COMP5331 Final Report: Implements and Applications of Toeplitz Inverse Covariance-Based Clustering

Group 13: Guo Wenshuo, Cheung Tsz Him and Ding Mu Cong
(Implementaton Type Paper)

The Hong Kong University of Science and Technology

Abstract. Time series clustering has shown its importance in providing useful information and applications in a wide range of research areas. Although a variety of clustering algorithms has been extensively explored, including the model-based clustering, density-based clustering and grid-based clustering, the difficulty in simultaneous segmentation and clustering, and the challenge in the interpretation of clustering results are remained. In this report, we investigated on the novel method designed by Dacid Hallac et al[3] which offers an alternative for simultaneous segmentation and clustering on multi-variate time series data. We implemented and verified the superiority of TICC solver on two open-source real-world datasets, including the human activity time series data and gesture sensor time series data. We studied the results produced by TICC in detail and provide comparisons with the baseline methods: K-means and GMM. We also compared TICC with some deep learning classification model, i.e recurrent neural network (RNN) with long short-term memory (LSTM) cell when labels are present. Based on that, we analyzed the potential short-comings of TICC. We hope that our work could bring benefits on the further generalizations and applications of the TICC model, and offer insights for future research on time series clustering.

1 Introduction

Time series clustering has shown its significance in providing effective information and useful applications in many research fields, including economics [2], medicine [8], bio-informatics [9] and multimedia [6].

The central objective of clustering is to discover patterns in an unlabeled data set. Although different clustering algorithms have been extensively investigate, where the clustering algorithms objectively organizing data into homogeneous groups so that the within-group-object similarity is minimized and the between-group-object dissimilarity is maximized [4], the unique temple characteristics of time series data are the causes that fail most of conventional clustering techniques. For instance, for time series data with a high dimensionality, very high feature correlation, and large amount of noise, the clustering tasks have been regarded as an challenging research problem [1].

Multivariate time series take up the majority in real-world time series datasets. Sub-sequence clustering of these long time series could serve as a useful tool to discover repeated patterns in the data. The breaking down of the original time series into a sequence of states requires segmenting and clustering the multi-variate time series simultaneously, which is an interesting and difficult challenge.

In this report, we specially investigated on the applications and improvements of Toeplitz Inverse Covariance-Based Clustering [3], which provides a novel way for simultaneous segmentation and clustering. We implemented TICC algorithm and tested it on two multi-variate time series data sets, the human activity data and gesture data. We then compared its performance with K-means and GMM and analyzed their prediction results in great details. Further more, We compared TICC with classification models like RNN using LSTM on these two labeled data sets, and revealed the short-comes of TICC is that It cannot handle frequent state transitions (or to say short latent sub-sequences equivalently). We also examined the effects of 4 hyper-parameters of TICC algorithm based ton the empirical study on human activity data. Based on these empirical summaries, we proposed our understanding on choosing the hyper-parameters.

2 Previous Work

Clustering problems are central to many data mining and knowledge discovery tasks. In this section, the literature review and discussions on some proposed papers on time series clustering and their applications are presented.

In recent years, mining repeated patterns in temporal data has been shown to serve as a useful way to gain effective information from time series datasets. Mining the patterns from time series could help simplify the original complicated datasets into a temporal sequence of small clusters. However, the simultaneous segmentation and clustering and comprehensive understanding of the clustering results have remained to be current difficulties in the research area.

Traditional methods including K-means have been applied on the clustering of different time series data. Recent work has demonstrated the usefulness of time series representations to the raw multimedia data [6], where k-medoids algorithm with Dynamic Time Warping (DTW) has been widely used.The Dynamic Time Warping could provide the ideal shape-based similarity measurement which can break the limitation of one-to-one mapping in the Euclidean distance metric. However, the K-means clustering may fail to provide correct results especially when Dynamic Time Warping (DTW) is used as the distance measure in averaging the shape of the time series. The potential causes and suggestions were discussed and presented by Vit Niennattarakul et al [6]. In 2003, [10], an expectation-maximization(EM) algorithm for learning the mixing coefficients and the parameters of the component models was presented with interesting outcomes. The Bayesian information criterion was used to determine the number of clusters in the data. And results showed that the method compares favorably with other previous methods for similar time series clustering tasks. However, one remained problem is that for EM-based methods, the clus-

tering performance could degrade significantly when the underlying clusters are close to each other. Therefore, possible extensions remain in need to improve the performance of EM-based clustering algorithms.

3 Implementation and Model Comparison

3.1 Data set

The TICC algorithm is applied to 2 multivariate, time-series data sets, namely Human Activity Data set [7] and Gesture Data set [5] to evaluate its performance.

Human Activity Data set: The Human Activity Data set consists of the accelerometer and gyroscope readings in smart phones and smart watches of 9 users. The readings are recorded while users are performing 6 different activities: Standing, Sitting, Biking, Walking, Stairs Up and Stairs Down. Each sensor generates 3 values in every time stamp, which are the xyz-axis reading from the sensors, forming altogether 6 features. The task is to cluster the 6 activities based on the sensor readings. Through observing the ground truth, the clusters are large in size, which means the same activity lasts for long period of time.

Gesture Dataset The Gesture Data set contains the xyz-axis reading of left and right hands, head, spine and left and right wrists from the Microsoft Kinect sensor, forming altogether 18 features. The readings are recorded while users are demonstrating 5 different gestures: Rest, Preparation, Stroke, Hold and Retraction. The task is to cluster the 5 gestures based on the position readings. Different from the Human Activity Data set, the clusters are smaller, which means the gesture changes frequently throughout the time series.

3.2 Data Pre-processing

For the Human Activity Data set, we picked a fixed user (user id = a from the dataset) as our target subject and smartphone model nexus 4 as our sensor. The smartphone accelerometer and gyroscope readings with the same time stamps are joined. The timestamps, index and the data with missing labels are discarded. Among the total 600K data points from user A, we evenly sampled the time series into around 20K data points.

For the Gesture Dataset, the entire dataset, around 10K data points, is used. Timestamps and missing data are dropped.

3.3 Implementation

We adopted most code from the TICC solver.py provided by the original paper [3] with slight modification on the input and output utilities. We also implemented the K-means clustering, Gaussian Mixture Model and some visualization

utilities to benchmark, visualize and evaluate TICC performance. Apart from those classical clustering algorithms, when labels are present, it is also possible to classify the sequence segments by classification methods. We also implement a RNN model using LSTM cells and compared its accuracy with TICC. Although it is not a fair competition, since TICC is an unsupervised learning algorithm, it answers the question that why TICC is still preferable even when we have enough labels to train a deep neural network.

3.4 Baseline Comparisons

We compared the accuracy of the cluster assignments from TICC, K-Means Clustering and Gaussian Mixture model, with the ground truth labels. We also compared it with LSTM RNN and discussed their differences.

3.5 Comparison on Human Activity Data

We run TICC, K-means and GMM on the human activity data. From the results, as shown in Fig. 1, we can observe that clearly TICC did a better job compared to K-means and GMM. Since it is a subsequence clustering algorithm, TICC classify each subsequence a a whole, which greatly reduce the error. Based on the clustering results, we also generate the confusion matrix where we label each clusters so that the accuracy is maximized. From the confusion matrices, Fig. 2, we can see that TICC perfectly clustered the 6 classes of activities, with very slight messing up between the "Stairs Up" and "Standing" states. While K-means and GMM perform much more poorly. From Fig. 2b we found that K-means tends to classify many subsequences as "Sitting" and "Standing", although it got good accuracy on recognizing these two states, it also miss interpret many other activities as "Sitting" and "Standing". For GMM, "Standing", "Stairs down" and "Walking" are best recognized, with around 0.5 accuracy. While "Sitting" is often misunderstand as "Standing".

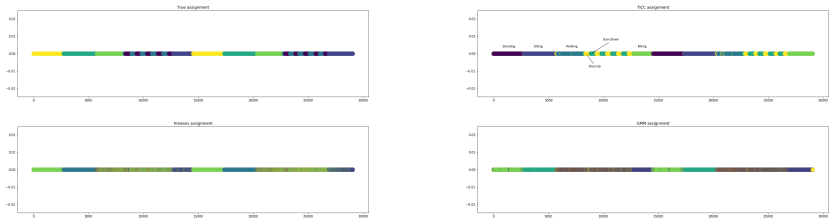

Fig. 1: Human Activity Cluster assignment of (a) True Labels (b) TICC (c) KMeans (d) GMM. Each different color represent a unique label in the time series.

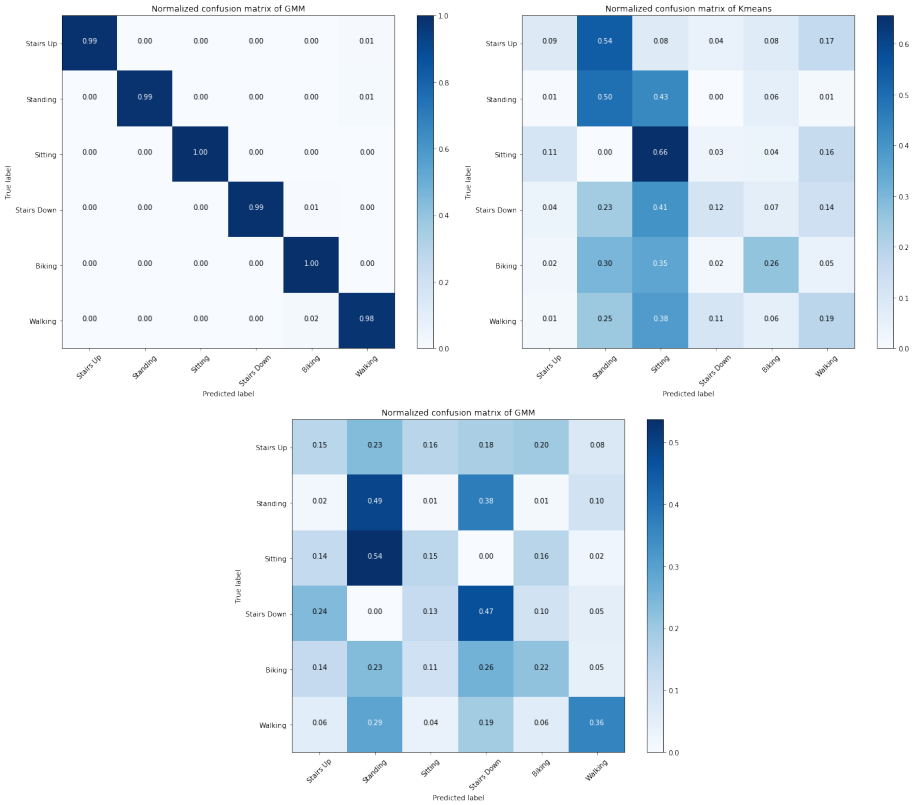

Fig. 2: Human Activity Cluster assignment of (a) True Labels (b) TICC (c) KMeans (d) GMM. Each different color represent a unique label in the time series.

3.6 Comparison on Gesture Data

We found that TICC, K-means and GMM clustered did not generate satisfactory results on the Gesture dataset. It is observed that the time series gesture changes frequently and each gesture lasts for a short period of time (Fig. 3a). However, even the TICC model is set to capture smaller window and use small smoothing penalty, which creates more segments, the model fails to cluster the gesture in the middle part (Fig. 3b). However, same as Human Data set, other benchmark models, like KMeans and GMM perform even worse (Table 1).

Model	Accuracy on Human Dataset	Accuracy on Gesture Dataset
TICC	92.88%	14.21%
KMeans	42.69%	3.11%
GMM	58.76%	4.98%

Table 1: Comparison between different models on two datasets

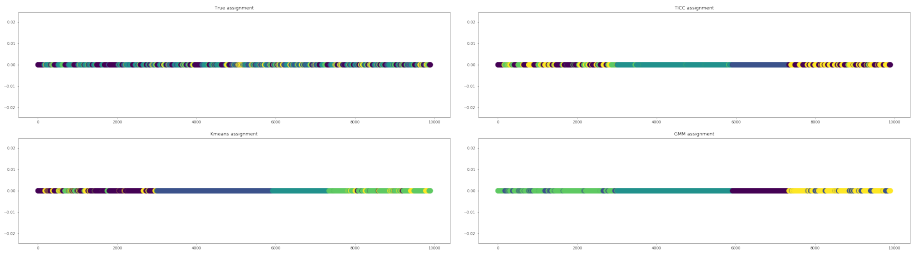

Fig. 3: Gesture Cluster assignment of (a) True Labels (b) TICC (c) KMeans (d) GMM

3.7 Comparison with RNN using LSTM

It is an interesting question that whether TICC could event beat some successful classification models using recurrent neural networks (RNNs) and long short-term memory cells (LSTM cells), even though TICC does not require the labels. Compared to RNN classification model, TICC is intrinsically an unsupervised learning method which preserves the inner consistency of sub-sequences. This feature is very valuable compared to other sequence-to-sequence learning approaches since many real word data maintains this sub-sequence consistency, e.g. our human activity and gesture data. In this subsection, we compare TICC with LSTM RNN and discuss their pros and cons.

We first trained a simple LSTM network on the gesture data. Since it is one long sequence, we split it into around 1K short sequences of length 25 before feed into the network. This splitting may already destroy the latent sub-sequence structure, but since it is unknown to us, we cannot do better. The input tensor is of shape (number of sequences, number of time units, number of features (sensor variables)), while the output shape is (number of sequences, number of time units, number of classes). We use softmax function at the output layer and cross entropy as loss function. The network structure is generally as shown in Fig. 4.

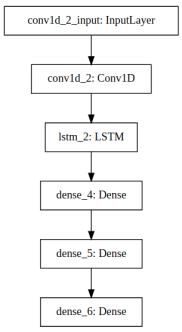

Fig. 4: Structure of LSTM RNN for comparison

Where the LSTM layer is put in between of convolution 1D layer and a block of 3 dense (fully-connected) layers. The 1D convolution layer (CNN) could effectively select representations along the time axis. While LSTM are capable to handle relatively long sequence inputs. The dense layers are merely for generating the predictions. We carefully tuned the hyper-parameters of this LSTM RNN and test it on the human activity data. The learning curve are shown in Fig. 5 as follows,

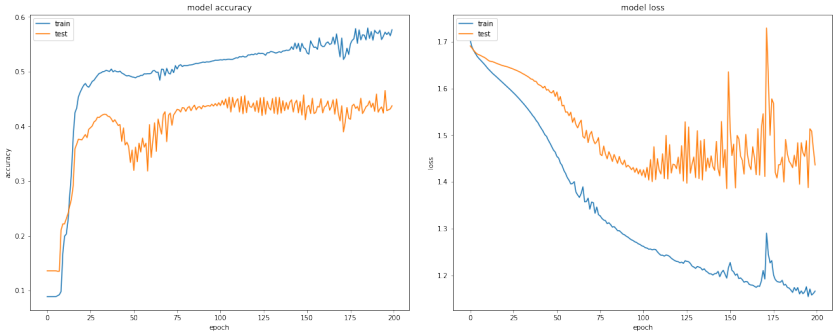

Fig. 5: Accuracy and loss v.s. number of epochs for the LSTM model on gesture data

Since the number of data after splitting is only 1K, there are some over-fitting problem when training this network. The validation accuracy is around 0.42. Compared to TICC's accuracy 0.14, it is already a lot more higher. The confusion matrix is shown in Fig. 6.

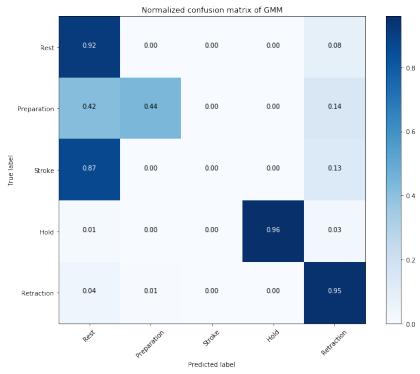

Fig. 6: Confusion matrix of prediction of LSTM on gesture data set

Based on these facts, we can conclude that LSTM methods generally works better on sequences whose consisting sub-sequences are small. As analyzed in the previous section, the labels of gesture data changes frequently so that TICC fails to capture the small sub-sequences and LSTM on truncated sample sequences generally work.

However, things change when the switching between states become less frequent, when we compare LSTM and TICC on human activity data. We train a LSTM network with the same structure on 1K truncated human activity sequence of lenth 25, and the best validation accuracy is only around 0.45.

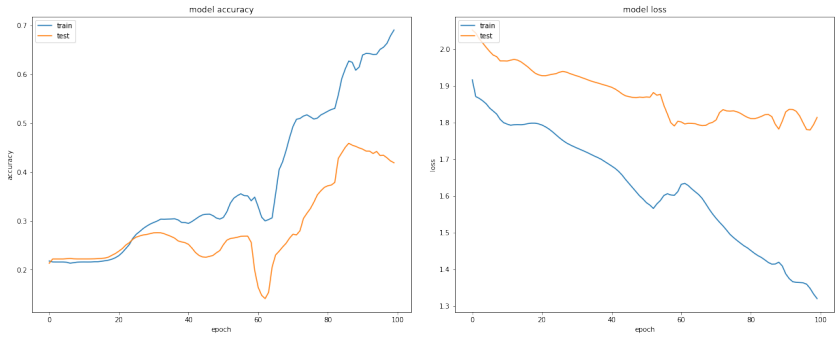

Fig. 7: Accuracy and loss v.s. number of epochs for the LSTM model on human activity data

Thus we can conclude that the TICC model is more capable to capture latent sub-sequence structure when the average length of them is enough long. Meanwhile, we have to split the full sequence before feeding into LSTM and this pre-processing step may already destroy may latent sub-sequence structure. The

power of LSTM is thus greatly affected since the sequences feed in to it are not correctly aligned to each others.

4 Hyper-Parameter Tuning

In this section, we examine the effect of different parameters to the accuracy and cluster property based on the empirical study on applying TICC model on Human Activities Dataset.

4.1 Effects of the Hyper-Parameters

Smoothing Penalty: This parameter determines the number of segment of a time series. When the smoothing penalty is higher, two small different contiguous clusters will merge to one cluster. As we can see in Fig. 8, smaller smoothing penalty yields more segments, with small strips in the time series. For a larger smoothing penalty, the amount of strips decreases (Fig. 8b) and eventually are smoothed to several pure large segments (Fig. 8c). This may reveal one limit of TICC algorithm in clustering frequently-changing time series.

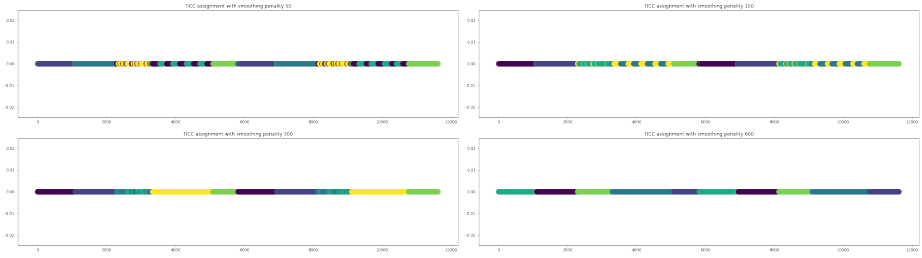

Fig. 8: Human Activity Cluster assignment of smoothing penalty equals to (a) 50 (b) 150 (c) 300 (d) 600

Regularization: This parameter regularizes the sparsity level of the MRF graph characterizing each cluster. Looking at the clustering assignment may not have immediate interpretation on the effect of changing the parameters (Fig. 9).

X

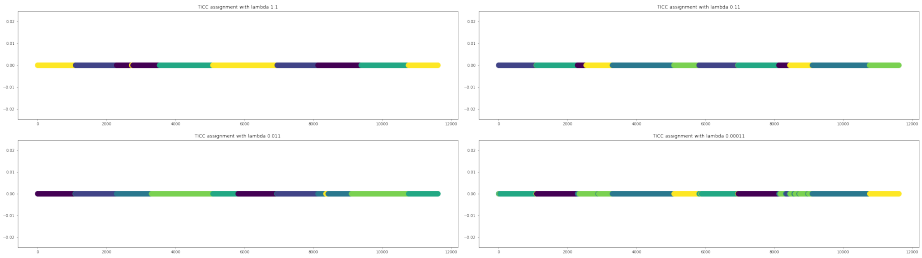

Fig. 9: Human Activity Cluster assignment of regularization equals to (a) 1.1 (b) 0.11 (c) 0.011 (d) 0.00011

Different clustering methods has its own representation of cluster, for example, KMeans can use mean of the data within that cluster, and GMM uses probability parameters. In TICC, each cluster is represented as a graph, called MRF. To observe the effect of regularizer parameter, we can observe the representation of each cluster. Fig. q0 shows the representation of the clusters when regularizer weight is set to 0.00011.

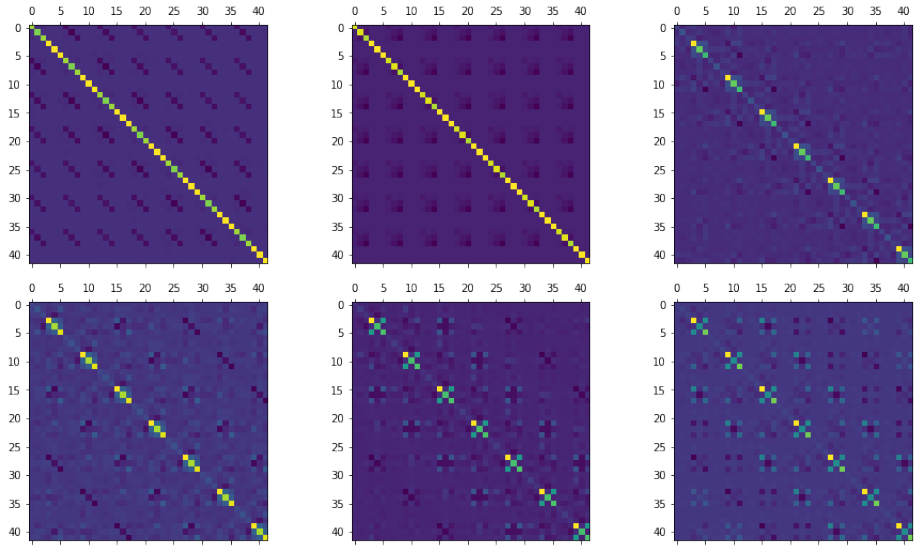

Fig. 10: The graphical plot of the correlation network (MRF) with regularization 0.00011 of (a) Stand (b) Sit (c) Bike (d) Walk (e) Stairs Up (f) Stairs Down

TICC represents Stand and Sit in similar graph similarly (Fig 10a, 10b). For Stairs Up and Stairs Down activities, they look also quite similar. (Fig 10e, 10f). For Biking (Fig. 10c), it looks closer to Stand and Sit than to stairs up and down;

while Walking is vice versa. This is in fact coincide with human interpretation of the events, which is not hard to imagine that, walking movement is similar to that of walking up and down stairs, while the movement of standing and siting are more or less the same.

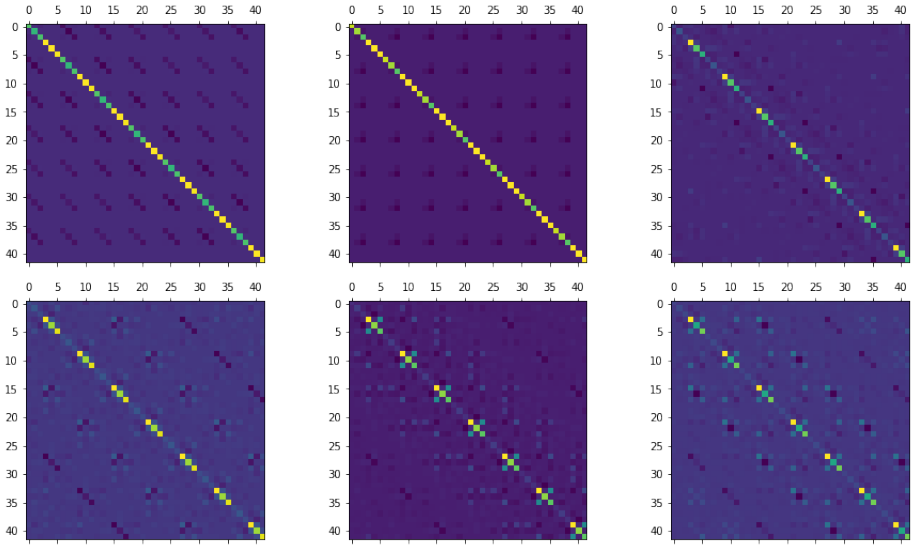

Fig. 11: The graphical plot of the correlation network (MRF) with regularization 0.011 of (a) Stand (b) Sit (c) Bike (d) Walk (e) Stairs Up (f) Stairs Down

When we carefully examine the cluster MRFs, the difference between with larger Fig. 10e and 11e is more observable. With larger regularization, the graph is less clear and the pattern are less sharp. The pattern of other plots from Fig. 11 is also slightly more blur than that of Fig. 10. This suggests that the more regularization leads to more general representation, possibly avoid over-fitting problem.

Window Size: This parameter controls the consideration of cross-time correlation, which means sensor reading at time t affect some sensor readings at time t + w, where w is the window size. A larger window size would reach a longer duration. As the window size increase, the shorter activities, that is the alternating stairs up and stairs down activity disappear and merge into one cluster. This is likely due to considering too long dependency between readings, causing local correlation to be diminished, thus fail to detect changes in shorter period activities.

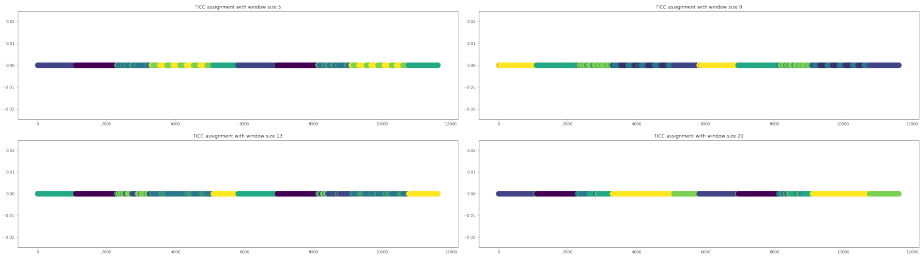

Fig. 12: Human Activity Cluster assignment of window size equals to (a) 5 (b) 9 (c) 13 (d) 20

Dataset Size: We also investigate TICC performance depending on the size of data. In data set of size 20K and 10K, it generates assignments with high accuracy up to 92%, while KMeans and GMM generates 43% and 58% respectively. We notice that to achieve same accuracy as KMeans and GMM, TICC requires much less data, around 1600 data points, in this particular Human Activity Dataset.

4.2 Choosing the Hyper-Parameters

Based on the empirical results we obtained, we see how clustering characteristic changes with different parameters. In general, to cluster dataset with large segments (activities with long period of time), the larger window size and higher smoothing penalty could be employed. For more general solution, larger regularization parameter can be set.

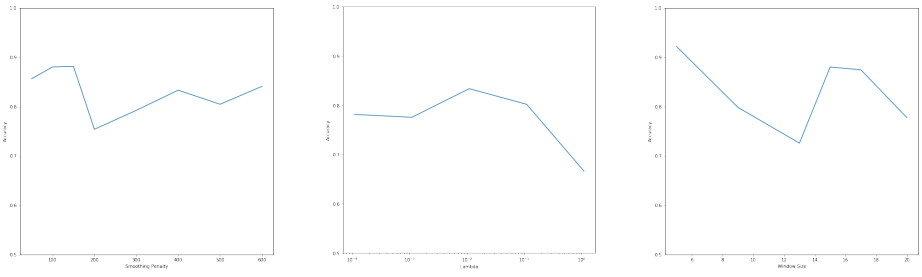

Fig. 13: Accuracy for different parameters values (a) smoothing penalty (b) regularization weight (c) window size

5 Conclusions

References

1. Aghabozorgi, S.R., Shirkhorshidi, A.S., Teh, Y.W.: Time-series clustering - A decade review. Inf. Syst. 53, 16–38 (2015), https://doi.org/10.1016/j.is.2015.04.007
2. Demetriades, P.O., Hussein, K.A.: Does financial development cause economic growth? time-series evidence from 16 countries. Journal of Development Economics 51(2), 387 – 411 (1996), http://www.sciencedirect.com/science/article/pii/S030438789600421X
3. Hallac, D., Vare, S., Boyd, S.P., Leskovec, J.: Toeplitz inverse covariance-based clustering of multivariate time series data. CoRR abs/1706.03161 (2017), http://arxiv.org/abs/1706.03161
4. Liao, T.W.: Clustering of time series data - a survey. Pattern Recognition 38(11), 1857–1874 (2005), https://doi.org/10.1016/j.patcog.2005.01.025
5. Madeo, R.C.B., Lima, C.A.M., Peres, S.M.: Gesture unit segmentation using support vector machines: Segmenting gestures from rest positions. In: Proceedings of the 28th Annual ACM Symposium on Applied Computing. pp. 46–52. SAC '13, ACM, New York, NY, USA (2013), http://doi.acm.org/10.1145/2480362.2480373
6. Niennattrakul, V., Ratanamahatana, C.A.: On clustering multimedia time series data using k-means and dynamic time warping. In: 2007 International Conference on Multimedia and Ubiquitous Engineering (MUE 2007), 26-28 April 2007, Seoul, Korea. pp. 733–738 (2007), https://doi.org/10.1109/MUE.2007.165
7. Stisen, A., Blunck, H., Bhattacharya, S., Prentow, T.S., Kjærgaard, M.B., Dey, A., Sonne, T., Jensen, M.M.: Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In: Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems. pp. 127–140. SenSys '15, ACM, New York, NY, USA (2015), http://doi.acm.org/10.1145/2809695.2809718
8. Tsujino, J., Oyama-Higa, M.: A time series change of biological information through chaos analysis in finger pulse waves after taking medicine with circulatory disease. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Istanbul, Turkey, 10-13 October 2010. pp. 1519–1523 (2010), https://doi.org/10.1109/ICSMC.2010.5642324
9. Wang, Z., Jin, S., Liu, G., Zhang, X., Wang, N., Wu, D., Hu, Y., Zhang, C., Jiang, Q., Xu, L., Wang, Y.: Dtwscore: differential expression and cell clustering analysis for time-series single-cell rna-seq data. BMC Bioinformatics 18(1), 270:1–270:14 (2017), https://doi.org/10.1186/s12859-017-1647-3
10. Xiong, Y., Yeung, D.Y.: Time series clustering with arma mixtures. Pattern Recognition 37(8), 1675–1689 (2004)