Data Mining in Personalizing Distance Education Courses

W. Hämäläinen, T. H. Laine, E. Sutinen
Department of Computer Science, University of Joensuu, Finland

Abstract

The need to personalize distance education courses stems from their ultimate goal: the need to serve an individual student independently of time, place, or any other restrictions. This means that a distance education system should be served by a data mining system (*DMS*) to monitor, intervene in, and counsel the teaching-studying-learning process. Compared to intelligent tutoring systems or adaptive learning environments where a teacher has only an occasional role, a *DMS* emphasizes the role of expert, in this case teacher, to interpret the findings obtained from analysing the data retrieved from the course. A *DMS* was designed and implemented to analyse the study records of two programming courses in a distance curriculum of Computer Science. Various data mining schemes, including the linear regression and probabilistic models, were applied to describe and predict student performance. The results indicate that a *DMS* can help a distance education teacher, even in courses with relatively few students, to intervene in a learning process at several levels: improving exercises, scheduling the course, and identifying potential dropouts at an early phase.

1 Introduction

The need to personalize distance education courses stems from their ultimate goal: the need to serve an individual student independently of time, place, or any other restrictions. This means that a distance education system should be served by a data mining system (*DMS*) to monitor, intervene in, and counsel the teaching-studying-learning process. Compared to current intelligent tutoring systems or adaptive learning environments where a teacher has only an occasional role, a *DMS* emphasizes the role of expert, in this case teacher, to interpret the findings obtained from analysing the data retrieved from the course.

ViSCoS (*Virtual Studies of Computer Science*) is a distance education program intended originally for high school students interested in continuing later at university with Computer Science (*CS*) as their major subject [1]. The reasons for the Department of *CS* at the University of Joensuu to initiate the program in 2000 were threefold: to recruit high school student to study *CS* at university, particularly in Joensuu; to design an experimental platform with real learners to explore the opportunities of technology in education; and to attract young people to choose information and communication technology (*ICT*) as their future career. One could call these interests recruiting, academic, and industrial ones, respectively. Therefore, the success of the *ViSCoS* program could be measured by the fulfillment of the three expectations, and to some extent, data mining could be used for analysing and supporting at least purposes one and three. In this chapter, we focus on how data mining techniques can contribute to better student performance, benefitting also our recruiting purpose.

An important part of recruiting the students via *ViSCoS* is to have them to complete their studies in a due manner. However, as in many distance education courses in almost any if not every academic field and in almost any programming course, face-to-face or distance, the problem has been the large proportion of dropouts [2]. It seems to be that at least part of the students could be supported, if their learning process were traced and analysed for a sufficiently early intervention – either by course tutors or an intelligent tutoring system. This, however, requires a predictive model of their future performance, given the data from their previous study outcomes, like assignment and exercise points collected this far.

Instead of picking up an *ad hoc* model – which is common manner in educational field – our main motive has been to construct models from real data. This requires interaction between *descriptive* and *predictive modelling* – or data mining and machine learning. In the following, we will first consider the general paradigms for such data-driven modelling. Then we will describe our datasets and the models constructed from this data. We will introduce four simple modelling paradigms for analysing the dynamics of different factors in two distance learning courses. As a result, we construct two descriptive-predictive model pairs, namely correlation analysis followed by linear regression models and association rules followed by Naive Bayes models. These paradigms were selected for their simplicity, easy interpretation and suitability to small data sets. The emphasis is in descriptive modelling and selecting the most promising modelling paradigms and model structures for future intelligent tutoring systems. The predictive accuracy of the selected models will be compared by cross-validation. Finally we will conclude our results and make suggestions for future research.

2 General paradigms for intelligent tutoring systems

The idea *intelligent tutoring systems* (*ITSs*) (see e.g. [3, 4, 5, 6, 7]) is to adapt the teaching according to individual skills, knowledge, and needs, and give personal feedback just-in-time. The core of the system is a *tutoring module*, which is responsible for selecting suitable actions like generating tests and exercises, giving

hints and explanations, suggesting learning topics, searching learning material and collaborative partners.

The main idea of *ITS* is that they should be *adaptive* – i.e. adapt to the individual student's needs, but so far the current *ITS*s are far from real adaptivity. Rather, the current systems are very stable: they are either based on a fully deterministic rule-based system (e.g. [8]) or even if some uncertain rules are used, they are predefined by experts [9, 10, 11]. A typical *ITS* consists of a predefined learning path, which a student proceeds by studying a concept and passing a test before entering the next topic. If the student fails in tests, s/he is advised to study more. In so called Bayesian learning models (e.g. [12, 13]), the model structure is also fixed and teachers have assigned probabilities for passing a test, if the concept are actually not known. The only Bayesian thing in such models is the method of updating probabilities by Bayes rule. For example *SModel* [14] offers a general framework for Bayesian ITS, which the teacher can tailor for her/his own course setting by assigning the desired parameters.

A better approach would be to learn the model from real data, but only few experiments have been done on this direction. Kotsiantis et al. [15] have done a pioneer work in predicting dropouts by machine learning techniques. The data consisted of detailed personal data (including family and occupation obligations) and course activity in the beginning of course. In the comparison of six modelling paradigms – Naive Bayes model, decision trees (C4.5), back-propagation neural networks, support vector machines, 3-nearest neighbours algorithm and logistic regression – the Naive Bayes model and back-propagation performed best with over 80% prediction accuracy in the middle of course. Shin et Kim [16] have analysed the most influencing factors for course grades by logistic regression. In their experiment, the data was collected by a query, which contained questions about job load, social integration, willingness to study, amount of study time etc. The analysis revealed important factors, but they were not used further for prediction.

The unwillingness to use data mining and machine learning techniques in educational context is partly due to domain-specific problems. The data sets are typically very small (size of a class) and very often the courses change every year. Thus, we can accurately learn only very simple (and less informative) models, with decreased number of attributes and small domains to prevent overfitting. Still, it is likely that even the training error is quite large, because students do not behave deterministically. Special care should also be given for model validation, but in the educational setting it is not straight-forward: two classes never represent the same population, because even the smallest changes in material, tutors, assessment practices etc. affect the measured learning.

In this chapter, we clarify general modelling paradigms for developing truly adaptive *ITS*s and test the suitability of certain techniques with real course data. As a contrast to previous approaches, we do not suppose any background data, but try to analyse and predict course outcomes based on the data available during the course. We suppose that suitable prerequisite queries would improve the results, but students are usually not willing to fill any queries.

We have adopted a dual principle of *descriptive* and *predictive modelling* [17],

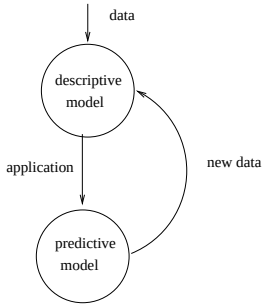

Figure 1: The iterative process of descriptive and predictive modelling.

which combines classical paradigms of data mining and machine learning (Figure 1). The idea is that in the descriptive phase, we analyse the previous year course data and search for local patterns like correlations and association rules. These results guide in selection of suitable modelling paradigm and especially in determining the model structure for predictive models. In the predictive phase, the modelling paradigm or paradigms (e.g. linear regression, Bayesian networks) are fixed and we define the model structure (variables and their relations). Then the model parameters are learnt from data. The resulting models are applied in the next course and updated in the light of new evidence. This is also the final test for validity of the model and guides the construction process in the next cycle.

In this framework, we can evaluate the descriptive and predictive power of models separately. For descriptive models, we can use the standard statistic measures like χ^2-test and F-values. In the search for associative rules, this can be taken into account beforehand, by determining the minimum frequency thresholds according to χ^2-test values. These evaluations confirm that we have found statistically exceptional dependencies, but their utility in prediction is harder to validate. In educational setting, the testing itself affects on course results, either directly (if the test is transparent, as recommendable) or indirectly, through tutors and/or test setting. Cross-validation with the current data set gives good guidelines on how well the model generalises to new data, supposing that the future course setting stays quite stable.

3 Data Description

In model construction we used the course records from two programming courses, (*Prog.1* and *Prog.2*), which are taught under *ViSCoS* distance learning program. The course records contained student identifiers, gender, exercise task points for 19 weeks, exam task points, total points, and grades. The exercise tasks were divided into six categories, according to topics covered (Table 1).

The data was collected in academic years 2001-2002 and 2002-2003. Since the course has stayed quite unchanged during both years, we simply combined both records after some normalisations. The resulting data sets were divided into two

Table 1: Exercise task categories.

Cat.	Description
A	Basic programming structures (*Prog.1*, weeks 1-3)
B	Loops and arrays (*Prog.1*, weeks 4-6)
C	Applets (*Prog.1*, weeks 7-9)
D	Object-oriented programming (*Prog.2*, weeks 1-3)
E	Graphical applications (*Prog.2*, weeks 4-8)
F	Error handling (*Prog.2*, weeks 9-10)

parts: *viscos1*, which contained all students (122 rows), and *viscos2*, which contained only those students, who had passed *Prog.1* course (91 rows).

The selected attributes, their domains and descriptions are presented in Table 2. In the original data set, all attributes were numerical, but for association rules and probabilistic models we converted them to binary-valued qualitative attributes. In exercise categories $A, ..., F$, we simply split the numeric domain in two: $little = \{0, ..., max/2\}$, $lot = \{max/2+1, ..., max\}$. However, in total points, $max/2 = 15$ is the passing limit, and we were interested in students, who had passed the*Prog.1* course. Thus, we defined that the binary value is $little$, when $TP < 23$, and lot, otherwise.

Table 2: Selected attributes, their numerical domain (*NDom*), binary-valued qualitative domain (*QDom*), and description.

Attr.	NDom.	QDom.	Description
G	0, 1	F, M	Student's gender.
A	$\{0, .., 12\}$	{little, lot}	Exercise points in A.
B	$\{0, .., 14\}$	{little, lot}	Exercise points in B.
C	$\{0, .., 12\}$	{little, lot}	Exercise points in C
D	$\{0, .., 8\}$	{little, lot}	Exercise points in D
E	$\{0, .., 19\}$	{little, lot}	Exercise points in E
F	$\{0, .., 10\}$	{little, lot}	Exercise points in F
$TP1$	$\{0, .., 30\}$	{little, lot}	total points of *Prog.1*
$TP2$	$\{0, .., 30\}$	{little, lot}	total points of *Prog.2*
$FR1$	$\{0, 1\}$	{fail, pass}	final result of *Prog.1*
$FR2$	$\{0, 1\}$	{fail, pass}	final result of *Prog.2*

Table 3: Correlation coefficients between the attributes.

$corr(A, TP1)$	0.61	$corr(D, E)$	0.59
$corr(B, C)$	0.69	$corr(D, F)$	0.42
$corr(B, E)$	0.61	$corr(D, TP2)$	0.60
$corr(B, TP1)$	0.75	$corr(E, F)$	0.73
$corr(B, TP2)$	0.54	$corr(E, TP2)$	0.73
$corr(C, D)$	0.51	$corr(F, TP2)$	0.61
$corr(C, TP1)$	0.69	$corr(TP, TP2)$	0.61

4 Correlations and Linear Regression Models

We started the research by (Pearson) correlation analysis to find the most strongly correlating attributes. The emphasis was to identify the most important factors for predicting final results $FR1$ and $FR2$. The results are presented in Table 3.

It can be recognised that the exercise points in B category correlate strongly with the amount of total points, especially in *Prog. 1* ($TP1$), but also in *Prog.2* ($TP2$) course. The latter correlation can be partially reduced to the correlation between B and E, which suggests that skills in loops and arrays are important prerequisites for graphical applications. However, the exercise points in A category have only a weak correlation with the total points of the courses. This may be due to the fact that the tasks in A were the easiest and nearly all students solved a lot of them.

Another interesting observation is that the exercise points in category E have a very strong correlation to the amount of total points on *Prog.2* course. These results suggests that the students' performance in the middle of the course has a strong impact on the outcomes.

The strong correlation between $TP1$ and $TP2$ can either tell about the general learning tendency, or prove the importance of managing basic programming skills before proceeding to more difficult topics.

Gender (G) was excluded from the correlation table, because it did not have any significant correlations with other attributes. For example, the correlation coefficients between *gender* and *TP1*/*TP2* were only approximately 0.16.

The natural complement of correlation in predictive modelling is *linear regression* (see e.g. [18]). In our research, we used multiple linear regression model, in which the predicted variable can depend on several factors. Formally we define:

Definition 1 *Let $X_1, ..., X_k$ and Y be discrete variables, where X_is are independent (explanatory) variables and Y is a dependent (response) variable. Then the expected value $\hat{Y}$ of Y can be predicted by linear equation*

$$\hat{Y} = \alpha + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_k X_k .$$

Table 4: Verification measures of regression models

Model	True pos.	True neg.	F-Signif.
$A, B \Rightarrow FR1$	0.978	0.333	$5.92e{-19}$
$A, B, C \Rightarrow FR1$	0.989	0.167	$1.70e{-20}$
$TP1 \Rightarrow FR2$	0.860	0.641	$6.94e{-9}$
$B, E \Rightarrow FR2$	0.840	0.845	$1.15e{-9}$
$B \Rightarrow FR2$	0.820	0.560	$2.70e{-6}$
$TP1, D \Rightarrow FR2$	0.840	0.744	$1.08e{-10}$
$TP1, D, E \Rightarrow FR2$	0.920	0.872	$6.76e{-13}$
$TP1, D, E, F \Rightarrow FR2$	0.920	0.872	$2.53e{-12}$

in which α and $\beta_1, ..., \beta_k$ are real-valued regression coefficients.

We constructed several linear regression models for the selected attributes that proved to have meaningful correlations in the first phase of our study. Analyses concentrated on predicting the final results of both courses ($FR1$ and $FR2$), with emphasis on predicting the outcome of *Prog.2*. Actually, the linear regression model predicted the total points ($TP1$ and $TP2$), and the final results ($FR1$ and $FR2$) were obtained by rule: $FR = 1$, if $TP \geq 15$, and $FR = 0$, otherwise.

The constructed models, with some validation measures, are presented in Table 4. The model name tells which attributes were used as independent variables to predict the final results ($FR1$ or $FR2$). *True positive* and *true negative* values tell the classification accuracy, i.e. the rate of correctly predicted outcomes from all passed/failed students. *F-significance* tells the significance level of the *F-value* for the whole model, i.e. the probability that such model would appear by chance. All the *F-values* are much less than 0.001, which means that the dependencies are very significant (there is much less than 0.1% probability that they would occur by chance).

Models $A, B \Rightarrow FR1$ and $A, B, C \Rightarrow FR1$ were constructed from the *viscos1* data set and the rest of the models from *viscos2* data set.

Validation measures of regression models suggest that each of these models can be used to predict quite reliably either success or failure of a student. Some models, like $TP1, D, E, F \Rightarrow FR2$, require information from the end of the other course (exercise category F particularly) in order to be usable. On the other hand model $B \Rightarrow FR2$ can already make good predictions for the outcomes of the second programming course based on the information on the exercise category B. Models where exercise category E has been used yield particularly good results for both success and failure predictions. This indicates great significance of E and therefore it could be used for instance to predict examination failure beforehand in order to provide a student with help to pass the course. We see also that attribute F has no

effect in the model as predictions from $TP1, D, E \Rightarrow FR2$ equal to predictions from $TP1, D, E, F \Rightarrow FR2$, thus influence of F to course outcome is minimal.

5 Association Rules and Probabilistic Models

Association rules offer a nice way to model nonlinear dependencies between attributes. Generally, association rules are of form $X \Rightarrow Y$, where X is a set of attributes and Y a single attribute. The *confidence* of the rule, $cf(X \Rightarrow Y)$, tells how strong the rule is, and the *frequency* or support of the rule, $fr(X \Rightarrow Y)$, tells the coverage of the rule (i.e. the portion of dataset it covers). Formally we define:

Definition 2 *Let $R = \{A_1, ..., A_k\}$ be a set of binary-valued attributes, and $r \in R$ a relation in R. The confidence and frequency of rule $X \Rightarrow Y$, $X \subseteq R$, $Y \in R$, $Y \notin X$ are defined by*
$cf(X \Rightarrow Y) = \frac{P(X,Y)}{P(X)} = P(Y|X)$ *and*
$fr(X \Rightarrow Y) = P(X, Y)$.
Given the user-defined thresholds $min_{cf}, min_{fr} \in [0, 1]$, we say that the rule is confident, if $cf(X \Rightarrow Y) \geq min_{cf}$, and frequent, if $fr(X \Rightarrow Y) \geq min_{fr}$.

For example, an association rule "$B = lot, E = lot, F = lot \Rightarrow FR2 = 1$" with confidence $cf = 0.956$ and frequency $fr = 0.516$ tells that about 96% of students who have done a lot of tasks in categories B, E and F have passed *Prog.2* course. In addition, the frequency tells that the rule covers about 52% of the students.

In *viscos* data we have found several interesting association rules. In Table 6 we have listed all frequent rules for predicting the final results of *Prog.1* and *Prog.2* courses. The minimum frequency thresholds min_{fr} were defined according to χ^2-test to catch only those rules, which are statistically significant on level 0.01 (Table 5). This decision is critical (compared to user-defined, constant frequency thresholds), because for small datasets with few attributes the frequencies have to be really high, before the rules have any statistical value. In addition, the thresholds decrease fast, when the number of attributes grows. For minimum confidence threshold we used value $min_{cf} = 0.7$.

The association rules can already be used for prediction. E.g. we can predict that students who have done a lot of A and B tasks will pass the *Prog.1* course with 92% probability. However, the frequent rules cover only some subsets of students, and a more general model would be needed. For this purpose we have constructed simple *Naive Bayes models*, *NB1* and *NB2*, for both *Prog.1* and *Prog.2* courses. The model structures are presented in Figure 2 and the parameters are given in Table 7.

In Naive Bayes model, we make *Naive Bayes assumption* that the leaf nodes depend only on the root node. In reality this assumption holds very seldom, but in practice the Naive Bayes model has proved to work well. In fact, Domingos et al. [19] have shown that this is only a sufficient but not a necessary condition for

Table 5: min_{fr} values for association rules $X \Rightarrow Y$, based on χ^2-values on 0.01 significance level. I.e. we demand that the probability that the rule holds by chance is less than 1%. Values are calculated for both datasets used.

$\|X, Y\|$	$n = 122$	$n = 91$
2	0.656	0.676
3	0.348	0.363
4	0.188	0.198
5	0.103	0.109

Table 6: The strongest frequent rules, and their frequencies and confidences for predicting final results for *Prog1* and *Prog.2*. The min_{fr} values were selected according to χ^2-test to guarantee significance on level 0.01. min_{cf} was 0.7.

Rule	fr	cf
$A = lot, B = lot, C = lot \Rightarrow FR1 = 1$	0.270	1.000
$A = lot, B = lot \Rightarrow FR1 = 1$	0.582	0.922
$A = lot \Rightarrow FR1 = 1$	0.697	0.773
$TP1 = lot, D = lot, E = lot, F = lot \Rightarrow FR2 = 1$	0.121	0.846
$TP1 = little, E = little, F = little \Rightarrow FR2 = 0$	0.198	0.947
$TP1 = lot, D = lot, E = lot \Rightarrow FR2 = 1$	0.275	0.926
$TP1 = lot, D = lot \Rightarrow FR2 = 1$	0.363	0.846
$TP1 = lot, E = lot \Rightarrow FR2 = 1$	0.363	0.943
$A = lot, B = lot, F = lot \Rightarrow FR2 = 1$	0.473	0.956
$B = lot, E = lot, F = lot \Rightarrow FR2 = 1$	0.516	0.959
$B = lot, F = lot \Rightarrow FR2 = 1$	0.473	0.956
$C = littleD = little \Rightarrow E = little$	0.407	0.974
$D = little, E = little, F = little \Rightarrow FR2 = 0$	0.308	0.824
$D = little, F = little \Rightarrow FR2 = 0$	0.363	0.744
$E = little, F = little \Rightarrow FR2 = 0$	0.404	0.720

optimality Naive Bayes classifier. Our experiments suggest that a more accurate condition might be that the attributes are not correlated, i.e. they are not linearly dependent.

In our model, the root nodes tell the probability of passing/failing the course and

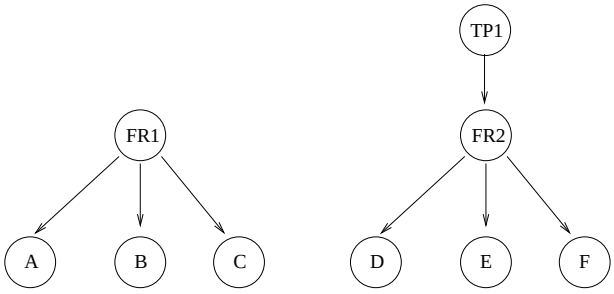

Figure 2: Naive Bayes models for predicting final results of *Prog.1* course, given the task points in A, B and C (left), and final results of *Prog.2* course, given the task points in D, E and F (right). $TP1$ (total points of *Prog.1*) has been used as a background variable, which defines the prior probability distribution of $FR2$.

the leaf nodes are used to update the probabilities, when exercise task points are gathered. Actually, the exercise task points in different categories depend on each other, but this is not taken into account.

In $NB1$, the prior probability of passing the course is simply assigned to 0.5. In $NB2$, we can use $TP1$ as a background variable, and define the prior probability of passing the course given that the student has got a lot/little of total points in *Prog.1*. This proved to be the most frequent rule for predicting $FR2$ according to *Prog.1* performance. The conditional probabilities have been simply calculated from the data with rule $P(Y|X) = \frac{P(X,Y)}{P(X)}$. The complement probabilities can be derived by rule $P(X = lot) = 1 - P(X = little)$.

Table 7: Probabilities for Naive Bayes models $NB1$ and $NB2$.

$P(FR1 = 1)$	0.500
$P(A = little\|FR1 = 0)$	0.167
$P(A = little\|FR1 = 1)$	0.045
$P(B = little\|FR1 = 0)$	0.833
$P(B = little\|FR1 = 1)$	0.180
$P(C = little\|FR1 = 0)$	0.917
$P(C = little\|FR1 = 1)$	0.584

$P(FR2 = 1\|TP1 = little)$	0.0.80
$P(FR2 = 1\|TP1 = lot)$	0.727
$P(D = little\|FR2 = 0)$	0.732
$P(D = little\|FR2 = 1)$	0.300
$P(E = little\|FR2 = 0)$	0.829
$P(E = little\|FR2 = 1)$	0.320
$P(F = little\|FR2 = 0)$	0.878
$P(F = little\|FR2 = 1)$	0.660

When the course proceeds, the prior probabilities are updated in the light of new evidence (exercise points in A, B, C, D, E and F) by *Bayes rule*:

$$P(FR|X) = \frac{P(FR) \times P(X|FR)}{P(X)}.$$

The predictions by Naive Bayes models after each new evidence are presented in Table 8. In the beginning of *Prog.1* course, nearly alls students have done a lot of exercises and thus they are predicted to pass the course. However, when exercise points in B category are known, the predictions are already very good, both for passing and failing. C points do not improve the results much, because only few students had done a lot of those tasks.

In *Prog.2* course, we can predict the outcomes already before the course has begun, based on *Prog.1* outcomes. These predictions are already surprisingly good – better than *Prog.1* predictions, when A points were known. The predictions improve fast, when D and E points are known. With F points we recognise a strange phenomenon: the classification accuracy decreases, even if we have got more evidence! This is totally correct, because F and E are highly correlating and thus the Naive Bayes assumption is clearly violated. This is the only case, when we have met restrictions of Naive Bayes model, and in practice these last predictions (when the course is over) are not so important. However, this demonstrates that Naive Bayes should be used with care, when attributes are correlated. A better approach might be a general Bayesian network with dependencies between attributes, as well. Friedman et al. [20] suggest that the best accuracy can be achieved by letting each attribute depend on at most one other attribute. Dempster-Shafer theory [21] offers another alternative: we can update the beliefs with only strong association rules (e.g. $F = lot \rightarrow FR2 = 1$), but leave the other predictions untouched.

Table 8: Predictions by Naive Bayes models $A \Rightarrow FR1$, $A, B \Rightarrow FR1$, $A, B, C \Rightarrow FR1$, $TP1 \Rightarrow FR2$, $TP1, D \Rightarrow FR2$, $TP1, D, E \Rightarrow FR2$, and $TP1, D, E, F \Rightarrow FR2$.

Model	true pos.	true neg.
$A \Rightarrow FR1$	0.96	0.30
$A, B \Rightarrow FR1$	0.82	0.80
$A, B, C \Rightarrow FR1$	0.82	0.84
$TP1 \Rightarrow FR2$	0.96	0.56
$TP1, D \Rightarrow FR2$	0.96	0.56
$TP1, D, E \Rightarrow FR2$	0.82	0.85
$TP1, D, E, F \Rightarrow FR2$	0.80	0.78

Table 9: Comparison of prediction accuracy of *LR* and *NB* models. All models were evaluated by cross-validation and the classification rates on all test sets were summed.

Model	LR rates		NB rates	
structure	True pos.	True neg.	True pos.	True neg.
1.	0.91	0.79	0.80	0.82
2.	0.78	0.85	0.92	0.55
3.	0.75	0.90	0.82	0.76

6 Evaluating the Predictive Power by Cross-Validation

Cross-validation is a standard way to evaluate the predictive power of a model, when no new test data is available. The idea is that the original data set of size n is divided k times to a training set and a test set of sizes $n - n/k$ and n/k. Each time a new model is learnt from the training set and tested with the test set. Finally the classification rates are summed to get the total (average) classification rates.

In our experiment, we divided our original data set into 10 training set–test set pairs. We constructed the corresponding linear regression and Naive Bayes models for the following model structures:

1. $A, B \Rightarrow FR1$
2. $TP1, D \Rightarrow FR2$
3. $TP1, D, E \Rightarrow FR2$

These model structures were selected, because they allow us to predict potential failing already in the middle of course. In the end of course (when all task points are known), the predictions are of course more accurate, but it does not benefit so much any more.

Each model was tested with corresponding test set and the classification rates were calculated. For NB models, the student's status was classified as *passing*, if the passing probability was ≥ 0.5, and *failing*, otherwise. The actual probabilities contain of course more information, but it was not used is this comparison. The average classification rates are presented in Figure 9.

The most interesting and encouraging result is that both modelling paradigms were able to predict course performance for more than 80% of students, when the course was still on. This is especially surprising in the first test, when no previous information was available, and the predictions were totally based on exercise points. In test cases 2 and 3 the *Prog.1* performance was already known. Test 2 gives especially valuable information, because dropout and failing are bigger problem in *Prog.2* course and these models are able to predict the outcomes when only three weeks of the course has passed.

When we compare the models, we see that LR model gives more "pessimistic"

predictions, i.e. it predict failing better than passing, while NB model is more "optimistic". The general classification rate is also better in LR model. This is mainly due to NB's simpler model structure – the attributes contained only binary-valued information (whether student has got a little or a lot of points in a given category). However, if we consider only the general classification rates, we see that the NB model can utilise the new information (E points in test 3) better than LR model. NB model is also more general and we expect it to adapt better in a new course setting with different tasks and maximum points.

According to our initial tests with those students, who had filled the course prerequisite query, we expect even better classification accuracy in the future. Currently only 60% of students had filled the query, but we could find strong dependencies between the previous programming skills and course performance in both courses. Another important factor that should be queried is the student's knowledge in mathematics, which is known [22, 23] to have strong correlation with programming skills.

7 Conclusions

In this chapter, we have introduced a general paradigms for tackling intelligent tutoring systems. We have focused on predicting the course performance in a distance learning setting, when only some of the exercise points are known. As an example, we have introduced two descriptive-predictive model pairs, correlations–linear regression and association rules–Bayes model. Both descriptive models have revealed statistically significant dependencies, which can be used to construct predictive models. In addition, when the predictive power of the models was tested by cross-validation, both predictive models were able to predict the outcomes with more than 80% accuracy during the course.

The applicability of both predictive modelling paradigms depends on some assumptions, which are revealled in the descriptive modelling. The linear regression models presuppose strong correlations between numeric attributes. Exercise points and total points satisfy this condition very well, but it should be noticed that the model can overfit easily, if the attribute values vary between different classes. To minimize this variance we recommend to group similar attributes, in our case exercise tasks on similar topics.

The Naive Bayes models do not catch only linear dependencies, but any conditional depndencies. They can be easily applied to any discrete data, but to restrict model complexity and thus overfitting, we discretised all attribute values to binary ($little/lot$). This makes the model very general, but on the other hand the Naive Bayes model does not fit as well as the linear regression. An important issue is the Naive Bayes assumption that attributes should be independent given the class. It has been shown that this condition is only sufficient but not necessary for optimality of Naive Bayes classifier, but our experiments suggest that the model suffers for strong correlations (i.e. linear dependencies). This situation occurs very often with course data, where certain topics are prerequisite for new topics, and managing former is reflected by latter. That is why we are currently studying suitability

of other similar paradigms like general Bayesian networks and Dempster-Shafer theory. Another interesting question is whether the lack of correlation is generally a sufficient condition for Naive Bayes optimality.

In the current research, we have simply tried to predict dropouts and failing as early as possible, but both proposed models can be easily generalised to predict also talented students, who might desire more challenges. In addition, it might be useful to distinguish dropouts and those who fail in the exam, because they may need different kind of tutoring.

Before implementing the intelligent tutoring system, we will carry on some more experiments to fully utilise all usable data. For example, our aim is to design a good prerequisite query, which all students should answer before the course. Other surveys [24, 16, 15] have reported that such prerequisite queries can predict the dropout in quite an early phase. Especially the ”locus of control” – whether the student believes to have the control over performance in her/his own hands or not – has proved to be the most important factor affecting dropout [24, 25]. This could be further divided into three factors, which are asked in the query: motivation, self-esteem and taking responsibility of one's own learning.

To summarise, the results from our experiments show the feasibility of data mining in order to personalize distance education courses. The data mining approach can also open the black box used in many adaptive or intelligent tutoring systems or learning environments. It is important for everyone in the teaching-studying-learning process to understand the explicit models used for description or prediction. For learners, this develops their meta-cognitive skills to observe, analyse and improve their learning process. For teachers, a model helps to understand and interpret the ongoing course at two levels: those of the whole learning group and of an individual learner. Therefore, data mining schemes can help the whole educational technology research community to move into the direction of semi-automation from a somewhat simplistic idea to automate the whole learning process.

References

[1] Haataja, A., Suhonen, J., Sutinen, E. & Torvinen, S., High school students learning computer science over the web. *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning (IMEJ)*, **3(2)**, 2001.

[2] Meisalo, V., Sutinen, E. & Torvinen, S., Choosing appropriate methods for evaluating and improving the learning process in distance programming courses. *Proceedings of 33rd ASEE/IEEE Frontiers in Education Conference*, 2003.

[3] Boulay, B.d., Can we learn from ITSs? *Intelligent Tutoring Systems*, pp. 9–17, 2000.

[4] Kabassi, K. & Virvou, M., Personalized adult e-training on computer use based on multiple attribute decision making. *Interaction with computers*, **(16)**, pp. 115–132, 2004.

[5] Chou, C.Y., Chan, T.W. & Lin, C.J., Redefining the learning companion: the past, present, and future of educational agents. *Computers & Education*, **(40)**,

pp. 225–269, 2003.

[6] Wasson, B., Advanced educational technologies: the learning environment. *Computers in human behavior*, **(4)**, pp. 571–594, 1997.

[7] Weber, G., Episodic learner modeling. *Cognitive Science*, **20(2)**, pp. 195–236, 1996.

[8] Cheung, B., Hui, L., Zhang, J. & Yiu, S., SmartTutor: an intelligent tutoring system in web-based adult education. *The journal of systems and software*, **(68)**, pp. 11–25, 2003.

[9] Ioannis, H. & Prentzas, J., Using a hybrid rule-based approach in developing an intelligent tutoring system with knowledge acquisition and update capabilities. *Expert systems with applications*, **(26)**, pp. 447–492, 2004.

[10] Vos, H., Contributions of minmax theory to instructional decision making in intelligent tutoring systems. *Computers in human behavior*, **(15)**, pp. 531–548, 1999.

[11] Hwang, G.J., A conceptual map model for developing intelligent tutoring systems. *Computers & Education*, **(40)**, pp. 217–235, 2003.

[12] Mislevy, R. & Drew, H., The role of probability-based inference in an intelligent tutoring system, 1996.

[13] Butz, C., Hua, S. & Maguire, R., Web-based intelligent tutoring system for computer programming. *Web Intelligence and Agent Systems: An International Journal*, **4(1)**, 2006. To appear.

[14] Zapata-Rivera, J.D. & Greer, J., Inspectable Bayesian student modelling servers in multi-agent tutoring systems. *International Journal of Human-Computer Studies*, **61(4)**, pp. 535–563, 2004.

[15] Kotsiantis, S., Pierrakeas, C. & Pintelas, P., Preventing student dropout in distance learning using machine learning techniques. *KES*, eds. V. Palade, R. Howlett & L. Jain, Springer, volume 2774 of *Lecture Notes in Computer Science*, pp. 267–274, 2003.

[16] Shin, N. & Kim, J., An exploration of learner progress and drop-out in Korea National Open University. *Distance Education*, **20(1)**, pp. 81–95, 1999.

[17] Hämäläinen, W., General paradigms for implementing adaptive learning systems. *Proceedings of IADIS Virtual Multi Conference on Computer Science and Information Systems (MCCSIS)*, 2005. Http://www.cs.joensuu.fi/∼whamalai/articles/paradigm.pdf.

[18] Draper, N. & Smith, H., *Applied regression analysis*. Wiley, 1981.

[19] Domingos, P. & Pazzani, M., On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, **29(2-3)**, pp. 103–130, 1997.

[20] Friedman, N., Geiger, D. & Goldszmidt, M., Bayesian network classifiers. *Machine learning*, **29(2-3)**, pp. 131–163, 1997.

[21] Shafer, G., *A mathematical theory of evidence*. Princeton University Press, 1976.

[22] Page, R., Software is discrete mathematics. *ACM SIGPLAN Notices*, **38(9)**, pp. 79–86, 2003.

[23] Devlin, K., Why universities require computer science students to take math. *Communications of the ACM*, **46(9)**, pp. 37–39, 2003.

[24] Parker, A., A study of variables that predict dropout from distance education. *International Journal of Educational Technology*, **1(2)**, 1999.

[25] Dille, B. & Mezack, M., Identifying predictors of high risk among community college telecourse students. *The American Journal of Distance Education*, **5(1)**, pp. 24–35, 1991.