Spectral Clustering in Educational Data Mining

SHUBHENDU TRIVEDI, ZACHARY A. PARDOS,
GÁBOR N. SÁRKÖZY, NEIL T. HEFFERNAN
Worcester Polytechnic Institute, United States

Spectral Clustering is a graph theoretic technique for metric modification such that it gives a much more global notion of similarity between data points as compared to other clustering methods such as k-means. It thus represents data in such a way that it is easier to find meaningful clusters on this new representation. It is especially useful in complex datasets where traditional clustering methods would fail to find groupings. In previous work we have shown the utility of using k-means clustering for exploiting structure in the data to affect a significant improvement in prediction accuracy on educational datasets. In this work we show that by using Spectral Clustering we are able to further improve the student performance prediction. We evaluate an educational data mining prediction task: predicting student state test scores from features derived from a tutor and also present some preliminary results on some other EDM tasks using spectral clustering.

Categories and Subject descriptors: I 2.7 **[Artificial Intelligence]**

Key Words and Phrases: Educational Data Mining, Intelligent Tutoring Systems, Bootstrap Aggregating, Clustering, Spectral Clustering, Ensemble Learning, Mixture of Experts

1. INTRODUCTION

The highly inter-disciplinary field of Educational Data Mining (EDM) has resulted from a fusion of many different areas, some of which include Machine Learning, Cognitive Science and Psychometrics. The main task in EDM is to construct computational models and tools to mine data that originated in an educational setting. With rapidly increasing data repositories from different educational contexts (paper tests, e-learning, Intelligent Tutoring Systems etc.), good practices in EDM can potentially answer important research questions about student learning. This goal of EDM is proving instrumental in combining the knowledge derived from the data to combine with theories from cognitive psychology to formulate the best learning settings and methodologies.

Within data mining, clustering is perhaps one of the most important tools for both exploratory and confirmatory analysis. It is a technique to discern meaningful patterns in unlabeled data by grouping together data points that are "similar". In EDM, clustering has been used in a variety of contexts: Ritter *et al.* In an already influential work essentially used the implicit information compression (albeit lossy) handed by clustering to reduce the Knowledge Tracing parameter space [Ritter 09] without compromising the performance of the system. Dominguez *et al.* used clustering as a tool to generate individualized hints for students [Dominguez 10]. In another interesting work, Shih *et al.* employed clustering for unsupervised discovery of student learning tactics [Shih 10]. Clustering has also been used for curriculum planning [Maull 10], for estimating skill set profiles [Nugent 10] amongst numerous other tasks. However, interestingly most of these works employ k-means clustering, expectation maximization based clustering or subspace clustering. This paper aims to introduce to the field of EDM the utility handed by spectral clustering over other clustering algorithms, which is also an easy to implement algorithm with numerous toolboxes available as well [Chen 10].

Authors' addresses: S. Trivedi, e-mail: shubhendu_trivedi@ieee.org ; Z. A. Pardos, e-mail: zpardos@cs.wpi.edu, G. N. Sárközy, e-mail: gsarkozy@cs.wpi.edu, N. T. Heffernan, e-mail: nth@cs.wpi.edu., Department of Computer Science, Worcester Polytechnic Institute, 100 Institute Road, Worcester, MA – 01609. United States.

To understand the weakness of methods such as k-means, a useful way of looking at clustering is the following: Consider a set of K distributions, $\mathcal{D} = \{D_1, D_2 \ldots D_K\}$ such that each of these distributions has an associated weight, the collection of which is given by $\{w_1, w_2 \ldots w_K\}$ such that $\sum_i w_i = 1$. Suppose a dataset is generated by sampling these K distributions, such that a point in this dataset might be picked from distribution D_i with probability w_i. The objective of clustering methods is to identify these K distributions given a dataset. Methods such as k-means and Expectation Maximization (EM) are based on estimating explicit models of the data. While k-means finds the clusters by assuming that the set of distributions $\mathcal{D}$ that generated the data was a set of spherical Gaussians, EM algorithms in general learn a mixture of Gaussians with arbitrary shapes. More formally, k-means finds the clusters by minimizing the distortion function:

$$J(c, \mu) = \sum_{i=1}^{m} \| x^{(i)} - \mu_{c^{(i)}} \|^2 \tag{1}$$

Where μ_c is the cluster centroid to which a point x has been assigned. In spite of the great popularity of the k-means algorithm very few theoretical guarantees on its performance are known [Dasgupta 99]. In practice however, k-means performs well on data that at least approximately follows its assumption of being generated by a mixture of well-separated spherical Gaussians [Chaudhuri 09]. This, coupled with its simplicity makes it a handy tool for a data-miner. However, k-means performs poorly when these assumptions of data generation are not met, which is usually the case in real world datasets. Fig. 1 illustrates this problem by a toy synthetic dataset.

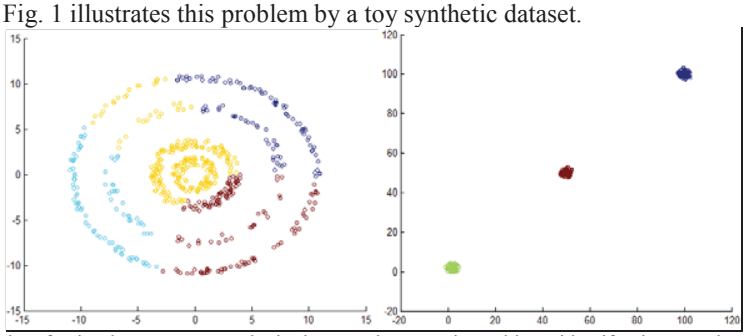


Fig 1: Results of using k-means on synthetic datasets. k-means is unable to identify clusters when the data is distributed in concentric groups (left), while it clearly finds the clusters in well separated and tight spherical Gaussians (right). The clusters identified are indicated by different colors. Both sets have 600 points.

Spectral Clustering makes no such assumptions for data generation. It instead finds groupings by analyzing the top eigenvectors of the affinity matrix and hence usually returns better results.

The rest of the paper is organized as follows: The next section discusses Spectral Clustering, giving a tutorial overview of the same. Section 3 uses the spectral clustering method to improve the prediction of post-test scores employing student features from an Intelligent tutor using a bootstrap aggregating method developed by the authors [Trivedi, Pardos 11] [Trivedi, Pardos 11]. Section 4 is a discussion of results and future work.

2. SPECTRAL CLUSTERING

One of the most important developments in Machine Learning in the past decade has been the use of spectral methods in clustering. They have created a new wave of excitement to understand the problem of clustering and the notion of similarity between points better and formulate it precisely. One major reason for this excitement is that

spectral clustering is based on solid graph theoretic principles. Given its strengths, it would be highly beneficial to the EDM community if it is used more widely in the same.

The broad idea of clustering is essentially to group points that are "similar" in one cluster and points that are "dissimilar" into different clusters. The notion of similarity that is employed in k-means is the Euclidean distance between data points and the cluster centroids to which they are assigned to (which get updated in each iteration). In a sense, the idea of similarity used in k-means restricts what could be known about the geometry of the data. In k-means we work with the data directly, in spectral clustering however, we work with a representation of the data that gives a more global (and hence better) encoding of the similarities between points. This "similarity" in spectral clustering is represented in the form of a graph called the similarity graph, represented by $\mathcal{G} = (V, E)$ where V is the set of vertices and E is the set of edges. The idea is that points in the dataset can be represented by a graph with each data point as a vertex of the graph $\mathcal{G}$ and the edges connecting them encoding a notion of similarity $w_{ij} > 0$ between them. Two points are connected in the graph if the similarity or weight between them is either non-zero or above some threshold. The clustering problem can then be re-stated using information from the similarity graph as: We want to find partitions of this graph such that weights between points in the same group are high and those between points in different groups are low. Before talking how we cluster using this representation, we introduce some notation and discuss how the graph $\mathcal{G}$ is used to represent the dataset.

Given the similarity graph $\mathcal{G}$ of n data points $\{x_1, x_2 \dots x_n\}$, there are essentially two things about it that tell us something about the global structure of the data:

1. The degree of a vertex (a data-point in our case): The degree of a vertex tells us the sum of weights of all the edges that originate from a vertex i to all other vertices j. It is given by:

$$d_i = \sum_{j=1}^{n} w_{ij}$$

This definition is somewhat non-standard but more general. The standard definition for degree of a vertex is only defined for $w_{ij} = \{0, 1\}$, and thus is only the count of vertices a given vertices is connected to. Given this definition, the degree matrix of the similarity graph is the diagonal matrix D with the degrees d_i on the diagonal.

$$D = \begin{pmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \end{pmatrix}$$

Intuitively the degree matrix of a graph tells us how many points each point is connected to (we could connect all points, or choose to connect k-nearest neighbors of each point) and by "how much" (hence the summation of the weights).

2. The weighted similarity matrix or the affinity matrix of the similarity graph, W on the other hand is a representation of similarity between all the points. Each element in the affinity matrix is given by w_{ij}, which is the weight or edge between two points i and j. A common way of representing the weight is:

$$w_{ij} = exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma^2}\right) \tag{2}$$

Notice that w_{ij} is simply the exponentiated Euclidean distance between two points (points in $\mathcal{R}^n$) scaled by a parameter called the scaling or weighing parameter σ. This parameter is to be tuned and varying it changes the weight between points. A point to note is that if all the points are connected then all w_{ij} such that $i \neq j$ will be non-zero values. If points are connected to only their k-nearest neighbors and not every other point, then most of the matrix W will be populated by zeros.

The matrices W and D tell us something about the global structure of the data, but we don't work with them directly. We instead work with the graph Laplacian matrix given by

$$L = D - W$$

The above is the un-normalized version of the Laplacian. There are two normalized versions that are represented as:

$$L_{sym} = D^{-1/2} W D^{-1/2}$$
$$L_{rw} = D^{-1} W$$

The first is called the symmetric Laplacian while the second is called the random-walk Laplacian. The Laplacian in a way combines both the degree and the affinity matrix and also has some mathematically interesting properties (such as being positive semi-definite) that make it easier to work with [Mohar 91]. Since the Laplacian is a representation of the similarity between the data-points, we can now work with it to find groups in the data.

Given the above background, clusters in a dataset can be found by the following method [Ng 01]:

1. For the dataset having n data points, construct the similarity graph $\mathcal{G}$. The similarity graph can be constructed in two ways: by connecting each data point to the other $n-1$ data points or by connecting each data point to its k-nearest neighbors. A rough estimate of a good value of the number of nearest neighbors is $log(n)$. The similarity between the points is given by equation 2. This will give the matrix W.
2. Given the similarity graph, construct the degree matrix D.
3. Using D and W find L_{sym}.
4. Let K be the number of clusters to be found. Compute the first K eigenvectors ofL_{sym}. Sort the eigenvectors according to their eigenvalues.
5. If $u_1, u_2 ... u_K$ are the top eigenvectors of L_{sym}, then construct a matrix U such that $U = \{u_1, u_2 \dots u_K\}$. Normalize rows of matrix U to be of unit length.
6. Treat the rows in the normalized matrix U as points in a K dimensional space and use k-means to cluster these.
7. If $c_1, c_2 \dots c_K$ are the K clusters, Then assign a point in the original dataset s_i to cluster c_K if and only if the i^{th} row of the normalized U is assigned to cluster c_K

It is noteworthy that we don't cluster the original dataset directly. We first transform it to find its top K eigenvectors. These being the most important eigenvectors of L, encode the maximum information about it. At the same time, this reduces the dimensionality which without throwing away much information which makes the task of clustering much easier. To illustrate the power given by this change of representation, we demonstrate it on a toy dataset (Fig. 2). A detailed tutorial that explains various spectral clustering

algorithms and some point of views on why it works is by Luxburg [Luxburg 07]. In the next section we discuss a specific application of spectral clustering in EDM.

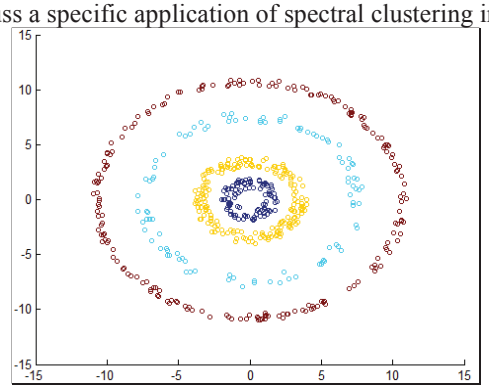


Fig 2: Result of using spectral clustering on a synthetic dataset. This synthetic set has 600 points. The colors indicate the clusters found by spectral clustering. Such groups cannot be found by k-means clustering.

3. IMPROVING PREDICTION ON STUDENT PERFORMANCE IN POST TESTS

Bayesian Knowledge Tracing [Corbett 95] has long been used to model student knowledge in an intelligent tutoring system (ITS). This knowledge estimate is used to calibrate the amount of training a student gets to ensure skill mastery. One of the goals of such modeling is to ensure that students perform well on actual post tests. In fact it is reasonable to say that perhaps one of the most important measures of success of an ITS is how well performance on it transfers to actual post tests.

Traditionally, performance on a post-test is predicted by using practice tests. The percentage of questions answered correctly on these practice tests give a crude estimate of how well a student would perform on the actual post test. Improving this estimate would be highly beneficial to both students and educators. For the improvement of such assessment, dynamic assessment [Grigerenko 98] has been advocated as an effective method. The big idea of dynamic assessment is that assessment is based on the amount of help students require to get questions correct and it enables the tutor to assess as it assists. This is a major advantage as it not only not only allows students to learn while being assessed, but can also predict student performance on post-tests better. Traditional testing, in which only the percentage of questions is considered is called static assessment. The notion of dynamic assessment makes intuitive sense as it gives a finer grained estimate of a student's knowledge. If a student gets a question wrong, it might not imply that the student has no knowledge pertaining to the question. The level of knowledge that the student has might be estimated by measuring the amount of help that the student required to get the question correct. Given the interactive nature of ITS, they are the ideal test bed for measuring the utility of dynamic assessment.

Feng *et al.* [Feng 09] reported the result that data from an ITS could better predict state test scores (MCAS or Massachusetts State Test Scores in their experiment) if it only considered the extra measures collected in dynamic assessment as compared to the static assessment condition. The paper had a weakness that time was never held constant. Feng & Heffernan went one step ahead and controlled for time in following work [Feng 10]. They reported better predictions on the MCAS state test scores by the dynamic condition, but not a statistically reliable difference. This work effectively showed that dynamic assessment led to better predictions on the post test. This prediction was done by fitting a linear regression model on the dynamic assessment features and making predictions on the MCAS test scores. They concluded that while Dynamic Assessment gave good assessment of students, the MCAS predictions made using those features were only

marginally statistically significant as compared to the static condition. Trivedi *et al.*[Trivedi 11] investigated further if the dynamic assessment data could be better utilized to increase prediction accuracy over the static condition (and hence establish the superiority of dynamic assessment). They used a newly introduced method [Trivedi 11] that clusters students using the k-means algorithm and uses multiple cluster models and then ensembles the predictions made by each cluster model to achieve a reliable improvement. Here we show that by using spectral clustering we further improve the prediction on the MCAS post-test based on the dynamic features. The improvement obtained by using spectral clustering is not only significant over the static condition, but also over results obtained using k-means after K = 3 (p-value < 0.03 on a paired t-test).

3.1 Data and Methodology

The data used for this study was the same as used by Feng *et al.*[Feng 10] and Trivedi *et al.* [Trivedi 11]. The data is from the 2004-05 school year and was collected using the ASSISTments tutor in two schools in Massachusetts. ASSISTments [Razzaq 05] is an ITS developed at Worcester Polytechnic Institute, MA, USA. The data is for 628 students and the features included the various dynamic features [Feng 10]. These features were: 1) Student's percent correct on main problems 2) Number of problems done 3) Percent correct on the help questions 4) Average time spent per item 5) Average number of attempts per item 6) Average number of hints per item. The first feature was a static feature and was used to make predictions on the static condition, while the others were used to make predictions in the dynamic condition. The prediction made was for the MCAS test scores that was available for the same students in the following year. A 5 fold cross validation was done.

The methodology used for making the prediction is a new bootstrap aggregation ensemble method [Trivedi, Pardos 11]. The procedure is summarized as follows:

1. Cluster the training data into K clusters.
2. For each cluster train a separate linear regression model using the points from that cluster as the training set.
3. Each such trained predictor (such as Linear Regression) represents a model of the cluster and is hence appropriately called a cluster model.

This is represented in figure 3 below:

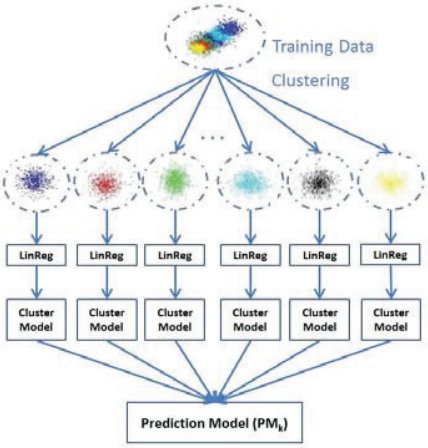


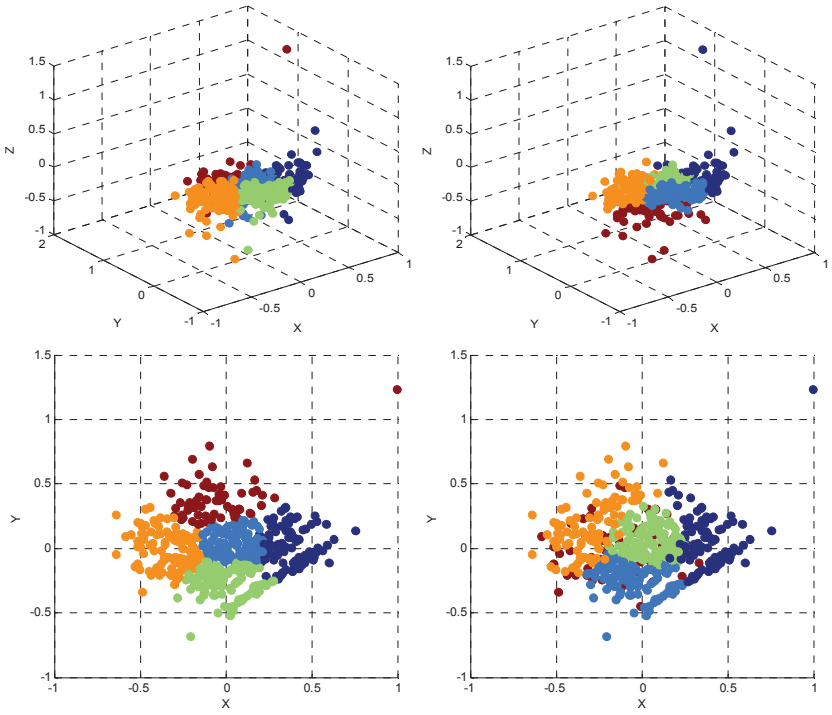
Fig 3: The first step in the methodology for using clustering to bootstrap and making a prediction on the training set. The scale of clustering can be varied to generate a number of predictions that can then be aggregated.

This collection of cluster models that make a prediction on the entire test set is called a prediction model (PM_K , the subscript denotes the number of clusters in each Prediction Model). Making a prediction for a test point would involve: Locating the cluster to which the point belongs, and then using the model trained for that cluster to make the prediction for it. However, by using the number of clusters as a free parameter we generate a set of K prediction models (PM_2, PM_3 … PM_K), such that each has a different number of cluster models. And thus, we can obtain K different predictions on the test set. These predictions are then averaged to obtain a single strong prediction.

This method can be thought of like an adaptive mixture of local experts [Jacobs, Hinton 91] that uses clustering to bootstrap. But unlike in other bagging methods, which select a random subset to bootstrap, this method has a specific expert for each cluster of the data. By varying the granularity of the clustering we are able to obtain a mixture of experts on the data at different levels each of which gives a prediction on the test set which are then averaged to get one prediction.

3.2 Results Using Spectral Clustering

The results of clustering the data using both kmeans and spectral clustering are represented in figure 4 below. Since the data is high dimensional and the actual partitions cannot be pictured, this visualization is done by doing a multi-dimensional scaling on the dataset to three dimensions with each cluster identified by a different color. This visualization is for K = 5.



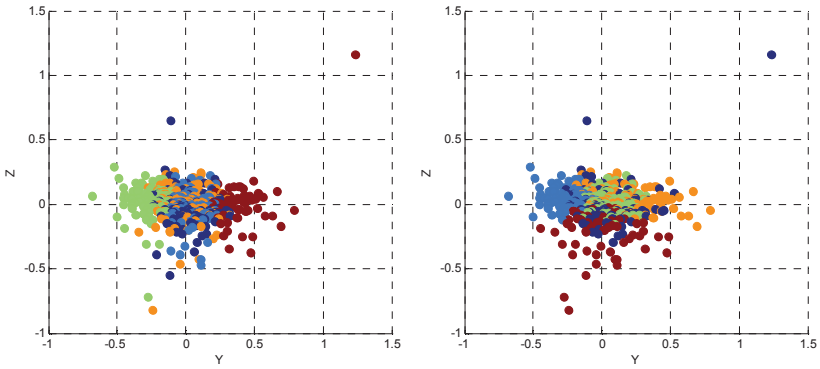


Fig 4: The images on the left column are for k-means and those on the right are for spectral clustering. The top row represents the plot of the ASSISTment data scaled down by multi-dimensional scaling to three dimensions and the clusters identified by both k-means and spectral clustering. The rows below are simply different planar views of the plot in row 1.

The spectral clustering ensemble results are not only significant over the static condition (K = 1 in figure 6) but also are significant for the kmeans generated ensemble beyond K = 3 with $p < 0.03$ in each case on a paired t-test.

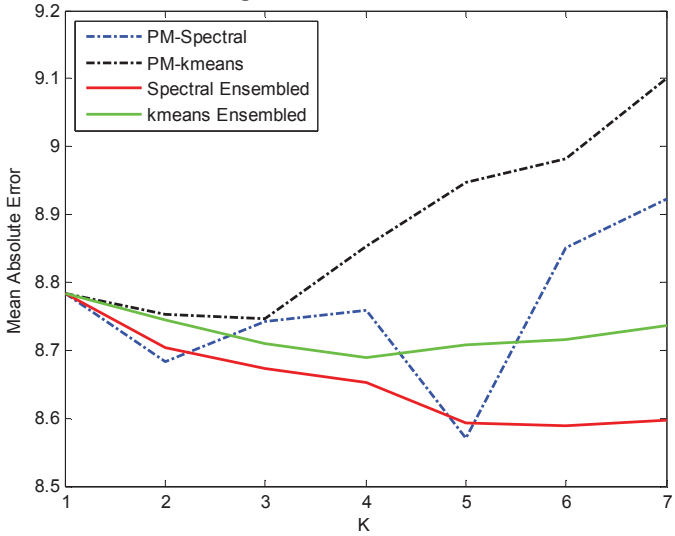


Fig 5: The plots of the 5 fold cross validated errors by the various prediction models and ensembles (from 1 to K = 7) for both kmeans and spectral clustering. The K ensemble prediction is the average of predictions returned by prediction models from 1 to K.

4. CONCLUSION, DISCUSSION AND FUTURE WORK

The methodology described in the paper was employed on some other EDM tasks as well, such as making an in-tutor prediction on the KDD Cup 2010 dataset and on the Performance Factor Analysis (PFA) task [Gong 10]. Preliminary results (summarized for PFA below) have indicated an improvement in the prediction accuracies.

Table 1: Preliminary work on Performance Factor Analysis

Spectral Ensemble	**K =1**	**K = 2**	**K = 3**	**K = 4**	**K = 5**
AUC	0.5861	0.6153	0.6252	0.6291	0.6307

The results indicate an improvement over the base condition as more prediction models are averaged. But this result is not cross-validated and is a work in progress. Also, given the prohibitive size of the dataset, spectral clustering was not used for all the rows in the training set, but a random subset of them was used. This was done to save time, however this is the reason the detailed results are not reported in this work. Also, more work needs to be done to use spectral clustering methods more efficiently for massive datasets such as the KDD cup dataset.

A deeper way of looking at clustering is essentially as a scheme for lossy data compression. Improvement in prediction accuracy using spectral clustering over k-means indicates that spectral clustering is a better information compression method than k-means and hence tells something deeper about the structure of the data that k-means misses. This would mean an interesting application to reduce the knowledge tracing space like by Ritter *et al* [Ritter 2009] and see how it compares with performance returned by k-means clustering.

The objective of this work was to introduce to the domain of EDM the great utility of using spectral clustering. We used spectral clustering to enhance the performance of a new ensemble method proposed in an earlier work by the authors. While the objective was to introduce the use of spectral clustering, a very significant result of the work is proving the efficacy of Dynamic Assessment as compared to static assessment. These results show that an ITS that can assess as it assists offers a significant advantage to students and teachers. This is important because it can not only save time that is wasted on assessment for instruction, but it can also be a better predictor of their performance in post-tests.

The results for the task of predicting the post test scores have been very encouraging; however there are some areas that need further work and could improve prediction accuracy further. One such area of possible improvement is allowing for fuzzy clustering. To make a prediction, the cluster closest to a test a point was identified and then the expert for that cluster was used to make a prediction on it. In many real world examples, membership of a data point to a particular cluster is a tricky question to answer. A more realistic view is to allow for fuzzy clustering. That is, given a test point, we determine its probability of occurring in each of the clusters. Then, we can obtain predictions by the cluster model /expert for each of the clusters and obtain one prediction for one test point that is a weighted average of the predictions returned by each cluster model (earlier a prediction was made on the test point by only one cluster model), with the weights being the probability that the point lies in that cluster. While fuzzy counterparts to the k-means algorithm such as fuzzy c-means are well known, the idea of doing fuzzy spectral clustering is something to be explored. Clearly, spectral clustering uses k-means at a lower dimensional representation of the laplacian and fuzzy c-means can be used at this level. However, the effectiveness of doing the same is not known.

Another possible area of improvement is using methods to merge clusters that are sparsely populated [Cheng 06]. By this method we could improve both the quality of clustering (if the task is purely unsupervised) and prediction accuracy (if the task like in the application discussed is a prediction task).

In this work we combine predictions by averaging them. Clearly this is a sub-optimal choice. Ideally, we would want to pick those predictions (made by prediction models) which are good in prediction and have less correlation with each other (are diverse). Since the method used to make the post-test predictions was an ensemble method, it can be used to combine the predictions themselves. Preliminary work utilizing this idea of using clustering to boot-strap the predictions returned by various prediction models has shown promise.

ACKNOWLEDGEMENTS

We are grateful to the following funders for supporting this research: http://www.webcitation.org/5ym157Yfr. We would also like to thank the Pittsburgh Science of Learning Centre for the Cognitive Tutor KDD dataset. Comments about this work by Dr Carolina Ruiz, Dr Sergio Alvarez and Dr Alexandru Niculescu-Mizil were especially helpful and are appreciated.

REFERENCES

CHAUDHURI., K., DASGUPTA S., VATTANI, A., 2009, Learning Mixture of Gaussians using the k-means Algorithm, In *CoRR vol.abs/0912.0086, http://arxiv.org/abs/0912.0086, 2009*

CHEN., W. Y., SONG, Y., BAI, H., LIN, C. J, CHANG, E. Y., 2010, (ACCEPTED) Parallel Spectral Clustering in Distributed Systems, In *IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010.*

CHENG D., KANNAN, R., VEMPALA, S., WANG, G., 2006, A Divide and Merge Methodology for Clustering. In *The Journal of the ACM, Vol V, 2006*

CORBETT A. T, ANDERSON, J. R., 1995, Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. In *User Modeling and User Adapted Interaction 4, pp. 253-278,1995*

DASGUPTA S., 1999, Learning Mixture of Gaussians, In *The 40th Annual IEEE symposium on Foundations of Computer Science. 349-358.*

DOMINGUEZ., A. K., YACEF, K., CURRAN, J. R., 2010, Data Mining for Individualized Hints in eLearning, In *Proceedings of the Third International Conference on Educational Data Mining, 2010, 91-100.*

FENG., M., HEFFERNAN, N. T, KOEDINGER, K. R., 2009, Addressing the assessment challenge in an online system that tutors as it assesses. *User Modeling and User-Adapted Interaction: The Journal of Personalization Research.* 19(3). 2009.

FENG., M., HEFFERNAN, N. T, 2010, Can We Get Better Assessment From A Tutoring System Compared to Traditional Paper Testing? Can We Have Our Cake (better assessment) and Eat it too (student learning during the test). *Proceedings of the 3rd International Conference on Educational Data Mining*, *41-50.*

GONG., Y., BECK, J. E, HEFFERNAN. N. T., 2010, (Accepted). How to Construct More Accurate Student Models: Comparing and Optimizing Knowledge Tracing and Performance Factor Analysis. In *International Journal of Artificial Intelligence in Education. Accepted, 2010.*

GRIGERENKO., E. L, STEINBERG, R. J, 1998, Dynamic Testing. In *Psychological Bulletin 124 pp. 75-111, 1998.*

JACOBS., R. A, JORDON, M. I, NOWLAN. S. J., HINTON. G.E., 1991, Adaptive Mixture of Local Experts. In *Neural Computation, Vol 3. No 1,79-87, 1991.*

LUXBURG U., 2007, A Tutorial on Spectral Clustering, In *Statistics and Computing, Kluwer Academic Publishers, Hingham, MA, USA. Vol 17. Issue 4, 2007.*

MAULL. K. E., SALVIDAR. M. G., SUMNER. T., 2010, Online Curriculum Planning Behavior of Teachers, In *Proceedings of the Third International Conference on Educational Data Mining, 2010, 121-130.*

MOHAR. B., The Laplacian Spectrum of Graphs, In *Graph Theory, Combinatorics and Applications, 871-898, 1991.*

NG. A. Y, JORDON. M. I, WAISS. Y., 2001, On Spectral Clustering: Analysis and an Algorithm. In *Advances in Neural Information Processing Systems 14, 2001.*

NUGENT. R., DEAN. N., AYERS. E., 2010, Skill-Set Profile Clustering: The Empty K-Means Algorithm with Automatic Specification of Starting Cluster Centers. In *Proceedings of the Third International Conference on Educational Data Mining, 2010, 151-160.*

RAZZAQ. L., FENG M., NUZZO-JONES. G., HEFFERNAN. N. T., KOEDINGER. K. R., JUNKER. B., RITTER S., KNIGHT A., ANISZCZYK. C., CHOKSEY. S., LIVAK. T., MERCADO. E., TURNER. T. E., UPALEKAR.R., WALONOSKI. J.A., MACASEK M. A., AND RASMUSSEN. K. P., 2005, The Assistment Project: Blending Assessment and Assisting. In C. K. Looi, G. McCalla, B. Bredeweg, & J. Breuker (Eds). *Proceedings of the 12th International Conference on Artificial Intelligence in Education, Amesterdam.* ISO Press, pp 555-562.

RITTER. S., HARRIS. T. K., NIXON. T., DICKISON. D., MURRAY. R.C., TOWLE. B., 2009, Reducing the Knowledge Tracing Space, In *Proceedings of the Second International Conference on Educational Data Mining, 2009, 151-160.*

SHIH. B., KOEDINGER. K. R., SCHEINES. R., 2010, Unsupervised Discovery of Student Learning Tactics, In *Proceedings of the Third International Conference on Educational Data Mining, 2010, 201-210.*

TRIVEDI. S., PARDOS. Z. A., HEFFERNAN. N. T, 2011, (submitted) The Utility of Clustering in Prediction Tasks, In *The Seventh ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2011.*

TRIVEDI. S., PARDOS. Z. A., HEFFERNAN. N. T, 2011, (accepted) Clustering Students to Generate an Ensemble to Improve Standard Test Score Predictions, In *The Fifteenth International Conference on Artificial Intelligence in Education, 2011.*