Analysis of clustering algorithms for group discovery in a web-based intelligent tutoring system

D. Bunić*, I. Jugo*, B. Kovačić*

* University of Rijeka Department of Informatics, Rijeka, Croatia
{dbunic;ijugo;bkovacic}@inf.uniri.hr

Abstract - DITUS is a web-based intelligent tutoring system developed and used at our institution as an additional learning platform. To make the system even more adaptive we expanded its architecture with several modules that perform educational data mining tasks such as clustering, to discover groups of students that use the system in a similar manner, and high-utility sequential pattern mining to discover efficient learning paths through the knowledge domain. The results of these modules enable the system to offer hints to students on which knowledge units to learn before or after the currently selected unit. One of the main pre-conditions of the quality of hints is the clustering phase in which we discover groups of students that are using the system in a similar manner in terms of learning activity and efficiency. In this paper, we analyze the results of several well-known clustering algorithms on our datasets, to determine which one is best suited for the needs of our system.

Keywords - intelligent tutoring system, clustering, educational data mining

I. INTRODUCTION

Intelligent tutoring systems (ITSs) are valuable teaching tools not only for distance education but also as a complementary teaching/learning tool in traditional (face-to-face) education. They are often used in teaching 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 [1], especially for teaching in ill-defined domains [2]. 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. At our institution, we developed one such system, called the Department of Informatics TUition System (DITUS) to serve as an additional learning platform.

Our web-based intelligent tutoring system (WITS), described first in [3] and in [4], 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 [5]. To improve the system’s overall efficiency further, we proposed a new architecture that added educational data mining (EDM) features like clustering and sequential pattern mining (SPM) [6][7].

A valuable source of data for our system lies in the records of student interactions with the system. By interactions, we mean all the students actions within the system that we store in the interactions database such as: log-in time (starting point for measuring time spent), log-out time (stop point), performing a learning action, performing a repetition action (explained later in this paper). We apply educational data mining (EDM) processes [8][9][10] 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 one was an integration layer [11] that creates a continuous communication channel to DM tools Weka [12] and SPMF [13] which are used to execute DM algorithms on data gathered within our ITS. Second was the module for clustering students based on their activity and learning progress. Finally, a SPM [14] module that is used to discover frequent patterns (FPs) students take through the knowledge domain.

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 and speed of advancement) levels. In order to help guide the students during their progress through the knowledge domain the system first clusters students that use the system in a similar way and consequently determines which cluster represents the highest achieving students, average achieving ones, etc. Then the SPM module finds productive learning paths through the knowledge domain for each group/cluster of students. Finally, the tutoring module offers students hints on which KU to learn next or before a selected one. 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.

An important aspect of this system is that it performs automated student model improvements [15] 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.

II. RELATED WORK

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. In [16] Milligan published an overview of the clustering analysis critical steps. In cluster analysis, the fundamental problem is to determine the optimal number of clusters, which has a deterministic effect on the clustering results. This well-known optimization problem has received significant attention. A variety of methods for this problem have been analyzed by Gordon [17], 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, $^G \hat{g}$, 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 analyzed by Tibshirani et al.[18] and Symons [19]. Our method implements the silhouette statistic approach.

Once a clustering model is selected, it can be evaluated through various statistical methods [20] or a number of other, more complex methods [17], 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.

III. RESEARCH ENVIRONMENT AND METHODOLOGY

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. Students that learn offline can skew the results somewhat because they appear as highly efficient. Honor students mostly display this behavior, which would be highly efficient even if they did not learn offline. This is not a great concern because the main goal of the system is to help struggling students find a more efficient learning path through the knowledge domain. The best students will always create the smallest number of interactions.

To be able to engineer features that will exactly model the way the students interact with the system we need to extract the raw data from the database of student interactions using SQL queries. After we acquired the data, we engineered four features that represent each student. The process of feature engineering and the exact mathematical expressions we used are beyond the scope of this paper but some explanations are provided in the rest of this chapter.

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.

The (L)earning feature is engineered from a number of raw data values extracted from the database of all student interactions with the system. It’s basis is the number of learning actions (one of the main functionalities of the systems) a student has performed up to this point. A learning action consists of: a) reading the learning materials and b) answering one question about each of the KU’s that are one level below the current one in the knowledge domain hierarchy. In this way, we check if the student understands the underlying concepts. If the student answers incorrectly, he/she is redirected to learning about that KU and the process begins again. In the engineered feature L, the number of learning actions is calculated in relation to the number of KUs completed (learned) and number of KUs started (not completed but completeness index > 0) at that moment.

The second feature – (R)epetition – is based on the number of repetition actions, which is the other main functionality of the system. A repetition action consists of selecting a KU and answering questions related to this KU, without presenting the learning materials. This functionality enables students to complete a KU faster, especially it they have previous knowledge about it. A similar calculation as described for the L feature is performed to get the engineered feature value R.

The (T)ime feature is based on the sum of time spent interacting with the system, also calculated in a way that takes into account the number of KUs completed and started at the moment the EDM process of clustering is started. While the activity features represent the way a student interacts with the system, the fourth, effectiveness feature (E) represent his/her success in answering questions about each knowledge unit being learned within our system. Before sending them to Weka, the values of the mentioned features are standardized[21] which results in average values being close to zero, the lowest being negative and so on.

As mentioned in the introductory part of the paper, a new module called “the integration layer” has been developed earlier 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 analyzed 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 using the Euclidian distance. Upon completion, the algorithm returns a set of clustering models with rising number of clusters k=2, 3, etc., to the DITUS system.

Clustering is usually observed from different standpoints having in common the notion of grouping or
discovering groups in a dataset. In general, clustering can be seen as a multi-objective optimization problem, therefore many different approaches can be used to solve it, e.g. connectivity based (hierarchical), centroid based (kMeans), distribution based (Expectation Maximization), density based, etc. The goal of this paper was to test different clustering algorithms in order to determine the best algorithm for the needs of our system. To do so we considered all the clustering algorithms offered by Weka. From the eight possible algorithms, we selected four:

- kMeans (Euclidian distance)
- kMeans (Manhattan distance)
- Farthest First
- Expectation Maximization (EM)

The distance measure used in kMeans can affect the obtained results. The Euclidian distance is one that is most commonly used (it is also the default option in Weka) and it is based on the Pythagorean Theorem. Its main drawback is that if one of the input features has a relatively large range it can overpower other features in the dataset. Manhattan distance is the sum of the absolute differences of coordinates of two points. It should give results that are more robust.

Farthest-first traversal of a bounded metric space is a sequence of points in the space, where the first point is selected arbitrarily and each successive point is as far as possible from the set of previously selected points. The same concept can be applied to a finite set of geometric points, by restricting the selected points to belong to the set or equivalently by considering the finite metric space generated by these points. For a finite metric space or finite set of geometric points, the resulting sequence forms a permutation of the points, known as the greedy permutation. It is often adapted for creating clustering models by minimizing the maximum diameter of a cluster.

In statistics, an expectation–maximization (EM) algorithm is an iterative method to find maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models, where the model depends on unobserved latent variables. The EM iteration alternates between performing an expectation (E) step, which creates a function for the expectation of the log-likelihood evaluated using the current estimate for the parameters, and a maximization (M) step, which computes parameters maximizing the expected log-likelihood found on the E step. These parameter-estimates are then used to determine the distribution of the latent variables in the next E step. EM is often used to create clustering models especially if the features have a Gaussian-like distribution.

We used one dataset from our DITUS system consisting of 32 rows representing 32 students with student ID value and another 4 columns (aforementioned feature values) modeling them. Weka was setup to ignore the student ID column. Here is the dataset in ARFF[22] format (the native format for the Weka software package):

```
@relation c_39_7_model_99c3ffd3b464f8190432

@attribute STUDENT_ID numeric
@attribute LEARNING numeric
@attribute REPETITION numeric
@attribute TIME numeric
@attribute EFFICIENCY numeric

@data
461,-0.65,0.75,-0.52,0.71
457,-0.43,-0.04,-0.36,0.75
442,1.49,-0.9,0.46,0.49
427,-0.68,0.22,1.11,0.73
454,-1,-0.96,-1.6,0
420,0.14,-0.87,0.25,0.72
455,-0.88,0.22,-0.76,0.72
447,0.51,-0.48,-0.33,0.46
465,0.2,-0.93,0.2,0.71
446,0.33,1.52,0.67,0.38
417,1.06,-0.9,1.18,0.46
438,1.14,-0.75,0.3,0.29
452,0.49,-0.93,-0.56,0.43
473,-1,-0.96,-1.6,0
458,-0.01,-0.16,-0.92,0.5
414,0.2,-0.9,0.56,0.72
503,3.69,0.4,1.49,0.47
419,-0.89,0.61,0.28,0.5
476,0.18,-0.96,-1.09,0.75
501,0.2,0.24,1.45,0.27
444,-0.71,1.61,1.59,0.6
459,1.78,-0.84,0.49,0.38
467,-0.56,1.84,-0.35,0.21
453,-0.58,0.87,-0.01,0.54
460,-0.54,2.9,-0.4,0.25
413,0.54,-0.93,2.19,0.55
433,-1,-0.96,-1.6,0
430,0.08,-0.93,-0.84,0.64
423,-0.82,0.25,-1.05,0.76
498,-0.7,0.99,0.52,0.41
466,-0.74,0.52,0.76,0.24
443,-0.82,0.43,-1.49,0.91
```
Where needed, we manually set the expected number of clusters to 2, 3, on to 7 (that is the current maximum number of clusters in the DITUS system) and run the algorithm. The results with the added cluster affiliation column where copied to an Excel file and joined with the raw values that we basede the four engineered features on – the number of Learning actions, Repetition actions, Time spent and Efficiency. The reason for that was to facilitate the interpretation of results. Parts of the acquired results are displayed in Figure 1 (results for k=3) and Figure 2 (results for k=5). The first set of columns represent the results from kMeans (Euclidian), the second one kMeans (Manhattan), the third one Farthest First and the last one EM. The first column represents students ID, the second one the cluster affiliation, and the next four represent the raw values of learning (L), repetition (R), time spent learning (T) and efficiency (E) acquired from the database.

Finally, we compared and analyzed the results.

IV. DISCUSSION

The results displayed in Figure 1, show that kMeans with Euclidian distance grouped students that had used Learning and Repetition in a balanced way, had spent less Time using the system and had higher Efficiency in the cluster marked “0”. Students with higher number of Learning and Repetition actions that took more Time thus having lower Efficiency where grouped in the second cluster. The students that almost exclusively used the Learning functionality (which takes longer to advance through the knowledge domain) in the third. There are some outliers in each cluster. Outlier values of engineered features usually appear when students brute force the system, e.g. use the repetition functionality to answer questions without reading them, until the question appears enough times that they guess the answer. The same question does not appear consecutively but the questions database is limited so with enough tries the same question will appear eventually. Outlier detection in the engineered dataset BEFORE the clustering step is a feature we are currently developing and adding to our system.

In the context of this analysis we use the term “outlier” to describe a student (a set of four values) that stands out from other value sets in the cluster.

When compared to the second column (the results of kMeans with Manhattan distance) we can see that there number of members is similar but there is almost no overlap – most students are assigned to different clusters and the patterns are less clear than in the first column. In the third column, the results of FF show a clear separation of inactive students (cluster 2) and slow students (cluster 1) while all the students in between make up cluster 0 (with some outliers).

It is very important for our system to discover inactive students as well as slow students correctly because this will positively affect the hint generation. In the fourth column, the results of EM again show the correct grouping of inactive students.

The members of cluster “0” could be described as group of students that used the system in a balanced way (learning and repetition functionalities) while cluster “1” consist of users that favored the slower Learning functionality. Students that use the system in a balanced way tend to have higher efficiency and less time spent in the system. That is why this group represent our “ideal” student. To correctly identify this group is the best result we expected from the clustering phase.

The results displayed in Figure 2. show the first time kMeans with Euclidian distance correctly identified inactive students (it did not succeed in this for k=2,3 and 4) and grouped them in cluster “3”. We can also see clear patterns – cluster “0” can be described as “ideal” (balanced, fast, highly efficient), cluster “1” as “balanced but slower”, cluster “2” as “unbalanced” (favors the Learning functionality), and cluster “5” as “unbalanced but efficient” (high number of correct answers). Overall, the results represent clear(er) patterns than the results in Figure 1. In reality, the DITUS system most often selects models with k=3, but in the future we plan to enable the teacher to be able to intervene in this step. The second column the patterns are also quite clear with some outliers. The third column displays very different results with much less clarity except the “inactive” and “unbalanced” clusters “2” and “3”. These results would not help our tutoring module in providing efficient hints for most of the students. The fourth column (EM) has the exact same results as in the first (kMeans with Euclidian distance). Overall, when considering all results (for k=2,…,7) we can conclude that EM displayed the most robust and clear results, grouping students with much clearer usage patterns. The algorithm currently being used in the DITUS system – kMeans with Euclidian distance was second in our list of overall results. FF had clearer results than kMeans with Manhattan distance measure.

V. CONCLUSION

In this paper, we gave a short introduction to the architecture of the DITUS system which has been under development for a number of years. The clustering phase, which precedes the SPM phase and the hint generation, is of crucial importance for the quality of hints presented to students. If we do not succeed in discovering groups of students that use the system in distinct ways the performance of our hint selection algorithm will be diminished. Our results show that the algorithm we use (kMeans with Euclidian distance) has good overall results, surpassed only by the EM algorithm. In our future work we will perform further research on more datasets and try EM in the following semester and evaluate the final results.
Figure 1. The results for k=3

ID = STUDENT_ID, K = NUMBER OF CLUSTERS, UCENJ = LEARNINGS (RAW VALUE), PON = REPETITIONS (RAW VALUE), VRIJ = TIME (RAW VALUE (SECONDS)), EFIK = EFFICIENCY
### Table 1: Results for k=5

<table>
<thead>
<tr>
<th>ID</th>
<th>K</th>
<th>UČENJ.</th>
<th>PON.</th>
<th>VRIJ.</th>
<th>EFIK.</th>
</tr>
</thead>
<tbody>
<tr>
<td>429</td>
<td>0</td>
<td>43.00</td>
<td>41.00</td>
<td>72.00</td>
<td>0.76</td>
</tr>
<tr>
<td>443</td>
<td>0</td>
<td>44.00</td>
<td>47.00</td>
<td>14.00</td>
<td>0.91</td>
</tr>
<tr>
<td>455</td>
<td>0</td>
<td>29.00</td>
<td>40.00</td>
<td>110.00</td>
<td>0.72</td>
</tr>
<tr>
<td>457</td>
<td>0</td>
<td>134.00</td>
<td>31.00</td>
<td>163.00</td>
<td>0.75</td>
</tr>
<tr>
<td>461</td>
<td>0</td>
<td>83.00</td>
<td>58.00</td>
<td>142.00</td>
<td>0.71</td>
</tr>
</tbody>
</table>

**Legend:**
- ID = STUDENT_ID
- K = NUMBER OF CLUSTERS
- UCENJ = LEARNINGS (RAW VALUE)
- PON = REPETITIONS (RAW VALUE)
- VRIJ = TIME (RAW VALUE (SECONDS))
- EFIK = EFFICIENCY

Figure 2. The results for k=5
REFERENCES


