point it is advisable to present some basic themes about Educational data mining for predicting student performance and in online learning algorithms and incremental ensemble classifiers.

2.1. Educational data mining for predicting student performance

To implement real intelligence or adaptivity, the models for tutoring systems should be learnt from data. However, the data sets are so small that machine learning methods cannot apply directly. Hämäläinen and Vinni [15] tackled this problem, and gives general for creating accurate classifiers for educational data. They describe experiment to predict course success with more 80% accuracy.

Minaei-Bidgoli et al. [16] present an approach to classifying students in order to predict their final grade based on features extracted from logged data in an education web-based system and they demonstrates a genetic algorithm (GA) to successfully improve the accuracy of combined classifier performance, about 10–12% when comparing to non-GA classifier.

Some of the most useful data mining tasks and methods are classification, clustering, visualization, association rule and statistically mining [17]. These methods uncover new, interesting and useful knowledge based on students’ usage data [18]. Some of the main e-learning problems or subjects to which data mining techniques have been applied [19] are dealing with the assessment of student’s learning performance, provide course adaptation and learning recommendations based on the students’ learning behaviour, dealing with the evaluation of learning material and educational web-based courses, provide feedback to both teachers and students of e-learning courses, and detection of atypical student’s learning behaviour.

As for the classification (one of the most useful educational data mining tasks in e-learning), there are different educational objectives for using classification, such as: to group students who are hint-driven or failure-driven and find common misconceptions that students possess [20], to predict/classify students when using intelligent tutoring systems [15], etc. Some other examples are: predicting a student’s academic success (to classify as low, medium and high risk classes) using different data mining methods [21]; using neural network models from Moodle logs [22]. Finally, Lykourentzou et al. [23] use feed-forward neural networks, support vector machines and probabilistic ensemble simplified fuzzy ARTMAP for the prediction of student dropout.

2.2. Online learning algorithms and incremental ensemble classifiers

When comparing online and batch algorithms, it is worthwhile to keep in mind the different types of setting where they may be applied. In a batch setting, an algorithm has a fixed collection of examples in hand, and uses them to construct a hypothesis, which is used thereafter for classification without further modification. In an online setting, the algorithm continually modifies its hypothesis as it is being used; it repeatedly receives a pattern, predicts its classification, finds out the correct classification, and possibly updates its hypothesis accordingly.

The on-line learning task is to acquire a set of concept descriptions from labelled training data distributed over time. This type of learning is important for many applications, such as computer security, intelligent user interfaces, and market-basket analysis. For instance, customer preferences change as new products and services become available. Algorithms for coping with concept drift must converge quickly and accurately to new target concepts, while being efficient in time and space.

Desirable characteristics for incremental learning systems in environments with changing contexts are:

- the ability to detect a context change without the presentation of explicit information about the context change to the system;
- the ability to quickly recover from a context change and adjust the hypotheses to fit the new context;
- the capability to make use of previous experience in situations where old contexts reappear.

Online learning algorithms process each training instance once “on arrival” without the need for storage and reprocessing, and maintain a current hypothesis that reflects all the training instances seen so far. Such algorithms are also useful with very large datasets, for which the multiple passes required by most batch algorithms are prohibitively expensive.

Numerous surveys have been conducted to study the incremental ensemble classifiers for a number of applications [24–31].

Researchers as Swere et al. [32] have developed online algorithms for learning traditional machine learning models such as decision trees. Given an existing decision tree and a new example, this algorithm adds the example to the example sets at the appropriate non-terminal and leaf nodes and then confirms that all the attributes at the non-terminal nodes and the class at the leaf node are still the best.

Batch neural network learning is often performed by making multiple passes (known in the literature as epochs) through the
data with each training example processed one at a time. So neural networks can be learned online by simply making one pass through the data. However, there would clearly be some loss associated with only making one pass through the data [33].

There is a known drawback for all these algorithms since it is very difficult to perform learning with several examples at once. In order to solve this problem, some algorithms rely on windowing techniques [34] which consist in storing the n last examples and performing a learning task whenever a new example is encountered.

The Weighted Majority (WM) algorithm [35] forms the basis of many online algorithms. WM maintains a weight vector for the set of experts, and predicts the outcome via a weighted majority vote between the experts. WM online learns this weight vector by “punishing” erroneous experts. A number of similar algorithms have been developed such as [36].

Voted-perceptron [37] stores more information during training and then uses this elaborate information to generate better predictions on the test data. The information it maintains during training is the list of all prediction vectors that were generated after each and every mistake. For each such vector, the algorithm counts the number of iterations the vector “survives” until the next mistake is made; they refer to this count as the “weight” of the prediction vector. To calculate a prediction it computes the binary prediction of each one of the prediction vectors and combines all these predictions by a weighted majority vote. The weights used are the survival times described above. This makes intuitive sense, as “good” prediction vectors tend to survive for a long time and thus have larger weight in the majority vote.

The concept of combining classifiers is proposed as a new direction for the improvement of the performance of classifiers [14]. Unfortunately, in an on-line setting, it is less clear how to apply ensemble methods directly. For instance, with bagging, when one new example arrives that is misclassified, it is too inefficient to resample the available data and learn new classifiers. One solution is to rely on the user to specify the number of examples from the input stream for each base learner [38,39], but this approach assumes we know a great deal about the structure of the data stream. There are also on-line boosting algorithms that reweight classifiers [39,40], but these assume a fixed number of classifiers. In addition, online boosting is likely to suffer a large loss initially when the base models have been trained with very few examples, and the algorithm may never recover. In the following section, we propose an online ensemble of classifiers.

3. Proposed online ensemble

It is a well-known fact that the selection of an optimal set of classifiers is an important part of multiple classifier systems and the independence of classifier outputs is generally considered to be an advantage for obtaining better multiple classifier systems. In terms of classifier combination, the voting methods demand no prerequisites from the classifiers.

When multiple classifiers are combined using voting methodology, we expect to obtain good results based on the belief that the majority of experts are more likely to be correct in their decision when they agree in their opinion.

As far as the used learning algorithms of the proposed ensemble are concerned, three online algorithms are used:

- WINNOW [35] is a linear online algorithm. The heart of the algorithm is similar to the perceptron. In detail, it classifies a new instance x into class 1 if $\sum x_i w_i > \theta$ and into class 1 otherwise, however, if the predicted class is incorrect, WINNOW updates its weights as follows. If predicted value is $y' = 0$ and actual value is $y = 1$, then the weights are too low; so, for each feature such that $x_i = 1$, $w_i = w_i - \beta$, where $\beta$ is a number greater than 1, called the promotion parameter. If $y' = 1$ and $y = 0$, then the weights were too high; so, for each feature $x_i = 1$, it decreases the corresponding weight by setting $w_i = w_i - \beta$, where $0 < \beta < 1$, called the demotion parameter. WINNOW is an example of an exponential update algorithm. The weights of the relevant features grow exponentially but the weights of the irrelevant features shrink exponentially. For this reason, WINNOW can adapt rapidly to changes in the target function (concept drift).

- 1-Nearest Neighbour (1NN) is based on the principal that the examples within a data set will generally exist in close proximity with other examples that have similar properties. If the examples are tagged with a classification label, then the value of the label of an unclassified example can be determined by observing the class of its nearest neighbour. The absolute position of the examples within this space is not as significant as the relative distance between examples. This relative distance is determined using a distance metric. Ideally, the distance metric must minimize the distance between two similarly classified examples, while maximizing the distance between examples of different classes. In the present implementation, we use normalized Euclidean distance to find the training instance closest to the given test instance, and predicts the same class as this training instance.

- Naive Bayes classifier is the simplest form of Bayesian network since it captures the assumption that every feature is independent of the rest of the features, given the state of the class feature. The assumption of independence is clearly almost always wrong. However, simple naive Bayes method remains competitive, even though it provides very poor estimates of the true underlying probabilities [41]. The naive Bayes algorithm is traditionally used in “batch mode”, meaning that the algorithm does not perform the majority of its computations after seeing each training example, but rather accumulates certain information on all of the training examples and then performs the final computations on the entire group or “batch” of examples [41]. However, note that there is nothing inherent in the algorithm that prevents one from using it to learn incrementally. As an example, consider how the incremental naïve Bayes algorithm can work assuming that it makes one pass through all of the training data. In step #1, it initializes all of the counts and totals to 0 and then goes through the training examples, one at a time. For each training example, it is given the feature vector x and the value of the label for that. The algorithm goes through the feature vector and increments the proper counts. In step #2, these counts and totals are converted to probabilities by dividing each count by the number of training examples in same class. The final step (#3) computes the prior probabilities $p(k)$ as the fraction of all training examples that are in class k.

The proposed ensemble begins by creating a set of three experts (NB, WINNOW, 1-NN). When a new instance arrives, the algorithm passes it to and receives a prediction from each expert. In online setting, the algorithm continually modifies its hypothesis as it is being used; it repeatedly receives a pattern, predicts its classification based on majority vote of the expert predictions, finds out the correct classification, and possibly updates its hypothesis accordingly.

The proposed ensemble is schematically presented in Fig. 1, where $h_i$ is the produced hypothesis of each classifier, x the instance for classification and $y$ the final prediction of the proposed online ensemble. The number of model or runtime parameters to be tuned by the user is an indicator of an algorithm’s ease of use.
Data were collected from two distinct sources, the Students’ Registry of the HOU and the records of the tutors. This enabled the authors to collect data concerning almost all students. Finally, the ‘class attribute’ (dependent variable) represents the result on the final examination test with two values. ‘Fail’ represents students with poor performance. ‘Poor performance’ indicated students that suspended their studies during the academic year (due to personal or professional reason or inability to hand in 2 of the written assignments), students who did not participate in the final examination or sit for the final examination but got a mark less than 5. ‘Pass’ represents students who completed the INF10 module getting a mark of 5 or more in the final test.

5. Experiment results and comparisons

During the first phase (training phase) every algorithm was trained using the data collected from the academic year 2006–2007. The training phase was divided in four consecutive steps. The 1st step included the data from the first written assignment and the resulting class. The 2nd step included data used for the 1st step and second written assignment. The 3rd step included data used for the 2nd step and third written assignment. The 4th step included data used for the 3rd step and fourth written assignment.

Subsequently, a group of data for the new academic year (2007–2008) were collected. This group was used to measure the prediction accuracy (testing phase). The testing phase also took place in four steps. During the 1st step, the first written assignment was used in order to predict the class and the prediction accuracy is denoted in the row labeled ‘WRI-1’ in the Table 2.

The remaining steps use data of the new academic year in the same way as described above. The prediction accuracy is denoted in the rows labeled ‘WRI-2’, ‘WRI-3’, and ‘WRI-4’ concurrently in the Table 3 for each algorithm.

### Table 2

<table>
<thead>
<tr>
<th>Proposed online ensemble</th>
<th>NB</th>
<th>1-NN</th>
<th>WINNOW</th>
</tr>
</thead>
<tbody>
<tr>
<td>WRI-1</td>
<td>73.86</td>
<td>68.00*</td>
<td>73.86</td>
</tr>
<tr>
<td>WRI-2</td>
<td>78.39</td>
<td>76.39</td>
<td>78.24</td>
</tr>
<tr>
<td>WRI-3</td>
<td>81.73</td>
<td>79.43</td>
<td>78.24</td>
</tr>
<tr>
<td>WRI-4</td>
<td>81.81</td>
<td>80.84</td>
<td>78.47</td>
</tr>
<tr>
<td>Average accuracy</td>
<td>78.95</td>
<td>76.17</td>
<td>77.20</td>
</tr>
</tbody>
</table>

### Table 3

Comparing the proposed ensemble with C4.5, 3NN, RIPPER, SMO, BP and RBF algorithms.

<table>
<thead>
<tr>
<th>Proposed online ensemble</th>
<th>C4.5</th>
<th>3NN</th>
<th>RIPPER</th>
<th>SMO</th>
<th>BP</th>
<th>RBF</th>
</tr>
</thead>
<tbody>
<tr>
<td>WRI-1</td>
<td>73.86</td>
<td>73.86</td>
<td>73.86</td>
<td>69.19</td>
<td>71.56</td>
<td>72.38</td>
</tr>
<tr>
<td>WRI-2</td>
<td>78.39</td>
<td>77.35</td>
<td>78.09</td>
<td>77.65</td>
<td>77.95</td>
<td>78.17</td>
</tr>
<tr>
<td>WRI-3</td>
<td>81.73</td>
<td>80.02</td>
<td>78.99</td>
<td>80.02</td>
<td>80.10</td>
<td>80.62</td>
</tr>
<tr>
<td>WRI-4</td>
<td>81.81</td>
<td>81.14</td>
<td>78.99</td>
<td>80.09</td>
<td>81.73</td>
<td>80.92</td>
</tr>
<tr>
<td>Average accuracy</td>
<td>78.95</td>
<td>78.09</td>
<td>77.48</td>
<td>78.06</td>
<td>77.24</td>
<td>77.81</td>
</tr>
</tbody>
</table>
During the first experiment, each online learning algorithm (Naïve Bayes, 1-NN, WINNOW) is compared with the proposed ensemble.

It must be mentioned that we used the free available source code for these algorithms by Witten and Frank [42] for our experiments. We have tried to minimize the effect of any expert bias by not attempting to tune any of the algorithms to the specific dataset. Wherever possible, default values of learning parameters were used. This approach may result in lower estimates of the true error rate, but it is a bias that affects all the learning algorithms equally.

In Table 2, as it is obvious, the specific algorithm performed statistically better than the proposed according to t-test with \( p < 0.05 \). Furthermore, in Table 2, **” indicates that proposed ensemble performed statistically better than the specific classifier according to t-test with \( p < 0.05 \). In all the other cases (Fig. 2), there is no significant statistical difference between the results (Draws).

To sum up, the proposed ensemble is significantly more precise than WINNOW algorithm in four out of the four test steps. In addition, the proposed algorithm is significantly more accurate than 1-NN algorithms in two out of the four test steps. Moreover, the proposed ensemble is significantly more precise than NB algorithm in one out of the four test steps.

During the second experiment, a representative algorithm for each of batch sophisticated machine learning techniques was compared with the proposed ensemble. The batch algorithm as an upper measure of the accuracy of learning algorithms was used. Most of the incremental versions of batch algorithms are not lossless [32–34]. A lossless online learning algorithm is an algorithm that returns a hypothesis identical to what its corresponding batch algorithm would return given the same training set. The C4.5 algorithm [43] was the representative of the decision trees in our study. A well-known learning algorithm to estimate the values of the weights of a neural network – the RBF algorithm [42] – was the representative of the neural nets. In this study, the 3-NN algorithm that combines robustness to noise and less time for classification than using a larger \( k \) for \( k \)NN was also used [44]. The RIPPER [45] was the representative of the rule learners in our study. Finally, the Sequential Minimal Optimization (or SMO) algorithm was the representative of the SVMs in our study [46].

As it obvious in Table 3, the proposed ensemble is significantly more precise than RBF, BP and SMO algorithms in one out of four test steps. In addition, the proposed algorithm is significantly more accurate than 3NN algorithms in two out of four test steps. The proposed algorithm is also significantly more precise than RIPPER and C4.5 algorithms in one out of four test steps. Finally, the proposed algorithm is also significantly more precise than Voted Perceptron in four out of four test steps.

During the third experiment, some well known ensembles of classifiers were compared with the proposed ensemble. It must be mentioned that the other ensembles can be only used in batch mode. We used batch ensembles as an upper measure of the accuracy of ensembles. For the third experiment, we used for the comparison: (a) Adaboost algorithm with decision stump as base learner and 10 iterations [47], (b) Random Forest ensemble with 10 trees [48], (c) Voted-Perceptron algorithm [37], (d) Rotation Forest algorithm with C4.5 as base learner and 10 iterations [49]. As it obvious in Table 4, the proposed ensemble is significantly more precise than the other tested batch ensembles in at least one out of four test steps.

As we have already mentioned, the main advantage of the proposed ensemble is that it can easily be adapted in an online envi-

<table>
<thead>
<tr>
<th>Proposed online ensemble</th>
<th>AdaBoost decision stump</th>
<th>Random forest</th>
<th>Voted perceptron</th>
<th>Rotation forest</th>
</tr>
</thead>
<tbody>
<tr>
<td>WRI-1</td>
<td>73.86</td>
<td>73.34</td>
<td>67.33</td>
<td>73.34</td>
</tr>
<tr>
<td>WRI-2</td>
<td>78.39</td>
<td>78.40</td>
<td>74.31</td>
<td>77.58</td>
</tr>
<tr>
<td>WRI-3</td>
<td>81.73</td>
<td>80.04</td>
<td>79.81</td>
<td>79.88</td>
</tr>
<tr>
<td>WRI-4</td>
<td>81.81</td>
<td>80.99</td>
<td>80.70</td>
<td>80.85</td>
</tr>
<tr>
<td>Average accuracy</td>
<td>78.95</td>
<td>78.19</td>
<td>77.96</td>
<td>77.91</td>
</tr>
</tbody>
</table>
environment in which not only new marks from the same WRI become available but also marks from the next WRI become available. If we had used another of the tested learners and ensembles then we should have retrained the classifier from the scratch when marks from the next WRI become available. Nevertheless, we present the training times in Table 5, in case all the algorithms were used as batch learners in our dataset.

Clearly, incremental updating would be much faster than rerunning a batch algorithm on all the data seen so far, and may even be the only possibility if all the data seen so far cannot be stored or if it is needed to perform online prediction and updating in real time or, at least, very quickly.

There is a great interest in minimizing the required training time because, as we have already said, a major research area of data analysis is the exploration of accurate techniques that can be applied to problems with millions of training instances.

6. Conclusions and future goals

This paper aims to fill the gap between empirical prediction of student performance and the existing MI techniques in a distance education environment. The proposed technique, aims to investigate phenomena in educational procedure from the point of causal interpretation point of view. In situations where data is being generated continuously as in an educational environment, storing data for batch learning is impractical. To this end, an online learning algorithm has been trained and found to be useful tools for identifying predicted poor performers in an open and distance learning environment.

Online learning is the area of machine learning concerned with learning each training example once (perhaps as it arrives) and never examining it again. Online learning is necessary when data arrives continuously so that it may be impractical to store data for batch learning or when the dataset is large enough that multiple passes through the dataset would take too long.

Ideally, it would be best to be able to identify or design the single best learning algorithm to be used in all situations. However, both experimental results and theoretical work indicate that this is not possible [14]. Recently, the concept of combining classifiers is proposed as a new direction for the improvement of the classification accuracy. However, most ensemble algorithms operate in batch mode. There are several avenues that could be explored when designing an online ensemble algorithm. A naive approach is to maintain a dataset of all observed instances and to invoke an offline algorithm to produce an ensemble from scratch when a new instance arrives. This approach is often impractical both in terms of space and update time for online settings with resource constraints. To help alleviate the space problem, the size of the dataset by only storing and utilizing the most recent or most important instances could be limited. However, the resulting update time is still often impractical.

Consequently, an online ensemble that combines three online classifiers: the Naïve Bayes, the 1-NN and the WINNOW algorithms using the voting methodology has been proposed. With the help of the proposed technique the tutors are in a position to know which of their students will complete a module or a course with sufficiently accurate precision. This precision reaches the 73% in the initial forecasts, which are based on demographic data of the students and beats the 82% before the final examinations. Dataset is from the module ‘Introduction in informatics’ but most of the conclusions are wide-ranging and present interest for the majority of HOU modules. It would be interesting to compare our results with those from other open and distance learning programs offered by other open Universities. Given that the HOU is not a conventional university (with the features of a homogenous student community) but it addresses to adults with special educational needs and incongruity (both as far as their age, their professional and family obligations are concerned), the future research access to such issues becomes particularly important.

References

[27] H.S. Mohammed, J. Leander, M. Marbach, R. Polikar, Comparison of ensemble techniques for incremental learning of new concept classes under hostile non-stationary environments, in: Proceedings of the IEEE International Conference on...


