A Weight-Based Clustering Method

Zhao-Yu Wang[1], Shie-Jue Lee[2], Shing-Tai Pan3

[1]*Department of Electrical Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan*
[2]*Department of Electrical Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan*
[3]*Department of Electrical Engineering, National University of Kaohsiung, Kaohsiung, Taiwan*
[1]*zywang@water.ee.nsysu.edu.tw,* [2]*leesj@mail.ee.nsysu.edu.tw,* [3]*stpan@nuk.edu.tw*

Abstract: This paper proposes a weight-based self-constructing clustering method for time series data. Self-constructing clustering processes all the data points incrementally. If a data point is not similar enough to an existing cluster, then (1) if the point currently does not belong to any cluster, it forms a new cluster of its own; (2) otherwise, the point is removed from the cluster it currently belongs to before a new cluster is formed. However, if a data point is similar enough to an existing cluster, then (1) if the point currently does not belong to any cluster, it is added to the most similar cluster; (2) otherwise, it is removed from the cluster it currently belongs to and added to the most similar cluster. During the clustering process, weights are learned and considered in the calculations of similarity between data points and clusters. Experimental results show that our proposed approach performs more effectively than other methods for real world time series datasets.

1. Introduction

Clustering is an unsupervised classification technology, with a purpose of forming meaningful clusters for the objects under consideration. Usually, similar objects are grouped in the same cluster, and different objects are grouped in different clusters. Clustering techniques play a very important role in the field of artificial intelligence [1] [2][3][4]. In particular, they are widely applied in times series data analysis in a variety of areas, such as bioengineering [5], environmental monitoring [6], economic applications, and so on. In the process of clustering time series data, using the same weight for each dimension may cause bad effects. To deal with this difficulty, Huang et al. proposed TSKmeans [7], which is K-means with weights, to assign different weights to different dimensions of the data. A similarity measure based on the weighted Euclidean distance was adopted. Through quadratic programming, smooth subspace in time stamps can be produced. It was shown that TSK means can result in better clusters than the original K-means for time series data.

This paper proposes another weight-based clustering method for time series data. Instead of using K-means, an iterative self-constructing clustering method is adopted. The method performs several rounds of clustering until convergence is reached. In each round, all the data points are processed incrementally. If a data point is not similar enough to an existing cluster, then (1) if the point currently does not belong to any cluster, it forms a new cluster of its own; (2) otherwise, the point is removed from the cluster it currently belongs to before a new cluster is formed. However, if a data point is similar enough to an existing cluster, then (1) if the point currently does not belong to any cluster, it is added to the most similar cluster; (2) otherwise, it is removed from the cluster it currently belongs to and added to the most similar cluster. During the clustering process, weights are learned and considered in the calculations of similarity between data points and clusters. If the cluster assignment of one instance has been changed in the current round, the next round of clustering continues. Otherwise, the cluster assignments are stable and the whole clustering process stops with a desired number of clusters.

The rest of this paper is organized as follows. TSKmeans is briefly reviewed in Section II. Our proposed method is presented in Section III. Experimental results are shown in Section IV. Section V gives a conclusion.

2. Related Work

Many clustering methods have been proposed for time series data [8] [9][10][7]. Among them, TSKmeans [7] is the most recently published. TSKmeans is a K-means incorporated with weights. It tries to make the distance between the data points contained in a cluster and the center of the cluster small through the use of weights of

time stamps. Given X = $\{X_1, X_2, \ldots, X_n\}$ is a set of n time series patterns. Each pattern $X_1 = \{x_{i1}, x_{i2} \ldots, x_{im}\}$ is the ith pattern characterized by m values, i.e., m time stamps. The membership matrix U is a n × k binary matrix, k is the total number of clusters, with $u_{ip} = 1$ indicating that X_i belongs to cluster p and u_{ij} , j ≠ p, is 0. The centers and weights of clusters are represented by two sets of k vectors Z = $\{Z_1, Z_2, \ldots, Z_k\}$ and W = $\{W_1, W_2, \ldots, W_k\}$, with w_{pj} being the weight of the jth time stamp for the pth cluster. The purpose of TSKmeans is to minimize the following objective function:

$$P(U, Z, W) = \sum_{p=1}^{k} \sum_{i=1}^{n} \sum_{j=1}^{m} u_{ip} w_{pj} \left(x_{ij} - z_{pj}\right)^2 + \frac{1}{2}\alpha \sum_{p=1}^{k} \sum_{j=1}^{m-1} \left(w_{pj} - w_{p,j+1}\right)^2 \quad (1)$$

Subject to

$$\begin{cases} \sum_{p=1}^{k} u_{ip} = 1\,,\ u_{ip} \in \{0\,, 1\} \\ \sum_{j=1}^{m} w_{pj} = 1\,, 0 \ \le\ w_{pj}\ \le 1 \end{cases} \quad (2)$$

by the application of quadratic programming. Using these weights in each iteration of Kmeans until convergence is reached. At the beginning, TSKmeans generates randomly the centers of clusters and sets initial values for the weights of clusters. Then the following three steps are done iteratively:

◎ Step 1. For each pattern Xi, compute the distance Dpi between it and cluster p by

$$D_{pi} = \sum_{j=1}^{m} w_{pj} (x_{ij} - z_{pj})^2 \quad (3)$$

for 1 ≤ p ≤ k, 1 ≤ i ≤ n. A pattern is assigned to the cluster with the smallest distance. If pattern i is assigned to cluster p, then u_{ip} is set to 1 and u_{ij} is set to 0, j ≠ 0.

◎ Step 2. Update the centers of all clusters by

$$Z_p = \frac{\sum_{x_i \in \text{cluster } p} x_i}{\text{total number of patterns in cluster } p} \quad (4)$$

for 1 ≤ p ≤ k.

◎ Step 3. Use known U and Z to update W by applying quadratic programming to Eq.(1) with

$$\alpha = \sum_{i=1}^{n} ||X_i - \frac{\sum_{j=1}^{n} X_j}{n}||^2 \quad (5)$$

If clusters have changed in the current iteration, then go back and Steps 1–3 are performed again. Otherwise, TSKmeans stops. However, TKmeans suffers from the same problem as K-means does. The number of clusters has to be specified in advance. Our proposed approach can overcome this shortcoming.

3. Proposed Method

We propose a self-constructing clustering (SCC) method which does not require the number of clusters to be specified by the user in advance. We describe the clustering in detail. Also, we improve the method by incorporating weights in the calculation of similarity, just as TKmeans does to K-means. SCC performs several rounds of clustering until convergence is reached. In each round, one full training cycle on the training set of N patterns $X^{(1)}$, $X^{(2)}$, . . . , $X^{(N)}$, is done. Let K be the number of existing clusters. Each cluster C_p, 1 ≤ p ≤ K is characterized by its center Z_p, deviation V_p, size S_p, and weight W_p. Initially, K is 0. Suppose we are in the rth round, r ≥ 1. For pattern i, $X^{(i)}$, 1 ≤ i ≤ N, we calculate the similarity between $X^{(i)}$ and each existing cluster by

$$\mu_p\left(X^{(i)}\right) = \prod_{j=1}^{m} \exp\left[-w_{pj}\left(\frac{x_j^{(i)} - z_{pj}}{v_{pj}}\right)^2\right] \quad (6)$$

for 1 ≤ p ≤ K. Two cases are considered:

Case 1. If

$$\mu_p\left(X^{(i)}\right) < \epsilon \quad (7)$$

for 1 ≤ p ≤ K, we do the following:

1) If $X^{(i)}$currently does not belong to any cluster, it forms a new cluster $C_{K+1}$of its own. And we have K = K + 1, $Z_K = X^{(i)}$, $V_K = \{v_0, v_0, \ldots, v_0\}$, S_K= 1, and W_K containing m randomly generated numbers.

2) If $X^{(i)}$ currently belongs to cluster Ca, we remove $X^{(i)}$ from Ca and update the characteristics of Ca. And a new cluster C_{K+1} containing only $X^{(i)}$ is created as previously.

Case 2. If

$$\mu_p\left(X^{(i)}\right) \ge \epsilon \quad (8)$$

for some existing clusters, we do the following:

1) If $X^{(i)}$ currently does not belong to any cluster, it is added to the most similar cluster, say C_t , and the characteristics of $C_t$are updated by

$$Z_t^n = \frac{S_t^o Z_t^o + X^{(i)}}{S_t^o + 1} \quad (9)$$

$$v_{aj}^{n\,2} = \frac{1}{S_a^o}\{(S_a^o - 1) v_{aj}^{o\,2} + S_a^o z_{aj}^{o\,2} + x_j^{(i)2} - (S_a^o + 1) z_{aj}^{n\,2}\} \quad (10)$$

$$S_a^n = S_a^o + 1 \quad (11)$$

2) If $X^{(i)}$currently belongs to cluster Ca, we remove $X^{(i)}$ from Ca, updating the characteristics of Ca, and we add $X^{(i)}$to the most similar cluster Ct as before.

After all the patterns are considered, if none of the cluster assignments has changed, SCC stops with K clusters. If the cluster assignments of some patterns have changed, we update the weights W by minimizing

the objective function of Eq.(1). Then we proceed with the next round of clustering.

4. Experimental Results

In this section, we present and compare the experimental results of several clustering methods on six real world time series datasets: SynControl, Trace, CBF, ECGFiveDays, FaceFour, and OliveOil [11]. For convenience, our proposed method is called SCC with weights, abbreviated as SCC-W. The characteristics of the six datasets are listed in Table I. In this table, column 1 indicates the name of the dataset, and the remaining columns indicate the number of instances, the number of features, and the number of classes, respectively, in each dataset. Note that these datasets are single-labeled, i.e., an instance belongs to only one class.

Table I
DESCRIPTIONS OF DATASETS

Dataset	# instances	# features()	# classes
SynControl	600	60	6
Trace	200	275	4
CBF	930	128	3
ECGFiveDays	884	136	2
FaceFour	350	112	4
OliveOil	60	570	4

Table II
PERFORMANCE COMPARISONS

Dataset	Algorithm	Fscore	RI	NMI	#clusters	CPU time
CBF	SCC	0.5451	0.5526	0.2240	3	0.03
	SCC-W	**0.7452**	0.6899	**0.4834**	3	1.70
ECGFiveDays	SCC	0.5454	0.5020	0.0040	2	0.03
	SCC-W	0.555	0.5058	0.0094	2	1.25
FaceFour	SCC	0.5977	0.5842	0.4524	4	0.01
	SCC-W	**0.7967**	**0.8379**	**0.6954**	4	1.10
SynControl	SCC	0.5254	0.6813	0.5759	6	0.03
	SCC-W	0.6544	0.8262	0.7480	6	0.60
Trace	SCC	0.6070	0.7505	0.5613	4	0.01
	SCC-W	**0.6393**	**0.7610**	**0.5844**	4	0.63
OliveOil	SCC	0.5532	0.468	0.3356	4	0.01
	SCC-W	**0.8679**	**0.8847**	**0.7460**	4	2.30

For the sake of fairness in comparison, a method was applied on every dataset ten times and the average of the results of the ten runs is then presented. To evaluate the effectiveness of these methods, the following performance measures are adopted [12]: Fscore , RI and NMI . All these measures have a common property: a higher measure indicates a better clustering performance.

The results after clustering are shown in Table II. It can be seen that SCC-W outperforms SCC for all the six datasets. However, more CPU time is required SCC-W.

5. Conclusion

We have presented a weight-based self-constructing clustering method for time series data. Self-constructing clustering processes all the data points incrementally. If a data point is not similar enough to an existing cluster, then (1) if the point currently does not belong to any cluster, it forms a new cluster of its own; (2) otherwise, the point is removed from the cluster it currently belongs to before a new cluster is formed. However, if a data point is similar enough to an existing cluster, then (1) if the point currently does not belong to any cluster, it is added to the most similar cluster; (2) otherwise, it is removed from the cluster it currently belongs to and added to the most similar cluster. During the clustering process, weights are learned and considered in the calculations of similarity between data points and clusters. Experimental results have shown that our proposed approach performs more effectively than other methods for real world time series datasets.

References

1. S. Haykin. 1999. *Neural Networks – A Comprehensive Foundation*. Upper Saddle River, NJ, USA: Prentice-Hall.
2. D. L. Olson and Y. Shi. 2007. *Introduction to Business Data Mining*, vol. 10. McGraw-Hill/Irwin Englewood Cliffs.
3. C.-S. Ouyang, W.-J. Lee, and S.-J. Lee. 2005 .A TSK-type neurofuzzy network approach to system modeling problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 35(4):751-767.
4. S.-J. Lee, C.-S. Ouyang, and S.-H. Du. 2003. A neuro-fuzzy approach for segmentation of human objects in image sequences. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 33(3):420-437.
5. T.-Y. Chiu , T.-C. Hsu , C.-C. Yen , J.-S. Wang. 2015. Interpolation based consensus clustering for gene expression time series. *BMC bioinformatics*, 16(1):117.
6. B. DeVries , J. Verbesselt , L. Kooistra , M. Herold. 2015. Robust monitoring of small-scale forest disturbances in a tropicalmontane forest using landsat time series. *Remote Sensing of Environment*, 161:107-121.
7. X. Huang , Y. Ye , L. Xiong , R.Y. Lau. , N. Jiang , S. Wang. 2016 . Time series k-means: A new k-means type smooth subspace clustering for time series data. *Information Sciences*, 367:1-13.
8. L. Rabiner , B.-H. Juang. 1993 . *Fundamentals of speech recognition*. Prentice hall.
9. Y. Chen , M. Nascimento , B.C. Ooi , A.K. Tung . 2007. SpADe: On shape-based pattern detection in streaming time series. *Proceedings of the 23rd IEEE International Conference on Data Engineering*, pages 786-795, IEEE.
10. T. W. Liao.2005. Clustering of time series data—a survey. *Pattern recognition*, 38(11):1857-1874.
11. Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, G. Batista. 2015.The UCR time series classification archive.
12. X. Huang , Y. Ye , H. Guo , Y. Cai , H. Zhang , Y. Li. 2014. DSKmeans: a new kmeans-type approach to discriminative subspace clustering. *Knowledge-Based Systems*, 70:293-300.