A Time Series Forecasting Method

Zhao-Yu Wang[1], Yu-Chun Lin[2], Shie-Jue Lee[3], Chih-Chin Lai[4]

[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 Sun Yat-sen University, Kaohsiung, Taiwan*
[4]*Department of Electrical Engineering, National University of Kaohsiung, Kaohsiung, Taiwan*
1zywang@water.ee.nsysu.edu.tw, 2yuchun@water.ee.nsysu.edu.tw, 3leesj@mail.ee.nsysu.edu.tw, 4 cclai@nuk.edu.tw

Abstract: This paper proposes a novel time series forecasting method based on a weighted self-constructing clustering technique. The weighted self-constructing clustering processes all the data patterns incrementally. If a data pattern is not similar enough to an existing cluster, it forms a new cluster of its own. However, if a data pattern is similar enough to an existing cluster, it is removed from the cluster it currently belongs to and added to the most similar cluster. During the clustering process, weights are learned for each cluster. Given a series of time-stamped data up to time t, we divide it into a set of training patterns. By using the weighted self-constructing clustering, the training patterns are grouped into a set of clusters. To estimate the value at time t + 1, we find the k nearest neighbors of the input pattern and use these k neighbors to decide the estimation. Experimental results are shown to demonstrate the effectiveness of the proposed approach.

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 ofartificial 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 TSKmeans 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 forecasting. 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, it forms a new cluster of its own. However, if a data point is similar enough to an existing cluster, it is removed from the cluster it currently belongs to and added to the most similar cluster. In each iteration of clustering, weights are learned for each cluster. When convergence is reached, the clustering process stops with a desired number of clusters.

The weight-based self-constructing clustering technique is then applied to time series forecasting. Given a series of time-stamped data up to time t, we divide it into a set of training patterns. By using the weight-based clustering, the training patterns are grouped into a set of clusters. To estimate the value at time t + 1, we find the k nearest neighbors of the input pattern and use the weighted sum of the centers of these k neighbors to be the estimate. Experimental results are shown to demonstrate the effectiveness of the proposed approach.

The rest of this paper is organized as follows. The weight-based self-constructing clustering technique is described in Section 2. Our proposed time series prediction method is presented in Section 3. Experimental results are shown in Section 4. Section 5 gives a conclusion.

2. Weight Self-Constructing Clustering

We describe the weighted self-constructing clustering algorithm in detail. Also, we improve the method by incorporating weights in the calculation of similarity,

just as TKmeans does to K-means. First of all, we briefly introduce TSKmeans [7]. 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 Xi = $\{X_{i1}, X_{i2} \ldots, X_{im}\}$ 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 Xi belongs to cluster p and u_{ij}, j $\neq$ 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 jW_{pj} being the weight of the jth time stamp for the pth cluster. The purpose of TSKmeans is to minimize the following objective function:

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

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 D_{pi} 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 $\neq$ 0.

◎Step 2. Update the centers of all clusters by

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

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.

We perform several rounds of clustering until convergence is reached. In each round, one full training cycle on the training set of N patterns $(X^{(1)}, Y^{(1)})$, $(X^{(2)}, Y^{(2)})$, . . . , $(X^{(N)}, Y^{(N)})$, is done. Two parts are involved in a pattern. The first part is the input vector and the second part is the corresponding output. Note that the dimensionality of the input vector is m. Let K be the number of existing clusters. Each cluster C_p, 1 ≤ p ≤ K is characterized by its center $Z_p = \{Z_{p1}, Z_{p2}, \ldots, Z_{pm}\}$, deviation $V_p = \{V_{p1}, V_{p2}, \ldots, V_{pm}\}$, height H_p, size S_p, and weight $W_p = \{W_{p1}, W_{p2}, \ldots, W_{pm}\}$. Initially, K is 0. Suppose we are in the rth round, r ≥ 1. For pattern i, $(X^{(i)}, Y^{(i)})$ 1 ≤ i ≤ N, where $X^{(i)} = \{x_1^{(i)}, x_2^{(i)}, \ldots, x_m^{(i)}\}$, we calculate the similarity between $(X^{(i)}, Y^{(i)})$ and each existing cluster by

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

$$L_p(Y^{(i)}) = |H_p - Y^{(i)}| \quad (7)$$

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

Case 1. If

$$\mu_p(X^{(i)}) < \epsilon \ \text{ or } L_p(Y^{(i)}) > \psi \quad (8)$$

for 1 ⩽ p ⩽ K, we do the following:

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

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

Case 2. If

$$\mu_p(X^{(i)}) \ge \epsilon \ \text{ or } L_p \le \psi \quad (9)$$

for some existing clusters, we do the following:

1. If $(X^{(i)}, Y^{(i)})$ currently does not belong to any cluster, we add it to the most similar cluster, say Ct, and update the characteristics of Ct.

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

After all the patterns are considered, if none of the cluster assignments has changed, we stop 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.

3. Time Series Prediction

Suppose we are given a time series A that is indexed by the natural numbers as

$$A1, A2, \ldots. At \qquad (10)$$

where Ai denotes the value taken at time i. We'd like to estimate the value at time $t + 1$ based on these t values.

Firstly, we take each window of continuous q values and the value following the window as a training pattern. As a result, we have $n = t−q$ training patterns. Each training pattern looks like (X, Y). For example, the first training pattern is (X_1, Y_1) where $X_1 = \{A_1, A_2, \ldots, A_q)$ and $Y_1 = A_{q+1}$, the second training pattern is (X_2, Y_2) where $X_2 = \{A_2, A_3, \ldots, A_{q+1})$ and $Y_2 = A_{q+2}, \ldots$, and the last training pattern is (X_2, Y_2) where $X_{t−q} = \{A_{t−q}, A_{t−q+1}, \ldots, A_{t−1}\}$ AND $Y_{t−q} = A_t$.

Secondly, we use the weight-based clustering method of the previous section to cluster these n training patterns into J clusters $C_1, C_2, \ldots, C_J$. Each cluster C_i, $1 \le i \le J$ has its center $Z_i = \{z_{i1}, z_{i2}, \ldots, z_{im}\}$, deviation $V_i = \{v_{i1}, v_{i2}, \ldots, v_{im}\}$, height H_i, size S_i, and weight $W_i = \{w_{i1}, w_{i2}, \ldots, w_{im}\}$.

Thirdly, we take $X = \{x_1, x_2, \ldots, x_q\} = \{A_{t-q+1}, A_{t-q+2}, \ldots, A_t)$ and compute

$$\mu_p(X) = \prod_{j=1}^{m} \exp\left[-w_{pj}\left(\frac{x_j - z_{pj}}{v_{pj}}\right)^2\right] \qquad (11)$$

for Cp, $1 \le p \le K$. Then we find k clusters with the largest k μ's. Let these clusters be $C_{p1}, C_{p2}, \ldots, C_{pk}$.

Finally, we estimate the value at time $t + 1$ as

$$\hat{A}_{t+1} = \frac{\mu_{p1}(X)H_{p1} + \mu_{p2}(X)H_{p2} + .. + \mu_{pk}(X)H_{pk}}{\mu_{p1}(X) + \mu_{p2}(X) + .. + \mu_{pk}(X)} \qquad (12)$$

and we are done.

4. Experimental Results

In this section, we present and compare the experimental results of several clustering methods on the Poland dataset [8]. There are 1,500 instances in the dataset. The first 1,000 instances are taken for training and the remaining instances are taken for testing. The methods to be compared with include ANFIS [9], NN-MAT [10], ARMA [11], and Sorjamaa [12]. the original K-means, TSKmeans, and SCC. For convenience, our proposed method is called SCC with weights, abbreviated as SCC-W. To evaluate the effectiveness of these methods, the following performance measures are adopted [13]:

Maximum absolute error (MAE). It is defined as

$$MAE = \max |y_i - \hat{y}_i| \qquad (13)$$

where yi is the desired value and $\hat{y}_i$ is the predicted value. MAE takes the maximum difference magnitude among all the testing instances.

Root-mean-square error (RMSE). It is defined as

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N_t}(y_i - \hat{y}_i)^2}{N_t}} \qquad (14)$$

where yi is the desired value and $\hat{y}_i$ is the predicted value, and Nt is the number of testing instances.

These measures have a common property: a lower value indicates a better prediction performance. We have found that SCC-W provides the best performance in terms of both MAE and RMSE. The window size = 5 and k = 2 are used for BCC-W. However, From this table, different window sizes may affect the performance of SCC-W, as shown in Table 1. As can be seen, the performance of SCC-W slightly varies as window size varies. The forecasted values are very close to the target values. We can conclude that SCC-W has a good prediction capability.

Table1. effect of different window sizes(WS)

	MAE	RMSE
WS = 3	0.0415	0.0614
WS = 5	**0.0355**	**0.0556**
WS = 6	0.0394	0.0613
WS = 7	0.0380	0.0614
WS = 9	0.0399	0.0620

5. Conclusion

We have presented a weight-based self-constructing clustering method and applied it to time series forecasting. Given a series of time-stamped data up to time t, we divide it into a set of training patterns. By using the weight-based clustering, the training patterns are grouped into a set of clusters. To estimate the value at time $t + 1$, we find the k nearest neighbors of the input pattern and use the weighted sum of the centers of these k neighbors to be the estimate. Experimental results have shown that the proposed approach is effective and promising for time series forecasting.

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.   Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, G. Batista. 2015. The UCR time series classification archive.

9. J.-S. R. Jang. 1993. ANFIS: Adaptive-network based fuzzy inference systems. IEEE Transactions on Systems, Man, and Cybernetics, 23(3):665-685.

10.   M. T. Hagan, H. B. Demuth, and M. H. Beale. 1995. Neural Network Design. PWS Pub. Co..

11.   G. E. P. Box, G. M. Jenkins, and G. C. Reinsel. 2008 .Time Series Analysis: Forecasting And Control. WILEY.

12.   A. Sorjamaa, J. Hao, N. Reyhani, Y. Ji, and A. Lendasse. 2007. Methodology for long-term prediction of time series. Neurocomputing, 70(16-18):2861-2869.

13 . 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.