Time2Cluster: Clustering Time Series Using Neighbor Information

Shima Imani [1] **Alireza Abdoli** [2] **Eamonn Keogh** [3]

Abstract

Time series clustering is an important task in its own right, and often a subroutine in other higher-level algorithms. However, clustering subsequences of a time series is known to be a particularly hard problem, and it has been shown that naive clustering of subsequences yields "meaningless" results under common assumptions. In this work, we introduce Time2Cluster, a novel representation and accompanying algorithm that meaningfully clusters time series subsequences. Our key insight is to avoid depending solely on relative distance information between subsequences, and instead to exploit information about the "neighborhood" subsequences. Our algorithm uses neighborhood information to mitigate the negative effects of small variations, such as phase shift, between the subsequences of time series data.

1. Introduction

Clustering is a prominent research area in data mining. It is essentially the process of partitioning data into meaningful groups to gain new insights into problems. Ideally this partitioning happens such that data within each cluster is similar to each other, while dissimilar to data in other clusters. Many data mining algorithms use clustering as a subroutine to solve problems in their domain (Ye & Li, 2005)(Jiménez-Pérez & Mora-López, 2016)(Lopez et al., 2012).

In this work we consider subsequence time series clustering. Our proposed representation allows every index of a time series to be labeled in one of K classes and each index is annotated by a score that reflects our confidence in the labeling. As we will show, this more expressive representation, in conjunction with our algorithm, allows us to cluster datasets that stymie state-of-the-art algorithms. In Fig. 1 we show one sample issue that our representation can mitigate.

[1]shimaimani@microsoft.com [2]aabdo002@ucr.edu [3]eamonn@cs.ucr.edu. Correspondence to: Shima Imani <shimaimani@microsoft.com>.

Time Series Workshop at the 38^{th} International Conference on Machine Learning (ICML), 2021.

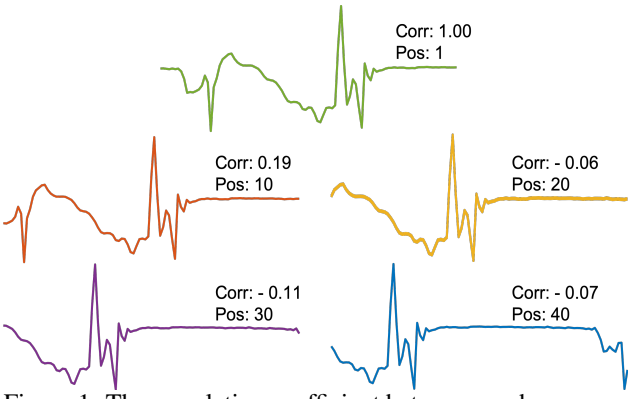

Figure 1: The correlation coefficient between a subsequence of walking behavior (green) and its neighbors is very low because of the phase shift between subsequences. All these subsequences describe walking behavior. *Pos* denotes position in respect to the position of green subsequence.

With classic algorithms/representations it is often the case that a subsequence is partitioned into one cluster and its "close" neighbors are partitioned into another cluster, even though they are describing the same behavior. Fig. 1 shows a subsequence of walking behavior green and its corresponding neighbors, along with correlations to the green subsequence. Note that correlation coefficient between the walking subsequence and its neighbors is low (i.e. distance is large). One would expect that subsequences that are neighbors and describe the same behavior to have a high correlation coefficient (i.e. low distance) with each other, but this is not the case here. The small amount of phase shift that exists in the green subsequence and its neighbors leads to low correlation.

Here we outline major contributions in this work:

- We introduce a novel representation for time series clustering. Our model is expressive enough to represent sparse conserved clusters, essentially time series motifs (Chiu et al., 2003), and, at the other extreme, very dense conserved clusters, essentially time series snippets (Imani et al., 2018).

- We introduce *Time2Cluster* algorithm to meaningfully cluster subsequences in our representation by utilizing "temporal locality" information for each subsequence.

We organize rest of the paper as follows: In Section 2 we introduce necessary notation and definitions. Section 3 reviews related works. We explain the proposed Time2Cluster algorithm, in Section 4. In Section 5, we perform empirical evaluation and compare our results with different baselines. Section 6 draws conclusions and suggests directions for future work.

2. Definitions and Notation

We begin by describing necessary definitions and notation.

Stride length: In time series data, stride length is the number of data points that we shift the position of current subsequence to extract position of the next subsequence.

In our algorithm we define *BAG* as:

BAG: $BAG^{ks}_{(i,m)}$ is a continuous ordered subset of subsequences which consists of ks subsequences (kernel size = ks) starting at position i and the subsequence length of m.

$$BAG^{ks}_{(i,m)} = T_{i,m}, T_{i+1,m}, ..., T_{i+ks-1,m}$$

Kernel size is the number of subsequences that we want to consider in each BAG which we denote as ks.

We can store distances between a subsequence of a time series with all the other subsequences from the same time series in an ordered array called *distance profile* [16].

Correlation matrix: A correlation matrix M stores the Pearson correlation coefficient between all subsequences of the time series. Note that the distance profile calculates the Pearson correlation of each subsequence with the time series. This means we can compute distance profile for all subsequences of the time series and create a correlation matrix by storing each distance profile in one row of the correlation matrix.

3. Related Work

Clustering algorithms can be classified into three categories of "whole time series clustering", "calendar-based clustering" [1], and "subsequence time series clustering".

For subsequence time series (STS) clustering we can further refine the taxonomy into three different approaches.

- Shape-based clustering: This method uses raw time series data and a distance measure such as Euclidean distance (ED) to measure similarity between data. For example, in (Paparrizos & Gravano, 2015) authors introduce k-shape clustering algorithm, which is similar to K-means, and it uses cross-correlation as a distance measure.
- Model-based clustering: In this approach, raw time series is used to find parameters of some model. For example, in (Hallac et al., 2017) authors used model-based clustering approach for multivariate time series data. Moreover, model-based clustering can suffer from scalability issues (Vlachos et al., 2004).

[1]Note that the "calendar" here does not need to be the familiar days or weeks. It can be anything that produces unambiguous cycles, for example tides or batch-cycles in batch processing.

4. Time2Cluster

In brief, Time2Cluster consists of the following steps: Computing correlation matrix, Computing augmented correlation, and , Using kmean++ algorithm on the augmented correlation matrix.

Correlation matrix: The correlation matrix is Pearson correlation coefficient between each two subsequences in the time series. Basically the correlation matrix can be written as:

$$M = \begin{bmatrix} \rho^m_{0,0} & \rho^m_{0,1} & \cdots & \rho^m_{0,i} & \rho^m_{0,n} \\ \rho^m_{1,0} & \rho^m_{1,1} & \cdots & \rho^m_{1,i} & \rho^m_{1,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \rho^m_{j,0} & \rho^m_{j,1} & \cdots & \rho^m_{j,i} & \rho^m_{j,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \rho^m_{n,0} & \rho^m_{n,1} & \cdots & \rho^m_{n,i} & \rho^m_{n,n} \end{bmatrix} = \begin{bmatrix} DP_0 \\ DP_1 \\ \cdots \\ DP_j \\ \cdots \\ DP_n \end{bmatrix}$$

and

$$\rho^m_{i,j} = corr(T_{i,m}, T_{j,m})$$

Where corr is the Pearson correlation coefficient between two subsequences of length *m* starting at position i and j. Note that $\rho^m_{i,i}$ is equal to one since each subsequence is perfectly correlated to itself. We can calculate M using distance profile DP as above, where DP_j is distance profile of subsequence j and changing the distance to Pearson correlation by using the formula of distance profile definition in Section 2.

Augmented correlation matrix: Due to phase shift between two subsequences that are close to each other, correlation coefficient of such subsequences would have a low value, even though the subsequences represent the same behavior. We would like to have an algorithm that gives us a high correlation coefficient if any two subsequences represent the same semantic behavior in spite of the phase shift. To understand this concept, consider some time series comprised of walking followed by running as shown in Fig. 2. The time series represents X-acceleration data of a sensor mounted on the shoe (Ainsworth et al., 2000).

As hinted at in Fig. 1, due to existence of a phase shift between a subsequence and its neighbors ("neighbor" in

the sense of similar time index), the correlation coefficient between them is low, thus using the correlation matrix as the input for any clustering method will likely not generate correct results. An ideal clustering algorithm partitions all the walking subsequences in one cluster and all the running subsequences in another cluster, given that these are two distinct behaviors. However, if we use any classic clustering algorithms such as Kmean++ (Arthur & Vassilvitskii, 2006) with K = 2 (K is the number of clusters), we do not obtain ideal results as shown in Fig. 2.

For visualization purposes, we show part of the time series from Fig. 2 with labeling of Kmean++ algorithm for walking-running time series. We show labeling results in red (walking) and blue (running) colors. The ground truth label (green) consists of walking behavior for about 2.5 minutes and then followed by running behavior for about 2.5 minutes. As Fig. 2 shows the algorithm randomly labels each subsequence as red or blue. This means the Kmean++ algorithm produces a random transition of walking and running labels. The reason is that the correlation coefficient between each subsequence and its neighbors is low due to phase shift, which in turn makes the subsequences appear distinct from each other. However, time series are not independent and identically distributed (i.i.d) random variables, and neighbors are not independent from each other. We argue that we should take this into account when clustering time series data.

In this section, we introduce augmented correlation matrix which will address the aforesaid problem. We need to mitigate the phase shift effect, for which one simple solution might be that if each behavior repeats exactly after a certain period of time, we can extract subsequences with the stride length of that certain period. This approach can give us subsequences with phase shift equal to zero. However, there are only a handful of special time series for which this solution might work. One example is the pedestrian count data in Melbourne (Set, 2009). Since the natural subsequence length in this dataset is *exactly* 24 hours, for each day we can extract one subsequence starting at 12:00 AM which means the stride length is one day.

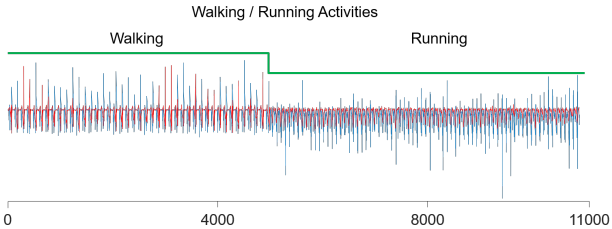

Figure 2: top) The green binary vector represents the ground truth bottom) The labels of Kmean++ for walking and running subsequences are shown in red and blue color, respectively.

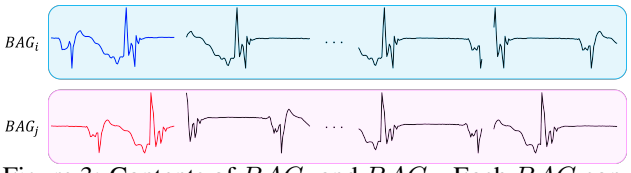

Figure 3: Contents of BAG_i and BAG_j. Each BAG contains one behavior but different subsequences of the same behavior (different phase shifts). Comparing two BAGs is meaningful because allows us to compare different behaviors, since each BAG contains different phase shifts of one behavior.

Thus, we need to create a clustering representation that does not assume all subsequences of the time series have exactly the same period. Our proposed method, Time2Cluster, computes the best stride length for each subsequence and therefore we have a variable stride length for time series. To achieve this, we use Bag definitions from Section 2.

$BAG^{ks}_{(i,m)}$ consists of a subsequence i and its neighbors where ks is the kernel size. We use a stride length of one to extract the next subsequence and m is the subsequence length. Consider two bags of walking behavior, $BAG^{ks}_{(i,m)}$ and $BAG^{ks}_{(j,m)}$ as shown in Fig. 3.

We compute correlation coefficient between each two BAGs rather than two subsequences. This helps to solve the phase shift problem. The correlation coefficient between $BAG^{ks}_{i,m}$ and $BAG^{ks}_{j,m}$ can be computed by calculating correlation coefficient between subsequences of two BAGs and then finding maximum correlation between the two BAGs.

Fig. 4 shows the result of using augmented correlation matrix for clustering of walking and running time series shown in Fig. 2. As shown in Fig. 4 using augmented correlation as an input to the Kmean++ clustering algorithm generates correct results. Using BAGs instead of subsequences helps with assigning subsequences and their neighbors to the same cluster and forms a temporal consistency.

Confidence Score: In the last step, we compute confidence score for each label. This score indicates confidence of the result of Time2Cluster algorithm for each label. We compute the confidence score for each index i by summing up the augmented correlation coefficient at index i for all neighbors of index i that are clustered as in the same class. The confidence score is a real number between zero and one. A score of one for index i means we are very confident about the label of that index, while a score of zero means, that the labels are probably random.

5. Experimental Evaluation

To ensure that our experiments are easy to reproduce, we have created a website that contains all data/code/raw

Table 1: The performance of Time2Cluster and baselines.

Clustering MethodDataset	Chicken	Tilt Table
Time2Cluster	**0.90**	**0.97**
TICC (Hallac et al., 2017)	0.74	0.81
GMM (Banfield & Raftery, 1993)	0.87	0.51
Euclidean Kmeans	0.75	0.44

Figure 4: top) The green vector represents the ground truth bottom) Clustering walking and running behaviors using Time2Cluster algorithm using red and blue colors, respectively.

spreadsheets for all the experiments (Author, 2021). To measure the performance, we fix the number of clusters for our algorithm and all baseline algorithms and evaluate clustering performance by measuring Macro-F1 score:

Macro-F1 score In order to compute Macro-F1 score, first for each cluster we compute F1 score which is the harmonic mean of the precision and recall as follows:

Then we compute Macro-F1 score which is the average of F1-score for all clusters. F1-score and Macro-F1 score are in the range of [0, 1] with F1-score equal to one means perfect clustering.

5.1. Chicken Behavior

It is believed that frequency and timing of behaviors such as pecking, preening and dustbathing can be good indicators of chicken health (Abdoli et al., 2018). Pecking refers to act of chicken striking at the ground with its beak for feeding purposes; while, cleaning and aligning of feathers with the beak is referred to as preening (Daigle et al., 2014).

Fig. 5 shows a time series of chicken data beginning with pecking behavior followed by preening behavior. The green binary vector shows the ground truth for chicken data. Using clustering can help us to analyze, understand and extract useful information from this data set.

5.2. Tilt Table

We use the arterial blood pressure (ABP) signal to find clusters in the time series as shown in Fig. 6. The ABP signal is a key source of information for determining hemodynamic state of the patient.

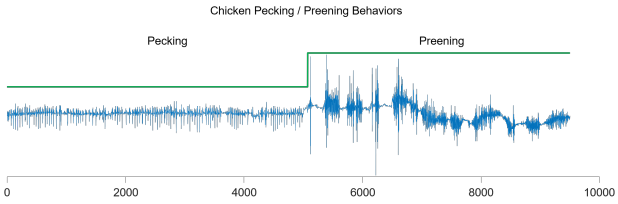

Figure 5: top) The green binary vector represents ground truth for chicken behavior bottom) Time series corresponding to X axis of chicken data.

As shown in Fig. 6, the clustering result of Time2Cluster algorithm is very similar to the ground truth label. Let us revisit the confidence score for Time2Cluster algorithm as we explained in Section 4. Note, the confidence score of all labels is high (close to one) except for a small region. The reason is that the original data contains a small region in which the sensor failed to record physiological data, and instead reported a square-wave calibration signal (Samaniego et al., 2003).

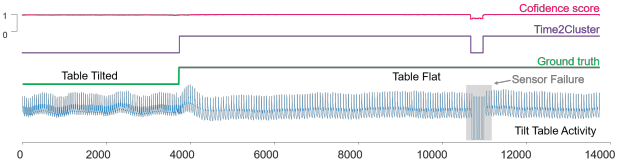

Figure 6: top) The confidence score (pink), labeling of Time2Cluster algorithm (purple) and ground truth (green), respectively. bottom) Time series of arterial blood pressure.

As we can see from Table 1, Time2Cluster significantly outperforms the baseline algorithms. On average Time2Cluster has a Macro-F1 score of 0.93. For chicken dataset, there is a big gap between Time2Cluster and baselines performance. In future we want to compare our algorithms with other baselines, with a larger collection of time series dataset.

6. Discussion and Conclusions

We have introduced Time2Cluster, an expressive representation/algorithm for clustering time series subsequences. Our method is a shape-based clustering algorithm that is invariant to phase shift effects, allowing us to find meaningful clusters where other algorithms struggle. In future work, we will investigate pruning, indexing and early abandoning techniques to further scale up our approach.

References

Abdoli, A., Murillo, A. C., Yeh, C.-C. M., Gerry, A. C., and Keogh, E. J. Time series classification to improve poultry welfare. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 635–642. IEEE, 2018.

Ainsworth, B. E., Haskell, W. L., Whitt, M. C., Irwin, M. L., Swartz, A. M., Strath, S. J., O Brien, W. L., Bassett, D. R., Schmitz, K. H., Emplaincourt, P. O., et al. Compendium of physical activities: an update of activity codes and met intensities. *Medicine and science in sports and exercise*, 32(9; SUPP/1):S498–S504, 2000.

Arthur, D. and Vassilvitskii, S. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006.

Author. Project website, 2021. https://sites.google.com/site/time2cluster/.

Banfield, J. D. and Raftery, A. E. Model-based gaussian and non-gaussian clustering. *Biometrics*, pp. 803–821, 1993.

Chiu, B., Keogh, E., and Lonardi, S. Probabilistic discovery of time series motifs. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 493–498. ACM, 2003.

Daigle, C. L., Banerjee, D., Montgomery, R. A., Biswas, S., and Siegford, J. M. Moving gis research indoors: Spatiotemporal analysis of agricultural animals. *PLoS One*, 9(8), 2014.

Hallac, D., Vare, S., Boyd, S., and Leskovec, J. Toeplitz inverse covariance-based clustering of multivariate time series data. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 215–223, 2017.

Imani, S., Madrid, F., Ding, W., Crouter, S., and Keogh, E. Matrix profile xiii: Time series snippets: A new primitive for time series data mining. In *2018 IEEE International Conference on Big Knowledge (ICBK)*, pp. 382–389. IEEE, 2018.

Jiménez-Pérez, P. F. and Mora-López, L. Modeling and forecasting hourly global solar radiation using clustering and classification techniques. *Solar Energy*, 135:682–691, 2016.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Lopez, M. I., Luna, J. M., Romero, C., and Ventura, S. Classification via clustering for predicting final marks based on student participation in forums. *International Educational Data Mining Society*, 2012.

Paparrizos, J. and Gravano, L. k-shape: Efficient and accurate clustering of time series. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pp. 1855–1870, 2015.

Samaniego, N., Morris, F., and Brady, W. Electrocardiographic artefact mimicking arrhythmic change on the ecg. *Emergency medicine journal*, 20(4):356–357, 2003.

Set, M. P. D. Melbourne pedestrian data set, 2009. http://www.pedestrian.melbourne.vic.gov.au/.

Vlachos, M., Gunopulos, D., and Das, G. Indexing time-series under conditions of noise. In *Data mining in time series databases*, pp. 67–100. World Scientific, 2004.

Ye, N. and Li, X. Method for classifying data using clustering and classification algorithm supervised, June 14 2005. US Patent 6,907,436.