An approach based on statistical features for extracting human actions from multivariate time series

Jun Gui, Zhaobo Qin, Danping Jia and Jianguang Zhu
School of Information Science and Engineering, Shenyang University of Technology, Shenyang, Liaoning, 110870, China
*Corresponding author’s e-mail: jiadp@sut.edu.cn

Abstract. Extracting system behavior from the multivariate time series collected by sensors is more challenging. If regarded people as system, the various physical actions can be considered as different states of the system. Our intention was to produce a simple, flexible and accurate method for clustering multivariate time series to obtain the system state. Previously, we proposed a distribution-based state extraction method (DBSE) that can effectively extract motion from the multivariate time series of human leg activities. In this paper, we propose an Auto-segmented state extraction method (AS-DBSE) by improving the size of the segmented window to adapt to more complex situations. The method uses vectors composed of statistical feature parameters to represent time series. It uses distance as the similarity measure and clusters these vectors to get the state of the system. We have proved the effectiveness and simplicity of this method through research on human arm activity data. Finally, compared with clustering based on Toeplitz inverse covariance (TICC) and symmetric non-negative matrix factorization (SymNMF), the results of the new method are more satisfactory.

1. Introduction
The driving vehicle can be regarded as a system, and the state of the system can be expressed as acceleration, turning, braking, etc[1]. Similarly, people are regarded as a system, various human activities can be considered as the state of the system. The paper assumes that the system behavior is composed of multiple stable states, which may repeatedly appear in a period of time. In a moment, the state will not change suddenly. During this time, it will tend to stabilize and comply with statistical distribution.

Time series (TS) refers to a series of observations denoted as $TS_i(t)$, where $TS_i(t)$ indexes the measurement on the $i^{th}$ variable at time point $t$. It is called univariate time series (UTS) when $i$ is equal to 1, and multivariate time series (MTS) when $i$ is greater than 1[2]. At present, MTS is widely used in biology, industry, finance and other fields. Our intention is to quickly and efficiently extract the system behavior from multiple and complex series, which is of great significance for analyzing system, anomaly detection, and using life prediction.

When studying UTS, methods such as Fourier transform, wavelet transform and constructing ARIMA model are usually used. However, some of these methods are not suitable for processing MTS[3].

Some methods based on each sampling point, i.e., Symmetric NMF (SymNMF) is proposed as a general framework for clustering, which is based on a similarity measure between data points[4]. This method has good performance for shorter sequences, but consumes too much computing resources for long sequence matrix. In addition, the data cannot appear missing values.
Sliding windows designed to reduce the length of series that need to be processed. A multi-stage clustering method for MTS using dynamic sliding window is proposed. The first stage clusters the data in each sliding window, and the second stage clusters the results of the first stage[5]. In another article with greater impact, the author extracts data in different sliding windows and conducts similarity analysis through the Toeplitz Inverse Covariance matrix[1]. Sliding window method considers each point and surrounding several points, which can enhance details, but consumes time and resources.

More methods process segmented sequences. Each subsequence is represented as a set of coarse-grained fuzzy information. Convert the original problem from the digital level to the granular level to get the sequence status more effectively[6]. A model-based clustering algorithm is proposed, which uses confidence values to represent the relationship between different variables, thereby discovering different patterns of subsequences. Then cluster according to the pattern[7].

Although the ability to obtain states is improving, there are still many imperfections. In order to improve accuracy, many algorithms greatly increase the complexity of the operation. In addition, many documents have not explored the segmented size of series.

This paper proposes an Auto-segmented distribution-based multivariate time series state extraction method (AS-DBSE). This method can extract state or system behavior without any relevant domain knowledge or model assumptions. It can process high-frequency and noisy data. The method continuously adjusts the size of the segmented sequence. Because the state in a short time is stable, the sampled values conform statistical distribution. The segmented multivariate series can be expressed by distribution parameter vector. The method uses a matrix to represent the processed multivariate time series, where each vector is a subsequence. Based on the obtained parameters, using distance as the similarity measure, cluster the vectors to get the state of the system. Finally, we compared the performance with TICC (Toeplitz Inverse Covariance-Based Clustering) and SymNMF (Symmetric Nonnegative Matrix Factorization) algorithms.

The contributions of this paper can be summarized as follows:

1. Verified the feasibility of using distribution parameter vectors (Mean, Variance, Skewness) instead of subsequences to extract the state of multivariate time series.

2. The method combining Silhouette Coefficient and Purity is used to select the segmented window size of subsequence.

3. Analyze the experimental results and compare with recent popular algorithms (TICC, SymNMF).

2. Theory and design of method

The multivariate time series is composed of multiple sequences, \( MTS = \{S_1, S_2, ..., S_m\} \). Where \( S \) represents the sequence. \( T \) represents the observation point on the time series \( S \). Each sequence has \( n \) observation points, each time series \( S_i = \{T_1, T_2, ..., T_n\} \). The method uses non-overlapping windows segmented series to obtain subsequence of dimension \( M \) and length \( \omega \). The MTS after segmented can represent \( \{\text{Sub}_1, \text{Sub}_2, ..., \text{Sub}_{n/\omega}\} \). Each subsequence is a short period of time, \( n \gg \omega \). In a short period of time, the system will be in a steady state, can be represented by statistical distribution. Figure1(a) Shows the distribution of sampling points for different actions over a period of time. Figure1 (b) Shows the distribution of sampling points in different periods of rolling bearings[8].

![Figure 1](image_url)

Figure 1. (a) The distribution of sampling points for actions in a short time. (b) Distribution of sampling points in rolling bearings at different short times.
We hope to express multivariate subsequences through statistics feature parameters. The mean (1) can represent the concentrated position of a certain variable in a subsequence over a period of time. Each variable of the multivariate subsequence will get an average value. Variance (2) can measure the dispersion of observations of a variable over a period of time. Each variable of the multivariate subsequence will get a variance. The entire subsequence can also be represented by a set of deviations indicating the degree of dispersion of the observations in each variable. Skewness (3) is a measure of the asymmetry of the probability distribution of a real-valued random variable about its mean. The skewness value can be positive, zero, negative, or undefined.

\[
\mu_{\text{sub}} = \frac{1}{\omega} \sum_{i=1}^{T_{\text{sub}}} t_{\text{sub}}^i (1), \quad \sigma_{\text{sub}} = \frac{1}{\omega} \sum_{i=1}^{T_{\text{sub}}} (T_{\text{sub}}^i - \mu_{\text{sub}})^2 (2), \quad \text{Skew}_{\text{sub}} = \frac{1}{\omega^{3/2}} \sum_{i=1}^{T_{\text{sub}}} (T_{\text{sub}}^i - \mu_{\text{sub}})^3 (3)
\]

As above, each subsequence can be represented by a vector. The vector in MTS (4) contains various statistics feature parameters that can describe the distribution of observations in the subsequence. Each vector in the matrix can be understood as a point in a multi-dimensional space. A clustering method based on Euclidean distance is adopted for these points. Use \( x \) to represent each vector in the matrix.

The two hyperparameters of the method, cluster centers number \( k \) and window size \( \omega \) are unknown. The method obtains hyperparameters through the combination of Silhouette Coefficient and Purity. The Silhouette Coefficient (\( S(i) \)) is used as the basis for evaluating clustering (5). The value of \( S(i) \) is \([-1, 1]\). Value closer to 1, the better the result [9].

\[
\text{MTS} = \begin{bmatrix}
\mu_{11} & \mu_{12} & \cdots & \mu_{1n} \\
\sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\
\text{Skew}_{11} & \text{Skew}_{12} & \cdots & \text{Skew}_{1n} \\
\vdots & \vdots & \ddots & \vdots \\
\mu_{m1} & \mu_{m2} & \cdots & \mu_{mn} \\
\sigma_{m1} & \sigma_{m2} & \cdots & \sigma_{mn} \\
\text{Skew}_{m1} & \text{Skew}_{m2} & \cdots & \text{Skew}_{mn}
\end{bmatrix}
\]

(4), \( S(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \), \( S(i) = \begin{cases} 
1 - \frac{a(i)}{b(i)}, & a(i) < b(i) \\
0, & a(i) = b(i) \\
\frac{a(i)}{b(i)} - 1, & a(i) > b(i)
\end{cases} \) (5)

We use iterative distance-based clustering. Assuming that the number of clusters is \( k \), \( k = \{u_1, ..., u_k\} \). Traverse different window sizes, and then traverse different \( k \) to obtain clustering results. Distance-based clustering optimizes the objective function \( f(y, u) \) through iteration. The variables that need to be optimized are \( y_{ij} \) and \( u \). In Assignment Step (7), Vector \( x \) in MTS is assigned as the cluster represented by the nearest cluster center. Obviously, this step will reduce the value of \( u_j \). In Update Step (8), the center position of all vectors \( x \) of this cluster is taken as the new cluster center. It will eventually converge.

\[
f(y, u) = \sum_{i=1}^{n/\omega} \sum_{j=1}^{k} y_{ij} \|x_i - u_j\|^2 (6), \quad y_{ij} = \begin{cases} 
1, & \text{argmin} \|x_i - u_j\|^2 (7), \quad u_j = \frac{\sum_{i=1}^{n/\omega} y_{ij} x_i}{\sum_{i=1}^{n/\omega} y_{ij}} (8), \quad \text{purity} = \sum_{i=1}^{m} \frac{m}{m} P_i (9)
\end{cases}
\]

According to this clustering method and evaluation index, the optimal number of clusters \( k \) and window size \( \omega \) are obtained, each subsequence (Sub_{\text{ij}}) will be assigned cluster label. About window size \( \omega \) will be discussed in the next chapter.

3. Experiment and research
In the previous work[10], using the method to process leg sensor signal, we found four states of the leg(stable, vigorous exercise, leg lifting and dropping). During the previous work, the segmented window size was not discussed. But, the arm actions have higher degree of freedom. It is more difficult to extract the state. Therefore, the paper research segment window size.

3.1. Description of action
Accelerate value is generated by the sensor placed on the arm of the object. X-axis, Y-axis and Z-axis acceleration are recorded. The frequency is 100Hz, and the action includes 12 actions. Lying, sitting(sitting in a chair in whatever posture the subject feels comfortable), changing sitting postures is
also allowed), standing, running, cycling, Nordic walking, ascending stairs, descending stairs, vacuum cleaning, ironing, rope jumping[11].

3.2. Segmented Window Size Research

In the known literature, the size of the segmented window is rarely mentioned. In order to improve method efficiency, the subsequence is used to represent the transient stable state. If the window size is small, it means that the number of sampling points is not enough to effectively obtain transient statistics. Although an excessively large window size can greatly improve the operation efficiency, it will make stability worse. This section verifies the above estimate by constantly adjusting the window size to study the optimal value.

The frequency of sampling is 100Hz. It be used as baseline. The initial segmented size of the window is 1 second, expressed by $\omega$, and each test increases by $0.5\omega$. The Silhouette coefficients corresponding to different sizes of segmented windows and the number of cluster centers are shown in Figure 2.

![Figure 2. The Silhouette Coefficient changes when selecting different segmented windows and cluster centers numbers.](image)

Search for optimal hyperparameters. In Figure 2, the Silhouette Coefficient shows better results at $(3\omega, k = 8)$, $(2.5\omega, k=13)$ and $(3.5\omega, k=8)$. The optimal size of the window can be confirmed between 2.5$\omega$ and 3$\omega$. Using Purity(9) to determine the best hyperparameter is $(3\omega, k = 8)$. Purity is a simple clustering evaluation method [12]. Where $P_i = \max(P_{ij})$, $P_{ij}$ refers to the probability that the member in cluster $i$ belongs to class $j$.

The dataset used in the experiment has labels, which can intuitively measure the clustering effect. Table 1 shows the processing results. The column of the table is the action of the measured object, that is, the label of the dataset. The index of the table is the cluster label. If almost all points of an action in the column are concentrated in a label of index, the state can be successfully extracted. To simplify the expression, the percentage is filled in the table. It reflects the degree to which the method extracts actions.

| Action          | lying | sitting | standing | walking | running | cycling | Nordic walk | ascending stairs | descending stairs | vacuum cleaning | ironing | rope jumping |
|-----------------|-------|---------|-----------|---------|---------|---------|-------------|------------------|------------------|-----------------|---------|-------------|
| 0               | 2.688 | 10.60   | 4.692     | 0.945   | 0.576   | 4.409   | 0.142       | 10.29            | 3.945            | 5.720           | 9.49    | 2.279       |
| 1               | 3.841 | 2.125   | 91.79     | 97.16   | 0       | 0       | 5.928       | 30.77            | 68.17            | 2.248           | 0       |             |
| 2               | 6.402 | 26.84   | 2.342     | 0.006   | 0       | 4.779   | 1.035       | 5.184            | 5.918            | 5.802           | 8.309   | 0           |
| 3               | 0     | 6.709   | 1.165     | 1.883   | 0       | 0       | 90.80       | 6.747            | 15.97            | 4.351           | 2.078   | 2.286       |
| 4               | 0     | 0       | 0         | 0       | 0       | 0       | 92.63       | 0                | 0                | 0                | 0       | 0.864       |
| 5               | 1.276 | 53.70   | 0         | 0       | 0       | 0       | 0           | 90.81            | 0                | 31.54           | 13.05   | 72.71       |
| 6               | 85.79 | 0       | 0         | 3.392   | 0       | 1.045   | 0           | 0                | 0                | 2.078           | 0       |             |
| 7               | 0     | 0       | 0         | 3.392   | 0       | 1.045   | 6.920       | 11.83            | 2.901            | 3.117           | 94.56   |             |
The size of the segmented window will have a greater impact on the final result of the method. The optimal hyperparameters will stabilize within a range. The Silhouette Coefficient combined with the Purity can effectively select the optimal segment size and the number of cluster centers.

3.3. Analysis of results
Except for sitting and descending stairs, most of the sampling values of actions are gathered into a certain state label. From the description of the actions, the sitting subject's hand move free. descending stairs is different from ascending stairs to overcome gravity, and the arms swing is freer. Therefore, the extraction effect of these two actions is not ideal.

Label 6 represents the state where the arm is almost stationary. Figure 3(a) shows that the acceleration value of the three axis fluctuate steady. The lying is individually divided into this state. It is easy to distinguish from other actions. label 1 is closer to the label 6. It indicates a slight sway of the arm. Standing, walking, ascending stairs and vacuum cleaning are classified into this state. Figure 3(b) can be seen from the probability density that the range of motion is larger than lying. But the movement is not violent and is more regular.

Label 5 represents the state that the arm remains raised. This state is also a non-vigorous compared to the state described above, but the arm is swaying irregularly. Both cycling and ironing are divided into this state. When cycling, the arm needs to be raised to control the driving of the bicycle through the handlebar. When ironing, the arm needs to constantly control the ironing tool. Figure 3(c) shows that the fluctuations of the sampling points from the X-axis and Y-axis are more obvious.

Running, Nordic walking and rope jumping are representative of vigorous actions of the arms. The method can successfully distinguish the three actions. Nordic walking is classified into state 2, running is state 3, and rope jumping is state 6. Figure 4 shows that the activities of running and Nordic walking perpendicular to the direction of the arm are small, and the activities of the arm on the X axis and the Y axis are vigorous. But the amplitude of running is greater. Rope skipping means that the activities in the three directions of the arm are very violent, especially the movements perpendicular to the direction of the arm.
4. Comparative experiment
We selected the five movements of the arm with obvious characteristics of lying (state 1), walking (state 2), running (state 5), cycling (state 4) and Nordic walk (state 3) as the comparison data set. The five actions represent five states. The method distinguishes the actions by extracting the states in the time series and compares the indicators with other algorithms.

4.1. Experiments and results
The paper use two recently popular algorithms to compare with our method. SymNMF is a general framework for graph clustering, which is based on a similarity measure between data points, and it factorizes a symmetric matrix containing pairwise similarity values. SymNMF inherits the advantages of NMF by enforcing non-negativity on the clustering assignment matrix.

Toeplitz Inverse Covariance-based Clustering is a new method of model-based clustering. Each cluster in the TICC method is defined by a correlation network, or Markov random field (MRF), describing the interdependencies between different observations in a subsequence of that cluster. TICC simultaneously segments and clusters the time series data. The problems are solved through alternating minimization and variation of the Expectation Maximization (EM) algorithm. Two resulting subproblems are solved through dynamic programming and the alternating direction method of multipliers (ADMM). Figure 5 shows the results of different algorithms. It can be seen that AS-DBSE (d) is closest to the label sequence (a) over time, which means that our method has the highest accuracy.

![Figure 5](image-url)

Figure 5. (a) Label sequence of data set. (b) State sequence generated by SymNMF algorithm. (c) State sequence generated by TICC algorithm. (d) State sequence generated by AS-DBSE.

We used multiple indicators to compare different methods. The Rand Index (10) and Adjust Rand Index (11) need to be given the actual category C. Assuming K is the clustering result, \( \alpha \) means that both C and K are in the same category, \( \beta \) means that both C and K are in different categories.

Where \( C_{K}^{\text{rs}} \) represents the total number of data sets that can be formed in the data set. The value range of RI is \([0,1]\). The larger the value, the better the clustering result is. However, for random results, RI does not guarantee that the index is close to 0, so it is proposed to adjust the rand coefficient.

\[
RI = \frac{\alpha + \beta}{C_{K}^{\text{rs}}} \tag{10}
\]

\[
ARI = \frac{RI - \frac{1}{N} \mathbb{E}[RI]}{\max(\mathbb{E}[RI]) - \frac{1}{N} \mathbb{E}[RI]} \tag{11}
\]

The range of ARI value is \([-1,1]\], the larger the value, the more consistent with the real situation. Mutual information (12) and adjusted mutual information (14) are used to measure the degree of correlation between two vectors in multivariate time series. The amount of information contained in one vector about another vector. Let U and V be the distribution of MTS, then the entropies of the two distributions are formula 12.
\[
\text{MI}(u, v) = \sum_{i=1}^{\mid U \mid} \sum_{j=1}^{\mid V \mid} P(i,j) \log \left( \frac{P(i,j)}{P(i)P(j)} \right) \tag{12}, \text{NMI}(u,v) = \frac{\text{MI}(u,v)}{\sqrt{H(u)H(v)}} \tag{13}, \text{AMI} = \frac{\text{MI} - E[\text{MI}]}{\max(H(u),H(v)) - E[\text{MI}]} \tag{14}
\]

Where \( P(i) = \frac{\mid U \mid}{N} \) and \( P'(j) = \frac{\mid V \mid}{N} \). The value range of MI and NMI is \([0,1]\), and the value range of AMI is \([-1,1]\). The larger the value, the better the clustering result matches the real situation.

We use two tables to display the processing results of the above indicators:

Table 2. Purity of each action

| SymNMF   | TICC    | AS-DBSE  |
|----------|---------|----------|
| Lying    | 0.5171531 | 0.75765077 | 0.90909091 |
| Walking  | 0.98109641 | 0.88964148 | 0.9857683 |
| Running  | 0.64809783 | 0.94136236 | 0.94695822 |
| Cycling  | 0.97370937 | 0.90452862 | 0.95977378 |
| Nordic walking | 0.7720619 | 0.74203666 | 0.88224217 |

Table 3. Accuracy of different algorithms

| SymNMF   | TICC    | AS-DBSE  |
|----------|---------|----------|
| ARI      | 0.48818 | 0.66971 | 0.835881 |
| NMI      | 0.604571 | 0.669711 | 0.840262 |
| AMI      | 0.56534 | 0.68222 | 0.835921 |
| Purity   | 0.699501 | 0.834227 | 0.936734 |

4.2. Analysis of result

Table 2 and Table 3 show that AS-DBSE has a satisfactory effect. In Table 2, taking Lying as an example, according to the action introduction, there will be some activities on the arm of the people when lying down. These activities can be regarded as disturbing movements relative to lying down. At this time, the advantages based on statistical distribution become obvious, the algorithm takes a short period of time as the processing unit, and tolerates some noise. However, the SymNMF with each point as a processing unit and the TICC with a center point and surrounding points as a processing unit cannot filter out the noise very well.

Figure 1 shows that in a short time the sampling points of many systems will follow an approximately normal distribution. By Auto-segmented, a suitable period of time (window) can be obtained. The size of the time period affects the distribution and determines the accuracy of the method. The speed of extracting distribution features from these time periods is much faster than other algorithms establishing point-by-point similarity associations. And after extracting features, the data volume is greatly reduced, and the clustering speed is greatly improved. In addition, because the method uses statistics, it is not sensitive to missing values, and there is no need to complete the data in the preprocessing.

5. Discussion

The paper uses the data of arm actions collected by multiple channels to verify the feasibility of the method to extract states from multivariate time series. The method uses multiple statistics to represent multiple subsequences. And the clustering method based on Euclidean distance is used to cluster high-dimensional vectors, where each vector is a multivariate subsequence expressed by multiple statistics.

The article discusses the selection of the window size of the segment sequence. Comparing the method with the current popular algorithms proves that the method has a very good performance in time without reducing the accuracy. The method does not optimize the seeding of clusters. The running speed can be further improved. The method does not introduce weights to the statistics, and the accuracy is expected to be further improved.
Acknowledgement
This work is supported by Liaoning Provincial Education Department Scientific Research Fund Project (LFGD2019006). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

References
[1] Hallac, David., Sagar Vare., Stephen Boyd. (2017) 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.
[2] Fei-Bao, Z., Tian-Qiang, H., Gong-De, G. (2009) WMCA: a weighted matrix coverage based approach to cluster multivariate time series. In 2009 Fifth International Conference on Natural Computation, vol. 1, pp. 382-386.
[3] Wang, X., Wirth, A., Wang, L. (2007) Structure-based statistical features and multivariate time series clustering. In Seventh IEEE International Conference on Data Mining (ICDM 2007), pp. 351-360.
[4] Zhou, L., Du, G., Tao, D. (2018) Clustering Multivariate Time Series Data via Multi-Nonnegative Matrix Factorization in Multi-Relational Networks. IEEE Access 2018, 6, pp.74747-74761.
[5] Ren, L., Wei, Y., Cui, J. (2017) A sliding window-based multi-stage clustering and probabilistic forecasting approach for large multivariate time series data. Journal of Statistical Computation and Simulation 2017, 13, 2494-2508.
[6] Wang X., Yu F., Zhang H. (2013) Granulation-based fuzzy clustering of large-scale time series. Paper presented at: 10th International conference on fuzzy systems and knowledge discovery; 2013 Jul 23–25; Shenyang, China. pp. 466–471.
[7] Zhou, P.Y., Chan, K. C. (2014) A model-based multivariate time series clustering algorithm. In Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 805-817.
[8] Wang, B., Lei, Y., Li, N. (2018) A hybrid prognostics approach for estimating remaining useful life of rolling element bearings. IEEE Transactions on Reliability 2018.
[9] Rousseeuw, P. J. (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics 1987, 20, pp. 53-65.
[10] Gui, J., Zheng, Z., Qin, Z. An Approach to Extract State Information from Multivariate Time Series. Journal of Computers (accepted).
[11] Reiss, A., Stricker, D. (2012) Introducing a new benchmarked dataset for activity monitoring. In 2012 16th International Symposium on Wearable Computers, pp. 108-109.
[12] Rao, K. R., Josephine, B. M. (2018) Exploring the Impact of Optimal Clusters on Cluster Purity. In 2018 3rd International Conference on Communication and Electronics Systems (ICCES), pp. 754-757.