A Method for Automatic Selection and Interpretation of Student Clustering Models According to Their Activity on E-Learning System

Igor Jugo, Božidar Kovačić
University of Rijeka
Department of Informatics
Radmile Matejcic 2, 51000 Rijeka, Croatia
{ijugo, bkovacic}@inf.uniri.hr

Abstract. *The paper proposes a method that is part of a new, extended architecture of our web-based intelligent tutoring system (ITS). It was developed to provide hints to students during learning through the application of educational data mining (EDM). The architecture consists of three modules – a) a communication module that enables seamless communication with data mining tools; b) a clustering module that discovers clusters in student data based on their activity and c) a sequential pattern mining (SPM) module that finds efficient frequent learning patterns of students in each cluster. Finally, the obtained results are used by the tutoring module to provide hints to a student on which item to learn next (or previous to the selected one). To improve the hint selection process we developed a method for cluster grading to determine which cluster represents the group that has been using the ITS in the manner closest to an envisioned optimum. We verify the method on data gathered from two groups of students who used the system to master a knowledge domain, and present the obtained results.*

Keywords. Clustering, Educational data mining, Intelligent tutoring system, E-learning

1 Introduction

Intelligent tutoring systems (ITSs) are a valuable teaching tool not only for distance education but also as a complementary teaching/learning tool in traditional (face-to-face) education. Many such systems have been developed for teaching in well-defined domains (e.g., math, physics, etc.) and are mostly standalone desktop applications. On the other hand, the number of web-based ITSs is much smaller (Brusilovsky, 1999), especially for teaching in ill-defined domains (Lynch et al., 2006). Ill-defined domains consist of a number of knowledge units (KUs) that do not have a strictly defined order in which they have to be taught/learned, but instead the system relies on a domain expert to define the structure of the domain. One such system has been developed at our institution to serve as an additional learning platform.

Our web-based intelligent tutoring system (WITS), described first by (Kovačić & Skočir, 2003) and (Kovačić & Jugo, 2009), currently provides teachers with functionalities for creating KUs, teaching materials, various types of questions for assessing acquired knowledge, and an editor to create the KU hierarchy. Each KU is described by a start and a threshold value which students reach by answering the questions correctly. Furthermore, the system features a descriptive statistics module for students and teachers (Kovačić et al., 2012). To further improve the system's overall efficiency we proposed a new architecture which integrates data mining (DM) algorithms (Jugo et al., 2014; Jugo et al., 2015).

A valuable source of data for our system lies in the records of student interactions with the system. We apply educational data mining (EDM) processes (Romero & Ventura, 2007; Romero & Ventura 2010; Fernandez et al., 2014) to these records and use the obtained information to enrich the system's student model and improve the tutoring model. In order for improvements to take place we added several new modules to the tutoring module. The first was an integration layer (Jugo et al., 2013) that creates a continuous communication channel to DM tools Weka (Hall et al., 2009) and SPMF (Fournier-Viger et al., 2014) which are used to execute DM algorithms on data gathered within our ITS. Second was the module for automatic clustering model selection and interpretation (which implements the method described in this paper) to discover different groups of students based on their activity and efficiency levels. Finally, a sequential pattern mining (SPM) (Srikant & Agrawal, 1996) module used to discover frequent patterns (FPs) students take through the knowledge domain. SPM algorithms can produce a large number of FPs. An algorithm that differentiates between productive patterns (those that yield better gains in the student's knowledge levels) and non-productive patterns (those that represent possible issues with the knowledge domain structure, difficult questions, etc.) was developed. In our system we use both types of patterns: productive patterns are used to enable dynamic creation of learning structures by our tutoring model, while non-productive patterns are presented to teachers so they can make actions to correct their causes.

This paper focuses in particular on the second module – clustering of students based on their activity (ways of using the system) and efficiency (correctness of answers to questions presented by the system) levels. In order to guide the students during their progress through the knowledge domain the system needs to discover clusters of students that use the system in a similar way and consequently determine which cluster represents the highest achieving students, average achieving ones, etc. An important aspect of this system is that it performs automated student model improvements (Koedinger, 2012) by running both clustering and SPM routines at scheduled intervals while the students are progressing through the knowledge domain and increasing the size of the interactions dataset.

With results from both clustering and SPM modules, we can make the tutoring model more adaptive: cluster ordering and discovered productive FPs of each cluster are used to guide students from a lower grade cluster towards the activity levels and learning paths of a higher grade cluster, thus improving the students' learning experience and overall results.

The rest of this paper is structured as follows: Section 2 introduces related work on EDM and focuses on applying clustering in e-learning systems. Section 3 gives an overview of the basic functionality of our WITS which is the environment used for research at hand. The process through which the system obtains clustering models using communication with DM tools is described in Section 4, while Section 5 presents our method for automatic clustering selection and interpretation. In Section 6 we present results of the proposed method and conclude the paper in Section 7.

2 Related work

Combining clustering (Kaufman & Rousseeuw, 1990) and SPM is an approach that has been applied for the purpose of recommending web pages in e-learning systems (Romero et al., 2007) and analysing online collaborative learning data (Perera et al., 2007). An example of integration of data mining tools with an e-learning system can be found in (Zorilla et al., 2010), where authors improved the existing Black Board monitoring tool and in (Romero et al., 2013), where authors presented a Moodle block that enables the users to perform clustering, classification and association rule mining, and export the raw results to a file. This is a useful extension of Moodle functionalities for teachers even though it presents them with "raw" output of DM algorithms which can prove to be difficult to understand. Our system integrates the results of DM algorithms into the process of increasing the system's adaptivity and further presents the results using interactive, in-browser, visualizations that are easy to comprehend and provide immediate insights.

Student clustering is an important research topic in EDM. There are a number of approaches to clustering: connectivity-based, centroid-based, distribution-based, density-based, etc., and an even larger number of algorithms that can be applied on student data. An overview of the clustering analysis critical steps was published by Miligan (1989). In cluster analysis, the fundamental problem is to determine the optimal number of clusters, which has a deterministic effect on the clustering results. It is a well-known optimization problem that has received significant attention. A variety of methods for this problem have been analysed by Gordon (1999), where the author divided them into two categories: global and local methods. The local methods are intended to test the hypothesis that a pair of clusters should be amalgamated. They are suitable for assessing only hierarchically-nested partitions. With global methods, the quality of clustering given a specific number of clusters, g, is measured by a criterion, and the optimal estimate of g, ^G , is obtained by comparing the values of the criterion calculated in a range of values of g. Some of these methods analysed were: Calinski and Harabasz's method, Hartigan's method, Krzanowski and Lai's method, Silhouette statistic and the Gap method. Their performance has been analysed by Tibshirani (2001) and Symons (1981). Our method implements the silhouette statistic approach.

Once a clustering model is selected, it can be evaluated through various statistical methods (Bouchet et al., 2013) or a number of other, more complex methods (Gordon, 1999), while the interpretation depends on the research area and the nature of data. Our method relies on descriptive statistics to sort the clusters based on cluster members' activity levels as well as their learning efficiency.

3 Student data and feature engineering

Each student has his/her own personal approach to using the system. Some students copy the learning materials, study them offline and then come back to answer the questions later, others try to learn from the bottom up or from the top down, yet others try to brute force the system by answering the questions until they find the right answer. To model the way the students interact with the system, two sets of features were used. The first set of "database" features can be obtained directly from the database using SQL queries. The second set consists of engineered features developed from database features in order to better represent the student's current activity levels and his/her efficiency. The first three features (learning (L), repetition (R) and time spent (T)) are called activity features and represent each student's interactions with the system. While the activity features represent the way a student interacts with the system, the effectiveness feature (E) represent his/her success in answering questions about each knowledge unit being learned within our system. The engineered feature set F_{EF} represents the student's activity in relation to: a) the quantity of content he/she has covered up to this moment and b) the activity of other students in the group. This evens out the activity level of students who use the system in a similar manner in situations where one student has already covered the whole domain and the other has just started using it. Before sending them to Weka, the values of the mentioned features are standardized.

As mentioned in the introductory part of the paper, a new module called "the integration layer" has been developed to enable communication with two DM tools. In this way, re-implementing any specific algorithm into our application has been avoided, which ensures that data can be analysed by a DM expert on another machine running the same DM tools, with absolute confidence that the results will be the same (where it is possible, depending on the algorithm). An important advantage of this architecture is that the administrator can easily change and use any clustering or SPM algorithm provided by either tool. In our system we communicated with Weka to run the kMeans clustering algorithm, and SPMF to run the USPAN algorithm.

Algorithm 1 uses as input the engineered feature set along with some additional information (domain, domain group) and system settings necessary for correct formatting of API calls towards DM tools. With each increment of the number of clusters *k*, a new API call is created and executed. It starts from *k*=2 and increases the value until the DM tool returns a model that satisfies the cut-off condition: "exit the loop if the model contains two or more clusters with just one student in it".

The reasoning behind the condition is that our overall goal is to discover groups of students that use the system in a similar way and then provide hints based on frequent patterns discovered **in their cluster or the cluster graded above** to create a better learning experience for students in each group. Having that in mind, it is not particularly useful to have single member clusters. We did however allow for solutions in which one single-member cluster appears (a possible outlier) because our system is fully automated which complicates outlier detection.

Upon completion, the algorithm returns a set of clustering models for *k*=2, 3, etc.

Algorithm 1: clustering using the integration layer and DM tools

Input:
Domain identifier D_i,
Domain group identifierDG_j,
Engineered features set F_{EF}= [L,R,T,E]
Settings:
tool T $\in$ [weka*,spmf],
format FT$\in$ [arff*,spmf],
alg. A $\in$ [kMeans*, EM,…]
Output:
clustering models[centroids, members]

```
prepare dataset D = (Di, DGj, FEF)
generate filename
write dataset to file (fn)
k = sqrt(sizeof(D)/2)   // the "elbow" method
array models[]
for i = 2 to k + 5  // more clusters in small groups
   x = construct_api_call (A,T, i, fn)
   out = execute_api_call(x)
   [centroids, members] = process_output (out)
   models[i] = [centroids, members]
      if (number of clusters in model with no. members == 1) >= 2)
      break
   endif
endfor
return models[]
```

4 Automated clustering model validation and interpretation

The algorithm presented in the previous Section returns a set of clustering models with an increasing number of clusters. The goal of this method is to select the best model and evaluate each of the clusters in that model. The interpretation scores and orders the clusters so that the highest graded cluster represents the group that has used or is using the system in a way that the authors believe to be closest to optimal. The cluster with the lowest grade represents the group that was inactive at the time the clustering routine was executed. As a result, the student's activity model (cluster membership and grade of cluster) is continuously updated in the database with the latest results based on a growing dataset of students' interactions with the system.

4.1. Model validation

As mentioned in Section 2, many methods that determine the quality of distribution (optimal number of clusters in a model) have been developed and tested by Tibshirani (2001) and Symons (1981). Some of these methods require changes to be made in the source code of the clustering algorithm; other can be applied after the model was acquired. The silhouette statistic belongs to the latter group. The definition of the silhouette statistic is based on the notion of silhouettes introduced by Rousseeuw (1987). According to this approach, silhouettes are constructed to show graphically how well each object is classified in a given clustering output.

The silhouette index, denoted by *s(g)*, is defined as the average of the *s(i)* for all objects (*i*) in the data set. In this way, *s(g)* represents the average silhouette width for the entire data set, reflecting the within-cluster compactness and between-cluster separation of a clustering. The optimum value of *g* is chosen such that *s(g)* is maximized over *g*. Average *s(g)* over accumulated data of a cluster is a measure of how tightly grouped all the data points in the cluster are. Thus the average *s(g)* over accumulated data of the entire dataset is a measure of how appropriately the data has been clustered. The silhouette statistic is implemented so that it receives a variable number of clustering models obtained from Algorithm 1, and returns only one model: the model with the highest silhouette statistic value. This completes the first phase of our method. The second phase grades/orders the clusters in the selected model.

4.2. Model interpretation

The frequency of student results is usually distributed in the shape of the “bell” curve. The area around its highest point represents the majority or the “average” students, while the “wings” represent the highest and lowest achieving students. The basic idea of our algorithm can be summarized as follows: we divide the distribution of results for each activity feature into intervals of equal width, assign a score to each interval and then evaluate the position of each of the cluster centroid values in the distribution. Since the values of our engineered features are standardized, we know that the left-most values represent inactive students and the right-most values represent students struggling to master the learning content. Students that create less than average number of learning material views are regarded as “fast” learners, while those on the other side of the average can be considered “slow”, etc. Also, when scoring intervals, we have to give a higher score to the students that are trying (“struggling”) than to those that are inactive. This concept is presented in Figure 3.

The score distribution depends of what each feature represents. For instance, the scores scale for the learning feature is completely inverted from the time feature, where we reward the effort in reading the materials thoroughly as opposed to glancing over them. We chose to divide the results into 13 intervals meaning that a cluster centroid value for a feature can receive a maximum of 13 points and a minimum of 1 point. At the end, the cluster with the highest sum of points is the cluster representing the group of students that are/were using the system in a manner we consider optimal.

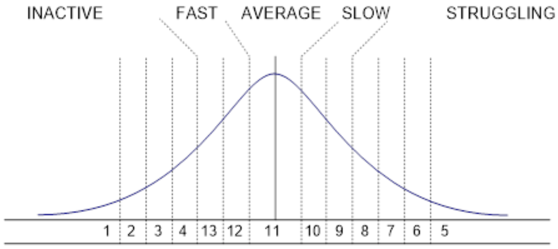

Figure 1. Scoring the intervals of feature values distribution

It is necessary to point out that a smaller or a larger number of intervals could also have been used. A larger number would reduce the chance of getting the same results for multiple clusters, while a smaller one would simplify the calculation process.

As there is no human intervention in our method and we cannot anticipate the distribution of feature values, we have no guarantee that they will follow a normal distribution. In fact, it is very unlikely, especially in the early stages (first day) of granting the students access to the system. Data is sparse during that period as only a few students have made significant progress through the domain. In such cases the curve can be very wide/narrow, skewed to left or right, etc. To account for the high variability, we have developed a flexible algorithm that adapts to the values of the dataset, prepares the grading process and examines each of the centroid values for each cluster.

The basic steps of our cluster grading/ordering method are presented in Figure 4 below.

The first step in the method is to determine the width of the distribution, i.e., the leftmost and rightmost data point in the dataset. Next, this width is divided by 13 to get the interval width, which is the first step in preparing the score-to-interval mapping. Then, the value of the probability density function of the normal distribution for each value in the dataset is calculated. The highest value becomes the centre position of our score-to-interval mapping “bar”.

Using this information and the interval width from the previous step, the left and right borders of each interval in our bar are calculated.

If the distribution is skewed to one side, several situations are possible: 1) part of the grading bar goes outside the width of the curve so grades in those intervals become unreachable; 2) the grading bar does not reach the end of the distribution, so any values that appear have to be approximated to the last interval in the bar. Both of these situations are demonstrated in Step 3 of Figure 2. Finally, when the grading bar is set we can read the centroid values of each of the activity features of every cluster in the model, find out which interval they fit in, read the score of the interval, and add that to the subtotal for each cluster.

We now turn to the description of the scoring process for the fourth feature: student efficiency.

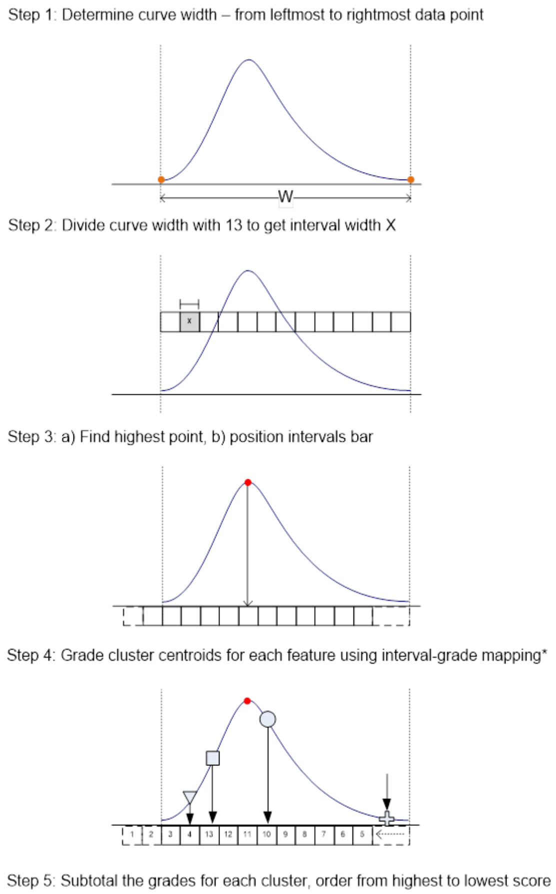

Figure 2. Steps of the selected model scoring algorithm

The students' efficiency feature is an objective measure of his/her ability to answer the questions about the learning materials correctly. The highest value represents the most efficient student. Therefore, the interpretation procedure for this feature is simpler. The first step is to sort the centroid values of each cluster from highest to lowest. Next, the highest value becomes the benchmark and gets assigned 13 points. The points for all other clusters are defined as a rounded percentage of the benchmark value. For example, if the highest value is 3.5 and the second best cluster has a centroid value of 1.8, the points will be calculated by the expression (1.8/3.5)*13 = 6.68 ~ 7 points. The same process is repeated until all cluster centroids for this feature are graded. The score is then added to the subtotal score for each cluster, which completes the scoring/ordering phase. The cluster ordering and the cluster attribute of each student are then updated accordingly in the database.

In this the system is able to continuously update the student's activity and efficiency model until the student masters the domain. This model represents the quality of the student's interaction with the system which reflects to his/her learning experience and the final course grade. Every time the system performs the clustering process the results are stored in the database so both the teacher and the DM expert can analyse the results. The teacher can use the WITS teacher interface to check visualizations of each student's cluster assignment changes in time, while the DM expert can review the results and identify possible improvements to our score-to-interval mappings. In Table 1, we present our current score-to-interval mappings for the three activity features.

Table 1. Score-to-interval mappings for F_{EF}

INT	1	2	3	4	5	6	7	8	9	10	11	12	13
$L\%_{std}$	1	2	3	4	13	12	11	11	9	8	7	6	5
$R\%_{std}$	1	2	3	4	13	12	11	10	9	8	7	6	5
$T\%_{std}$	1	2	3	4	9	10	11	12	13	8	7	6	5
E_{total}	calculated [1…13]												

These are used by Algorithm 2 to evaluate centroid values of clusters for activity features (L, R, T) and the efficiency feature (E). The cluster centroid value interpretation algorithm is presented below.

Algorithm 2: Cluster centroid value interpretation

```
Input:
  selected clustering model with k clusters:
     model=[ k, centroids, members, dataset ]
  score-to-interval mapping table for f ∈ F_EF: M[f][scores]
Output:
  evaluated/graded list of clusters

array clusterOrder[], array centroidScores[]
for each f ∈ F_EF
   if( f == E )
      score = calculate_efficiency_score(centroids(f));
      centroidScores[f][k]=score
   else
      ndf = calculate_pdf_ (dataset(f))
      cw = calculate_dataset_width()
      iw = calculate_interval_width(cw)
      peak = max(ndf)
      M[f][center] = center_grading_bar(peak)
M[f][borders]= calculate_border_values(peak, iw, cw, dataset(f))
       for each centroid ∈ centroids[f]
          i = find_interval(centroid)
          score = get interval score(i,M(f))
          centroidScores[f][k]=score
       endfor
    endif
 endfor
 for each cs ∈ centroidScores
    clusteringOrder[k] = sum(cs[k])
 endfor
 sort(clusteringOrder)
 return clusteringOrder
```

The highest ranked cluster should always represent the best group of students, while the lowest ranked usually represents the inactive students. Other clusters will be ranked between these two. As we have 13 intervals, the highest possible efficiency score is 13 and the lowest is 1. For the "time" (T) feature, which represents the amount of time the student spent reading learning materials, the grading scale is inverted relative to L and R features. **Therefore, the best student (group/cluster) is the one that needed lower than average number of learning and repetition actions, spent more than average time reading the learning materials and had the highest efficiency** (number of correct answers to questions).

5 Results

The method described in Section 5 was tested on data collected from a knowledge domain “Introduction to web application development” developed for third year undergraduate students at our Department. The domain consisted of forty knowledge units. Participants had access to the domain for 7 days in September, 2015, and were asked to complete it in the allotted period of time. In order to analyse differences in the way students interact with the system, the participants were divided them into two groups that had an equal average result in the pre-test. The basic statistical data is presented in Table 2.

Table 2. Basic statistics of student groups

Groups	G1	G2
Number of students	33	31
Number of active students (had > 0 interactions with the system)	33	30
Percentage of students that completed the domain	97%	98%

In order to test the algorithm for optimal k value selection based on the silhouette statistic we ran the clustering analysis for both groups using the engineered features set FEF after the access period had expired and the majority of students had completed the domain. The obtained results for G1 are presented in Table 3.

Table 3. Silhouette statistics and cluster sizes for G1

k	S(k)	1	2	3	4	5	6	7	8
2	0.34	16	17						
3	0.36	15	4	14					
4	0.33	11	2	8	11				
5	**0.47**	12	2	7	11	1			
6	0.45	9	2	7	9	1	5		
7	0.38	6	2	7	8	1	5	4	
8	0.29	6	2	7	7	1	3	4	3

For both groups the model with k=5 clusters had the highest value of the silhouette statistic. This model has then been employed in the second phase. The second phase began after the best clustering model had been selected. As mentioned earlier, the results presented here were obtained after the students had already finished using the system so the presented dataset is complete. Figures 3 and 4 represent the distribution of values of the two main activity features in F_{EF} for G1 (dots) and G2 (squares). They show that students in both groups had a similar distribution of results. The only exception is one student, who had a specific way of using the system (had almost zero repetitions). These exceptions compelled us to create a flexible solution that is able adapt to the differences between the ways each group interacts with the system.

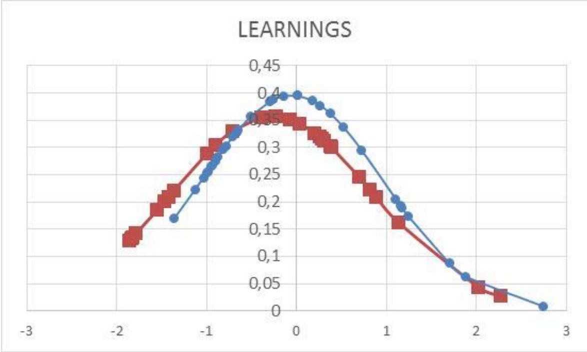

Figure 3. Learning (L%std) feature results distribution for G1 and G2

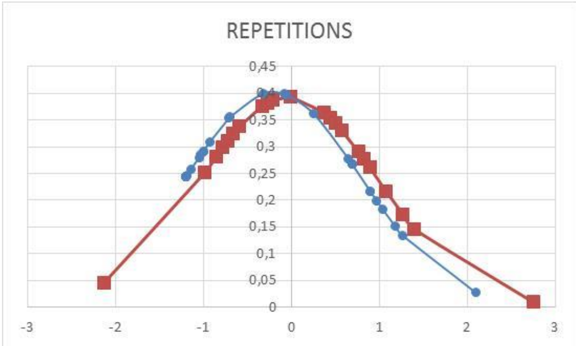

Figure 4. Repetition (R%std) feature results distribution for G1 and G2

Since this process runs in scheduled intervals during the entire time the student have access to the knowledge domain, the distribution of results will vary greatly as the students advance through the domain. The results for G1 are presented in Tables 4 and 5.

The placement row for the activity features gives us a simple visualization that can be useful for quick results interpretation. The best cluster (4) is consistently below average for L and R features, and average for the T feature.

When we add the efficiency feature scoring, it becomes clear that the members of this cluster are the closest to our definition of a high achieving student. It is also clear that cluster 2 represents students that made the least effort and had the lowest efficiency, which leads us to believe they kept guessing the answers to the questions.

Table 4. Interval distribution, scores and cluster placement for activity features for G1

							Center							
L Interv.	-1.86	-1.57	-1.28	-0.99	-0.71	-0.42	-0.13	0.15	0.44	0.73	1.02	1.31	1.59	1.88
L Scores	1	2	3	4	13	12	11		10	9	8	7	6	5
Placement			2		1	4,5							3	
R Interv.	-1.97	-1.72	-1.46	-1.21	-0.95	-0.70	-0.44	-0.19	0.18	0.56	0.93	1.31	1.68	2.06
R Scores	1	2	3	4	13	12	11		10	9	8	7	6	5
Placement				2,3,5		4						1		
T Interv.	-1.89	-1.62	-1.35	-1.08	-0.81	-0.54	-0.27	-0.00	0.26	0.53	0.80	1.07	1.34	1.61
T Scores	1	2	3	4	9	10	11		12	13	8	7	6	5
Placement				2		1	4			3				5

Table 5. Final scores for clusters (G1)

f	*k=5*	Centroid values/scores				
	Clusters	1	2	3	4	5
L	Values	-0.598	-1.150	1.494	-0.220	-0.260
	Score	13	3	6	12	12
R	Values	0.980	-1.025	-1.100	-0.692	-1.200
	Score	7	4	4	12	4
T	Values	-0.488	-0.845	0.329	-0.048	3.660
	Score	10	4	13	11	5
E	Values	0.352	0.155	0.280	0.446	0.060
	Scores	8	4	7	10	2
Subtotal		**38**	**15**	**30**	**45**	**23**
Grading		**B**	**E**	**C**	**A**	**D**

This is supported by the lowest value for feature T (time spent reading/learning the materials).

6 Conclusion

In this paper we presented a method for automatic selection and interpretation of clustering results obtained from a database of students' interactions with our web-based intelligent tutoring system. The proposed method was implemented in a clustering module that is part of a system built to improve the adaptivity of the tutoring module. The clustering module discovers groups of students that interact with the system in a similar manner. This information, combined with the results of the SPM module, is used to provide useful hints to students on which units in the domain to learn next.

The presented method was tested on four different knowledge domains, over the least two years. During this time we identified some aspects of this method that can be improved. First, the method tends to choose a model with two clusters when most of the students have completed the domain, because the strongest difference exists between inactive students and all other students. We will resolve this problem by removing all inactive students from the dataset up front and placing them in a separate, additional cluster graded zero. Second, we will try to implement outlier detection before executing the clustering algorithm. Thirdly, we will develop a visualisation of centroid valued of all obtained cluster models and present them to a teacher to test whether they would select the same model that our method selected based on the silhouette index. Lastly, other clustering algorithms with different distance measures will also be used and their results compared to the ones produced by the algorithm used in this research.

Acknowledgments

This work was supported in part by project "Enhancing the efficiency of an e-learning system based on data mining", code: 13.13.1.2.02., funded by the University of Rijeka, Croatia.

References

Brusilovsky, P. (1999). Adaptive and intelligent technologies for web-based education. In: Rollinger C. et al. (Eds.) Künstliche Intelligenz, Special Issue on Intelligent Systems and Teleteaching, 4, 19-25.

Bouchet, F., Harley, J. M., Trevors, G. J. & Azevedo, R. (2013). Clustering and Profiling Students According to their Interactions with an Intelligent Tutoring System Fostering Self-Regulated Learning. Journal of Educational Data Mining, 5(1), 104-146.

Bostock, M. (2015) D3 JavaScript Library. Retrieved August 10, 2015 from http://d3js.org/.

Fernandes, A., Peralta, D., Benitez, J. M. & Herrera, F. (2014). E-learning and educational data mining in cloud computing: an overview. International Journal of Learning Technology, 9(1), 25-52.

Fournier-Viger, P. et al. (2014). SPMF: a Java Open-Source Pattern Mining Library. Journal of Machine Learning Research, 15, 3389-3393.

Gordon, A. D. (1999). Classification, 2nd ed., Chapman & Hall/CRC, Boca Raton, Florida.

Hall, M. et al. (2009). The WEKA Data Mining Software: An Update. SIGKDD Explorations, 11(1), 10-18.

Jugo, I., Kovačić, B. & Slavuj, V. (2013). A proposal for a web based educational data mining and visualization system. In ITIS2013: International Conference on Information Technologies and Information Society, Novo Mesto, 59-64.

Jugo, I., Kovačić, B. & Slavuj, V. (2014). Using data mining for learning path recommendation and visualization in an intelligent tutoring system. In MIPRO2014: Information and Communication Technology, Electronics and Microelectronics, 37th International Convention on, Opatija, 924-928.

Jugo, I., Kovačić, B. & Slavuj, V. (2015). Integrating a Web-based ITS with DM tools for Providing Learning Path Optimization and Visual Analytics. In EDM2015: Proceeding of the 8th International Conference on Educational Data Mining Madrid, Madrid, 574-575.

Kaufman, L. & Rousseeuw, P. J. (1990). Finding Groups in Data. An Introduction to Cluster Analysis. Wiley-Interscience, New York.

Koedinger, K. R., McLaughlin, E. A. & Stamper, J. C. (2012). Automated student model improvement. In EDM2012: Proceedings of the 5th International Conference on Educational Data Mining, Chania, 17–24.

Kovačić, B & Skočir, Z. (2003). Development of distance learning system based on dialogue. In EUROCON 2003: Computer as a Tool. The IEEE Region 8, Romania, 224-228.

Kovačić, B. & Jugo, I. (2009). Applying a Distance Learning System Based on Dialogue in e-commerce. In MIPRO2009: Proceedings of 32nd International Conference on information and communication technology, electronics and microelectornics, Opatija, 36-39.

Kovačić, B., Jugo, I. & Slavuj, V. (2012). Improvement of system for distance learning based on dialogue by appliance of statistical analysis. In MIPRO2012: Proceedings of the 35th International Convention, Opatija, 1475-1478.

Lynch, C. et al. (2006). Defining Ill-Defined Domains; A literature survey. In ITS2006: Proceedings of Intelligent Tutoring Systems Ill-Defined Domains Workshop, Taiwan, 1-10.

Milligan, G. W. (1989). A validation study of a variable weighting algorithm for cluster analysis. Journal of Classification, 6, 53-71.

Perera, D., Kay, J., Koprinska, I., Yacef, K. & Zaiane, O. R. (2007). Clustering and Sequential Petern Mining of Online Collaborative Learning Data. Knowledge and Data Engineering, IEEE Transactions on, 21(6), 759-772.

Romero, C. and Ventura, S. (2007). Educational Data Mining: a Survey from 1995 to 2005. Expert Systems with Applications, 1(33), 135-146.

Romero, C. et al. (2007). Personalized links recommendation based on data mining in adaptive educational hypermedia systems. Creating New Learning Experiences on a Global Scale. Springer Berlin Heidelberg, 292-306.

Romero, C., Castro, C. & Ventura, S. (2013). A Moodle Block for Selecting, Visualizing and Mining Students' Usage Data. In EDM 2013: Proceedings of the 6th International Conference on Educational Data Mining, 400-401.

Romero, C. & Ventura, S. (2010). Educational data mining: a review of the state of the art. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 40(6), 601-618.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics, 20, 53-65.

Symons. M. J. (1981). Clustering criteria and multivariate normal mixtures. Biometrics, 37, 35-43.