Clustered Knowledge Tracing

Zachary A. Pardos, Shubhendu Trivedi, Neil T. Heffernan, Gábor N. Sárközy

Department of Computer Science, Worcester Polytechnic Institute, United States
{zpardos,s_trivedi,nth,gsarkozy}@cs.wpi.edu

Abstract. By learning a more distributed representation of the input space, clustering can be a powerful source of information for boosting the performance of predictive models. While such semi-supervised methods based on clustering have been applied to increase the accuracy of predictions of external tests, they have not yet been applied to improve within-tutor prediction of student responses. We use a widely adopted model for student prediction called knowledge tracing as our predictor and demonstrate how clustering students can improve model accuracy. The intuition behind this application of clustering is that different groups of students can be better fit with separate models. High performing students, for example, might be better modeled with a higher knowledge tracing learning rate parameter than lower performing students. We use a bagging method that exploits clusterings at different values for K in order to capture a variety of different categorizations of students. The method then combines the predictions of each cluster in order to produce a more accurate result than without clustering.

Keywords: Bayesian Knowledge Tracing, Clustering, Bagging.

1 Introduction

A recent work that involved clustering of the knowledge tracing (KT) space was that by Ritter *et al.* [1]. Their work focused on clustering the parameter space of KT [2] and essentially showed that the information compression offered by clustering was enough to significantly reduce the parameter space without compromising the performance of the system. Ritter *et al.* also mention this as their motivation. It thus cannot be considered an extension to KT per se, but it raises important questions about the nature of the parameter space. Trivedi *et al.* [3] used clustering to make better out-of-tutor predictions and didn't deal with knowledge tracing at all. They clustered students based on features of tutor usage and then used those features to fit a model to predict performance on a test that students are given at the end of the school year. In our case, we cluster students based on some tutor usage features and then use these distinct clusters to train KT on them. We use a technique by Trivedi *et al.* [3] that exploits the information handed down by varying the granularity of the clustering to learn a more distributed representation.

A longer version of this paper is available online at: http://web.cs.wpi.edu/~gsarkozy/Cikkek/57.pdf

p. 1, 2012.

2 Clustered Knowledge Tracing

For each student we have a number of features that measure his/her interaction with the tutor. Students could be clustered on the basis of these features and once the groups have been found the item sequences for these groups of students could be used for training KT separately. Below we briefly review the clustering algorithms and the bootstrapping method used.

2.1 Clustering Algorithms used and Strategy for Bootstrapping

In our experiments we clustered students based on the features on tutor usage based on two algorithms: k-means and spectral clustering [4]. The basic k-means algorithm finds groupings in the data by randomly initializing a set of K cluster centroids and then iteratively minimizing a distortion function and updating these K cluster centroids and the points assigned to them. This is done till a point is reached such that sum of the distances of all the points with their assigned cluster centroids is as low as possible. Clustering methods such as k-means estimate explicit models of the data (specifically spherical gaussians) and fail spectacularly when the data is organized in very irregular and complex shaped clusters. Spectral clustering on the other hand works quite differently. It represents the data as an undirected graph and analyses the spectrum of the graph laplacian obtained from the pairwise similarities of the data-points. This view is useful as it does not estimate any explicit model of the data and instead works by unfolding the data manifold to form meaningful clusters. Usually spectral clustering is a far more "accurate" clustering method as compared to k-means except in cases where the data indeed confirms to the model that the k-means estimates. This leads to another interesting question – Which of the two works better in our scenario? This question is more interesting than just the comparison of two algorithms. If the per-user-per-skill KT parameters are arranged in approximately spherical clusters then the k-means algorithm might do better and vice versa. Note that this should happen even though we are clustering tutor usage features and not the per-user-per-skill KT parameters themselves. This is because student groupings in the feature space should correspond to the groupings found in the KT parameter space unless the features collected are irrelevant. An exploration of this correspondence could be used to collect or engineer better features. These features should also be more useful for out-of-tutor predictions as well.

Using the methodology due to Trivedi *et al.* [3] we use clustering for bagging predictors. Using the features from tutor usage we initially employ clustering to find K student groups. Corresponding to each group identified we train KT models separately, thus getting K different models (Trivedi *et al.* call each such model trained on one cluster a "cluster model"). All of these models together will make one set of predictions on the test data (all of the cluster models together for a given K are called a "prediction model" PM_K). This process is schematically described in Fig. 1. The number of clusters K is then varied and the above process is repeated iteratively from K -1 to 1 (K = 1 corresponds to KT trained on the entire dataset, this should serve as

the baseline KT). By this process we get a set of K different predictions. These predictions are then averaged to get a single final prediction.

3 Empirical Validation

In this section we present results of experiments to evaluate the performance of "Clustered Knowledge Tracing" as described above and compare it with the baseline. Both k-means and spectral clustering are used. Specifically we used the classical k-means with random initialization and for spectral clustering we used self-tuned spectral clustering with a fully connected graph of data-points.

3.1 Dataset Description

The data comes from the 2010 KDD Cup competition on educational data mining. We used the Algebra 2005-2006 and the Bridge to Algebra 2006-2007 datasets. These represent two different Algebra tutoring systems which are part of the Cognitive Tutor family of tutors [5]. The number of students in the Algebra set was 575 with 813,661 total logged responses over 387 skills. There were 1,146 students in the Bridge to Algebra set with 3,656,871 total logged responses over 470 skills. These datasets included skill information for each response and no response was tagged with more than one skill. The Cognitive Tutor divides its online curriculum into units. Skills which appear in different units, even if they have the same name, are considered different skills. Within units there are many problems which students try to solve. Each problem consists of many sub questions called steps. Steps are the level at which the responses in this dataset were logged. Our training and test set is the same as defined by the competition organizers [6]. We stick to the competition's train and test set format so that comparisons can be made between the error levels we find and the error levels of other published work with this dataset. The various tutor features that were used to cluster the students were: number of skills completed, total number of datapoints, user prior, user learn rate, user guess, user slip, number of EM iterations, Log likelihood improvement, percent correct, average response time. In experiments, students were clustered using *all* these features and also only using the user tutor features (user prior, user learn rate, user guess, user slip). These user specific KT parameters were generated like in [6] by training a separate KT model per student based on all of that student's data in the training set (across all skills).

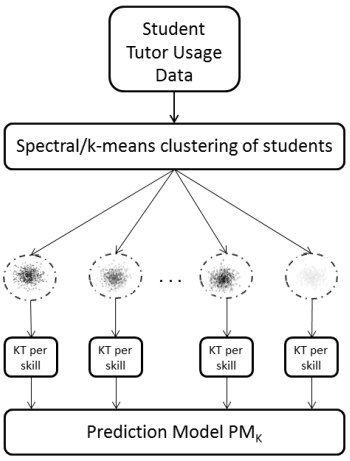

Figure 1. Construction of a Prediction Model for a given K. In each case a new PM_K is obtained and thus a prediction on the test data.

3.2 Results of the Bagging Strategy to Knowledge Tracing

For both datasets we report results using *all* the features described above and also by only using the user features. The results while using all features are with both kmeans and spectral clustering, and while using the user features are only by kmeans. We report the results for both the individuals prediction models (i.e. the model obtained by training KT on each cluster for a given K i.e. PM_K) and the ensembled results (results obtained by averaging from PM_1 to PM_K). For results we report the RMSE defined per user. The justification to use the RMSE per user is that it equally weighs the benefit to each student without biasing it to students who have contributed more data points.

Initially we tried spectral clustering for the purpose of bootstrapping. This was motivated by the fact that spectral clustering is generally better than k-means clustering as discussed in section 2.1. Fig 2 shows the results for bagging using spectral clustering considering all the features on both the datasets. We see the declining trend in error when the results are ensembled and also notice that the individual prediction models don't do too well showing that clustering alone does not help but blending the predictions does. Fig 3 indicates that a similar result is repeated in the same scenario with k-means (all features) in the algebra dataset. Such a result is not observed in the bridge dataset however. In fact in the bridge dataset both the various PM_k and the ensembled results do worse than the baseline (which is PM_1 i.e. KT trained on the entire dataset). But in further experiments we see that we can do better even on the bridge dataset if we consider only the user features. For the algebra dataset the baseline (i.e PM_1) RMSE is 0.32185, which represents standard KT with no clustering. The best result in the Algebra dataset for spectral (Fig 2) is obtained on averaging the first ten prediction models (0.31706). The best result for k-means (Fig 3) on this dataset is 0.31696, also after averaging the first ten prediction models. The result is surprising as kmeans seems to do better than spectral clustering in this case. Perhaps this might be explained by the intuition in section 2.1. The trend however is reversed in the Bridge to algebra data-set, however we still note that the ensemble using spectral clustering does better than the baseline for all the K's considered in this dataset. Given that k-means appeared to do well in one dataset and also given its speed, the above procedure was repeated in both the datasets with k-means using only the user specific features. We also cluster to a much higher K and see that the error trend line only decreases as K is increased as is shown in Fig 4. Here again, for the Algebra dataset, PM_1 has an RMSE of 0.32185. The best prediction accuracy on averaging is attained at K = 20 where the RMSE is 0.3149. This accuracy is even better as was reported earlier considering both the clustering methods indicating that the user features are much richer for clustering the students. When only the user features are considered a similar error profile is also observed in the bridge to algebra dataset too (PM_1 RMSE = 0.28397 and RMSE of the average from PM_1 to PM_{30} is 0.28225). Except for the case when kmeans was run on the bridge to algebra set considering all the features, all the improvements are statistically significant over the baseline ($p < 0.05$). In another experiment in which all the above models are combined, the best accuracy that we obtain for the algebra dataset is 0.31506 and 0.2827 for the bridge to algebra dataset.

Like we noted earlier, we report the RMSE per user. However even if we considered the RMSE on the leaderboard we get a statistically significant improvement over the baseline with PM_1 being 0.32408 and the best prediction being 0.32318.

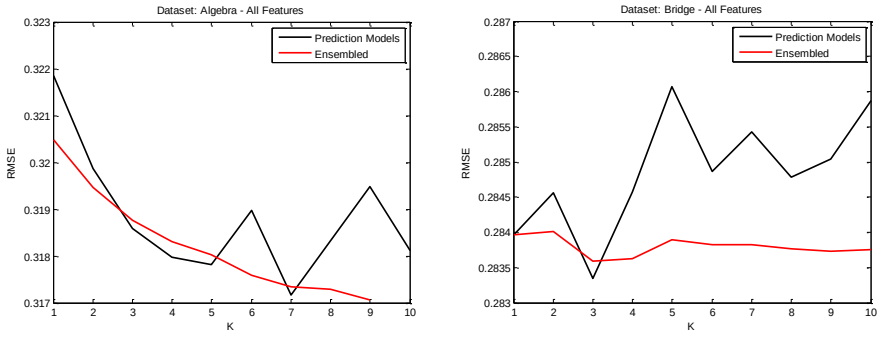

Fig. 1. Results on the Algebra (L) and the Bridge to Algebra (R) datasets with spectral clustering when all the features are considered. The red line shows the ensembled results after averaging from PM_1 to PM_K while the black one shows the results for each Prediction Model (PM_K).

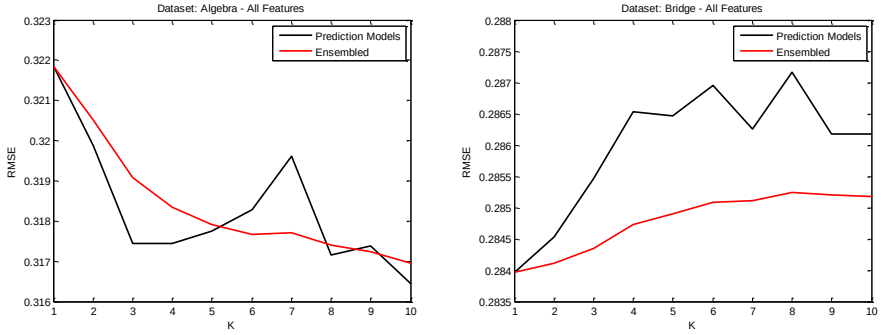

Fig. 2. Algebra (L) and the Bridge to Algebra (R) with k-means clust. considering all features.

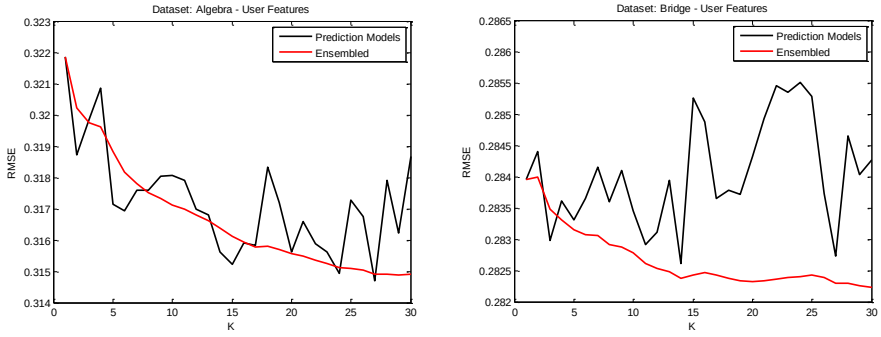

Fig. 3. Algebra (L) and the Bridge to Algebra (R) with k-means clust. considering user features.

4 Discussion and Future Work

While various extensions to the base KT model have focused on adding new features to the base model, in this work we took a slightly different view. Instead of trying to model new parameters we try to learn a more distributed representation of the KT input space. We achieve this by using clustering for bootstrapping. In extensive validation we show that our strategy indeed works very well. We report an improvement in prediction accuracy in most cases. We also report that the user features are much richer for clustering than the features of interaction of a student with a tutor. We believe that this leads to an interesting research problem. Often, the interaction of students with a tutor is measured and recorded as features. These features should be such that if students were clustered on this feature space, the clustering should correspond to one on the KT parameter space. If it is not the case then it indicates that the task of feature generation in the tutor is noisy and could be improved in a more principled manner. An improvement in methodology here would be greatly useful in getting features that would be most helpful in making better out-of-tutor predictions. An interesting problem would be to consider a case study in which the various clusters are analyzed and an attempt is made to interpret them on the basis of the associated KT parameters. Such a study could be quite useful, especially in making some data driven inferences and pedagogy. Lastly, this exploration concerning the KT input space, especially concerning learning a more distributed representation could be quite useful even when used in conjunction with KT variants such as [6] that are known to be stronger predictors than the base KT.

References

1. Ritter, S., Harris, T., Nixon, T., Dickison, D., Murray, R., Towle, B., (2009) Reducing the knowledge tracing space. *In Proceedings of the International Conference on Educational Data Mining*, Cordoba, Spain, pp. 151-160.
2. Corbett. A. T. & Anderson, J. R. (1995). Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. *User Modeling and User Adapted Interaction*, 4, 253-278.
3. Trivedi S, Pardos Z. A., Heffernan N. T., (2011) Clustering Students to Generate an Ensemble to Improve Standard Test Score Predictions, G. Biswas et al. (Eds.): AIED 2011, LNAI 6738, *In The proceedings of the 15th International Conference on Artificial Intelligence in Education 2011*, Auckland, New Zealand, pp. 377–384.
4. Luxburg. U., (2007) A Tutorial on Spectral Clustering, In *Statistics and Computing*, Kluwer Academic Publishers, Hingham, MA, USA, Vol 17, Issue 4, 2007
5. Koedinger, K. R., Corbett, A. T., (2006) Cognitive tutors: Technology bringing learning science to the classroom. In K. Sawyer (Ed.), The Cambridge handbook of the learning sciences (pp. 61-78). New York: Cambridge University Press.
6. Pardos, Z.A., Heffernan, N. T. Accepted (2011) 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, Special Issue on The Knowledge Discovery and Data Mining Cup 2011.