Using knowledge components for collaborative filtering in adaptive tutoring systems

Peter Halkier Nicolajsen
University of Copenhagen
srb816@alumni.ku.dk

Barbara Plank
University of Copenhagen
bplank@cst.dk

ABSTRACT

In adaptive tutoring systems, accurately assessing the ability of a student is central to prescribing the tasks that best facilitate learning. For the 2010 KDD Cup challenge a data set of logs from the Cognitive Tutor system was made available, and contestants were asked to predict the correctness of a student's attempt to answer questions. A successful approach included a collaborative filtering system which predicted student performance on the basis of the performance of similar students. In this paper, we present an extension of this approach. Rather than finding similar students on the basis of their performance on specific questions, we based our similarity measure on the performance on questions that require the same "knowledge components" (or skills). This approach increases the amount of users with whom it is possible to compare performance, which in turn increases the likelihood of finding similar students. The experiments using our question type-based distance measure yield promising results.

Keywords

Adaptive tutoring systems, collaborative filtering, distance measure

1. INTRODUCTION

Education is getting more expensive, a reason for this is that the spread of technology-based improvements in productivity have been very limited compared to other industries. If technological advances allow the same amount of labor to be more productive, that production will become less expensive. Education is a sector where the amount of *output* (i.e., students taught) per hour of teacher labor has been relatively constant. This means that relative to sectors with more technology-based productivity gains–most sectors–education becomes more expensive. In economics this is referred to as *Baumol's cost disease* [3].

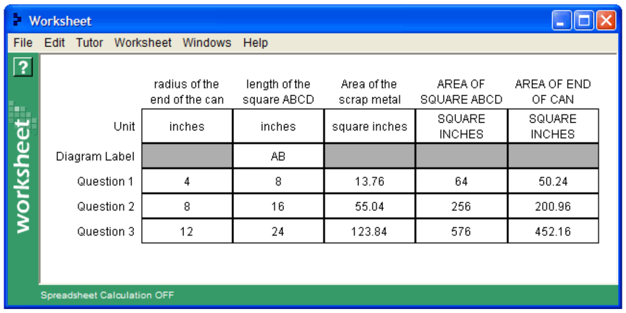
Assessment is an element of teaching that is amongst the most labor intensive and thus calls most for technological advancement. In order to give appropriate feedback, it is necessary for a teacher to have an accurate, and up to date picture of the ability of the students. Assessing students requires attention, which naturally limits the number of students that can be effectively supervised. If assessments could be made more efficient, more of the teacher's time could be spent giving appropriate feedback. An educational technology that is based on this idea is the adaptive tutoring system (ATS). An ATS is a platform that delivers educational materials (e.g. lectures, problems etc.) while assessing the student and—as the student uses the system—adapting the material to best suit each student. One instance of an adaptive tutoring system is Carnegie Learning's Cognitive Tutor. This system is based on the ACT-R model of cognition [1, 2]. Logs of interactions with this system for Algebra courses were made available in the 2010 KDD Cup [7], where the task was to predict student performance based on logs of previous interactions. If we use performance as a proxy for ability, a more accurate performance prediction corresponds to a better ability assessment.

In this paper, we propose to extend the work of Töscher & Jahrer [9] (referred to as TJ). Part of their solution was a k-nearest neighbor system that predicted scores based on a weighted average of the 41 most similar students. Here, we propose using a different distance measure, by looking at the students with highly correlated performance scores on similar problems, rather than on identical problems.

2. ADAPTIVE TUTORING SYSTEMS

The Cognitive Tutor is an adaptive tutoring system that provides practice for different subjects. The system assigns specific problems for the user to rehearse on. The student's performance on these problems then allows the system to suggest the appropriate level of additional problems. Figure 1 shows an example screenshot from the system.

The Cognitive Tutor has been developed on the basis of the ACT-R model of cognition [1, 2]. There are two elements of ACT-R that are particularly relevant to learning. The first element is the idea that all complex knowledge is the combination of smaller, discrete, pieces of knowledge, so-called *knowledge components* (KCs). The second element is that a student improves a KC by rehearsing it often and in different contexts. When using the Cognitive Tutor, a student will acquire some complex knowledge by incrementally rehearsing each of the required KCs. Any subject for which


Worksheet
File Edit Tutor Worksheet Windows Help
?
worksheet
Spreadsheet Calculation OFF

	radius of the end of the can	length of the square ABCD	Area of the scrap metal	AREA OF SQUARE ABCD	AREA OF END OF CAN
Unit	inches	inches	square inches	SQUARE INCHES	SQUARE INCHES
Diagram Label		AB			
Question 1	4	8	13.76	64	50.24
Question 2	8	16	55.04	256	200.96
Question 3	12	24	123.84	576	452.16

Figure 1: Screen shot from Cognitive Tutor. Each field is one step, while each column consists of three steps that share one knowledge component.

a Cognitive Tutor is implemented, must first be subject to a decomposition analysis, where subject experts identify all of the required KCs, and arrange them in a hierarchy based on the order in which they must be learned. The idea is that, once the subject has been mapped, the student will then be assigned problems that rehearse KCs appropriate to the current level of the student. To assess the student's level, the system keeps track of whether or not the student is able to consistently and correctly, solve problems associated with each KC, in different contexts. For example, being consistently able to solve $4 + 3$ is not the same thing as being consistently able to solve single digit addition. In the course of a session, the system will thus assign several different problems, that require the same KCs.

3. COLLABORATIVE FILTERING FOR ATS

Next, we outline the collaborative filtering approach that was part of [9] (ranking 3rd in the competition) and our proposed extension.

3.1 Töscher and Jahrer

Töscher and Jahrer [9] adopted a collaborative filtering solution, used in the field of recommender systems (e.g., Netflix challenge), and adapted it to the KDD cup challenge. Conceptually the challenges have similarities. The task in the Netflix Prize competition was to recommend movies based on ratings that different users would give different movies, based on the other movies they had rated. The KDD Cup task also required assigning values to different items (steps) for different users based on their previous data. Given these similarities, they proposed a user-based collaborative filtering approach based on the k-nearest neighbor algorithm with correlation shrinkage, described next.

The k-nearest neighbor algorithm found the 41 most similar students for each student, based on how correlated their results were on the basis of the steps they had in common. The prediction is then made on the basis of this group of similar students by using a weighted average (see details below). The stronger a neighbor correlated with the student, the more weight was given to their contribution to the prediction. If the correlation with the whole group was not very strong, the prediction would be corrected toward the student's own mean score. Despite creating the groups based only on correlations in the correct first attempt rate for the different steps, this classifier reached good performance.

The distance measure TJ used was Pearson correlation. Because there was a lot of variation in how many steps each pair of students had in common, the correlation value was transformed to reflect the support for each correlation, giving higher value to correlations based on more common steps. For the sake of consistency, we will use the same terminology as TJ in the algorithm description. They use the terms *students* and *items* to describe the main elements of the model. The items here are the step names. The students s are in the set $\mathbb{S}$, while the steps i are in the set $\mathbb{I}$. The variable to be predicted is whether a student s answered correctly on the first attempt at a step i, is called c_{is}, while the predicted value for this is $\hat{c}_{is}$.

To find the most similar students, the Pearson correlations are calculated between all pairs of students for the steps that both students s_1 and s_2 have answered. The set of steps for s_1 is $\mathbb{I}s_1$, so the set of common steps is $\mathbb{I}s_1 s_2 = \mathbb{I}s_1 \cap \mathbb{I}s_2$. Then, the Pearson correlation ρ between s_1 and s_2 is given by:

$$\rho_{s_1 s_2} = \frac{\frac{1}{|\mathbb{I}_{s_1 s_2}|} \sum_{i \in \mathbb{I}_{s_1 s_2}} (c_{s_{1i}} - \mu_{s_1})(c_{s_{2i}} - \mu_{s_2})}{\sqrt{\frac{1}{|\mathbb{I}_{s_1 s_2}|} \sum_{i \in \mathbb{I}_{s_1 s_2}} (c_{s_{1i}} - \mu_{s_1})^2} \sqrt{\frac{1}{|\mathbb{I}_{s_1 s_2}|} \sum_{i \in \mathbb{I}_{s_1 s_2}} (c_{s_{2i}} - \mu_{s_2})^2}}$$

where
$\mu_{s_1} = \frac{1}{|\mathbb{I}_{s_1 s_2}|} \sum_{i \in \mathbb{I}_{s_1 s_2}} c_{s_{1i}}$ and $\mu_{s_2} = \frac{1}{|\mathbb{I}_{s_1 s_2}|} \sum_{i \in \mathbb{I}_{s_1 s_2}} c_{s_{2i}}$.
To account for the large variability in the number of common steps, they perform a shrinkage transformation that adjusts the correlation by scaling it to the number of common steps $| \mathbb{I}_{s1s2} |$, this transformation of the correlations is calculated as:

$$\bar{\rho} = \frac{| \mathbb{I}_{s_1 s_2} | \cdot \rho_{s_1 s_2}}{| \mathbb{I}_{s_1 s_2} | + \alpha}$$

They set the meta parameter α to a value of 12.9. In the KDD Cup paper [9] they do not describe how they obtain α, but in the Netflix Prize competition paper [8]–where they use an identical shrinkage transformation–they explain that they used a random search method in which they iterate through parameter values selected from a normal distribution, until they find the value that minimizes error. This method is also used to find the other meta-parameters K (set to 41), β (set to 1.5), δ (set to 6.2) and γ (set to -1.9). We here use the same parameters throughout the paper, and leave parameter optimization for future work.
Finally, another transformation is performed on the correlations, in order to minimize the error. The transformation uses the sigmoid function[1]: $\sigma(x) = \frac{1}{1+e^{(-x)}}$ The sigmoid function is then applied to the correlations according to:

$$\tilde{\rho}_{s_1 s_2} = \sigma(\delta \cdot \bar{\rho}_{s_1 s_2} + \gamma)$$

To calculate a predicted score, the scores of the 41 most similar students (k=41) are averaged for the relevant step. Each student's average for the step is then weighted by how strong the correlation is.

$$\tilde{c}_{is} = \frac{\sum_{\tilde{s} \in \mathbb{S}_i(s;K)} \tilde{\rho}_{s\tilde{s}} c_{i\tilde{s}}}{\sum \tilde{s} \in \mathbb{S}_i(s;K) \mid \tilde{\rho}_{s\tilde{s}} \mid}$$

where $\mathbb{S}_i(s;K)$ is the set of nearest neighbor.
The last element of the algorithm is a final correction of the

[1] Note that the original paper [9] contains a typo, describing the sigmoid function as $\sigma(x) = \frac{1}{1-e^{(-x)}}$

prediction towards the mean score μ_s of student s. This is also necessary in case there is not enough support among the neighbors to make a prediction. The β term ensures that the summed correlation to the neighbors is strong enough that the prediction can be based on it, if the correlation is 0, the prediction will simply be the average score for the student.

3.2 Extension of the approach

The system described in the following is a replication and extension of the k-nearest neighbor model described above. In contrast to the TJ model, we here propose to find similarities based on knowledge components rather than just steps. This idea can be seen as abstracting from concrete question instances to basic concepts of knowledge.

The distance measure used was the correlation between students on correct answer rates for steps sharing the same knowledge component, rather than the same step name. The fact that KCs each represent several step names, means that on average, each pair of students will have more steps on which to be compared. Referring to Figure 1, this would correspond to comparing performance on steps in the same column, rather than on identical steps. In the internal training set the average number of common steps between any pair of students is 40.66, while the average number of common KCs is 52.20. Using this distance measure can be advantageous both by expanding the number of other students with which it is possible to test correlation, and by providing a broader base of problems from which to predict a score.

The procedure for finding the neighbors is the same as above, only with the compared items being different. They are now KC names rather than step names. So the knowledge components KC are in the set $\mathbb{KC}$. The predicted value for the to be predicted CFA (cf. section 4.1) then becomes: $\hat{c}_{KCs}$. The Pearson correlations are again calculated between all pairs of students, this time for the steps that have KCs that both students s_1 and s_2 encounter. The set of KCs for s_1 is $\mathbb{KC}s_1$, so the set of common steps is $\mathbb{KC}s_1 s_2 = \mathbb{KC}s_1 \cap \mathbb{KC}s_2$ The Pearson correlation ρ between s_1 and s_2 is given by:

$$\rho_{s_1 s_2} = \frac{\frac{1}{|\mathbb{KC}_{s_1 s_2}|} \sum_{i \in \mathbb{KC}_{s_1 s_2}} (c_{s_1 KC} - \mu_{s_1})(c_{s_2 KC} - \mu_{s_2})}{\sqrt{\frac{1}{|\mathbb{KC}_{s_1 s_2}|} \sum_{KC \in \mathbb{KC}_{s_1 s_2}} (c_{s_1 KC} - \mu_{s_1})^2} \sqrt{\frac{1}{|\mathbb{KC}_{s_1 s_2}|} \sum_{i \in \mathbb{KC}_{s_1 s_2}} (c_{s_2 KC} - \mu_{s_2})^2}}$$

where
$\mu_{s_1} = \frac{1}{|\mathbb{KC}_{s_1 s_2}|} \sum_{KC \in \mathbb{KC}_{s_1 s_2}} c_{s_1 KC}$ and
$\mu_{s_2} = \frac{1}{|\mathbb{KC}_{s_1 s_2}|} \sum_{KC \in \mathbb{KC}_{s_1 s_2}} c_{s_2 KC}$. The shrinkage transformation is also changed to reflect the number of steps with common KCs:

$$\bar{\rho} = \frac{| \mathbb{KC}_{s_1 s_2} | \cdot \rho_{s_1 s_2}}{| \mathbb{KC}_{s_1 s_2} | + \alpha}$$

The correlations again undergo the same sigmoid transformation as in the case of the stepwise algorithm:

$$\tilde{\rho}_{s_1 s_2} = \sigma(\delta \cdot \bar{\rho}_{s_1 s_2} + \gamma)$$

The calculation of the predicted score is altered to use the all of the steps of the most similar students that share a KC with the to be predicted score, again weighted by each neighbor $\tilde{s}$'s correlation to s:

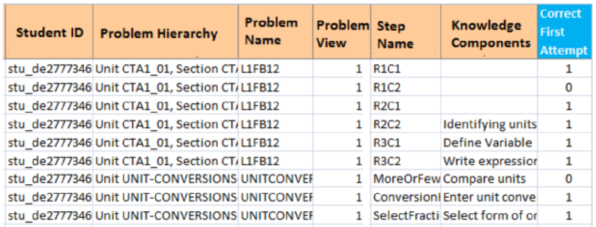
$$\tilde{c}_{KCs} = \frac{\sum_{\tilde{s} \in \mathbb{S}_{KC}(s;K)} \tilde{\rho}_{s\tilde{s}} c_{KC\tilde{s}}}{\sum \tilde{s} \in \mathbb{S}_{KC}(s;K) \mid \tilde{\rho}_{s\tilde{s}} \mid}$$

where $\mathbb{S}_{KC}(s;K)$ is the set of nearest neighbors.

4. EXPERIMENTS

4.1 KDD Cup 2010

In 2010 a large amount of log files from the Cognitive Tutor system for algebra was made available for the KDD Cup competition held in conjunction with a data mining conference. These logs contained data on interactions for more than 3,000 students over the course of a school year. Every entry was an interaction of a student with the system. For each student there was an an average of 2700 interactions.



Student ID	Problem Hierarchy	Problem Name	Problem View	Step Name	Knowledge Components	Correct First Attempt
stu_de2777346	Unit CTA1_01, Section CT/	L1FB12	1	R1C1		1
stu_de2777346	Unit CTA1_01, Section CT/	L1FB12	1	R1C2		0
stu_de2777346	Unit CTA1_01, Section CT/	L1FB12	1	R2C1		1
stu_de2777346	Unit CTA1_01, Section CT/	L1FB12	1	R2C2	Identifying units	1
stu_de2777346	Unit CTA1_01, Section CT/	L1FB12	1	R3C1	Define Variable	1
stu_de2777346	Unit CTA1_01, Section CT/	L1FB12	1	R3C2	Write expressior	1
stu_de2777346	Unit UNIT-CONVERSIONS	UNITCONVEF	1	MoreOrFew	Compare units	0
stu_de2777346	Unit UNIT-CONVERSIONS	UNITCONVEF	1	Conversion	Enter unit conve	1
stu_de2777346	Unit UNIT-CONVERSIONS	UNITCONVEF	1	SelectFracti	Select form of or	1

Figure 2: Structure of data, from [5].

The information provided in the data set (see excerpt in Figure 2) included unique identifiers for the student and the interaction, identifiers for the task, information on the success of the student on this interaction, as well as time-stamp information. Every interaction was also marked with an indicator for whether the user solved the step correctly at the first attempt (CFA). The task of the competition was then to predict the "correct first attempt" value of each student for each step, on the basis of the data describing the previous interactions with the system. The step on which the CFA was to be predicted was always drawn from an interaction occurring later than the interactions in the data set.

Since the official test set is not available, we follow standard data splitting practices [10, 4]. In the same way that the organizers had created their test set by taking the last instance of each problem, we created an internal test set by separating out the last two instances of each step within the training set to create an internal test set roughly one tenth the size of the training set. As a result of this split, any step name that occurred fewer than three times was discarded. Ultimately, that left an internal data-set of 6,596,059 training instances, with 662,074 instances in the test set. This internal set then contains 13,128 distinct steps. This also meant that some students with very few lines were discarded, which left 3,079 students.

Due to time constraints it was only possible to test the predictions on a sample of 50 students. Results for the baseline systems are reported on these same 50 students, which means that they are tested on 11,888 rows in the test set (note, results are similar to the entire data set, cf. Section 4.3). The k-nearest neighbor systems still find the 41 most similar students among all 3,079, just like the average based baselines are still calculated from all 3,079 students.

4.2 Evaluation Method

We here use the same evaluation measure as in the KDD cup, i.e., root mean squared error: $RMSE = \sqrt{\frac{\sum(\tilde{c}-c)^2}{n}}$ where $\tilde{c}$ is the predicted score, c is the actual score, and n is the number of predictions.

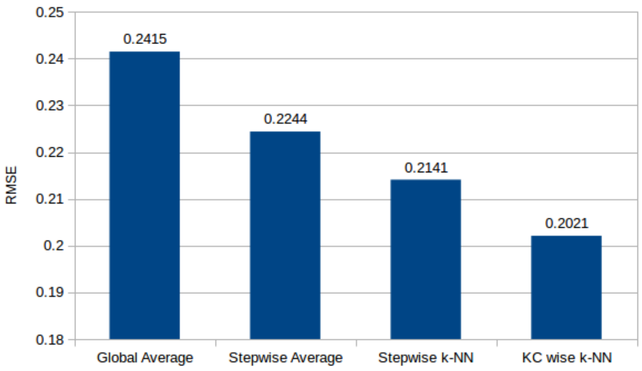


Figure 3: RMSE scores on the 50 student sample from the internal test set.

4.3 Results

Global average baseline. The first, and most basic baseline simply predicts the same score for every problem, 0.8494. The prediction is the average rate of correct first attempts, from the whole training set: $\tilde{c} = \frac{\sum c_{train}}{n}$
For first 50 students in the test set (11,888 predictions) this gave a score of: $RMSE = 0.2415$. For comparison, if we consider the entire test sets (662,074 rows), this system gave a score of $RMSE = 0.2394$.

Stepwise average baseline. The second baseline was already a clear improvement. This system distinguishes between stepnames, and uses the average score for the step in the training set to predict: $\tilde{c_i} = \frac{\sum c_{train_i}}{n_i}$
For first 50 students in the test set this gave a score of: $RMSE = 0.2244$ ($RMSE = 0.2255$ on the full set).

k-nearest neighbor (stepwise) baseline. The third baseline system is the replication of TJ's nearest neighbor system, which makes predictions by taking a weighted average of the scores on the predicted steps for the 41 students with the most similar results in the training set (cf. Section 3.1) This baseline gave further improvement on the second baseline. For the first fifty students in the development set this gave a score of: $RMSE = 0.2141$.

Knowledge component-based k-nearest neighbor system. Our expanded version of the k-nearest neighbor system also predicts on the basis of a weighted average of the scores for the 41 most similar students, but measures proximity on common steps with the same KCs rather than on common steps with the same names. For the first fifty students in the development set this gave a score of: $RMSE = 0.2021$. The results are visualized in Figure 3.

5. RELATED WORK

The 2010 KDD cup received submissions based on a large variety of approaches, many of the highest scoring system being ensemble methods such as [10] (ranking first). Another approach which also accounts for differences between students and problems combines HMMs with bagged decision trees, ranking fourth [6].

6. CONCLUSIONS

We propose to use the performance on similar steps instead of performance on identical steps as a novel distance measure in a collaborative filtering approach to ATS. So far, we only evaluated it on a reduced but reasonably large sample (11,888), but we hypothesize that the prediction error would remain low on the full set, particularly with optimization of hyper-parameters. One potential argument against using KCs is that an expert is needed to decompose the subject material and annotate the KCs. In order to provide learning material, it is necessary have a overview of what the material consists of and the order in which the different elements should be prescribed to best facilitate learning. It would be interesting to automatically learn such a structure, as in fact exploiting latent content is important for improved prediction [4]. However, the aim of this paper is to gauge whether exploiting KC information is sensible, and our preliminary results show that KCs are a potentially valuable source of information. They provide an opportunity to leverage the higher-level structure of the material to gain information about the learning process.

7. REFERENCES

[1] J. Anderson and C. Schunn. *Implications of the ACT-R learning theory: No magic bullets. Advances in instructional psychology.* Educational design and cognitive science, 2000.

[2] J. R. Anderson. ACT: A simple theory of complex cognition. *American Psychologist*, 51(4):355, 1996.

[3] W. J. Baumol. *The next twenty-five years of public choice*, chapter Health care, education and the cost disease: a looming crisis for public choice, pages 17–28. Springer Netherlands, 1993.

[4] S. Cetintas, L. Si, Y. P. Xin, and R. Tzur. Probabilistic latent class models for predicting student performance. *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, 2013.

[5] T.-N. Nguyen. *Predicting Student Performance in an Intelligent Tutoring System.* PhD thesis, University of Hildesheim, 2011.

[6] Z. A. Pardos and N. T. Heffernan. Using hmms and bagged decision trees to leverage rich features of user and skill from an intelligent tutoring system dataset. *Journal of Machine Learning Research W & CP.*, 2010.

[7] J. Stamper, A. Niculescu-Mizil, S. Ritter, G. Gordon, and K. Koedinger. Algebra I 2008-2009. challenge data set from kdd cup 2010 educational data mining challenge., (2010). Find it at http://pslcdatashop.web.cmu.edu/KDDCup/downloads.jsp.

[8] A. Töscher and M. Jahrer. The bigchaos solution to the netflix prize 2008. *Netflix Prize Report*, 2008.

[9] A. Töscher and M. Jahrer. Collaborative filtering applied to educational data mining. *Proceedings of the KDD Cup 2010 Workshop*, 2010.

[10] H.-F. Yu, H.-Y. Lo, and et al. Feature engineering and classifier ensemble for kdd cup 2010. *Proceedings of the KDD Cup 2010 Workshop*, Feature engineering and classifier ensemble for KDD cup 2010.