Predicting Long Term Student Retention using Expanded PFA and Clustering

Fei Song, Shubhendu Trivedi, Yutao Wang, Gábor N. Sárközy and **Neil T. Heffernan**
Worcester Polytechnic Institute
TTI at UChicago

Abstract

The regularity lemma is a fundamental result from the extremal graph theory. It is to claim the existence of a regular partition, from which we can construct the reduced graph, hence preserve the consistent behavior inside the same partition, and also decrease the order of input graph significantly. It is a very important tool in theoretical proofs, but due to the requirement of very large graph, it has no practical applications so far. In this paper we discuss the possible modifications to make the regularity lemma applicable in practical setting. This leads to a brand new clustering algorithm: Spectral regularity algorithm. We apply it to an Educational Data mining task: predicting student test result from features derived from tutors, we also compare the result with using standard spectral clustering algorithm. The experimental results are very promising.

Introduction

An important concept in student modelling is of "mastery learning" - that is, a student continues to learn a skill till mastery is achieved. While the exact definition of mastery varies, it is usually defined in terms of the most recent student performance. For example, in the Knowledge Tracing (Corbett and Anderson, 1995) framework that has come to dominate student modelling in many contexts, mastery in a skill is said to have been achieved when according to the model the probability that the student knows the skill exceeds 0.95. In many actual tutoring systems this definition is relaxed but still relies on the idea of recent performance. In a recent work (Wang and Beck, 2012) draw our attention to the question whether such a near singular focus is important after all. Intuitively, whether a student will remember enough to answer a question after taking a break is a better definition of mastery as compared to a local measure based on next item response, particularly in subjects such as Mathematics which are cumulative.

In particular, in their investigations Wang and Beck, while expanding the notion of mastery learning to incorporate the long term effect of learning, report some additional evidence for the spaced practice effect (for example see (Spitzer, 1939). That is, they found that features such as the number of distinct days that the student practised a skill was more important than features that accounted for how many questions they got correct. It is noteworthy that models such as Knowledge Tracing are in stark contrast to this, they only rely on the patterns of questions that students get correct or incorrect to make a prediction of their response on the next item, and hence factors such as how many questions they get correct are more important. This difference is not surprising since the factors that reflect long term retention might be quite different from factors that cause good short term performance.

While the goal of expanding mastery learning to incorporate long term retention makes intuitive sense, it is first important to consider the following questions: Is long term student retention actually predictable? and secondly, does some construct beyond performance, such as forgetting, vary by student? Wang and Beck indicate that the answer to both the questions appears to be in the affirmative. They give a roadmap for further research on this question and also suggest that the Performance Factor Analysis framework (Pavlik *et al.* 2009) could be expanded with features that are more relevant to retention. They list some such features that appear to indicate that student retention is predictable, but stop short of a study towards building such a detector.

In this work we take a step towards building a detector that could predict student performance after a delay of 5-10 days. As a baseline, we consider the expanded Performance Factor Analysis model as mentioned above (which is basically a logistic regression model) with additional features more relevant to retention. We also consider a simple bootstrapping strategy that uses clustering the data to generate a mixture of experts while using the expanded PFA model as a subroutine. The clustering methods used for the above are k-means clustering, Spectral Clustering and a clustering algorithm recently developed by the authors called Regularity Clustering, which is derived from the Regularity Lemma (Szemerédi, 1976), a fundamental result in graph theory. We report that the predictions obtained by the bootstrapping strategy are significantly better over the baseline and more specifically the predictions obtained by bootstrapping while using the Regularity Clustering method were significantly better as compared to the cases when k-means and spectral clustering was used. In the next section we provide details on the strategy used for prediction and also discuss the regularity clustering method in more detail.

Clustering Students and Strategy for Bootstrapping

The idea that students are perhaps quite different when it comes to forgetting makes it quite apparent that it is perhaps not a good idea to fit a global model on all of the data. Inspite of individual differences, we hypothesize that broadly the patterns and underlying reasons of forgetting would fall into several coarse groups, with each such group having students more "similar" to each other in regard to forgetting. Honing on this intuition, it might make more sense to cluster students into somewhat homogeneous groups and then train a predictor separately on each such group, which considers only the points from that cluster as the training set for itself. It is clear that each such predictor would be a better representative for that group of students as compared to a single global predictor trained on all the students at one time. While this idea sounds compelling, there is a major issue with it. While it is useful to model students as belonging to different groups, it is perhaps not a good idea to divide them into hard clusters. This is because the groupings are usually quite fuzzy. For example, a student might be extremely good at retaining information about certain aspects of Trigonometry but not other aspects, while at the same time might be strong with retaining algebra. Such complex characteristics can not be modelled by a simplistic solution as only clustering the data to some upper limit and then training predictors on each cluster. The "fuzzy" nature of such a process, which is like a spread of features across groups needs to be captured to make a distributive model such as the above more meaningful. This issue can be fixed by varying the granularity of the clustering and training separate models each time so the such features can be accounted for. A simple strategy to do so was proposed by the authors and was found quite useful in various tasks in student modelling (Trivedi *et al.*, 2011a), (Trivedi *et al.*, 2011b).

The idea is a simple ensemble method. The basic idea behind ensemble methods is that they involve running a "base learning algorithm" multiple times, each time with some change in the representation of the input (e.g. considering only a subset of the training examples or a subset of features etc) so that a number of diverse predictions can be obtained. This process also gives a rich representation of the input, which is one of the reasons why they work so well. In the particular case of our method, unlike many other ensemble methods that use a random subset to bootstrap, we use clustering to bootstrap. The training set is first clustered into k disjoint clusters and then a logistic regression model is trained on each of the clusters only based on the training points that were assigned to that cluster. Each such model, being a representative of a cluster is referred to as a *cluster model*. Thus for a given value of k there would be k cluster models. Note that since all the clusters are mutually exclusive, the training set is represented by all the k *cluster models* taken together. We refer to this as a *Prediction Model*, PM_k. For an incoming test point, we first figure out the cluster that point belongs to and then use the concerned cluster model alone to make a prediction on that point. Now also note that we don't specify the number of clusters above.

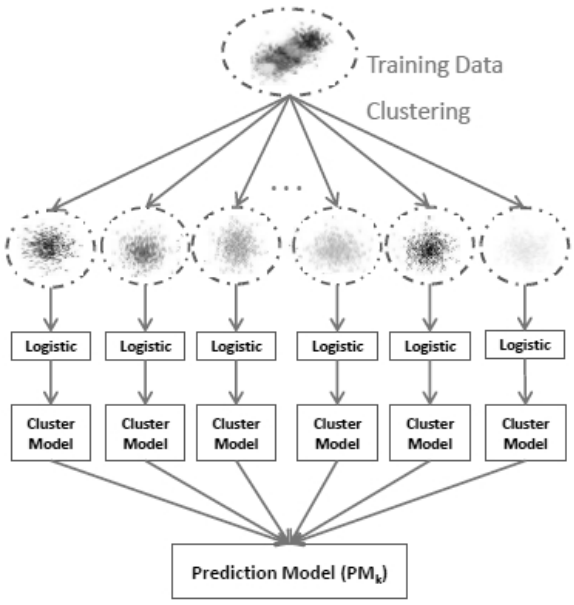

Figure 1: Construction of a Prediction Model for a given K. See text for details

Hence, we can change the granularity of the clustering from 1 (PM_1, which is the entire dataset as one cluster) to some high value K. In each such instance we would get a different *Prediction Model*, thus obtaining a set of K *Prediction Models*. Since the granularity of the clustering is varied, the predictions obtained would be diverse and hence could be combined together by some method such as averaging them together to get a single prediction.

Note that the clustering algorithm above is not specified and hence could be any clustering technique, as long as there is a straightforward way to map test points to clusters. In particular we clustered students using three algorithms: *k*-means (Hartigan *et al* 1979), Spectral Clustering (Luxburg, 2007) and a clustering technique recently introduced by the authors called Regularity Clustering (Sárközy *et al.* 2012). 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 (also called the similar matrix of the graph). 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. For more details we refer the reader the mentioned references. In the next section we describe the newly introduced clustering technique called Regularity Clustering.

Regularity Clustering Algorithm

In this section we briefly describe the Regularity Clustering algorithm. Among the various clustering techniques, one reason that spectral clustering has become quite popular is that it has a much stronger theoretical background. The objective in it is to approximately solve the balanced mincut problem. However, in the balanced mincut criteria, only the distance between current part and the complement to it is considered and ignores the information about the arrangement of the complement. More precisely, suppose we attach a weight value to each edge, the balanced mincut problem can be formalized as: to find a partition $A_1, A_2, \ldots, A_k$ which minimize the value

$$cut(A_1, A_2, \ldots, A_k) = \frac{1}{2}\sum_{i=1}^{k} \frac{1}{|A_i|} \cdot W(A_i, \bar{A}_i).$$

Now, consider the following scenario: we have two clusters A and B which have a lot of edges in between; however A also has a large number of connections to other parts of the graph, B doesn't have any such connections to the other parts. By the definition of balanced mincut, we might combine these two clusters; but indeed they have very different behaviour, certain applications would prefer that these two remain separated.

The Regularity Lemma (Szemerédi, 1976) is a fundamental result in Graph Theory that claims the existence of a regular partition, from which we can construct the reduced graph and hence decrease the order of input graph significantly. The criteria for evaluating a regular partition is quite different from the spectral clustering, it looks for the consistency behaviour within the same cluster, so a small portion would be enough as a representative to all. It is a very important tool in theoretical proofs, but due to the requirement of very large graph, it doesn't have practical applications. We recently introduced a clustering method that makes an attempt to harness the power of the Regularity Lemma. Before we describe the algorithm, we first introduce some notation.

Notation and Definitions

Let $G = (V, E)$ denote a graph, where V is the set of vertices and E is the set of edges. When A, B are disjoint subsets of V, the number of edges with one endpoint in A and the other in B is denoted by $e(A, B)$. When A and B are nonempty, we define the *density* of edges between A and B as

$$d(A, B) = \frac{e(A, B)}{|A||B|}.$$

The most important concept is the following.

Definition 1 *The bipartite graph $G = (A, B, E)$ is ϵ-regular if for every $X \subset A$, $Y \subset B$ satisfying*

$$|X| > \epsilon|A|, |Y| > \epsilon|B|$$

we have

$$|d(X, Y) - d(A, B)| < \epsilon,$$

otherwise it is ϵ-irregular.

Roughly speaking this means that in an ϵ-regular bipartite graph the edge density between *any* two relatively large subsets is about the same as the original edge density. In effect this implies that all the edges are distributed almost uniformly.

Definition 2 *A partition P of the vertex set $V = V_0 \cup V_1 \cup \ldots \cup V_k$ of a graph $G = (V, E)$ is called an equitable partition if all the classes $V_i, 1 \leq i \leq k$, have the same cardinality. V_0 is called the exceptional class.*

Thus note that the exceptional class V_0 is there only for a technical reason, namely to guarantee that the other classes have the same cardinality.

Definition 3 *For an equitable partition P of the vertex set $V = V_0 \cup V_1 \cup \ldots \cup V_k$ of $G = (V, E)$, we associate a measure called the index of P (or the potential) which is defined by*

$$ind(P) = \frac{1}{k^2}\sum_{s=1}^{k}\sum_{t=s+1}^{k} d(C_s, C_t)^2.$$

This will measure the progress towards an ϵ-regular partition.

Definition 4 *An equitable partition P of the vertex set $V = V_0 \cup V_1 \cup \ldots \cup V_k$ of $G = (V, E)$ is called ϵ-regular if $|V_0| < \epsilon|V|$ and all but ϵk^2 of the pairs (V_i, V_j) are ϵ-regular where $1 \leq i < j \leq k$.*

With these definitions we are now in a position to state the Regularity Lemma.

Original Regularity Lemma

Basically this lemma claims that every (dense) graph could be partitioned into a bounded number of pseudo-random bipartite graphs and a few leftover edges. Since random graphs of a given edge density are much easier to treat than all graphs of the same edge-density, the Regularity Lemma helps us to translate results that are trivial for random graphs to the class of all graphs with a given number of edges.

Theorem 1 (Regularity Lemma (Szemerédi, 1976)) *For every positive $\epsilon > 0$ and positive integer t there is an integer $T = T(\epsilon, t)$ such that every graph with $n > T$ vertices has an ϵ-regular partition into $k + 1$ classes, where $t \leq k \leq T$.*

In applications of the Regularity Lemma the concept of the *reduced graph* plays an important role.

Definition 5 *Given an ϵ-regular partition of a graph $G = (V, E)$ as provided by Theorem 1, we define the reduced graph G^R as follows. The vertices of G^R are associated to the classes in the partition and the edges are associated to the ϵ-regular pairs between classes with density above d.*

The most important property of the reduced graph is that many properties of G are inherited by G^R. Thus G^R can be treated as a representation of the original graph G albeit with a much smaller size, an "essence" of G. Then if we run any algorithm on G^R instead of G we get a significant speed-up.

Algorithmic Version of the Regularity Lemma

The original proof of the regularity lemma (Szemerédi, 1976) does not give a method to construct a regular partition but only shows that one must exist. To apply the regularity lemma in practical settings, we need a constructive version. Alon *et al.* (Alon *et al.*, 1994) were the first to give an algorithmic version. Since then a few other algorithmic versions have also been proposed (Frieze and Kanna, 1999), (Kohayakawa, Rödl and Thoma, 2003). Below we present the algorithm due to Alon *et al*.

To describe this algorithm, we need a couple of lemmas.

Lemma 1 *(Alon* et al.*, 1994) Let H be a bipartite graph with equally sized classes $|A| = |B| = n$. Let $2n^{-1/4} < \epsilon < \frac{1}{16}$. There is an $O(M(n))$ algorithm that verifies that H is ϵ-regular or finds two subset $A' \subset A$, $B' \subset B$, $|A'| \geq \frac{\epsilon^4}{16}n$, $|B'| \geq \frac{\epsilon^4}{16}n$, such that $|d(A,B) - d(A',B')| \geq \epsilon^4$.*

This lemma basically says that we can either verify that the pair is ϵ-regular or we provide certificates that it is not. The certificates are the subsets A', B' and they help to proceed to the next step in the algorithm. The next lemma describes the procedure to do the refinement from these certificates.

Lemma 2 *(Szemerédi, 1976) Let $G = (V, E)$ be a graph with n vertices. Let P be an equitable partition of the vertex set $V = V_0 \cup V_1 \cup \ldots \cup V_k$. Let $\gamma > 0$ and let k be a positive integer such that $4^k > 600\gamma^{-5}$. If more than γk^2 pairs (V_s, V_t), $1 \leq s < t \leq k$, are γ-irregular then there is an equitable partition Q of V into $1 + k4^k$ classes, with the cardinality of the exceptional class being at most*

$$|V_0| + \frac{n}{4^k}$$

and such that

$$ind(Q) > ind(P) + \frac{\gamma^5}{20}.$$

This lemma implies that whenever we have a partition that is not γ-regular, we can refine it into a new partition which has a better index (or potential) than the previous partition. The refinement procedure to do this is described below.

Refinement Algorithm: *Given a γ-irregular equitable partition P of the vertex set $V = V_0 \cup V_1 \cup \ldots \cup V_k$ with $\gamma = \frac{\epsilon^4}{16}$, construct a new partition Q.*
For each pair (V_s, V_t), $1 \leq s < t \leq k$, we apply Lemma 1 with $A = V_s$, $B = V_t$ and ϵ. If (V_s, V_t) is found to be ϵ-regular we do nothing. Otherwise, the certificates partition V_s and V_t into two parts (namely the certificate and the complement). For a fixed s we do this for all $t \neq s$. In V_s, these sets define the obvious equivalence relation with at most 2^{k-1} classes, namely two elements are equivalent if they lie in the same partition set for every $t \neq s$. The equivalence classes will be called atoms. Set $m = \lfloor \frac{|V_i|}{4^k} \rfloor$, $1 \leq i \leq k$. Then we choose a collection Q of pairwise disjoint subsets of V such that every member of Q has cardinality m and every atom A contains exactly $\lfloor \frac{|A|}{m} \rfloor$ members of Q. The collection Q is an equitable partition of V into at most $1 + k4^k$ classes and the cardinality of its exceptional class is at most $|V_0| + \frac{n}{4^k}$.

Since the index cannot exceed $1/2$, the algorithm must halt after at most $\lceil 10\gamma^{-5} \rceil$ iterations (see (Alon *et al.*, 1994)). Unfortunately, in each iteration the number of classes increases to $k4^k$ from k. This implies that the graph G must be indeed astronomically large (a tower function) to ensure the completion of this procedure. As mentioned before, Gowers (Gowers, 1997) proved that indeed this tower function is necessary in order to guarantee an ϵ-regular partition for *all* graphs. The size requirement of the algorithm above makes it impractical for real world situations where the number of vertices typically is a few thousand.

Spectral regularity algorithm

To make the regularity lemma applicable we first needed a constructive version that we stated above. But we see that even the constructive version is not directly applicable to real world scenarios. We note that the above algorithm has such restrictions because it's aim is to find a perfect regular partition. Thus, to make the regularity lemma truly applicable, we modify the Regular Partition Algorithm so that instead of constructing a regular partition, we find an *approximately* regular partition. Such a partition should be much easier to construct. We have the following 3 major modifications to the Regular Partition Algorithm.

Modification 1: We want to decrease the cardinality of atoms in each iteration. In the above Refinement Algorithm the cardinality of the atoms may be 2^{k-1}, where k is the number of classes in the current partition. This is because the algorithm tries to find all the possible ϵ-irregular pairs such that this information can then be embedded into the subsequent refinement procedure. Hence potentially each class may be involved with up to $k-1$ ϵ-irregular pairs. One way to avoid this problem is to bound this number. To do so, instead of using all the ϵ-irregular pairs, we only use some of them. Specifically, in this paper, for each class we consider at most one ϵ-irregular pair that involves the given class. By doing this we reduce the number of atoms to at most 2. We observe that in spite of the crude approximation, this seems to work well in practice.

Modification 2: We want to bound the rate by which the class size decreases in each iteration. As we have at most 2 atoms for each class, we could significantly increase m used in the Refinement Algorithm as $m = \frac{|V_i|}{l}$, where a typical value of l could be 3 or 4, much smaller than 4^k. We call this user defined parameter l the refinement number.

Modification 3: Modification 2 might cause the size of the exceptional class to increase too fast. Indeed, by using a smaller l, we risk putting $\frac{1}{l}$ portion of all vertices into V_0 after each iteration. To overcome this drawback, we "recycle" most of V_0, i.e. we move back most of the vertices from V_0. Here is the modified Refinement Algorithm.

Modified Refinement Algorithm: *Given a γ-irregular equitable partition P of the vertex set $V = V_0 \cup V_1 \cup \ldots \cup V_k$ with $\gamma = \frac{\epsilon^4}{16}$ and refinement number l, construct a new partition Q.*
For each pair (V_s, V_t), $1 \leq s < t \leq k$, we apply Lemma

1 with $A = V_s$, $B = V_t$ and ϵ. For a fixed s if (V_s, V_t) is found to be ϵ-regular for all $t \neq s$ we do nothing, i.e. V_s is one atom. Otherwise, we select one ϵ-irregular pair (V_s, V_t) randomly and the corresponding certificate partitions V_s into two atoms. Set $m = \lfloor \frac{|V_i|}{l} \rfloor$, $1 \leq i \leq k$. Then we choose a collection Q' of pairwise disjoint subsets of V such that every member of Q' has cardinality m and every atom A contains exactly $\lfloor \frac{|A|}{m} \rfloor$ members of Q'. Then we unite the leftover vertices in each V_s, we select one more subset of size m from these vertices and add these sets to Q' resulting in the partition Q. The collection Q is an equitable partition of V into at most $1 + lk$ classes.

Now we present our modified Regular Partition Algorithm. There are three main parameters to be selected by the user: ϵ, the refinement number l and h the minimum class size when we must halt the refinement procedure. h is used to ensure that if the class size has gone too small then the procedure should not continue.

Modified Regular Partition Algorithm :

Given a graph G and parameters ϵ, l, h, construct an approximately ϵ-regular partition.

1. **Initial partition:** *Arbitrarily divide the vertices of G into an equitable partition P_1 with classes $V_0, V_1, \ldots, V_l$, where $|V_1| = \lfloor \frac{n}{l} \rfloor$ and hence $|V_0| < l$. Denote $k_1 = l$.*
2. **Check size and regularity:** *If $|V_i| < h$, $1 \leq i \leq k$, then halt. Otherwise for every pair (V_s, V_t) of P_i, verify if it is ϵ-regular or find $X \subset V_s, Y \subset V_t, |X| \geq \frac{\epsilon^4}{16}|V_s|, |Y| \geq \frac{\epsilon^4}{16}|V_t|$, such that $|d(X,Y) - d(V_s,V_t)| \geq \epsilon^4$.*
3. **Count regular pairs:** *If there are at most ϵk_i^2 pairs that are not verified as ϵ-regular, then halt. P_i is an ϵ-regular partition.*
4. **Refinement:** *Otherwise apply the Modified Refinement Algorithm, where $P = P_i, k = k_i, \gamma = \frac{\epsilon^4}{16}$, and obtain a partition Q with $1 + lk_i$ classes.*
5. **Iteration:** *Let $k_{i+1} = lk_i, P_{i+1} = Q, i = i + 1$, and go to step 2.*

Two phase algorithm

To make the regularity lemma applicable in clustering settings, we still need to solve two issues.

The first is in practise we don't require equitable partition; the other is we do not have full control of cluster numbers in the final partition. To overcome these, we adopt the following two phase strategy (Figure 1):

1. **Application of the Regular Partition Algorithm:** In the first stage we apply the regular partition algorithm as described in the previous section to obtain an approximately regular partition of the graph representing the data. Once such a partition has been obtained, the reduced graph as described in Definition 5 could be constructed from the partition.
2. **Clustering the Reduced Graph:** The reduced graph as constructed above would preserve most of the properties of the original graph (see (Komlós *et al.*, 2002)). This implies that any changes made in the reduced graph would also reflect in the original graph. Thus, clustering the reduced graph would also yield a clustering of the original graph. We apply spectral clustering (though any other pairwise clustering technique could be used) on the reduced graph to get a partitioning and then project it back to the higher dimension. Recall that vertices in the exceptional set V_0, are leftovers from the refinement process and must be assigned to the clusters obtained. Thus in the end these leftover vertices are redistributed amongst the clusters using k-nearest neighbour classifier to get the final grouping.

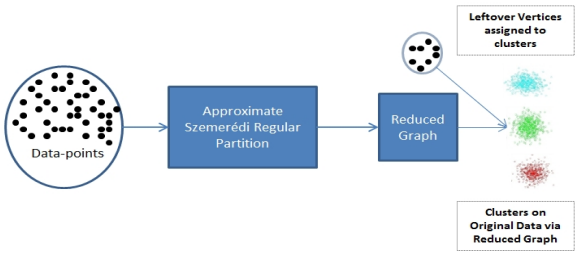

Figure 2: A Two Phase Strategy for Clustering

Dataset Description and Experimental Results

The data considered in this article comes from the ASSISTments system, a web-based tutoring system for 4th to 10th grade mathematics. The system is widely used in Northeastern United States by students in labs and for doing homework in the night. The dataset used is the same as used in (Wang and Beck, 2012). The only exception being that we considered the data for a unique 1969 students and did not consider multiple data points of the same student attempting something from a different skill. This was only done because we were interested in clustering students according to user-id.

The following features were used. The goal was to predict whether a response was correct i.e. 1 or incorrect or 0.

1. *n_correct:* the number of prior student correct responses on this skill; This feature along with *n_incorrect*, the number of prior incorrect responses on this skill are both used in PFA models.
2. *n_day_seen:* the number of distinct days on which students practiced this skill. This feature distinguishes the students who practiced more days with fewer opportunities each day from those who practiced fewer days but more intensely, and allow us to evaluate the difference between these two situations. This feature was designed to capture certain spaced practice effect in students data.
3. *g_mean_performance:* the geometric mean of students previous performances, using a decay of 0.7. For a given student and a given skill, use opp to represent the opportunity count the student has on this skill, we compute the geometric mean of students previous performance using formula: $g\_mean\_performance(opp) =$

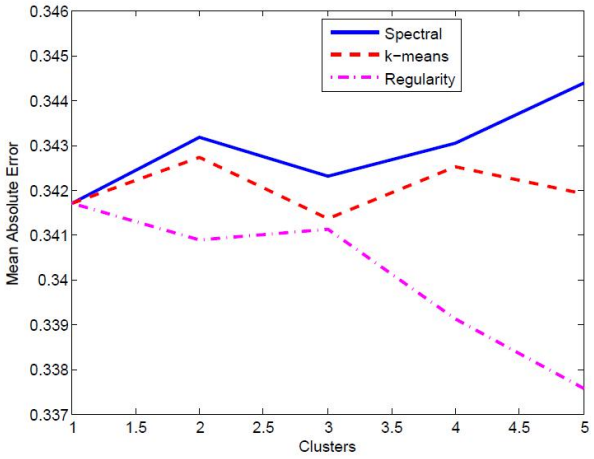

Figure 3: Mean Absolute Errors on Using the three Clustering Techniques for Bagging

$g\_mean\_performance(opp - 1) \times 0.7 + correctness(opp) \times 0.3$. The geometric mean method allows us to examine current status with a decaying memory of history data. The number 0.7 was selected based on experimenting with different values.

4. *g_mean_time:* the geometric mean of students previous response time, using a decay of 0.7. Similar with g_mean_performance, for a given student and a given skill, the formula of the geometric mean of students previous response time is: $g\_mean\_time(opp) = g\_mean\_time(opp - 1) \times 0.7 + response\_time(opp) \times 0.3$.

5. *slope_3:* the slope of students most recent three performances. The slope information helps capture the influence of recent trends of student performance.

6. *delay_since_last:* the number of days since the student last saw the skill. This feature was designed to account for a gradual forgetting of information by the student.

7. *problem_difficulty:* the difficulty of the problem. The problem_difficulty term is actually the problem easiness in our model, since it is represented using the percent correct for this problem across all students. The higher this value is, the more likely the problem can be answered correctly.

Out of these features it was reported that features such as *n_correct* and *n_incorrect* had very little influence on the prediction performance while the features *g_mean_performance* and *n_day_seen* appear to be reliable predictors of student retention. This observation is consistent with the spaced practice effect in cognitive science. Hence, in our experiments we don't consider *n_correct* and *n_incorrect* while training the model.

Table 1: Paired t-tests on the predictions obtained with Spectral and Regularity Clustering at different k

Pred. Models	Spectral & Regularity
1	-
2	0.1086
3	**0.0818**
4	**0.0045**
5	$\ll$ **0.005**

Table 2: Paired t-tests on the predictions obtained with the baseline (PM_1) and Regularity Clustering

Pred. Models	Baseline & Regularity
1	-
2	0.00531
3	**0.0401**
4	**0.0018**
5	**0.0044**

Conclusion and future work

References

A. T Corbett and J. R. Anderson. 1995. Knowledge Tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4: pp. 253-278.

N. Alon, R. A. Duke, H. Lefmann, V. Rödl, R. Yuster, The Algorithmic Aspects of the Regularity Lemma. *Journal of Algorithms*, 16, (1994), pp. 80-109.

A. M. Frieze, R. Kannan, A simple algorithm for constructing Szemerédi's regularity partition. *Electron. J. Comb*, 6, (1999).

W. T. Gowers, Lower bounds of tower type for Szemerédi's uniformly lemma. Geom. Funct. Anal 7, (1997), pp. 322-337.

J. A. Hartigan, M. A. Wong, Algorithm AS 136: A K-Means Clustering Algorithm, In *Journal of the Royal Statistical Society, Series C (Applied Statistics)*, 28 (1): pp. 100108.

Y. Kohayakawa, V. Rödl, L. Thoma, An optimal algorithm for checking regularity. *SIAM J. Comput*, 32(5), (2003), pp. 1210-1235.

J. Komlós, A. Shokoufandeh, M. Simonovits, and E. Szemerédi, The Regularity Lemma and Its Applications in Graph Theory. *Theoretical Aspects of Computer Science*, LNCS 2292, (2002), pp. 84-112.

R. A. Jacobs, M. I. Jordan, S. J. Nowlan and G. E. Hinton, Adaptive Mixtures of Local Experts, *Neural Computation*, Vol 3, No 1, 79-87, 1991

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

P. I. Pavlik, H. Cen and K. R. Koedinger. 2009. Performance Factor Analysis - A New Alternative to Knowledge Tracing. In the Proceedings of *the 14th International Conference on Artificial Intelligence in Education*.

G. N. Sárközy, F. Song, E. Szemerédi and S. Trivedi. A Practical Regularity Partitioning Algorithm and its Applications in Clustering. arXiv preprint arXiv:1209.6540.

H. F. Spitzer. 1939. Studies in Retention, In *Journal of Educational Psychology*, 30(9), pp. 641-657.

E. Szemerédi, Regular partitions of graphs, Colloques Internationaux C.N.R.S. N^{o} 260 - *Problèmes Combinatoires et Théorie des Graphes*, Orsay (1976), pp. 399-401.

S. Trivedi, Z. A. Pardos and N. T. Heffernan, Clustering Students to Generate an Ensemble to Improve Standard Test Predictions, The fifteenth international Conference on Artificial Intelligence in Education, 2011.

S. Trivedi, Z. A. Pardos, G. Sarkozy and N. T. Heffernan, Spectral Clustering in Educational Data Mining.Proceedings of *the 4th International Conference on Educational Data Mining*, 2011, pp. 129-138

Y. Wang and J. E. Beck. 2012. Incorporating Factors Influencing Knowledge Retention into a Student Model. In the Proceedings of *the 5th International Conference on Educational Data Mining*, pp. 201-203.