Use of SOM for Non-Static Data

Vilson Lu [1] and Judith Azcarraga[2, *]
[1] *De La Salle University*
[2] *De La Salle University*
**Corresponding Author: judith.azcarraga@dlsu.edu.ph*

Abstract: This paper presents how Self-Organizing Map (SOM) can be used to visualize the changes of non-static data or time-series data such as brainwaves or EEG data and music data. The change of states of these data may be represented as a trajectory plotted on a SOM. Analysis of multiple trajectories may be difficult especially those with long sequences. Finding similar trajectories may also pose a challenge when trajectories being compared are of different lengths. Various similarity measures may be used to compare the similarity of behavior between trajectories. In most cases, the alignment of trajectories must be performed before applying any similarity measure formula. This paper presents the use of edit-distance similarity measure where the alignment of the trajectories is not necessary. The use of edit-distance on the EEG and music datasets had shown better results in measuring the similarity between trajectories compared to a pairwise similarity approach which requires the alignment of the trajectories.

Key Words: Self-organizing map; trajectory analysis; similarity measure

1. INTRODUCTION

Self-Organizing Map (SOM) or Kohonen Map (Kohonen, 1990) is a type of unsupervised learning approach that arranges the neurons in a grid. Each neuron adjusts their weight based on the surrounding neighborhood, which allows the group of neurons to specialize in a particular domain. SOM is an effective method for reducing the dimension of the data, by reducing the space into 2 dimensions. It preserves the space of the data in the lower dimension grid, making it a good tool for exploratory data analysis (Vesanto, 1997).

There are other ways that the SOM can be used aside from dimensionality reduction or data visualization. Some researches explore SOM as a tool for time-series analysis. In Azcarraga (J. Azcarraga, 2012) works, they used Structured SOM (SSOM) to analyze the brainwave patterns and predict the emotion of the student studying. Plotting the time-series data to SOM allowed them to analyze how time-series data change from one state to the other by observing how the time-series data behave in SOM.

Analysis of the trajectories can be quite difficult as we rely on visual comparison of the trajectory. Given two time-series data, a good similarity measure should have a reliable technique in measuring how similar they are. It should be able to let the user perceive the similarity between the two, be consistent with human intuition, and be robust. A similarity measure is said to be robust if it satisfies the following four properties: scale, warp, noise, and outliers. Scale measures how it handles amplitude shifting, warp is for time-shifting, noise is for the added white noise in the data, and outliers are for the added random data (Esling & Agon, 2012).

This paper proposes an edit-distance approach for measuring the similarity between two trajectories plotted on a SOM.

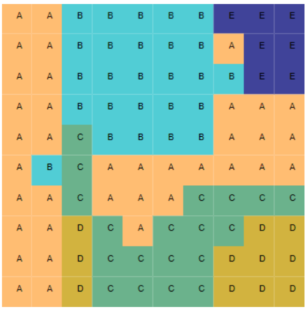
2. METHODOLOGY

2.1 Setup

Experiments were performed on two datasets to see how well the similarity measures perform on short and long trajectories. The first was the Reader's Affect (RA) dataset. The dataset contained EEG data collected from 32 participants, 23 females and 9 males while reading a short story. Their research tried to classify the emotions of the readers based on their brainwaves (Kalaw & Ong, 2017). The second dataset was the SOMphony dataset. The dataset consisted of 1-second segments from 45 symphonies. The symphonies were from five major musical periods (Baroque, Classical, 19th Century, Romantic, and 20th Century). Each period has 3 composers and each composer has 3 symphonies (A. Azcarraga & Flores, 2016).

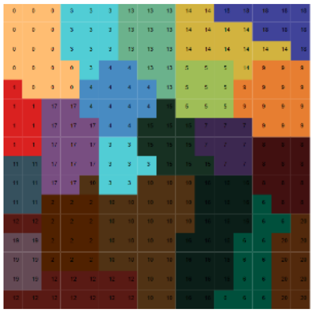
A SOM was trained for each of the datasets. For the RA dataset, the SOM was trained using only the data from the 9 male participants. Each participant was asked to read "The Veldt" by Ray Bradbury. The story was divided into 72 segments. The resulting EEG data that were collected from the participants were then split based on the segments. For the SOMphony dataset, the SOM was trained on a 1-second music segment from 45 symphonies.

After training the SOM, the nodes were clustered using K-Means Clustering. The clusters were then used. The SOM's weight will be served as inputs for the K-Means algorithm. K-Nearest Neighbor was used to labeling each node. Fig. 1 was the resulting SOM trained on the RA dataset and Fig. 2 was trained from the SOMphony dataset.



A	A	B	B	B	B	B	E	E	E
A	A	B	B	B	B	B	A	E	E
A	A	B	B	B	B	B	B	E	E
A	A	B	B	B	B	B	A	A	A
A	A	C	B	B	B	B	A	A	A
A	B	C	A	A	A	A	A	A	A
A	A	C	A	A	A	C	C	C	C
A	A	D	C	A	C	C	C	D	D
A	A	D	C	C	C	C	D	D	D
A	A	D	C	C	C	C	D	D	D

Fig. 1. SOM trained on RA dataset



0	0	0	3	3	3	13	13	13	14	14	18	18	18	18	18
0	0	0	3	3	3	13	13	13	14	14	14	14	18	18	18
0	0	0	3	3	3	13	13	13	14	14	14	14	14	14	18
0	0	0	0	3	4	4	13	13	5	5	5	14	9	9	9
1	0	0	0	4	4	4	4	13	5	5	5	9	9	9	9
1	1	17	17	4	4	4	4	15	5	5	5	9	9	9	9
1	1	17	17	17	4	4	15	15	15	7	7	7	9	9	9
1	1	17	17	17	3	3	15	15	15	7	7	7	8	8	8
11	11	17	17	17	3	3	3	15	15	15	7	7	8	8	8
11	11	17	17	10	3	3	10	10	10	16	16	16	8	8	8
11	11	2	2	2	10	10	10	10	10	16	16	16	6	8	8
12	12	2	2	2	10	10	10	10	10	16	16	16	6	6	20
19	19	2	2	2	10	10	10	10	16	16	16	6	6	20	20
19	19	2	2	2	10	10	10	10	16	16	16	6	6	20	20
19	19	12	12	12	12	12	10	10	16	16	16	6	6	20	20
12	12	12	12	12	12	12	10	10	16	16	6	6	6	20	20

Fig. 2. SOM trained on SOMphony dataset

The time series were then plotted on their respective trained SOM. To plot the time series in SOM, each instance in the time series is mapped to the closest neuron in the map, which was called the Best Matching Unit (BMU). The series of BMU is then called as the trajectory. This process was done for each time-series data to create the trajectory dataset.

The trajectory datasets were then used to measure the effectiveness of the two similarity measures. A visual check was then used to gauge the effectiveness of the similarity measures.

2.2 Similarity Measures

Two similarity measures with different approaches were compared: pairwise similarity and edit-distance approach.

In the pairwise similarity approach, the two trajectories to be compared must have the same length. To do this, the length of the longer trajectory needed to match the length of the shorter trajectory by pruning the excess points. This was done by lining up the two trajectories. The longer trajectory was cut based on the length of the shorter trajectory. After lining up the trajectories, they compared the K-Means cluster for each point. If the cluster matched in both position and cluster, it gains a score. The higher the score, the more similar the two trajectories are.

In the proposed approach, edit-distance was used to measure the similarity between two trajectories. Edit-distance approach is borrowed from strings. It measures the number of operations (addition, deletion, and substitution) needed to transform one string to another. Equation 1 shows how to calculate the similarity using edit-distance (Lipton, 2010). Edit-distance approach mainly solves the time-shifting or warping. It aligns the two strings by adding a gap element in the other strings (Chen & Ng, 2004).

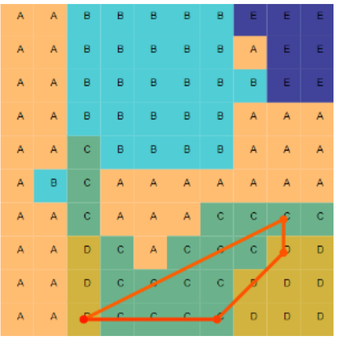
$$D(N,M) = min\begin{cases} D(i-1,j)+1 \\ D(i,j-1)+1 \\ D(i-1,j-1)+\begin{cases} 2; S(i) \neq S(j) \\ 0; S(i) = S(j) \end{cases} \end{cases} \quad (1)$$

The proposed approach used the K-means clusters of each BMU to compare the trajectories, resulting in a sequence of clusters. First, the sequence of clusters was transformed into its canonical form. For example, the cluster sequence "ABBBBC" is simplified to the string "ABBC". Note that "ABBC" is different from "ABC". The sequence "ABBC" represented that the trajectory stayed in the cluster for some time, wherein "ABC", showed that it simply passed through the cluster. The canonical string is used as the input to the edit-distance algorithm so that those trajectories with a huge difference in length would have a lesser impact on the similarity measure. The lower the score, the more similar the two trajectories were.

The trajectory datasets were used to measure the effectiveness of the two similarity measures. Given an unknown trajectory, the similarity measure must determine which among the trajectories in the dataset is the closest to the unknown trajectory. The resulting trajectories were compared to the unknown trajectory and were checked visually to gauge the effectiveness of the similarity measures.

3. RESULTS AND DISCUSSION

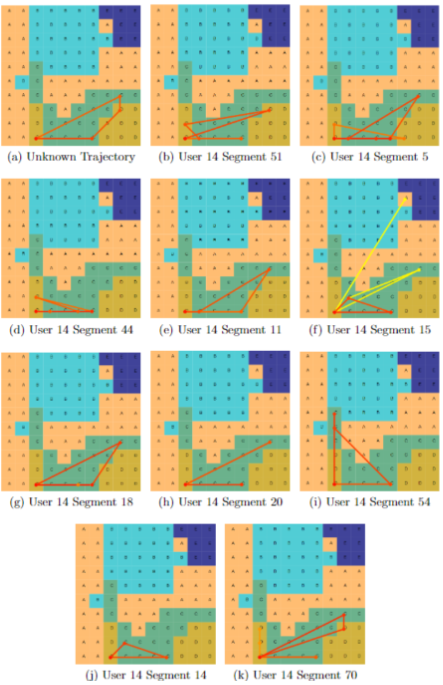
The performance of two similarity measures, Edit-Distance and Pairwise, were compared on the EEG RA dataset and music SOMphony dataset. To do this, an unknown trajectory was compared to other trajectories using each of the similarity measures. Fig. 3 showed the trajectory the similarity measure needed to compare. To show the time variable in the trajectory, the line started with a yellow color, then gradually changed to red at the end of the time series.



A	A	B	B	B	B	B	E	E	E
A	A	B	B	B	B	B	A	E	E
A	A	B	B	B	B	B	B	E	E
A	A	B	B	B	B	B	A	A	A
A	A	C	B	B	B	B	A	A	A
A	B	C	A	A	A	A	A	A	A
A	A	C	A	A	A	C	C	C	C
A	A	D	C	A	C	C	C	D	D
A	A	D	C	C	C	C	D	D	D
A	A	D	C	C	C	C	D	D	D

Fig. 3. The trajectory of User 14 Segment 23

Fig. 4 showed the top 10 trajectories based on the edit-distance approach. Based on Fig. 4, the resulting trajectories were similar to the unknown trajectories, visually. Also, the trajectories were quite similar to each other. Based on edit-distance, it could be seen that trajectories that came from the same user were considered as the most similar to the unknown trajectory.



(a) Unknown Trajectory (b) User 14 Segment 51 (c) User 14 Segment 5

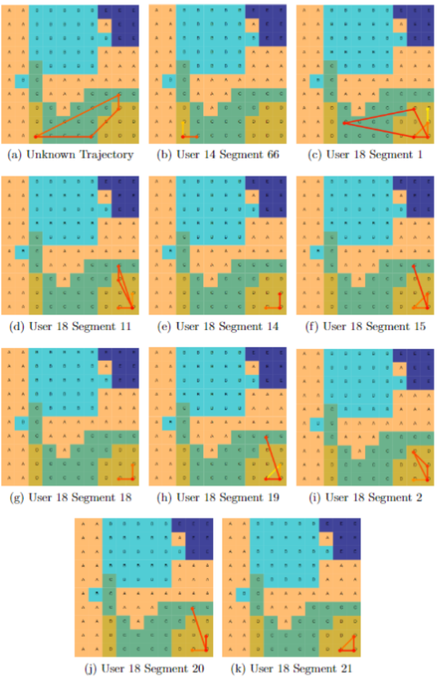
(d) User 14 Segment 44 (e) User 14 Segment 11 (f) User 14 Segment 15

(g) User 14 Segment 18 (h) User 14 Segment 20 (i) User 14 Segment 54

(j) User 14 Segment 14 (k) User 14 Segment 70

Fig. 4. Top 10 similar trajectory based on Edit-Distance approach

On the other hand, in the pairwise approach (Fig. 5), the resulting trajectories barely showed resemblance to the unknown trajectory. The trajectories mostly stayed on cluster D on the lower right side of the map. The pairwise approach considered User 18 as the most similar to the unknown trajectory as most of the trajectories in the top 10 came from User 18. The resulting trajectories tended to be simple in the sense that most of the trajectories generally tended to stay in only 1 cluster. This is because the pairwise similarity measure tends to choose long trajectories with a simple sequence. Such long but simple trajectories have a higher chance of matching the unknown trajectories.



(a) Unknown Trajectory (b) User 14 Segment 66 (c) User 18 Segment 1

(d) User 18 Segment 11 (e) User 18 Segment 14 (f) User 18 Segment 15

(g) User 18 Segment 18 (h) User 18 Segment 19 (i) User 18 Segment 2

(j) User 18 Segment 20 (k) User 18 Segment 21

Fig. 5. Top 10 similar trajectory based on the pairwise approach

For the SOMphony dataset, Mozart Symphony 41 "Jupiter" was set as the unknown trajectory as shown in Fig. 6.

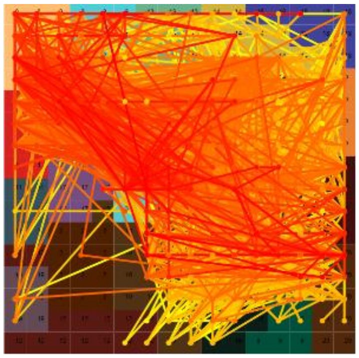


Fig. 6. The trajectory of Mozart Symphony 41

"Jupiter"

The trajectory of Mozart Symphony 41 "Jupiter" was described to have a major diagonal movement (A. Azcarraga & Flores, 2016). Fig. 7 showed the top 10 trajectories that are similar to Mozart Symphony 41. It showed that Stravinsky's Symphony in 3 Moves was considered as the most similar to the unknown trajectory and Stravinsky as the most similar to Mozart as three of his symphonies were in the top 10 results. In Azcarraga and Flores qualitative analysis (A. Azcarraga & Flores, 2016), they considered Mendelssohn, Schumann, Schubert, and Rachmaninov symphonies as the most similar to Mozart Symphony 41. Mendelssohn, Schumann, and Schubert made it to the top 10 for edit distance, except for Rachmaninov.

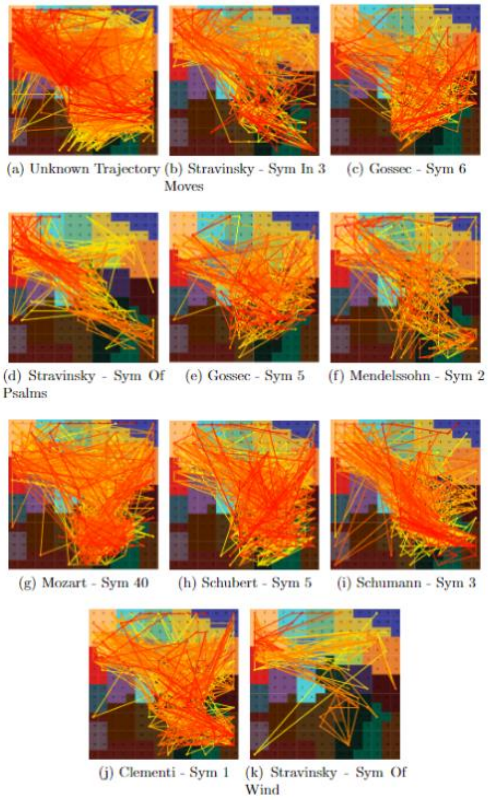

(a) Unknown Trajectory (b) Stravinsky - Sym In 3 Moves (c) Gossec - Sym 6
(d) Stravinsky - Sym Of Psalms (e) Gossec - Sym 5 (f) Mendelssohn - Sym 2
(g) Mozart - Sym 40 (h) Schubert - Sym 5 (i) Schumann - Sym 3
(j) Clementi - Sym 1 (k) Stravinsky - Sym Of Wind

Fig. 7. Top 10 trajectories similar to Mozart Symphony 41 using Edit-Distance

Fig. 8 showed the top 10 similar trajectories to Mozart Symphony 41 using the Pairwise Similarity Measure. It showed that Shostakovich – Symphony 8 "Stalingrad" was considered as the most similar trajectory to the unknown trajectory and Shostakovich as the most similar composer as three of his works were in the top 10 list. Comparing the results from Fig. 7, it could be seen that the diagonal movement was present in both the pairwise and edit-distance approach.

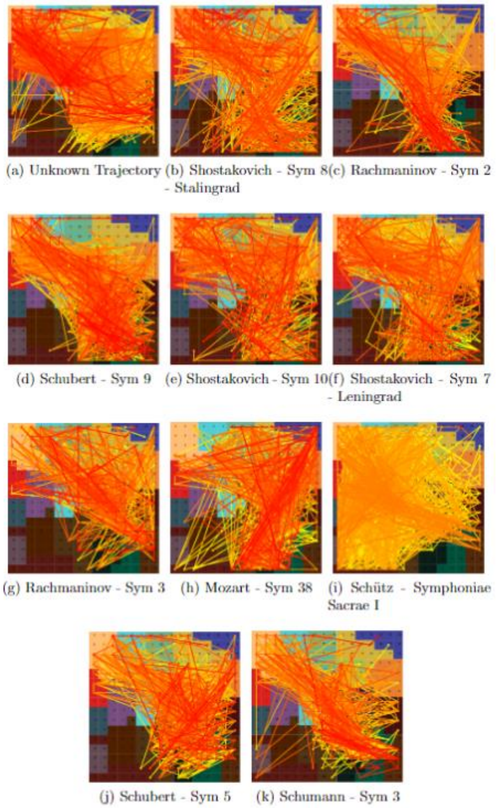

(a) Unknown Trajectory (b) Shostakovich - Sym 8 - Stalingrad (c) Rachmaninov - Sym 2
(d) Schubert - Sym 9 (e) Shostakovich - Sym 10 (f) Shostakovich - Sym 7 - Leningrad
(g) Rachmaninov - Sym 3 (h) Mozart - Sym 38 (i) Schütz - Symphoniae Sacrae I
(j) Schubert - Sym 5 (k) Schumann - Sym 3

Fig. 8. Top 10 trajectories similar to Mozart Symphony 41 using Pairwise

4. CONCLUSIONS

Analyzing the similarity of different

trajectories can be difficult, especially those with long sequences. Visual inspection may not be sufficient to specifically describe similarity among trajectories. Sanhi (Sanhi & Azcarraga, 2016) proposed a pairwise approach to measuring the similarity of two trajectories by using the clusters of the K-Means clustering algorithm. It simply counts all those who have the same cluster and position. There is still a problem of comparing trajectories that have different lengths, as the approach needed to align the trajectories first. The resulting trajectories barely resembled the unknown trajectory. In this research, using an edit-distance approach showed that alignment of trajectories is not necessary and it was able to handle scale and shifting, in which the pairwise similarity measure had difficulty. Based on the results, it showed that using an edit-distance approach resulted in a more similar trajectory, visually, than using the pairwise approach. Works on similarity measures for trajectories can open up new possibilities. It could be used as a preprocessing task for predicting the trajectories. However, there were still some limitations to this research. First, the research used qualitative analysis for the similarity measure. It would be best if there was a way to score the similarity measure. Second, the approach only used the clusters, where the proposed similarity measure did not consider the actual location on the map.

6. REFERENCES

Azcarraga, A., & Flores, F. K. (2016). SOMphony: Visualizing Symphonies Using Self Organizing Maps. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 9950 LNCS* (pp. 531–537). https://doi.org/10.1007/978-3-319-46681-1_63

Azcarraga, J. (2012). *Analysis and Visualization of EEG Data Towards Academic Emotion Recognition* (Issue May). De La Salle University.

Chen, L., & Ng, R. (2004). On The Marriage of Lp-norms and Edit Distance. In *Proceedings 2004 VLDB Conference*. https://doi.org/10.1016/B978-012088469-8/50070-X

Esling, P., & Agon, C. (2012). Time-series data mining. *ACM Computing Surveys*, *45*(1), 1–34. https://doi.org/10.1145/2379776.2379788

Kalaw, K., & Ong, E. (2017). *Recognizing Reader's Affect Using EEG Data*. De La Salle University.

Kohonen, T. (1990). The Self-Organizing Map. *Proceedings of the IEEE*, *78*(9), 1464–1480. https://doi.org/10.1109/5.58325

Lipton, R. J. (2010). Edit Distance. In *The P=NP Question and Gödel's Lost Letter* (pp. 179–183). Springer US. https://doi.org/10.1007/978-1-4419-7155-5_37

Sanhi, C., & Azcarraga, J. (2016). The Use of Trajectory Analysis on a Structured SOM for Behavior Analysis. In *De La Salle University*. De La Salle University.

Vesanto, J. (1997). Using the SOM and local models in time-series prediction. *Proc. Workshop on Self-Organizing Maps 1997*, *1*, 209–214. https://doi.org/10.1.1.55.642