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.

Related Work

Searching for ways to provide more effective learning experiences for students who learn in classroom environments has been a popular research topic, especially after the publication of [1]. There, the authors present evidence that, depending on the learning environment, students of the same performance level can differ in their mastery of the taught subject by as much as two Sigma, i.e., two times the standard deviation.

Subjects where the classes can be taught via virtual tutoring programs are especially interesting for this area of research, as they offer an opportunity for a relatively easy acquisition and detailed analysis of huge amounts of data. The authors of [2] and [3] show how the data sets of these problems can be used to make the learning more effective (meaning that a skill can be mastered with a smaller number of exercises) and to optimize the schedule in order to prevent under- and over-practicing.

An experimental evaluation of each possible adjustment of the teaching process would be extremely time consuming. Consequently, a model for the prediction of the student performance is a key component of approaches for the optimization of teaching systems. This model takes a description of the students and the exercise and provides an estimation of the student performance, hereby enabling the implementation of the optimization loop illustrated in Fig. 1.

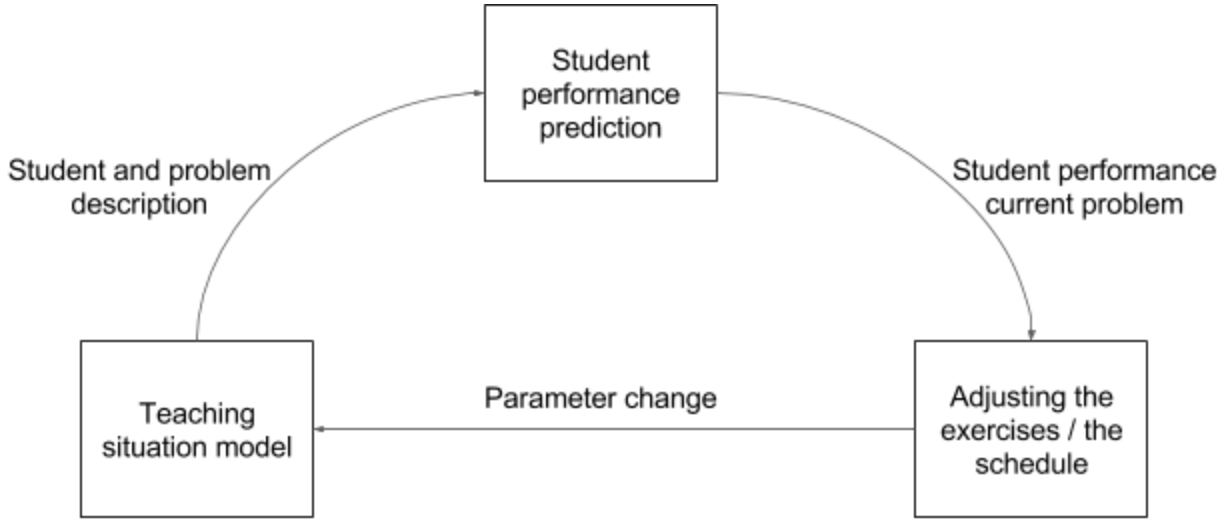

Figure 1: A model for the prediction of the student performance can be used for an iterative optimization of the teaching situation.

Here, the effect of each change of the teaching situation, like using a different exercise structure or presenting the exercises in a different order, on the investigated objective function, like the general performance of the students or their learning rate, is estimated with the help of the performance prediction. The teaching situation can then be iteratively changed into the direction of an improving objective function.

As a basis for the student performance prediction, most state-of-the-art approaches rely on the so-called *Learning Factors Analysis (LFA)* model. There, the prediction of the student performance is done based on the following equation [4]:

$$log\left(\frac{P_{ijt}}{1-P_{ijt}}\right) = \sum \theta_i X_i + \sum \beta_j Y_j + \sum \gamma_j Y_j T_{jt}$$

Equation 1: Calculation of successfully completing a problem step

In this equation, the sum of the overall “smarts” (i.e., strength in the different skill fields) of the student (θ_i), the “easiness” of a knowledge component (KC) and the amount of experience gained (γ_j) for each practice opportunity is used to calculate the probability that the i-th student

will get a problem based on the j-th KC right when presented with the t-th opportunity to practice the KC. When applying this model, a key assumption is that the solution of each problem can be seen as an opportunity for the application of one or multiple KCs. A KC is hereby a generalization of all pieces of information that can be used for the accomplishment of a task, like concepts or skills. Identifying the KCs that the given problems are based on then becomes the main challenge for the creation of the student performance model.

Figure 2 illustrates the state-of-the-art approach that is used for the creation of a student performance model.

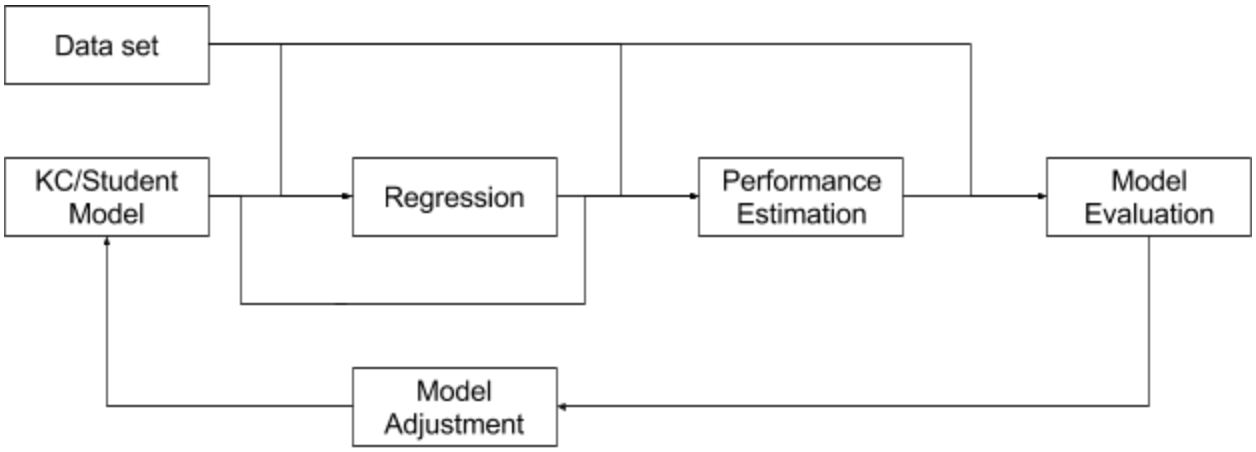

Figure 2: Iterative optimization. In the first step, a regression model is used to fit the current KC model to the data by adjusting the weighting factors for the students' smarts and the KCs' easiness. This fitted model is then used for the estimation of the investigated objective like, e.g., the learning rate. The difference to the objective calculated using the actual data set (the error rate of the current KC/student model) is then iteratively minimized by adjusting the KC/student model.

A KC/student that provides a small error relatively to the real data set enables a precise estimation of the student performance and can, consequently, be used for the estimation of the effect that certain changes of the teaching situation will have on the student performance.

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 and/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.

The framework that I want to create has two key differences to the state-of-the-art approach presented in the last section. On the one hand, I intend to apply a model trained with certain data (solving algebra problems) for the performance evaluation of problems from an entirely different field (computer science). On the other hand, I want this framework to be usable for tutors not having knowledge about the KC- or student models. The framework consequently has to work with inputs that can be intuitively provided by the tutors, rather than the features found in the data sets used for the training of the framework's predictions models.

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 [5].

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 relatively 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.

A short description of the attributes found in each entry of the data set can be found in the file attribute_description.pdf (in the git repository) as well as at the data set website.

The data in this data set is provided in two data sets. Each of the two data sets (“algebra” and “bridge to algebra”) consists of a training and a test set.

The training and the test set differ in that the training set provides more information. The test set contains only the information on the knowledge components contained in the problem, the number of the problem view and the opportunity count for each of the contained KCs. As it does not contain the information about the correctness of the step performed by the student, it can not be used as a test set in the classical sense, but rather as an input for the trained prediction model.

The training set on the other hand does contain a lot more information. Over 99 percent of the data entries provide information about time for the correct transaction, the time for the step, and the time for the correct or the erroneous step. All the data entries provide information about the correct first attempt, the time when the first transaction was taken and the time when the step ended.

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, and a KC model, 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).

Considering the fact that the problem I will be solving consists of three separate problems (clustering of the students, clustering of the problems/steps, and training the prediction model for the learning effectiveness of the given exercise), I am not 100% sure which attributes I will be using. An initial analysis of the data showed that the training sets contain information on all but the KC-related attributes in most of the entries (see the generated files fraction_trainA.txt and fraction_trainB.txt in the documentation folder of the git repository listing the fraction of the entries where each of the attributes is set). Please refer to the Solution statement section for my initial assumption about the parameters I will be using for the three subproblems.

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. A schematic illustration of the functionality of the framework is given in Fig. 3.

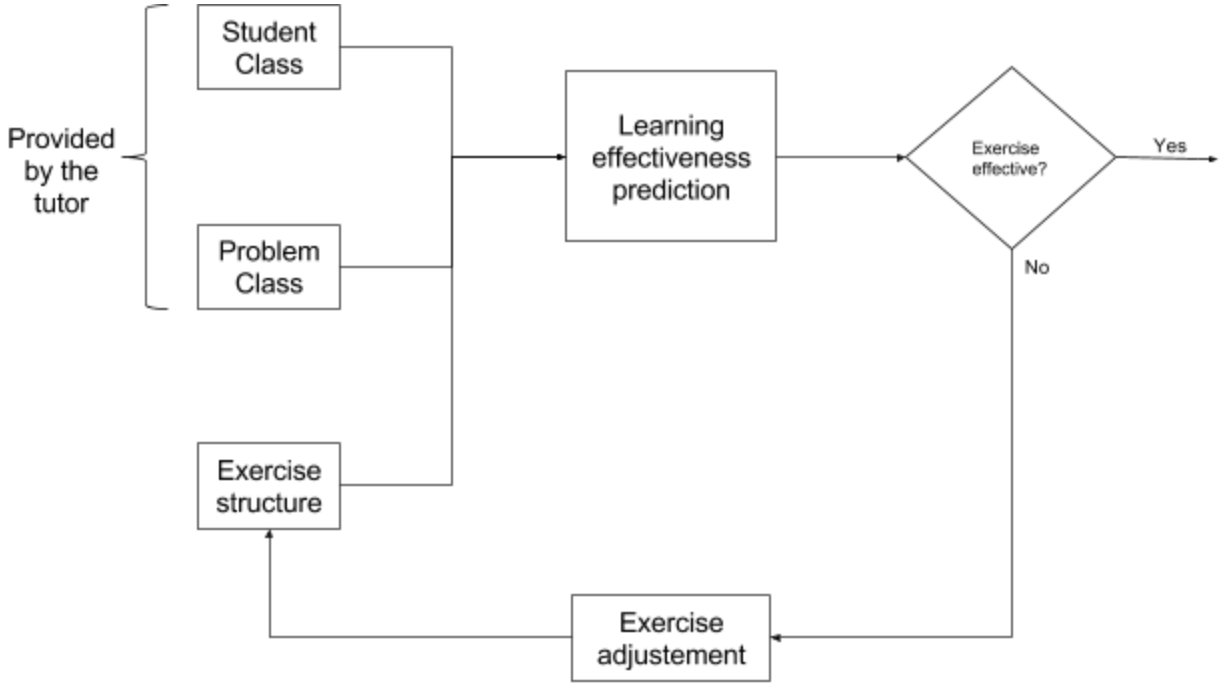

Figure 3: The tutor provides his assessment of the difficulty of the problem and the skill level of the students in the course along with the current structure of the exercise. The framework then predicts whether the exercise is effective in the current form or whether the tutor has to adjust its structure.

The **exercise (problem) structure** can be directly described by parameters such as the number of steps, the problem view number (is this a new subject or have the students seen this problem in previous courses) and the number of times the students have seen the underlying KCs (have the students solved different problems that targeted the same concepts?).

The **student class** is the subjective estimation of the skill level of the majority of the students of the course. While this estimation will be easy to obtain when using the framework (the tutor just has to pick a grade that feels right from his/her perspective), the first part of the project will be the clustering of the students in the data set and assigning skill labels to them. I intend to approach this problem by applying a clustering algorithm. During the first step, I will process the data in the given data set and create a student-centered data set. There, each student will be represented by a single entry. At the moment, I am not yet sure whether I will summarize the data for the individual steps or not (by summarizing I mean, i.e., using the

average correctness on all exercises instead of using an own feature for the correctness of every step/problem). Not summarizing the steps is probably preferable to prevent information loss. An important step, however, will be to analyze whether each student in the data set was working with each problem for the same number of times. If this is not the case, I will probably have to apply some kind of summarization in order to be able to use the data from all students for the clustering. For the clustering of the students, I intend to use features that provide information about the correctness of the performed steps and the time needed for the solution of the step as well as the number of hints that was needed. After applying min/max scaling on this features, I will use soft clustering to cluster the students, hereby experimenting with different numbers of clusters to find the number where the confidence for the assignment of students to the clusters is highest. After evaluating the clustering by comparing it to the benchmark described in the benchmark section, I will use it to create skill level labels for the students, which I will then write back into the original data set.

The problem class is the input that the tutor provides to describe the problem at hand. Similarly like the student class is a label for the skill level of the students, the problem class is a label for the difficulty of the problem. The generation of this label will also be similar to the generation of the student label. The first step is the creation of a problem-centered data set, where each problem is represented by a single entry. The features that are relevant for the difficulty of the problems are the features that describe how often the problem was solved correctly as well as the features that describe how much time the students needed to come up with a solution. Additionally, I will try to apply the provided information about the underlying KCs. On the one hand, I think that a higher number of underlying KCs makes the problem more difficult. I think I may also try to do a two step clustering, where I first cluster the KCs (to get a feeling for their difficulty and the amount of experience gained per opportunity, see Equation 1) and then use the hereby generated labels for clustering the problems into

different classes of difficulty. For the clustering itself, I intend to use the same overall approach as for the student class problem. As a result of this step, I will have difficulty labels that I can write into the original data set.

The learning effectiveness prediction model will be created by training a classifier that takes the described inputs 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. For this step, I will try applying different classification algorithms from the sciKit learn library and see which one has the best performance.

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.

References

[1] Bloom, Benjamin S. "The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring." *Educational researcher* 13.6 (1984): 4-16.

[2] Cen, Hao, Kenneth R. Koedinger, and Brian Junker. "Is Over Practice Necessary?-Improving Learning Efficiency with the Cognitive Tutor through Educational Data Mining." *Frontiers in Artificial Intelligence and Applications*158 (2007): 511

[3] Stamper, John, and Kenneth Koedinger. "Human-machine student model discovery and improvement using DataShop." *Artificial intelligence in education*. Springer Berlin/Heidelberg, 2011.

[4] Draney, Karen L., Peter Pirolli, and Mark Wilson. "A measurement model for a complex cognitive skill." *Cognitively diagnostic assessment* (1995): 103-125.

[5] 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.