**PREDICTING STUDENT PERFORMANCE FOR AN INTELLIGENT TUTORING SYSTEM USING SUPPORT VECTOR MACHINE**

Anjali Doneria (adoneri), Smit Doshi (stdoshi)

**Introduction**

Nowadays, the use of Intelligent Tutoring Systems in academic setups is becoming quite common. Students get the help of tutors to solve exercises and do their assignments or learn lessons. These tutors are equipped with technology to suggest problems to the student based on their learning capabilities, help them through the steps of solving a problem, suggest steps if the student asks for help or give feedback as and when needed. The use of intelligent tutors enables collection of huge amount of data that can be mined to predict to say if a student will get a question correct or incorrect, or if a student is proficient in a certain skill or task or knowledge component (KC). Thus, accurately predicting student performance based on their ongoing activities becomes crucial for effectively carrying out pedagogical interventions to ensure students’ satisfactory performance and help the instructors in knowing which student needs special attention in which area of the subject. Different data mining techniques have been applied to educational data to enhance teaching methods, improve the quality of teaching, identify weak students, and identify factors that influence a student’s academic performance. This helps in bringing the benefits to students, educators and academic institutions.

The problem is to develop an efficient algorithm to predict student performance for a given set of tasks while learning through an intelligent tutoring system as accurately as possible in order to enhance the teaching efficiency and thereby, help students learn better.

**Background**

The Association for Computing Machinery (ACM) used an educational dataset for hosting a competition in 2010. The datasets for this 2010 Knowledge Discovery and Data Mining Cup (KDD Cup) came from Intelligent Tutoring Systems (ITS) used by thousands of students over the course of the 2008-2009 school year. There were 30 million training rows and 1.2 million test rows in total. The task was to predict if a student answered a given math step correctly or incorrectly given information about the step and the students’ past history of responses. Predictions between 0 and 1 were allowed. The team from National Taiwan University [1] was the winner of the competition followed by students from Worcester Polytechnic Institute, Zachary A. Pardos and Niel Heffernan [2]. [1] employed combination of feature engineering, discretization, sparse matrix, weka, chromosome condensation, random forest, and logistic regression to solve the problem statement. They were not able to obtain good results by direct modeling of domain knowledge using Bayesian network, so they used traditional classification technique of logistic regression but the question of its extension to other problems in educational data mining to achieve the same purpose is under question. [2] used the combination of customised hidden markov models with features extracted along with Random forest-decision tree modelling creating a pristine ensemble method. Prediction error was very low for rows that had sufficient data to compile a complete user and skill feature set however the error was very high for rows where the user did not have sufficient skill data.

**Data Description**

According to the KDD cup dataset generated from the Carnegie Learning's Cognitive Tutor Geometry(2005 version), the data shows records of student interactions with the tutoring system while solving various problems for each unit to master certain set of skills. Certain terminologies used in this data are:

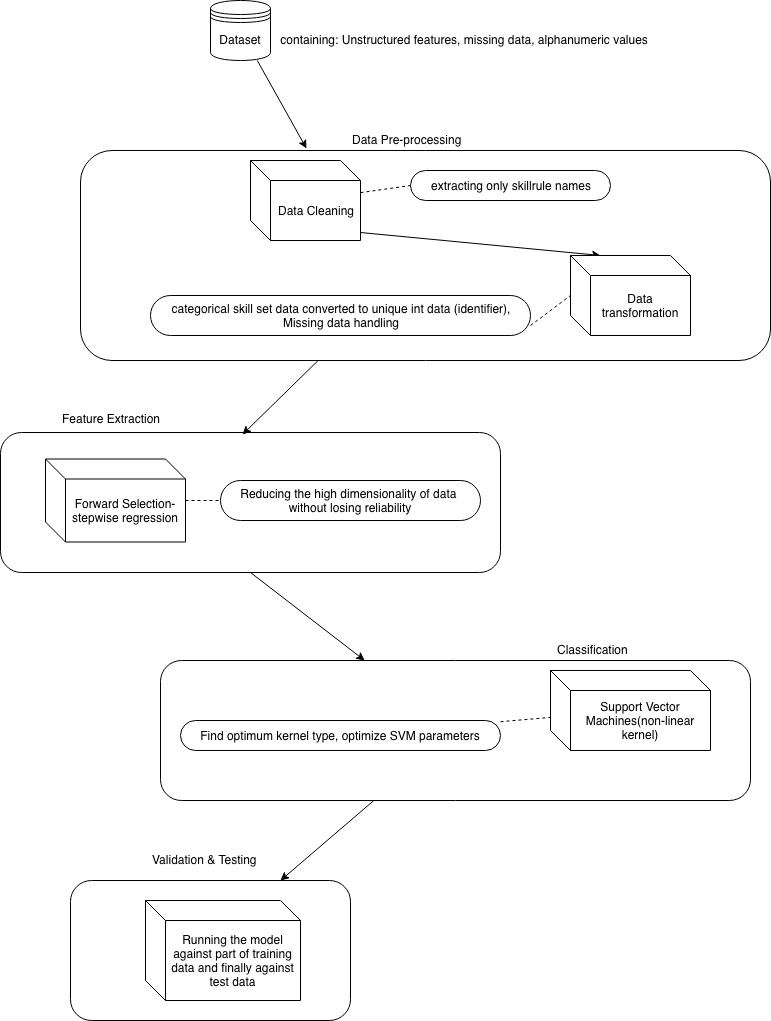
* Problem: It is a task for a student to perform that typically involves multiple steps. There are many such tasks for every unit/topic covered by the tutor for the student’s learning.
* Step: To solve the given problem, the student has to provide a step-by-step solution to reach the answer. Each step in this solution is recorded sequentially and matched against a predefined set of steps(ideal solution).
* Transaction: Each attempt, event, correct/incorrect action taken by the student as a form of an interaction in order to correctly attempt the required step towards the solution is recorded as a transaction.
* Knowledge Component: The skill-set/knowledge required to perform a certain step correctly to go forward towards the answer is pre-defined as the knowledge component. There maybe steps that require more than 1 knowledge component or even those steps which do not require any knowledge component to be carried out. If a student is able to perform a certain step, it means that the student possesses those knowledge components related to that step. The student’s task is to master in all these knowledge components to successfully solve those kind of problems and learn those corresponding units.
* Opportunity: An opportunity is a chance for a student to demonstrate whether he or she has learned a given knowledge component. A student’s opportunity count for a given knowledge component increases by 1 each time the student encounters a step that requires this knowledge component.

Given below is the description for the data columns in the KDD Cup 2010 dataset:

* Row: the row number
* Anon Student Id: unique, anonymous identifier for a student
* Problem Hierarchy: the hierarchy of curriculum levels containing the problem.
* Problem Name: unique identifier for a problem
* Problem View: the total number of times the student encountered the problem so far.
* Step Name: name of the step that the student is solving. These could collide with steps from other problems but the whole tuple is going to be unique
* Step Start Time: the starting time of the step. Can be null.
* First Transaction Time: the time of the first transaction toward the step.
* Correct Transaction Time: the time of the correct attempt toward the step, if there was one.
* Step End Time: the time of the last transaction toward the step.
* Step Duration (sec): the elapsed time of the step in seconds, calculated by adding all of the durations for transactions that were attributed to the step. Can be null (if step start time is null).
* Correct Step Duration (sec): the step duration if the first attempt for the step was correct.
* Error Step Duration (sec): the step duration if the first attempt for the step was an error (incorrect attempt or hint request).
* Correct First Attempt: the tutor’s evaluation of the student’s first attempt on the step (1 if correct, 0 if an error)
* Incorrects: total number of incorrect attempts by the student on the step.
* Hints: total number of hints requested by the student for the step.
* Corrects: total correct attempts by the student for the step
* KC(KC Model Name): the identified skills that are used in a problem, where available.
* Opportunity(KC Model Name): a count that increases by one each time the student encounters a step with the listed knowledge component.

The goal during testing will be to predict whether the student got the step right on the first attempt for each step in that problem. Each prediction will take the form of a value between 0 and 1, that indicates the probability of a correct first attempt for this student-step for the column Correct First Attempt.

Our test-data will have the information about only the KC-description and Opportunities attribute apart from the student identification and problem hierarchy along with its name, views and step-names.



Project Flow-Diagram

**Method**

1. **Data preprocessing**

For any data mining problem, data preprocessing is one of the very crucial steps since the quality of data that is being fed into the model can affect the model performance. Hence data cleaning, transformation and handling missing values becomes important to have a good and clean dataset to feed into our algorithm. The details about the training dataset are given in the section above, so we are performing following two steps on that data as part of data preprocessing phase which would then be fed into the next stage of Feature selection to extract relevant features from the dataset.

* Data cleaning - The data is cleaned to remove redundant information like the description of a skill in the Knowledge Component to just extract the corresponding skill value. For example, a field as below,

[SkillRule: Remove constant; {ax+b=c, positive; ax+b=c, negative; x+a=b, positive; x+a=b, negative; [var expr]+[const expr]=[const expr], positive; [var expr]+[const expr]=[const expr], negative; [var expr]+[const expr]=[const expr], all; Combine constants to right; Combine constants to left; a-x=b, positive; a/x+b=c, positive; a/x+b=c, negative}]

Is cleaned to have ‘Remove Constant’ as the value in the field. Similarly, we have broken down the value of the field ‘Problem Hierarchy’ to Unit name and problem number to have a proper record of the problem that the student is solving using the tutor. For example,

Unit ES\_04, Section ES\_04-15 is broken down to ES04 as Unit and 15 as the problem number.

* Data transformation - The categorical data like Skills obtained from KC field are converted to integer values by giving every skill/set of skills a unique identifier. We are also handling missing values as part of the data transformation process. Step start time and end time are two of the fields that have null/blank values. To handle missing values for them, we are amputating the start time of the given step to be the end time of the last step that is solved by the student and the start time of the following step as the end time of the given step.

For this project, we are using the Algebra-2008-2009 dataset with information recorded for 3310 students for 9,426,966 steps overall using the tutoring system. There are total 19 features out of which 13 are the numerical and the final label is categorical.

1. **Feature Selection**

Feature selection step can help us in identifying the set of features that have the biggest impact on our final output. The objective of this step is to resolve the data dimensionality problem by reducing the high dimensional data without losing its reliability for the next step of classification(SVM) as it will work best in low-dimensional data. All the features that are provided in the training dataset are not going to be of our use, so we are considering following ways to extract features-

* Automatic feature extraction using Forward Selection by incrementally adding features one-by-one and simultaneously testing at every increment at a set P-value for significance and then select the most significant feature to be the part of the model.
* Using few features that are not provided in the test dataset out of StepStartTime, CorrectTransactionTime, FirstTransactionTime, StepEndTime, ProblemView, StepDurationsec, CorrectStepDurationsec , ErrorStepDurationsec, Incorrects, Hints, Corrects as part of hand engineered feature extraction because we belive that the time that a student takes to solve a step correctly should not be discarded right away. Also, it has been argued in [2] that not neglecting these features leads to better prediction. So, as far as the decision goes right now, we will try to integrate extracting and using these features in the SVM model that would be built in next phase.

1. **Classification**

This step would be the final step of the process where we will build a model using Support Vector Machines (SVM) to predict the performance of the given student on a given step. SVM is known to be one of the good classification algorithms to work on linear and non-linear data. So, SVM searches for a linear optimal separating hyperplane which is also called the ‘decision boundary’ to separate one class from another using support vectors. SVM is known to be less over-fitting on the amount of data that we have. Mostly, we will use the non-linear approach to make the classification because the value between 0 to 1 is acceptable for the output label for the target dataset. We have decided to use SVM because it has been successfully applied before [6] to predict student grades in academia but was not applied on the KDDCup Dataset. Since the performance of this classification technique has been comparable to Bayesian Network and Random Forest, we hope it would perform pretty well on the given dataset as well.

Firstly, we’ll check the correlation coefficient between features with the default parameters for various types of SVM kernels and find the most optimum one. Later, we will be optimizing its parameters- cost, degree, eps, gamma and loss to fine tune our model. This shall help make the conversion of selected few features into multidimensional data-points optimum in order to achieve accurate predictions.

**Timelines**

1. Data preprocessing - by Oct 31
2. Midterm report and presentation - Nov 1
3. Feature selection - Nov 2 - 7
4. Model building - Nov 8 - 14
5. Training and testing the algorithm - Nov 15 - 20
6. Fine tune algorithm for better predictions - Nov 27-30
7. Final report - Dec 4
8. Final presentation - Dec 4/6

As opposed to the work distribution before, both of the team members are working on all the steps together collaboratively with [1,3] led by Anjali and [4,5] led by Smit.

**References**

[1] Yu, Hsiang-Fu et al. “Feature Engineering and Classifier Ensemble for KDD Cup 2010.” (2010)

[2] Pardos, Zachary A. and Neil T. Heffernan. “Using HMMs and bagged decision trees to leverage rich features of user and skill from an intelligent tutoring system dataset.” (2010)

[3] W. Ham ̈ al ̈ ainen, ̈ M. Vinni, Comparison of machine learning methods for intelligent tutoring systems, in: Intelligent Tutoring Systems, Springer, 2006, pp. 525–534.

[4] S. Sembiring, M. Zarlis, D. Hartama, S. Ramliana, E. Wani, Prediction of student academic performance by an application of data mining techniques, in: International Conference on Management and Artificial Intelligence IPEDR, Vol. 6, 2011, pp. 110–114.

[5] G. Gray, C. McGuinness, P. Owende, An application of classification models to predict learner progression in tertiary education, in: Advance Computing Conference (IACC), 2014 IEEE International, IEEE, 2014, pp. 549–554.

[6] Eashwar, K. B., R. Venkatesan, and D. Ganesh. "Student Performance Prediction Using Svm." *International Journal of Mechanical Engineering and Technology* 8.11 (2017): 649-662.