Machine Learning Engineer Nanodegree

Capstone Proposal

Fedor Smirnov

Proposal

Domain Background

My name is Fedor Smirnov and I am currently pursuing a Phd degree in Computer Science at the University of Erlangen-Nuremberg in Germany. Beside their research work, all phd students who work in our department also participate in teaching activities by giving exercise courses where the students work on problems that help them to understand and learn computer science topics necessary for their studies. Although preparing these courses takes quite some time and does not directly help me with my research work, I enjoy the teaching activities very much.

One thing that bothers me about the teaching organization is that all the people responsible for the teaching (the professors and the phd students) do not have any pedagogic background or education. Nevertheless, the phd students have to come up with exercises that can be used to effectively convey the topics addressed in the courses. I myself find it very hard to estimate how well an exercise that is organized in a certain way is suited to teach concepts to the students.

As my capstone project in the machine learning nanodegree, I would like to create a framework that can be used to evaluate the effectiveness of an exercise and (by altering the input) provide hints how altering the structure of the exercise (for example the number of steps, the number of problems that has to be practiced or the number of hints that the tutor should provide

for specific steps) can improve its effectiveness. When I am talking about the effectiveness of an exercise, I mean that an exercise is effective if at the end of it, the students have learned to handle the underlying problem and would be able to solve the problem correctly without requiring any help if they encounter the problem again.

Problem Statement

The framework that I want to create will take a description of the exercise that is being evaluated (number of steps, skill level of the students in the class, number or kind of knowledge components necessary to solve the problem) and classify the exercise as effective or not effective. If an exercise is effective, the students who have finished it (in the way specified by the input, for example through hints from the tutor) will be able to solve this problem correctly and without help when they encounter it the next time. Otherwise the exercise is considered to be ineffective.

Data sets and Inputs

The data set that I want to use for this project is the data set used during the KDD Cup 2010 Educational Data Mining Challenge:

Stamper, J., Niculescu-Mizil, A., Ritter, S., Gordon, G.J., & Koedinger, K.R. (2010). Algebra I 2008-2009. Challenge data set from KDD Cup 2010 Educational Data Mining Challenge. Find it at http://pslcdatashop.web.cmu.edu/KDDCup/downloads.jsp.

This set documents the learning process of students using an online program to learn algebra concepts. I think that this set is well suited to be used for my problem as I assume that learning mathematics and computer science is rather similar and as the teaching situation is not that different. Similarly to the online learning platform, we confront our students with problems that they have to solve (ideally by themselves). We divide the problems into substeps and monitor the students while they solve the

exercises, providing feedback when the solutions are incorrect and hints if the students get stuck.

I plan to use the data set to train the classifier on the question how the structure of the exercise affects its effectiveness. Of course, the tutors in our exercises do not have all the data that can be found in the data set. Consequently, one of the main parts of the project will be the creation of abstract categories that can be intuitively used by the tutors to describe the exercise under test (for example, instead of using concrete problem names, I would like to have the tutor perform a categorization of the problem by stating whether it is a hard, an average or an easy problem).

Solution statement

As the solution to the problem described above, I would like to create a tool that can be directly used to judge whether an exercise which can be described by inputs that an exercise tutor is able to provide is suited to teach the students (which are again described by the tutor) the knowledge components that the exercise contains. As most of the information about the data and the students will not be available for the tutors, I would like to let them describe both the students and the exercise components (that is the problem, the steps and the underlying knowledge components - I am not yet sure which of these is best suited to be used) by a subjective categorization. The students can hereby be described as gifted, average or less able while the exercise components are described as hard, average or easy. To be able to implement the tool in this way, I will first have to cluster the data appropriately to map the students and the problem components onto the aforementioned categories. The second part of the problem solution will be the training of a classifier that takes the described inputs (the subjective description by the tutors as well as quantifiable parameters like the number of steps or the number of times that the students encounter the problem during the exercise) and classifies the exercise as an exercise with a high learning effect (output is 1) or as one with a low learning effect

(output is 0). To create the labels for the training phase, I intend to consider an exercise (a problem) as effective, if a student was able to solve it correctly and without hints during the next time when he/she saw the problem.

Benchmark model

As the solution consists of multiple parts I also need different benchmarks to evaluate the models. To evaluate the mapping of the students onto the performance categories, I would like to compare the results of the clustering against a classification that I would perform by just creating a list of the students ordered by their average correctness rate and dividing the list into three parts with an equal number of students. A similar approach can be used to create the benchmark for the problem components (here, we can use the average correctness of the problem, that is the number of times it was solved correctly divided by the number of solution attempts). As a benchmark for the classification of the exercises, I would like to use a predictor trained with all the data that is available. Here, I would apply PCA to reduce the number of input features and then use a supervised learning technique to train the model. As the creation of the subjective categories usable by the tutors is likely to come with quite some information loss, I would expect this benchmark model to perform better than my tool and to act as a sort of upper bound.

Evaluation metrics

As the tool I want to create is based on an A/B-classification, I think that using the classification accuracy as the evaluation metric makes sense. To see whether the misclassifications favor a side (whether the tool rather tends to classify effective exercises as ineffective or the other way round) I will also have a look at the precision and the recall values.

Project design

Parts of the problem: From my current point of view, the solution to the problem can be divided into three parts where three classifiers are created. In the first part, I want to create a classifier that assigns a class to a data set representing a student. The class hereby represents how gifted the student is. In the second part of the problem, I would like to implement a classifier that gets the problem as input and classifies it as an easy, a medium or a hard problem. The third part of the problem solution is then the creation of a classifier which will take a data representation of a student and a problem and classify this learning situation (the combination of a student and a problem) as a situation with high or with low learning effect. **Note:** At this point I am not sure whether I should create the labels describing the training effect based on the problem or on the knowledge components found within. Using the knowledge components would make more sense from the practical point of view, as the purpose of the exercise is to learn the concepts that the problem is based on rather than the problem solution itself. However, considering the knowledge components introduces a lot of additional questions (How to handle problems which target multiple components? Is the supposition that if a student has learned a knowledge component, he will never make mistakes in exercises based on this component correct? etc.). For now the first approach will be based on the problems rather than the knowledge components. I do however intend on trying at least one approach based on knowledge components at a later point in the project when I have a better feeling for the data set.

General preprocessing of the data: Before deciding which features I will be using for the solution of the three problem parts, I will have to check whether the features can be found in all or at least most of the data samples.

Workflow problem one and two: As both problems require clustering, the approaches that will be used for them are similar. In both cases, I will start by applying soft (Gaussian) clustering algorithms. For the distance function,

I will use different combinations of attributes that logically correlate with the classification I want to make (e.g., for the classification of students, qualities like the average time to process the exercises or the average fraction of correct answers are likely to be relevant). Before applying clustering algorithms, I will normalize the chosen attributes. I will evaluate the results of the clustering with the approach I described in the benchmark model section.

Workflow problem three: For the classification problem I will try using multiple different classifiers and implement a structure to visualize their learning curves. As the amount of data is limited, overfitting will be a concern as I do not have the option of acquiring more data. I will compare the different learning algorithms to each other and to the benchmark model I described in the benchmark section.