DS-GA 1003: Machine Learning - Final Project Proposal

Group Name: "We are Machines Learning"

Members: Michael Higgins (mch529), Sebastian Brarda (sb5518), Felipe Ducau (fnd212)

1. The problem

Online education has already became one of the main education formats globally. It has several advantages over traditional education: lower costs, scalability and the possibility to take classes anytime at anyplace. However, it still has some disadvantages against traditional education. One of the main drawbacks, is the lack of personalization. It is very difficult to design a system that adapts to the different student profiles.

The pace at which the different students learn varies considerably across different topics and exercise types. If we could train a machine to understand how well a particular student has learned a particular topic, we could choose which is the right question to show next in order to optimize their rate of learning. In other words, the goal of the project is to predict as accurately as possible the probability that a student has to answer correctly any given question given his/her past performance on questions. If we estimated that the probability of answering correctly a certain question is very high, the problem would be too easy and showing that problem to the student would be a waste of time. On the other hand, if the problem had a very low probability of success, the problem would be too difficult, which would generate frustration without assisting learning.

In this project, we will focus on generating a good probability estimation and creating a personalization system based on these probabilities. Our goals for this project are to create a strong probabilistic model using novel feature creation and leverage the model to create an effective recommendation system. To give an example on how our recommendation system would work: after estimating the probability of succeeding in the next question, if we determine that it is too low, we would suggest which is the right question to show instead. If we find that the problem is going to be too easy for the student the system would skip it.

2. The Data

The datasets come from Intelligent Tutoring Systems (ITS) system used by thousands of algebra and pre-algebra high-school students. Originally, the dataset was used for the KDD cup, a competition[1] whose objective was to achieve the minimum Root Mean Squared Error (RMSE) in predicting probability of being successful in the first attempt of the next question.

The exercises of the system are separated into a four stage hierarchy; unit, section, problem and step. The collection of 3-5 steps comprises the problem and the last step can be considered the "answer" - the others are intermediate steps. For each step that a student attempts there is a row in our data frame that summarizes all of the students previous attempts at solving the step. This includes the quantity of **hints** requested, the **number of incorrect attempts at the problem**, the **unit** and **problem** to which the step belongs, the **start and stop time** for the students attempts, the **knowledge components** of the step along with the number of times a student has previously encountered a step that involved each knowledge component.

A knowledge component (KC) is a skill, fact or principle that is used to solve the step. One issue is the overlap of KC's, for example considering "write expression, positive slope" and "write expression, positive one slope" are considered as two separate skill sets even though they are closely related. We will investigate several methods (outlined in methods) to solve this issue.

An important thing to mention is that the original dataset was already split into training and test sets. The competitors had to upload their predictions for the test set to the webpage, and their RMSE would be populated in the leaderboard. Since we do not want to restrict ourselves to using the RMSE as evaluation metric (see "Performance Evaluation"), we will split the original training set into training, validation and test sets, preserving a similar distribution to the original datasets.

For a detailed description of the features, size of the dataset, and split methodology please review "Appendix - Dataset"

3. Performance Evaluation

Since our end goal is to estimate probabilities, we propose to use **Log-Loss** and **RMSE** as performance evaluations. Both loss functions will be minimized when the probability estimation is optimal. After training the models based on these performance metrics, we will decide which to use based on a calibration plot. **Sanity check**: Once we beat our baseline models, we will retrain our model with the full dataset (original training set for the competition) and upload the predictions for the original test set (which we are not using) to the KDD cup competition website in order to get the RMSE (they return only the metrics, but do not provide the labels of the test set). The purpose of doing that is to have a rough estimate of how well we are doing against the original competitors, but as we said, the goal is not to beat them. It is only a sanity check metric to understand if we are on the right path.

4. Baseline Algorithms

The simplest baseline model would be to predict based solely on the past performance of the student or the accuracy of all the students on a particular problem. We will use a method slightly more sophisticated than this called IRT (Item Response Theory) which combines these two simple models. IRT estimates the probability of a student answering a question correctly given their past performance and the difficulty of the question. IRT takes into account that different problems "difficulty score" is dependent on the skill of the students that attempt to solve it and the "skill score" of a given student is dependent on which problems they attempt to solve. We will fit the parameters of the model using maximum likelihood given the data.

5. Methods

Data preprocessing (03/24 to 04/07): Much of the missing time data is missing sporadically - not all the time data is missing for a given session by a student. When the start time of a problem is missing we can estimate it by taking the difference of the end and start times in the adjacent steps. In other cases we can fill the time based on the mean time (by all other students) for that step. There are also Null values for the knowledge components, we will probably treat them as

individual topics by unit (for example: Null-unit1, Null_unit2, etc.). Another important step in data preprocessing, is to create a sparse representation of the data set whereby the knowledge components (and the counts for cumulative appearances of each of them) each have their own feature as opposed to the current format that lists several skills under one feature.

Feature creation (04/07 to 04/21): Feature creation is key to obtain good results for our project. Some of the features that we are considering creating are: cumulative number of problems answered correctly, cumulative number of problems answered incorrectly, cumulative number of problems answered correctly for each knowledge component, cumulative number of problems answered incorrectly for each knowledge component, cumulative number of problems answered correctly with hint, cumulative number of problems answered incorrectly with hint, number of time spent in the problem (sec), student type (created with complementary model), problem type (created with complementary model). Afterwards, we will try creating combinations of these features; for example it would be interesting to create a ratio of answered_correctly / total_answered by knowledge component. We also plan to add features by creating "Complementary Models" (mentioned later).

ML algorithms to be used (04/27 to 05/05): We will initially consider 3 different models for our prediction. Keeping in mind that our main goal is to predict probabilities we have to restrict ourselves to ML algorithms that work for that purpose. One other thing to take into account at the time of choosing the model is the fact that the dataset is big (8M rows for one and 20M rows for the other). Also we expect our feature space to increase, and probably be sparse. Therefore we have to be careful about computational issues.

After some research on the topic we decided to start with Logistic Regression, Random Forest and Bayesian Networks. This last one is considered mostly because it was the main line of research in educational systems for several years now and because we believe we could gain insight by estimating the parameters. These parameters can be also used as features for other models.

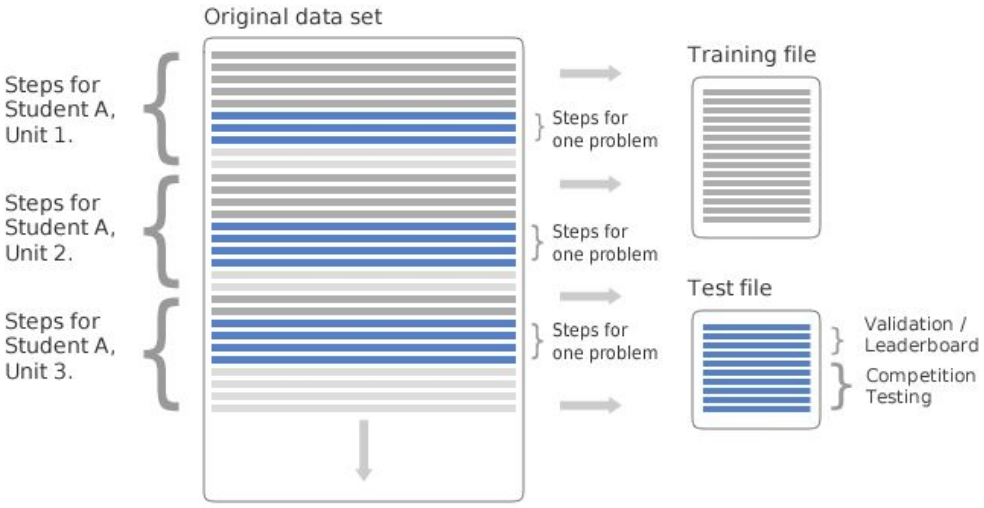
Complementary models (04/13 to 04/27): We would like to build a knowledge graph based on the attribute knowledge components. There are over 500 categories for knowledge but there is no data given as to how these pieces of knowledge relate to one another. Since many of these topics are highly related, for example: *factor-quadratic-neg-const*, *factor-quadratic-pos-const* we would like to perform dimensionality reduction. We would like to combine the words used to describe the skill as well as the performance of the students under these topics to arrive at a more condensed set of knowledge components. We will use this graph to aid in our recommendation system as well as be a tool in solving the original predicting probability problem. By grouping (using K-means) the students based on their accuracy on different topics and adding the students group as a feature of our training set we hope to gain generalization power.

Appendix - Dataset

Attributes

- **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: each problem consists of one or more steps (e.g., "find the area of rectangle ABCD" or "divide both sides of the equation by x"). The step name is unique within each problem, but there may be collisions between different problems, so the only unique identifier for a step is the pair of problem_name and step_name.
- **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. (Only increases if the step is encountered more than once.)
- **KC(KC Model Name)**: the identified skills that are used in a problem, where available. A step can have multiple KCs assigned to it. Multiple KCs for a step are separated by ~~ (two tildes). Since opportunity describes practice by knowledge component, the corresponding opportunities are similarly separated by ~~.
- **Opportunity(KC Model Name)**: a count that increases by one each time the student encounters a step with the listed knowledge component. Steps with multiple KCs will have multiple opportunity numbers separated by ~~.
- **Additional KC models**, which exist for the challenge data sets, will appear as additional pairs of columns (KC and Opportunity columns for each model).

Training and Test Split



Source: *https://pslcdatashop.web.cmu.edu/KDDCup/rules_data_format.jsp*

Size of the dataset and missing values

We will use the original training set as our full dataset. Each observation/row corresponds to one interaction (step) of a student with a certain student.
Number of observations = 8.918.054
Number of features = 23

The following features have Null values. We list the feature, the number of Nulls and the % that the Nulls represent in the dataset.

Step Start Time: 265516, 2.98%
Correct Transaction Time: 238090, 2.67%
Step Duration (sec): 442921, 4.97%
Correct Step Duration (sec): 1641028, 18.4%
Error Step Duration (sec): 7719947, 86.56%
KC(SubSkills): 2475917, 26.76%
Opportunity(SubSkills): 2475917, 26.76%
KC(KTracedSkills): 4498349, 50.44%
Opportunity(KTracedSkills): 4498349, 50.44%
KC(Rules): 322051, 3.6%
Opportunity(Rules): 322051, 3.6%

References

[1] https://pslcdatashop.web.cmu.edu/KDDCup/