Journal of Southeast University (English Edition) Vol. 24, No. 3, pp. 264 – 266 Sept. 2008 ISSN 1003—7985

Clustering algorithm for multiple data streams based on spectral component similarity

Zou Lingjun[1] Chen Ling[1,2] Tu Li[3]

([1]Information Engineering College, Yangzhou University, Yangzhou 225009, China)
([2]State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China)
([3]College of Information Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

Abstract: A new algorithm for clustering multiple data streams is proposed. The algorithm can effectively cluster data streams which show similar behavior with some unknown time delays. The algorithm uses the autoregressive (AR) modeling technique to measure correlations between data streams. It exploits estimated frequencies spectra to extract the essential features of streams. Each stream is represented as the sum of spectral components and the correlation is measured component-wise. Each spectral component is described by four parameters, namely, amplitude, phase, damping rate and frequency. The ε-lag-correlation between two spectral components is calculated. The algorithm uses such information as similarity measures in clustering data streams. Based on a sliding window model, the algorithm can continuously report the most recent clustering results and adjust the number of clusters. Experiments on real and synthetic streams show that the proposed clustering method has a higher speed and clustering quality than other similar methods.

Key words: data streams; clustering; AR model; spectral component

Massive volumes of data streams can be found in numerous applications. Stream data are massive, continuous, temporally ordered, dynamically changing, and potentially infinite[1-2]. For the stream data applications, the volume of data is usually too huge to be stored or to be scanned more than once. Furthermore, in data streams, the data points can only be sequentially accessed. Recently, an abundant body of researches on data stream clustering has emerged[3-11].

Some real streams or their subsequences may have lag correlations; they demonstrate highly similar rise/fall patterns, neglecting some lags or shifts on the time axis. Since in such stream clustering, traditional similarity measures such as the Euclidean distance or the correlation coefficient cannot be helpful in revealing lagged similarities among data streams, this problem may be challenging. Therefore, a new technique to cluster such streams is needed to effectively detect such lagged similarities overlooked by traditional methods.

A clustering algorithm for data streams based on the autoregressive modeling technique[12-13] is proposed. The algorithm uses estimated frequencies spectra to extract the essential features of streams. Each stream is represented as the sum of spectral components and the correlation is measured component-wise. We calculate the ε-lag-correlation between two spectral components and use such information as similarity measures in clustering data streams. Experimental results show that our algorithm has a better clustering quality than other algorithms.

Received 2008-04-15.
Biographies: Zou Lingjun (1984—), female, graduate; Chen Ling (corresponding author), male, professor, lchen@yzcn.net.
Foundation items: The National Natural Science Foundation of China (No. 60673060), the Natural Science Foundation of Jiangsu Province (No. BK2005047).
Citation: Zou Lingjun, Chen Ling, Tu Ii. Clustering algorithm for multiple data streams based on spectral component similarity[J]. Journal of Southeast University (English Edition), 2008, 24(3): 264 – 266.

1 AR Model and Spectral Component-Wise Correlation

For data streams $Y=(y_1, y_2, \ldots, y_n, \ldots)$ and $Y^{(d)}=(y_{d+1}, y_{d+2}, \ldots, y_{d+n}, \ldots)$, we call Y the d-lagged stream of $Y^{(d)}$. Given two streams X and Y and a threshold $\varepsilon \in (0, 1]$, for window size L, if $|\rho(X(L), Y^{(d)}(L))| \geqslant \varepsilon$, streams X and Y are called ε-lag-correlated data streams, $\rho(X(L), Y^{(d)}(L))$ is the correlation coefficient between $X(L)$ and $Y^{(d)}(L)$. Given the time horizon for clustering L, a threshold $\varepsilon \in (0, 1]$ and the number of clusters k, the clustering algorithm partitions n data streams into k clusters $C(L)=\{C_1(L), C_2(L), \ldots, C_k(L)\}$, so that data streams in the same cluster are ε-lag-correlated and minimize some objective function measuring the quality of clustering in the period $[t-L+1, t]$, here t is the time when the analysis is performed.

To detect such lagged correlations between the data streams, we use an AR modeling technique to extract relevant features and ignore some irrelevant ones which may corrupt our similarity search. Let a subsequence of a data stream be $x_i=\{x_{it}\}$, $t=1, 2, \ldots, n$. Its AR(n) is $x_{in}=a_{i1}x_{i,n-1}+a_{i2}x_{i,n-2}+, \ldots, +a_{in}x_{in-k}+c_{in}$, where c_{in} represents a noise drawn uniformly from $[0, 1]$ and $a_{i1}, a_{i2}, \ldots, a_{in}$ are coefficients. By solving the linear equation system represented by the Yule-Walker form, coefficients $a_{i1}, a_{i2}, \ldots, a_{in}$ can be obtained. Then the features such as the amplitude, the phase, the damping factor and the frequency of components can be estimated as follows: First, sequence x_i can be decomposed into a set of discrete complex exponentials or sinusoids. That is

$$x_i = \sum_{k=1}^{K} \alpha_{ik} e^{j\varphi_{ik}} e^{(\sigma_{ik}+j\omega_{ik})n} = \sum_{k=1}^{K} c_{ik} z_{ik}^{n} \quad (1)$$

where $\alpha_{ik}, \varphi_{ik}, \sigma_{ik}$ and ω_{ik} denote amplitude, phase, damping factor and frequency of the sinusoidal component k of stream I, respectively. Since the AR coefficients $a_{i1}, a_{i2}, \ldots, a_{in}$ are also the coefficients in a polynomial $P(z_i)$, these parameters can be obtained by solving the following equation of z_i.

$$P(z_i) = \prod_{k=1}^{K}(z_i - z_{ik}) = \sum_{k=0}^{K} a_{ik} z_i^{K-k} \qquad a_{i0}=1 \quad (2)$$

Therefore, after the roots z_{ik} of Eq. (2) being calculated, the complex amplitudes c_{ik} can be calculated by Eq. (1), and, hence, the spectral components $(\alpha_{ik}, \varphi_{ik}, \sigma_{ik}, \omega_{ik})$ of a sequence can be obtained. Then the spectral component based similarity between x_a and x_b can be computed by

$$\rho(x_a, x_b) = \arg \max_{\alpha_{ai}, \alpha_{bk} > \alpha_T} | R(x_a^{(i)}, x_b^{(k)}) | \quad (3)$$

where α_T denotes the spectral component amplitude threshold which is used to remove the noisy data. In Eq. (3), $R(x_a^{(i)}, x_b^{(k)})$ is actually the Pearson correlation coefficient between the two sequences in terms of their damping rates and frequencies.

2 Framework of the Algorithm SPE-Cluster

In our algorithm, we explore a sliding window technique model for clustering data streams. The framework of the proposed SPE-cluster algorithm is as follows:

Algorithm SPE-cluster
Input: New values at time t for n streams $x_1, x_2, \ldots, x_n$;
Output: Clustering results for each sliding window.
Begin:
read in the first w data from the streams and create initial clusters;
$t = w$;
while not end of streams do
read in $x_{ki}(t)$ for each stream x_k, $t = t + 1$;
if t mod $l = 0$ then
form a new basic window;
if the number of basic windows exceeds m then delete the oldest window;
compute lag correlations for each pair of streams in $(x_i[t - w + 1, t])$;
cluster the streams using k-medoids;
adjust k according to the clusters obtained;
output clustering results;
endif
Endwhile
End.

3 Experimental Results

To evaluate the performance of our algorithm, experiments are designed to compare the speeds and qualities of our algorithm with those of the DFT-cluster[8]. We perform experiments on real datasets of the daily stock prices obtained from http://finance.yahoo.com. We use a straightforward and easy metrics to evaluate the quality of the clustering results: if the structure we obtain is identical to that of the real one, we set the score to 1, otherwise the score is 0.

Our experiment performs 50 trials on the dataset and the trials use different lengths of basic widows. Fig. 1 shows the average quality of the clustering results using different lengths of basic widows. From Fig. 1, we can see that our algorithm has a higher clustering quality than the DFT-cluster.

We also test the processing speed of the SPE-cluster and compare it with the DFT-cluster (400 DFT coefficients). The experimental results show that the executing time for the SPE-cluster is shorter than that of the DFT-cluster for every data set. Fig. 2 shows that the average processing time per segment for the SPE-cluster is 0.928 s whereas it is 1.2 s for the DFT-cluster using 400 DFT coefficients. The DFT-cluster needs an even longer processing time when more coefficients are used. When using 1 500 DFT coefficients, the DFT-cluster takes an average of over 7 s. Reducing the number of DFT coefficients can save time but leads to poorer quality. The DFT-cluster with 250 DFT coefficients has much poorer quality than does the SPE-cluster on these data sets.

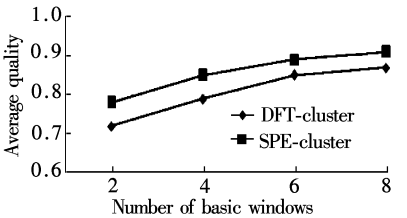

Fig. 1 Average clustering quality under different basic windows for DFT-cluster and SPE-cluster

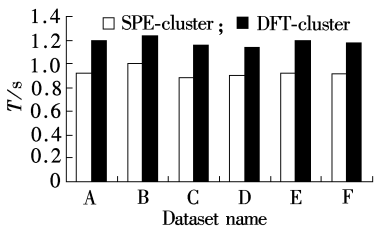

Fig. 2 Computation time of SPE-cluster and DFT-cluster

4 Conclusion

In this paper, we concentrate on the problem of clustering multiple data streams. In these data streams, there may be some streams that are highly correlated with others but with time delays. We use an AR modeling technique to decompose these streams into a set of sinusoids of various frequencies and measure correlation similarity using spectral component information. We cluster data streams within a sliding window, and continuously find cluster structures. Experimental results show that our algorithm can effectively cluster highly correlated data streams, some of which may have time delays.

References

[1] Han J, Kamber M. *Data mining*: *concepts and techniques* [M]. 2nd ed. Beijing: China Machine Press, 2006: 467 – 531.

[2] Babcock B, Babu S, Datar M, et al. Models and issues in data stream systems [C]//*Proceedings of the* 21*st ACM Symp on Principles of Databases Systems*. Madison: ACM Press, 2002: 1 – 16.

[3] Guha S, Meyerson A, Mishra N, et al. Clustering data streams: theory and practice [J]. *IEEE Transactions on Knowledge and Data Engineering*, 2003, **3**(15): 515 – 528.

[4] Aggarwal C C, Han J, Wang J, et al. A framework for projected clustering of high dimensional data streams [C]//*Proceedings of the VLDB*. Toronto: Morgan Kaufmann Publishers, 2004: 852 – 863.

[5] Nam H, Won S. Statistical grid-based clustering over data

streams [J]. *SIGMOD Record*, 2004, **33**(1): 32 – 37.
[6] Cao F, Ester M, Qian W, et al. Density-based clustering over an evolving data stream with noise[C]//*Proceedings of the* 2006 *SIAM Conference on Data Mining*. Springer, 2006: 326 – 337.
[7] Nasraoui O, Cardona C, Rojas C, et al. TECNO-STREAMS: tracking evolving clusters in noisy data streams with a scalable immune system learning model [C]//*Proceedings of the* 3*rd IEEE Intl Conf on Data Mining*. Melbourne, 2003: 235 – 242.
[8] Beringer J, Hullermeier E. Online clustering of parallel data streams[J]. *Data and Knowledge Engineering*, 2006, **58**(2): 180 – 204.
[9] Aggarwal C C, Han J, Wang J, et al. On demand classification of data streams [C]//*Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Seattle: ACM Press, 2004: 503 – 508.
[10] Dai B, Huang J, Yeh M, et al. Adaptive clustering for multiple evolving streams [J]. *IEEE Transactions on Knowledge and Data Engineering*, 2006, **18**(9): 1166 – 1180.
[11] Sakurai Y, Papadimitriou S, Faloutsos C. BRAID: stream mining through group lag correlations [C]//*Proceedings of the* 2005 *ACM SIGMOD Intl Conf on Management of Data*. Baltimore: ACM Press, 2005: 599 – 610.
[12] Yeung L K, Szeto L K, Liew A W C, et al. Dominant spectral component analysis for transcriptional regulations using microarray time-series data [J]. *Bioinformatics*, 2004, **20**(5): 742 – 749.
[13] Yeung L K, Yan H, Liew A W C, et al. Measuring correlation between microarray time-series data using dominant spectral component [C]//*Proceedings of the* 2*nd Asia-Pacific Bioinformatics Conference*. Dunedin: Australian Computer Society, 2004: 309 – 314.

一种基于谱分量相似度的多数据流聚类算法

邹凌君[1] 陈 崚[1,2] 屠 莉[3]

([1] 扬州大学信息工程学院,扬州 225009)
([2] 南京大学计算机软件新技术国家重点实验室,南京 210093)
([3] 南京航空航天大学信息科学与技术学院,南京 210016)

摘要:提出了一种新的多数据流聚类算法.该算法可以有效地对有相似行为但存在一定时间延迟的多数据流进行聚类.算法采用自回归模型技术度量数据流间的延迟相关,利用频谱估计来抽取数据流的特征.每一个数据流用其谱分量的和来表示,从而来计算每对数据流间的相关关系.每个谱分量用振幅、相位、衰减率、频率4个参数来描述.算法计算谱分量对之间的 ε-延时相关关系,并以此为基础来得到聚类分析中数据流间距离的度量.此外,算法采用滑动窗口技术对多数据流进行聚类,实时地得出聚类结果且动态地调节聚类的个数.在人工数据集和实际数据集上的实验结果表明,所提出的算法比其他类似的算法具有更快的速度和更好的聚类效果.

关键词:数据流;聚类;AR 模型;谱分量
中图分类号:TP311