Clustering Student Sequential Trajectories Using Dynamic Time Warping

Shitian Shen
Department of Computer Science
North Carolina State University
Raleigh, NC 27695
sshen@ncsu.edu

Min Chi
Department of Computer Science
North Carolina State University
Raleigh, NC 27695
mchi@ncsu.edu

ABSTRACT

One of the most challenging tasks in the field of Educational Data Mining (EDM) is to cluster students directly based on system-student sequential moment-to-moment interactive trajectories. The objective of this study is to build a general temporal clustering framework that captures the distinct characteristics of students' sequential behaviors patterns, that tracks whether a student's learning experience is *unprofitable*, and can identify such an individual as early as possible so personalized learning can be offered. The central idea of our framework is based on Dynamic Time Warping (DTW), which calculates distance between any two temporal sequences even with different lengths. In this paper, we explore both the original DTW and our proposed normalized DTW to generate distance matrix and apply Hierarchical Clustering to the resulted distance matrix. To fully evaluate the power of our temporal sequential clustering framework, we calculate distance matrix at three types of granularity in the increasing order of: problem, level, and session across three training datasets. As expected, results show that clustering moment-to-moment temporal sequences at problem granularity is more effective than level and session granularity. In addition, our proposed normalized DTW is more effective than both original DTW and the baseline Euclidean distance.

Keywords

Clustering, distance matrix, dynamic time warping

1. INTRODUCTION

The impetus for the development of many Intelligent Tutoring Systems (ITSs) was the desire to capture the effective learning experience provided by human one-on-one instruction. ITSs have shown positive impact on learning but the degree of their effectiveness often depends on individual student's motivation, incoming competence, etc. In ITSs, the system-student interactions can be viewed as a sequential action-response process. Each of these interactions will affect the system-student's subsequent interactions. As one of the great promises of ITS is to support personalized learning [15], the system-student moment-to-moment interactive trajectories often have vastly different lengths while most existing clustering approaches including K-means and Hierarchical Clustering are not designed to directly handle such temporal sequential datasets. Therefore, the main objective of this research is to build and evaluate a general clustering framework that captures the distinct characteristics of system-students' sequential interactive behavioral patterns, that tracks whether a student's learning experience is *unprofitable*, and can identify such an individual as early as possible so personalized learning can be offered.

Previously, various clustering methods have been widely applied for different Educational Data Mining (EDM) applications such as temporally coherent clustering [7], collaborative learning [9], reading comprehension [13], handwritten coursework [4], and personalized e-learning [8]. However, as far as we know, most of the prior research has used datasets that consist of per-student feature vectors that *summarize* a student's entire interaction trajectory but do not consider the sequential nature of the interactions; or sequential data where the student's behavior is extracted as a sequence of feature vectors but the length of the sequence is *fixed*. Neither approach directly handles the moment-to-moment temporal dependency and different length of interactive trajectories. Therefore, we implement Dynamic Time Warping (DTW) [11] which calculates the distance between any two sequences of different lengths and also considers moment-to-moment dependencies.

We proposed a general temporal clustering framework that would firstly construct a specified distance matrix on the sequential dataset and then apply clustering approach on the resulted distance matrix. We tested our framework across three datasets collected in Fall 2015, Spring 2016 and Fall 2016 semesters. All participants were trained on a logic tutor named Deep Thought (DT) and they were assigned to different conditions based on how the tutor decided whether to assign a *Problem Solving* or a *Worked Example* on next problem. Two-three weeks after the training, all participants took a in-class midterm as the PostTest. Much to our surprise, empirical results showed no significant difference among different conditions on PostTest scores across all three semesters. So we explored whether our proposed

general temporal clustering framework would generate effective clusters to predict student PostTest scores. To do so, we explored three types of granularity in increasing order of problem, level and session. More specifically, a session contained a student's entire training session on the tutor which involved six levels and each level contained multiple training problems. For three types of granularity: problem granularity recorded students' problem-by-problem behaviors and thus had different lengths for different students since the number of problems that students solved on DT varied greatly: from 19 to 65; level granularity contained the sequential data with a fixed length of six, one per level, for each student; and session granularity had one single summarized feature vector for each student. In our case, we treated session granularity as the baseline for early detection and investigated the impact of different types of granularity on clustering results.

In this work, we applied three distance functions including DTW, normalized DTW and Euclidean distance, and implemented Hierarchical Clustering with four different linkage functions. Finally, we evaluated the goodness of clusters on PostTests. Our results showed that significant difference was consistently found among the discovered clusters when clustering student trajectories at problem granularity rather than level and session granularity, and the best result is found when using the first four out of six levels of trajectories rather than using entire trajectories. Therefore it suggested that using fine-grained problem granularity was more suitable for clustering student interactive trajectories than coarse-grained level and session granularity.

2. RELATED WORK

2.1 Previous Research on Clustering

Previous research has showed the value of clustering for various applications in EDM. For example, clustering has been widely used in student modeling. Yue Gong et al [3] implemented k-means on to identify clusters with distinct students' skill and then applied knowledge tracing model to model students from each cluster separately in order to detect students' knowledge level. They found that clustering had positive impact on student modeling, providing a good representation of student knowledge. Furthermore, Terry Peckham and Gord McCalla [13] utilized k-means in reading comprehension tasks and determined four different clusters based upon cognition skills including positive or negative reading, scanning or scrolling behaviors.

Relatively little research has done to directly cluster student trajectories. Generally speaking, most of the prior research used either per-student feature vectors or the sequential data with fixed length on such task. For the former case, Ke Niu et al [12] extracted the feature vector per learner through analyzing his/her behavior and then applied spectral clustering algorithm to classify students' performance in order to provide benefit for personalized services. They categorized students' performance into nine classes and evaluated clustering results based on accuracy. Similarly, Gholam Montazer [10] proposed hybrid clustering method to group learners in E-learning systems and evaluated clustering results by comparing clustering labels with the ground truth labels.

For using sequential data but with fixed length, Severin Klingler et al [7] designed a pipeline for evolutionary clustering on student behavior sequential data with fixed length in order to group students at any time point and to identify the change of clusters over time. Particularly, Markov Chain model is applied to transfer the original behavior data as well as to capture the moment-to-moment temporal dependency. The optimal number of clusters is selected based upon the best model, evaluated by Akaike information criterion (AIC). Different from this work, we try to clustering the sequences with different lengths.

2.2 Application of DTW

DTW has been successfully applied to a variety of applications related to time series data, such as time series indexing [6], classification [14] and clustering in domains of astronomy, speech physiology, and medicine [1]. More specifically, Hesam Izakian et al [5] applied fuzzy clustering with DTW distance approach on UCR time series data sets and evaluated the performance of clustering methods based on precision value. In addition, Gançarski, Pierre et al [2] utilized DTW to capture the semantic proximity between urban blocks on spatial temporal topographic databases and implemented ascendant Hierarchical Clustering to detect the distinctive evolutions of urban blocks. Furthermore, Nurjahan Begum et al [1] explored DTW by adding pruning strategies and did the multidimensional time series clustering on different types of data sets in astronomy, speech physiology, medicine, entomology and astronomy domains. They evaluated performance of clustering approaches in term of accuracy.

As far as we know, this is the first study of applying DTW to the field of EDM by directly clustering student-system interactive sequential trajectories. Given the special nature of EDM, we further propose normalized DTW and find that normalized DTW is more effective to our task than original DTW.

3. METHODOLOGY

In this section, we first introduce the original and the proposed normalized DTW for calculating the distance matrix between any pair of student interactive trajectories, and then describe how we apply Hierarchical Clustering to identify clusters with distinctive behavior pattern and performance.

3.1 Distance Function

3.1.1 Dynamic Time Warping (DTW)

Given sequences $X = \{x_1, x_2, ..., x_N\}$ and $Y = \{y_1, y_2, ..., y_M\}$ with different lengths ($N \neq M$), a warping path W is an alignment between X and Y, involving *one-to-many* mapping for each pair of elements. The cost of a warping path is calculated by the sum of cost of each mapping pair. Furthermore, warping path contains three constraints: 1) *Endpoint constraint:* The alignment starts at pair $(1, 1)$ and ends at pair (N, M); 2) *Monotonicity constraint:* The order of elements in the path for both X and Y should be preserved same as the original order in X and Y respectively; 3) *Step size constraint:* the difference of index for both X and Y between two adjacent pairs in the path need to be no more than 1 step. In other words, pair (x_i, y_j) can be followed

by three possible pairs including (x_{i+1}, y_j), (x_i, y_{j+1}) and (x_{i+1}, y_{j+1}).

Dynamic Time warping (DTW) is a distance measure that searches the optimal warping path between two series. Particularly, we firstly construct a cost matrix C where each element $C(i,j)$ is a cost of the pair (x_i, y_j), specified by using Euclidean, Manhattan or other distance function. DTW is calculated based on dynamic programming. Initial step of DTW algorithm is defined as

$$DTW(i,j) = \begin{cases} \infty & \text{if } (i=0 \text{ or } j=0) \text{ and } i \neq j \\ 0 & \text{if } i=j=0 \end{cases}$$

The recursive function of DTW is defined as

$$DTW(i,j) = \min \begin{cases} DTW(i-1,j) + w_h \cdot C(i,j) \\ DTW(i,j-1) + w_v \cdot C(i,j) \\ DTW(i-1,j-1) + w_d \cdot C(i,j) \end{cases}$$

Where w_h, w_v, w_d are weight for horizontal, vertical and diagonal direction respectively. $DTW(i,j)$ denotes distance or cost between two sub sequences $\{x_1, ..., x_i\}$ and $\{y_1, ..., y_j\}$, and $DTW(N,M)$ indicates total cost of the optimal warping path.

In equally weighted case $(w_h, w_v, w_d) = (1,1,1)$, the recursive function has the preference on diagonal alignment direction because the diagonal alignment takes one-step cost while the combination of a vertical and a horizontal alignment takes two-steps cost. In order to counterbalance this preference, we can set $(w_h, w_v, w_d) = (1,1,2)$.

3.1.2 Normalized DTW

One potential issue of using the original DTW definition is that the longer the two sequences are, the larger their DTW value will be. Thus, its absolute value may not truly reflect the difference of the two sequences. Thus, we propose the normalized DTW, defined as dividing original DTW by the sum of lengths of two sequences as shown below:

$$DTW_{norm}(N,M) = \frac{DTW(N,M)}{N+M}$$

Each alignment in the warping path has a corresponding weight, selected from (w_h, w_v, w_d) and the sum of weights for all alignments equals to the sum of lengths of two sequences $(N+M)$. Therefore, the normalized DTW evaluates the average distance of alignments in the warping path for two sequences. We will empirically compare the effectiveness of the original DTW and our proposed normalized DTW.

3.2 Hierarchical Clustering

Our proposed framework uses Hierarchical Clustering because K-means cannot directly applied here. K-means needs to calculate the centroid of each cluster while we only have the DTW-based distance for each pair of trajectories.

To apply Hierarchical Clustering, we explore four linkage functions: average, median, complete and ward, which determine how to merge clusters based on the distance between the clusters. Our results show that the first three linkage methods generate extremely unbalanced clusters while the ward linkage discovers relatively balanced ones. Therefore, in the following, we will report our results using ward linkage only.

The optimal number of cluster is selected based upon the measurement called WCSS (*within cluster sum of squares*) [16] defined as

$$W(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(i')=k} d(x_i, x_{i'})$$

Our results show that the optimal number of clusters is 4.

4. EXPERIMENT

4.1 Training Datasets

Our datasets were collected by training students on a logic DT tutor across three semesters: 2015 Fall, 2016 Spring and 2016 Fall referred as DT15F, DT16S and DT16F respectively. For each semester, students were randomly assigned into different conditions based on the pedagogical strategies employed by the tutor. *Pedagogical strategies* were policies used to decide whether give Problem Solving (PS) or Worked Examples (WE) as the next problem. In WE, students were given a detailed example showing the expert solution for the problem. In PS, by contrast, students were tasked with solving a particular problem. For different versions of DTs, we applied different types of data-driven approaches to induce pedagogical strategies [15]. There were a total of four, six and five conditions for DT15F, DT16S and DT16F respectively. One-way ANOVA results showed that there was no significant difference on PostTest scores among conditions across all three semesters: $F(158,1) = 0.728, p = 0.537$ for DT15F, $F(196,1) = 0.644, p = 0.667$ for DT16S and $F(188,1) = 0.445, p = 0.776$ for DT16F. More details were eliminated due to the limitation of space. While no significant was found among different conditions, different pedagogical policies resulted in quite different student-system interactive trajectories and our goal was to investigate whether the proposed temporal clustering framework would be more effective to predict PostTest scores and to discover the true temporal patterns during student training than the condition.

To best describe student learning trajectory, we considered the following 36 continuous features which could be grouped into three categories:

1 **Autonomy (AM):** the amount of work done by the student: such as the number of problems solved so far (*PSCount*) or the number of hints requested (*hintCount*).
2 **Temporal Situation (TS):** the time related information about the work process: such as the average time taken per problem (*avgTime*), or the total time for solving a problem (*TotalPSTime*).
3 **Student Action (SA):** the statistical measurement of student's behavior: such as the number of non-empty-click actions that students take (*actionCount*), or the number of clicks of applying rules for logic proof (*AppCount*).

To fully evaluate our proposed framework, we explored three types of granularity: 1) **Problem granularity** considered students' behaviors problem by problem. When training on DT, the number of problems that each student solved differed greatly and as a result, the length of student interactive sequences varied. For example, about 8%, 4% and 1% of students had more than 40 problems in their interac-

DT	Level	Hierarchical Clustering with Ward Linkage					
		Problem		Level			Session
		Normalized DTW	DTW	Normalized DTW	DTW	Euclidean	Euclidean
DT15F	3	**5.05(.027)***	**6.48(.012)***	3.84(.051).	1.40(.238)	1.30(.256)	2.26(.135)
	4	**10.3(.001)****	1.18(.279)	**5.86(.017)***	**7.67(.006)****	0.75(.388)	2.96(.087).
	5	**6.06(.015)***	1.71(.193)	**4.03(.046)***	2.55(.112)	1.93(.166)	1.81(.181)
	6	3.19(.076).	1.37(.244)	3.79(.053).	0.50(.480)	1.21(.272)	0.76(.385)
DT16S	3	**12.4(.000)*****	0.63(.427)	**8.94(.003)****	1.89(.171)	0.41(.521)	1.00(.318)
	4	**13.7(.000)*****	0(.995)	**10.0(.002)****	2.49(.117)	0.99(.319)	0.38(.536)
	5	**7.1(.008)****	0.84(.359)	**6.36(.013)***	3.75(.054).	0.39(.532)	**15.8(.000)*****
	6	3.11(.079).	0.05(.821)	0.53(.466)	0.67(.412)	0.06(.806)	**8.33(.004)****
DT16F	3	0.28(.594)	0.96(.328)	0.94(.333)	2.38(.124)	0.89(.344)	2.90(.090).
	4	**3.93(.049)***	1.97(.163)	2.61(.108)	3.32(.070).	0.52(.471)	1.14(.288)
	5	**4.76(.030)***	3.64(.058).	2.64(.058).	1.74(.189)	1.65(.201)	0.06(.798)
	6	**3.95(.048)***	**9.67(.002)****	2.27(.134)	1.92(.168)	1.27(.261)	0.0(.997)

Note: significant codes: 0.000 :'****'; 0.001: '***'; 0.01: '**'; 0.05: '*'; 0.1: '.'

Table 1: One way ANOVA using PostTest score as dependent measure and cluster as a factor

tive sequences in DT15F, DT16S and DT16F respectively. 2) **Level granularity** summarized students' behaviors for each level as a singe feature vector; since DT has six levels, the length of level interactive sequence is six for each student. 3) **Session granularity** summarized the students' entire training behaviors by a single feature vector.

Furthermore, there were 158, 196 and 188 students that participated in DT15F, DT16S and DT16F respectively. Combining semesters with three types of granularity, we had a total of 9 data sets.

4.2 Data Preprocessing

Our data-preprocessing involved two steps: 1) *Standardization.* To ensure that our state features measured at different scales would contribute equally to the distance functions, we standardized all features by subtracting mean and dividing standard deviation; 2) *Principle Component Analysis* (PCA), which is widely used for dimensionality reduction. PCA is able to generate mutually independent principle components (PCs) which cover the majority of variance information. We selected PCs with the corresponding variance larger than 1, thus 6-8 PCs were chosen for different training data sets.

4.3 Clustering Process

While most of previous clustering research on sequential trajectory used the entire trajectory, we investigated whether it was more effective to only use sub-sequential trajectories rather than the entire trajectories. This was especially important because we wanted to identify students with different learning patterns, especially the students with *unprofitable* learning as early as possible so personalized learning could be offered.

To do so, we recursively generated our nine training datasets, three types of granularity across three semesters, using subsequential trajectories from the beginning of the training up to each of the six levels separately. For example, 'Level4-Problem-DT16S' training dataset was generated by using problem-by-problem trajectories from the beginning of training process up to level 4 using DT16S. Then we followed the following three steps:

Distance matrix. We explored three types of distance matrices: DTW, normalized DTW and Euclidean distance. Euclidean distance was used as the baseline here.

Outlier Detection. Given that many clustering methods are often sensitive to outliers, we applied filtering approach to remove them from our training data. More specifically, for each type of distance matrix, we calculated the average distance for each student to all others and then obtained the mean μ and standard deviation σ for all students' average distances. We filtered out students whose average distances were larger than: $\mu + 2 * \sigma$.

Cluster Evaluation. We applied Hierarchical Clustering on distance matrices calculated above, and used PostTest scores to evaluate the effectiveness of the resulted clusters.

5. RESULT

As mentioned above, while the assigned condition did not seem to be a crucial factor to predict student PostTest scores, we explored whether our proposed temporal clustering framework could do better.

5.1 Cluster Evaluation

Table 1 summarized clustering results. In Table1, each row denoted clustering results of using student interactive subsequential trajectories, varying from using the first three levels up to the entire six levels. For instance, 'Level 4' used sequential data or summarized data points from the beginning of training process up to level 4. Note that we did not get good clustering results when using only the first two levels so their results were eliminated from the table. This was probably because there were a lot of noises in the first two levels as some students were still getting used to the

DT	#Student	Dependent Measures: F-ratio(p-value)						
		PostTest	Interaction	WrongApp	hintCount	avgstepTime	avgTime	TotalTime
DT15F	155	10.31(**.001**)	40.54(**.000**)	21.55(**.000**)	6.79(**.010**)	0.01(.919)	17.15(**.000**)	20(**.000**)
DT16S	190	13.69(**.000**)	47.59(**.000**)	67.47(**.000**)	99.73(**.000**)	2.77(.097)	28.76(**.001**)	36.21(**.000**)
DT16F	178	3.93(**.048**)	2.28(.133)	0.16(.691)	5.99(**.015**)	13.45 (**.000**)	0.31(.58)	0.20(.655)

DT	Cluster	Size	Dependent Measures: Mean(Standard Deviation)						
			PostTest (score)	Interaction (count)	WrongApp (count)	hintCount (count)	avgstepTime (sec)	avgTime (min)	TotalTime (hour)
DT15F	C1	47	**84.84**(21.64)	**1052**(432)	**80**(47)	**21**(34)	6.01(1.86)	**5.45**(2.31)	**1.75**(0.73)
	C2	26	76.92(26.02)	1259(662)	110(79)	44(45)	**10.88**(3.21)	11.47(5.52)	3.76(1.97)
	C3	55	72.35(28.77)	**2021**(752)	**214**(155)	**76**(61)	8.31(3.00)	**12.64**(5.14)	**4.49**(1.76)
	C4	27	**66.58**(24.49)	1706(600)	154(101)	26(30)	**5.48**(1.73)	7.88(3.28)	2.60(1.04)
DT16S	C1	112	**91.04**(16.53)	**1242**(519)	**104**(64)	**13**(12)	5.89(2.32)	**5.98**(3.60)	**2.06**(1.22)
	C2	41	83.99(23.83)	1483(660)	140(91)	22(16)	**9.37**(3.73)	10.72(4.33)	3.66(1.42)
	C3	14	**70.98**(27.14)	**2186**(551)	275(170)	39(28)	**5.04**(1.84)	8.84(4.50)	3.16(1.73)
	C4	23	78.66(26.08)	2058(994)	**278**(205)	**65**(52)	6.81(2.08)	**10.91**(8.07)	**4.05**(2.98)
DT16F	C1	40	79.61(20.67)	**1216**(500)	122(92)	17(21)	**8.76**(2.53)	9.11(5.43)	3.15(1.94)
	C2	44	88.21(16.35)	1713(867)	147(98)	16(15)	**4.19**(0.94)	**5.71**(2.93)	2.03(1.09)
	C3	35	**78.57**(25.87)	**2335**(887)	**276**(182)	**43**(34)	6.26(1.74)	**11.25**(4.62)	4.09(1.84)
	C4	59	**90.09**(13.95)	1440(528)	**116**(66)	25(28)	5.99(1.48)	7.12(3.30)	2.47(1.19)

Table 2: result of one way anova on dependent measurements for best clustering assignment

tutor. Each cell in Table 1 denoted one-way ANOVA results using PostTest score as the dependent measure and clusters as the factor in the format of *F-ratio*(*p-value*). The bold numbers showed that significant differences were found among clusters on PostTest scores. Each column represented different types of granularity using different distance functions: DTW, normalized DTW and Euclidean. For problem granularity, we only applied DTW and normalized DTW approaches because Euclidean distance could not be applied on sequential trajectories with different lengths. For level granularity, we utilized all three distance functions. Note that when calculating Euclidean distance, we first calculated distance for each level separately and then summed them up. For session granularity, all three distance functions were equivalent in that all became Euclidean distance.

Granularity Comparison. Table 1 showed that among three types of granularity, problem granularity was most suitable for clustering because significant differences were found across all three datasets and across all levels of subsequences on PostTest scores when using problem granularity. This finding was consistent with our hypothesis that directly clustering student moment-to-moment fine grained trajectories indeed provide benefit to discover the underline characteristics of student learning processes.

Distance Function Comparison. To compare the three distance functions, we only focused on the level granularity since it was the only one that involved all three distance matrices. Table 1 showed that both original and normalized DTW outperformed Euclidean distance because no significant differences were found among the clusters using Euclidean distance. To compare the two types of DTW, we focused on both problem and level granularity. Table 1 showed normalized DTW could induce more robust and consistent results than DTW. In short, among the three distance functions, our proposed normalized DTW was the best.

Sub-sequences Comparison. Table 1 showed that consistently significant results were found for all problem granularity data sets using normalized DTW and sub-sequential trajectories up to first four or five levels. Interestingly, using the entire sequential data may be not as effective as using sub-sequences in that for DT15F and DT16S datasets, no significant difference was found when using problem granularity on the entire trajectories.

Variable	Definition
PostTest	the score of student's post test
Interaction	number of student's actions
WrongApp	number of wrong application of rules
hintCount	number of hints that students take
avgstepTime	average time per step
avgTime	average time per problem
TotalTime	time of completing the training process

Table 3: Variables and Definitions

5.2 Clusters Analysis

Table 1 showed that the consistent significant results was found when we clustered on problem granularity using normalized DTW on sub-sequences from beginning of training process to the level 4 across the three semesters' datasets. Therefore, in the following, we will shed some lights on characteristics of the discovered clusters.

Table 2 showed one-way ANOVA results on seven dependent measures using clusters as the factor. Particularly, we bolded p values which were less than 0.05. We found

that there was significant difference on all variables except *avgstepTime* for DT15F and DT16S. Additionally, significant difference existed on three variables including *PostTest*, *hintCount* and *avgstepTime* for DT16F. In order to investigate how much difference existed among clusters based on selected variables, we presented the mean and standard deviation for each pair of cluster and variable in Table 2. We highlighted the mean of variables that were significantly different from others, either extremely large or small. We analyzed the difference among clusters for three semesters separately shown as follows.

1. DT15F. C1 had the highest *PostTest* while C4 had the lowest one among four clusters. C1 had the lowest *Interaction*, *WrongApp*, *hintCount* and *TotalTime* among four clusters. Although C2 and C3 had similar *PostTest*, C2 contained dramatically larger *Interaction*, *WrongApp* and *hintCount* than C3. Furthermore, C3 had the largest value of *Interaction avgTime* and *TotalTime*.

2. DT16S. C1 had the highest *PostTest* and the lowest *Interaction*, on the contrary, C3 had the lowest *PostTest* and the highest *Interaction* among four clusters. Although C2 and C4 had the closed *PostTest*, C4 contained higher *WrongApp* and *hintCount* than C2.

3. DT16F. C4 had the highest *PostTest*, while C3 had the lowest one. Although C2 performed closed to C4, C2 had higher *WrongApp* than C4. Furthermore, C1 had the lowest *Interaction* and the highest *avgstepTime* while C3 contained the highest *Interaction* and *WrongApp*.

In short, our results showed that our discovered clusters indeed had the distinctive interactive patterns and could predict students PostTest better than their assigned conditions.

6. CONCLUSIONS & FUTURE WORK

In this paper, we proposed the temporal clustering framework to directly cluster student interactive trajectories. Particularly, we explored three different distance functions and three types of granularity. Results showed that normalized DTW is the most effective function for generating distance matrix; problem granularity is more effective than level and session granularity. More importantly, through clustering statistical analysis, we were able to identify distinctive patterns among clusters during the learning process, which could provide benefit to the personalized learning. For the future work, we will modify distance matrix by combining kernel function with DTW approach given sequential data containing both continuous and discrete features in order to generate effective distance matrix.

7. ACKNOWLEDGEMENTS

This research was supported by the NSF Grant 1432156 "Educational Data Mining for Individualized Instruction in STEM Learning Environments".

8. REFERENCES

[1] N. Begum, L. Ulanova, J. Wang, and E. Keogh. Accelerating dynamic time warping clustering with a novel admissible pruning strategy. In *Process of ACM SIGKDD*, 2015.

[2] P. Gançarski, A. Puissant, and F. Petitjean. Use of symbolic dynamic time warping in hierarchical clustering of urban fabric evolutions extracted from spatiotemporal topographic databases. *AI Communications*, 2016.

[3] Y. Gong, J. E. Beck, and N. T. Heffernan. Using multiple dirichlet distributions to improve parameter plausibility. In *Educational Data Mining*, 2010.

[4] J. Herold, A. Zundel, and T. F. Stahovich. Mining meaningful patterns from students' handwritten coursework. *Proceedings of EDM*, 2013.

[5] H. Izakian, W. Pedrycz, and I. Jamal. Fuzzy clustering of time series data using dynamic time warping distance. *Engineering Applications of Artificial Intelligence*, 2015.

[6] E. Keogh. Exact indexing of dynamic time warping. In *Proceedings of VLDB*, 2002.

[7] S. Klingler, T. Käser, B. Solenthaler, and M. Gross. Temporally coherent clustering of student data. In *Proceedings of EDM*, 2016.

[8] M. Köck and A. Paramythis. Activity sequence modelling and dynamic clustering for personalized e-learning. *UMUAI*, 2011.

[9] R. M. Maldonado, K. Yacef, J. Kay, A. Kharrufa, and A. Al-Qaraghuli. Analysing frequent sequential patterns of collaborative learning activity around an interactive tabletop. In *EDM*, 2010.

[10] G. A. Montazer and M. S. Rezaei. A new approach in e-learners grouping using hybrid clustering method. In *ICEELI*, 2012.

[11] M. Müller. Dynamic time warping. *Information retrieval for music and motion*, 2007.

[12] K. Niu and Z. Niu. A coupled user clustering algorithm for web-based learning systems. In *EDM*, 2016.

[13] T. Peckham and G. McCalla. Patterns in reading comprehension tasks. *EDM*, 2012.

[14] F. Petitjean, G. Forestier, G. I. Webb, A. E. Nicholson, Y. Chen, and E. Keogh. Faster and more accurate classification of time series by exploiting a novel dynamic time warping averaging algorithm. *KAIS*, 2016.

[15] S. Shen and M. Chi. Reinforcement learning: the sooner the better, or the later the better? In *Proceedings of UMAP*, 2016.

[16] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society*, 2001.