Journal of Computers Vol. 31 No. 6, 2020, pp. 1-11
doi:10.3966/199115992020123106001

An Approach to Extract State Information from Multivariate Time Series

Jun Gui[1,2,3,4,5*], Zeyu Zheng[1,3,4], Zhaobo Qin[5],
Danping Jia[5], Yuan Gao[1,3,4], Zhi Liu[1,2,3,4], Qingfeng Yao[1,2,3,4]

[1] Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China
{guijun, zhengzeyu, liuzhi, yaoqingfeng}@sia.cn

[2] University of Chinese Academy of Sciences, Beijing 100049, China

[3] Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110016, China

[4] Key Laboratory of Network Control System, Chinese Academy of Sciences, Shenyang 110016, China

[5] School of Information Science and Engineering, Shenyang University of Technology, Shenyang 110870, China
{jiadanping, qinzhaobo}@sut.edu.cn

Received 15 July 2019; Revised 24 January 2020; Accepted 2 March 2020

Abstract. System behaviors could be recorded as multivariate time series by smart sensors. It is challenging to interpret such high-dimensional dataset with a one-dimensional temporal state sequence. In this paper, we propose a new approach DBSE (Distribution-based States Extraction) which is based on statistical and clustering analysis. State extraction problem could be resolved through using distribution parameters to describe each subsequence in multivariate time series and applying clustering method to extract states based on distribution similarity in order to obtain a temporal state sequence and information of each state. We validate DBSE and demonstrate how to use DBSE in real-world by extracting state information from a wearable sensor dataset (PAMAP2_Dataset). By comparing DBSE with TICC (Toeplitz Inverse Covariance-based Clustering) and FCM (Fuzzy C-means Clustering), the new approach is more accurate and effective. Moreover, DBSE is also expected to facilitate future behavior analysis.

Keywords: DBSE, multivariate time series, state extraction, system behavior

1 Introduction

In recent years, from Meteorological monitoring [1], Medical equipment [2] to wearable sensors [3], many fields generate enormous amounts of data. These data are multivariate time series generated by multiple sensors. They collect data from measured objects [4]. Many regular or recurring segments in a long period of multivariate time series are defined as features [5] or states [6]. For example, in a driving dataset, states can be described as some behaviors such as turning, speeding up, slowing down, going straight, stopping at a red light [6], etc. States can discover important patterns that are significant for constructing predictive models [7], detecting anomalies [8], and interpreting large high-latitude data sets [9].

The state extraction methods have tended to more diverse. Some supervised methods based on single sample points, i.e., training the improved decision tree are used in the human motion dataset [10]. Some based time-sequence methods, i.e., DTW and DFW to extract the similarity features of the time series be used in extracting patient sleep states [11]. This method uses a standard sleep as a reference, but most

* Corresponding Author

datasets do not provide standards or labels. This is the limitation of supervised learning.

Unsupervised learning methods are more effective and commonly used to deal with such problems. Clustering is one of the representative methods. Some clustering methods based on sampling points, i.e., clustering by fuzzy technology is proposed to analyze the behavior of users during their interaction with the system [12]. Subsequently, they improved algorithm combining fuzzy technology and feature extraction [13]. Toeplitz Inverse Covariance-based Clustering (TICC) is also a state extraction method based on sample points [6]. It maps the subsequences into Markov networks (MRFs), extracting the driving states of the car such as turning, speeding up, slowing down, etc. They are almost based on sampling points. The method based on sampling points can get the status of each point, but it takes a lot of time and calculating power.

The other clustering methods based on subsequences, i.e., Local Mean Decomposition (LMD) and fuzzy C-means (FCM) are used to identify the idling and four coal seams cutting status of the Shearer Cutting [14]. Similarly, wavelet features can also be used for clustering [15]. Shape similarity clustering is used in rough cluster of very large time series datasets. States of the time series are then obtained by splitting and merging [16]. In the same year, they proposed a new hybrid clustering algorithm, also merging subsets based on shape similarity [17]. Besides, Univariate time series can be represented by vectors with statistically significant features. This method can be extended to multivariate time series and proved by human motion sequences [18]. The methods all based subsequences get approximate state, but it improves speed and enhances tolerance of missing value and abnormal value.

Neural networks are also combined with clustering. These methods are almost all based on sampling points. They need strict pre-processing on the data, such as filling missing values and removing abnormal. An adaptive subsequence clustering method based on single layer self-organizing incremental neural network (SOINN) [19] is proposed to use in an open Caltrans performance measurement systems extracting traffic flow states, which improves the Temporal Kohonen Map (TKM) [20] and Recurrent Self-Organizing Map (RSOM) [21]. The CNN network is also applied in the field of multivariate time series, called multivariate convolutional neural network (MVCNN) [22]. Multivariate time series can be converted multi-relational networks. Then use multi-non-negative matrix factorization (MNMF) clustering [23]. Some other methods, i.e., regression is proposed to deal with time series [24]. And improved the parameter problem soon [25].

However, state extraction in Multivariate time series has been still some improved aspects. Methods [6, 12-13, 19-25] are too complicated for processing the time series. They need a higher calculation. The processing of a single sample point is the main reason. In order to observe accurately, subjects often wear sensors with high sampling frequencies. The sampling frequency of acceleration sensors that capture human motion reaches 100Hz [10], vibration acceleration of monitoring bearings reaches 25.6KHz [26]. However, in many cases, people do not need to know what happened in one hundred of a second. It increases the computational burden and gets real-time results hard. Methods [6, 14-15, 18-25] require strict preprocessing for the input data to remove noise points and missing values. This may require manual or additional algorithms, but changing the data is not what we want. Methods [11-13] have a better processing effect for univariate time series, but more improvements are needed to adapt the multivariate time series. Methods [10-13, 19-25] have a better effect on processing low-frequency data, but high-frequency unknown. We found that although adding neural networks can improve the accuracy of state extraction, it requires more complex data preprocessing and higher calculating requirements, especially when processing high-frequency data. Although some algorithms that do not use clustering can also extract states, they are not as effective for multivariate time series processing as univariate time series.

Here, we proposed a state extraction method based on statistical distribution to process multivariate time series data. First, Multivariate time series are divided into subsequences. These subsequences are composed of sampling points. Compared with sample point processing, the subsequence method obtains the approximate state, which solves the problem of slow processing speed in large multivariate time series. It also has high tolerance for noise and missing values of the input data. Next, these sampling points in the subsequence conform to the statistical distribution. Compared with wavelet analysis, MRFs or neural networks, statistical principles can quickly extract the features of subsequences. We use the feature vector to represent these subsequences. In the third step, we cluster the feature vectors by distance-based clustering and assign the clustering results to each subsequence. The clustering algorithm is an unsupervised algorithm. Compared with decision trees, SVM(Support Vector Machine) and other supervised learning, it does not require references and labels. It is more suitable for processing multivariate time series. Finally, these multivariate time series can be represented by one-dimensional state series, the paper structure as the following. In sections II and III, we introduce the principle and the design of DBSE. In section IV, we demonstrate the algorithm to extract states of human actions on the open dataset (PAMAP2_Dataset). In section V, we validate our approach and compare DBSE with TICC and FCM algorithms. Finally, we make a discussion on DBSE in the last section.

The contributions of this paper can be summarized as follows:

(1) A state extraction method based on statistical distribution is proposed, called DBSE. It uses the statistical features (mean μ, variance δ^2, and skewness sk) to represent them and distance-based clustering to extract states.

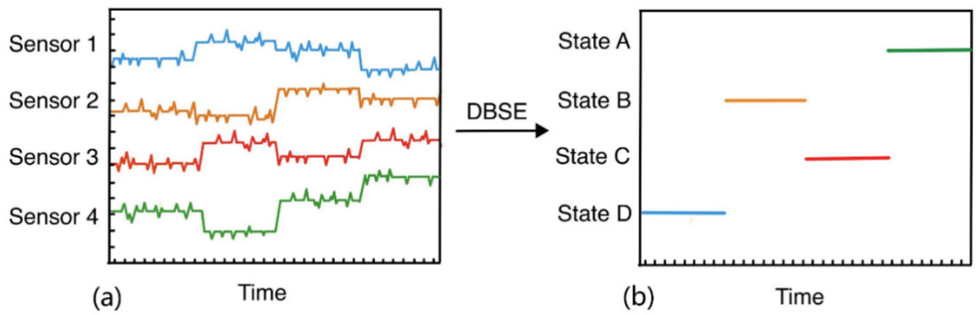
(2) Compared with the sliding window state extraction method (TICC) and the sampling point processing method (FCM), the DBSE performance is much better than these algorithms

(3) The algorithm requires less calculated performance. It has a better tolerance for missing values and noises. The statistical and distance-based clustering design makes the algorithm more suitable for high-frequency sampling data.

2 Principle and Design of DBSE

2.1 Principle of DBSE

Our approach (DBSE) is designed to interpret multivariate time series into a temporal state sequence, which is shown in Fig. 1. DBSE is based on statistical analysis and clustering method. We divide multivariate time series into subsequence and assume each subsequence follows normal distribution. Then, we calculate distribution parameters such as mean μ (Eq.1), variance δ (Eq.2), and skewness sk (Eq.3) for each subsequence. We use them as features to describe subsequence. We know that the mean can describe the central trend of the data. The variance can describe the degree of dispersion of the data. The skewness can describe the symmetry of the sample distribution. We believe that the vector consisting of three parameters can be used to describe the feature of the sample points in a short period of time series. The advantage of associating statistics with features is the increased versatility and interpretability of the algorithm.



(a) to a one-dimensional temporal state sequence

(b) which is more clearly and easy to be analyzed

Fig. 1. (color online) Our approach (DBSE) interprets multivariate time series

$$\mu = \frac{\sum_{i=1}^{n} T_i}{N}. \quad (1)$$

$$\delta = \sqrt{\frac{1}{N}\sum_{i=1}^{n}(T_i - \mu)^2}. \quad (2)$$

$$sk = E[(\frac{T_i - \mu}{\delta})^3] = \frac{\mu^3}{\delta^3} = \frac{E[(T_i - \mu)^3]}{(E[(T_i - \mu)^2]^{3/2}} = \frac{K_3}{K_2^{3/2}} \quad i \in (1, 2, \ldots, n). \tag{3}$$

In Eq.(1)-(3), μ represents the mean of a sequence, δ represents the variance of a sequence, sk represents the skewness of a sequence, and i represents the number of sampling points in a subsequence.

DBSE uses the idea of statistical distribution. The results obtained by this algorithm can be well interpreted. DBSE tolerates missing values when extracting the mean, deviation and skewness from the subsequence. We design the algorithm in the following sections and apply it in practical application. The results confirm the reliability of the algorithm.

2.2 Design of DBSE

In this section, we introduce the DBSE. The five steps are shown in Fig. 2. In first step, the DBSE algorithm need centralizes and normalizes the multivariate time series, as known as the preprocessing process. The algorithm does not need to deal with noise and missing values. In the second step, the DBSE algorithm divides the multivariate time series into a multivariate subsequence. Since the sampling time is constant, a fixed number of sampling points are selected as the subsequence duration. It is preferable to select a larger number of sampling points in order to constitute a sample distribution. In the third step, the DBSE algorithm extracts three parameters that can represent the sequence from each dimension subsequence according to the principle of statistical distribution, The N-dimensional subsequence constitutes 3N parameters. Each subsequence can be represented by a vector of 3N parameters. The algorithm groups these vectors into a state table as Table 1 in subsequent order.

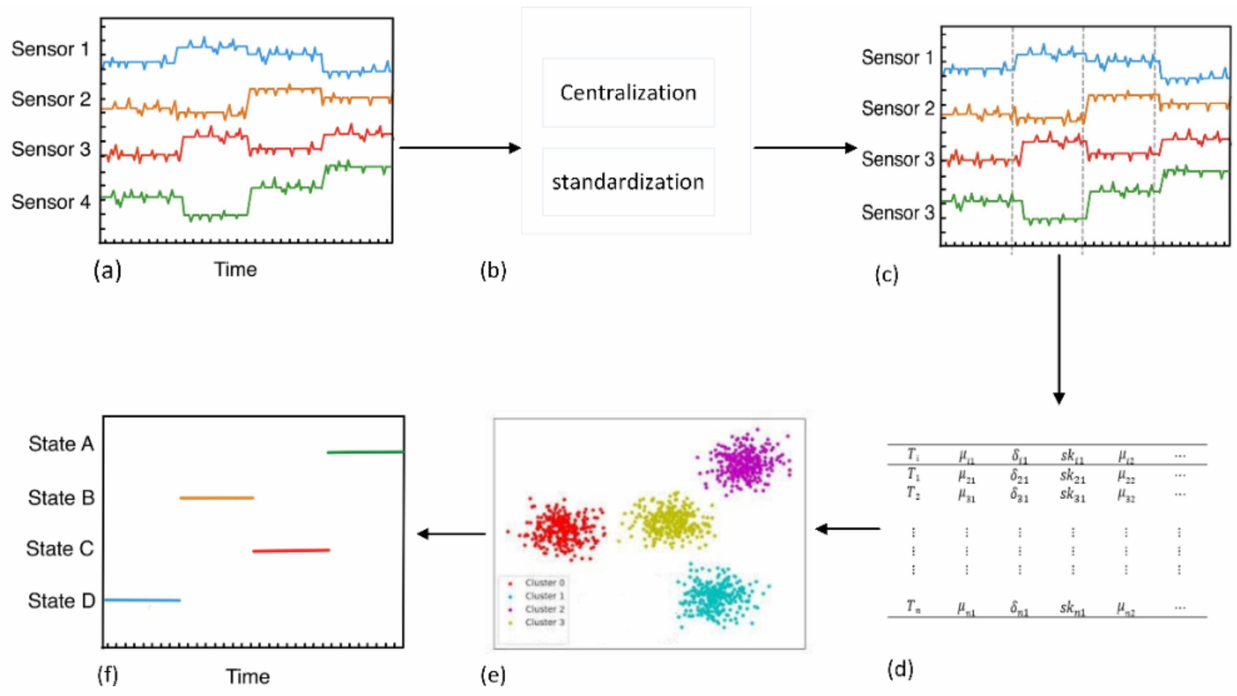


Fig. 2. (color online) Five steps process of DBSE. (a) The multivariate time series need to be processed. (b) Centralization and standardization are performed on multivariate time series. (c) Preprocessed multivariate time series are divided into subsequences. (d) Statistical distribution parameters of all subsequences are calculated and saved in Subsequence status table. (e) States are extracted by clustering based on statistical distribution similarity. (f) The result is a one-dimensional temporal state sequence

In the fourth step, the DBSE clusters these subsequences. We use k-means for clustering. K-means clustering needs to determine the optimal number of clusters. We can use the sum of squared error (SSE) or Silhouette Coefficient to determine. However, the number of clusters may be large. The BIC processing time is too long and the efficiency is low. Therefore we choose SSE [27] (Eq.4) combined with the Silhouette Coefficient to select the best number of clusters [28].

Table 1. The first column in the state table is the label of the subsequence ($T_1 \cdots T_n$). Each column after the first column is the corresponding parameter of the subsequence $T_1 \cdots T_n$

T_i	μ_{i1}	δ_{i1}	sk_{i1}	μ_{i2}	$\cdots$
T_1	μ_{21}	δ_{21}	sk_{21}	μ_{22}	$\cdots$
T_2	μ_{31}	δ_{31}	sk_{31}	μ_{32}	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
T_n	μ_{n1}	δ_{n1}	sk_{n1}	μ_{n2}	$\cdots$

$$SSE = \sum_{i=1}^{k} \sum_{j=1}^{m_i} (x_{ij} - C_i)^2. \tag{4}$$

In Eq.(4) k represents the number of groups, m represents the dimension of each group, x represents the objects in cluster and C represents the center point of the cluster.

Finally, each subsequence is assigned to the cluster by the label. A sequence of states has been generated according to the cluster label, which is a one-dimensional sequence and can describe system behaviors.

3 Application of DBSE on Human Action Analysis

3.1 Description of Dataset

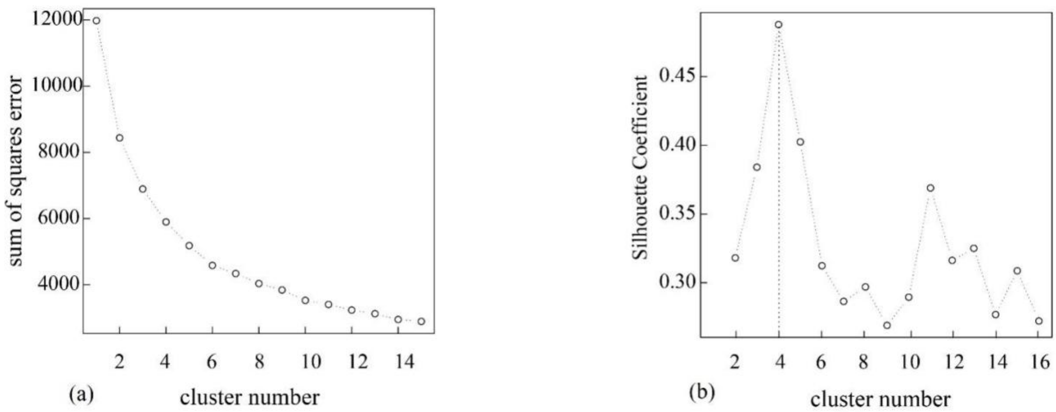
The subject wearing a multi-sensor performs corresponding actions according to the action label. The sensor collects data at a frequency of 100 Hz to obtain a dataset with an action label. It can be used to verify the DBSE algorithm. The dataset mixes a variety of actions. We can construct the dataset needed for the experiment.

This dataset has a total of 12 actions. These actions are recorded by the three-dimensional acceleration sensor of the leg, the specific actions as the following: 1. Lying (lying in a relaxed situation, lying quietly, there will be small movements such as changing the lying position), 2. Sitting (sit in a chair in any position, allowing a small movement, such as changing the sitting position), 3. Standing (including standing still and talking or gesturing); 4. Walking (walking at a speed of 4-6km / h); 5. Running (Running at an appropriate speed in a variety of postures); 6. cycling (cycle at a slow medium speed outside, not strenuous exercise); 7. Nordic walking (strong exercise on non-flat roads); 8. Ascending stairs (the height of five floors); Descending stairs (the height of five floors); 10. vacuum cleaning (cleaning the office with a vacuum cleaner, including placing or moving objects); 11. Ironing (ironing 1-2 shirts or T-shirts); 12. Rope jumping (including foot jump and alternate foot jump) [29-30].

3.2 Process of Processing Dataset

When using the DBSE algorithm to process human action data in chapter 3.1, the algorithm first makes Centralization and standardization in the entire multivariate time series. Centralization is the data in the dataset minus the mean of the dataset. Standardization is the data after centralization divided by the standard deviation of the data set. Since the sampling frequency of the data is 100 Hz. The states of normal human action are usually in units of seconds. The algorithm cuts the pre-processed multivariate time series into a multi-subsequence of 100 sampling points. The mean, standard deviation, and skewness of the subsequence features can be extracted from each dimension of the multivariate subsequence. Since it is a three-dimensional subsequence. Each subsequence can be represented by a set of nine-dimensional vectors. Each set of vectors is a row in the state table. Eventually, after the extraction operation, the multivariate subsequence becomes the state table. Next, use k-means to cluster the vectors from each row in the state table.

The optimal number of clusters would be selected through SSE and Silhouette Coefficient curves in Fig. 3. Both curves have good performance when the number of clusters is 4 and 11. But the data set is collected by the leg acceleration sensors. We tried to infer that it is difficult to divide many actions by simply using a single-site (leg), single-measurement sensor (acceleration sensor). Experimental data shows that some clusters are only assigned a small number of sample points when the number of clusters is too many. This situation is due to interference from abnormal points. Therefore, we choose 4 as the optimal number of clusters and assign the final clustering result to the subsequence.



(a) represents the value of SSE (b) represents the Silhouette Coefficient value

Fig. 3. The x-axis of both figure is the number of clusters. The two figures show clustering indicators that selecting different cluster numbers when using distance as a measure of similarity in nine-dimensional sequence

3.3 Analysis of Results

The status label (cluster label) will be created after the DBSE processed the dataset. Subsequences are divided into 4 clusters, name as A state, B state, C state, and D state. The results of the DBSE are presented in Table 2. The first row is the action item. The first column presents the four states after the data is processed by DBSE. The number in each cell represents the ratio sp (Eq.5).

$$SP_p = \frac{s_p}{\sum_{i=1}^{n} s_i} * 100\%,\ p \in (1, 2, \ldots, n). \quad \textbf{(5)}$$

Table 2. This table describes the results of the DBSE. The first column in the state table is a variety of actions. The numbers in the table show the percentage of action in a certain state (%)

	Lying	Sitting	Standing	Walking	Running	Cycling	Nordic walking	Ascendin g stairs	Descending stairs	Vacuum cleaning	Ironing	Rope jumping
A state	98.7	95.9	99.6	2.86	3.8	91.6	3.80	14.1	18.5	86.5	98.6	13.6
B state	0.43	0.9	0	85.1	95.1	0.8	48.0	19.6	12.5	1.93	0	17.4
C state	0.43	0.9	0	5.53	0	3.2	4.70	59.4	1.97	7.25	1.03	1.51
D state	0.85	2.24	0.39	6.45	1.1	4.4	43.3	6.92	67.1	4.35	0.35	67.1

In Eq. (5) p represents the number of states, S_p represents the number of sampling points of a certain action in the p state. Since the action is made by the person according to the regulations, the number of sampling points of a certain action $\sum_{i=1}^{n} s_i$ is known. The number of sampling points assigned to a certain state S_p is given by the results of the DBSE.

According to Table 2, some actions such as lying, sitting, standing, cycling, vacuum cleaning, ironing are clearly divided into A state. It is easy to find that the slower the leg action, the higher the chance of dividing into the A state. For example, in the A state, the ratio of the action standing is greater than the cycling, and the ratio of the cycling is greater than the vacuum cleaning. We name state A "state of slow action". The DBSE algorithm can accurately divide the state of slow action of the leg.

The action of walking, running, and Nordic walking, running is divided into B state. Through the description of the action in the data set, we know that state B can divide the state of strenuous action. The strenuous actions are very complicated. For example, in the Nordic walking, except for the strenuous action of the leg, the action of jumping is also included. It is found that more intense the action of the leg, more clearly the DBSE algorithm can divide the action into the B state. For example, the ratio of the action running in the B state is as high as 93.1%.

The Ascending stairs action is divided into the C state. According to the description of the action, the Ascending stairs is obvious leg uplifting movement. The action of lifting the leg is different from the action of the body upwards. The upward movement of the body needs to overcome the gravity of the whole body, and the lifting of the leg only needs to overcome the gravity of a part of the leg. The D state easily distinguishes between the two cases. The table shows that both the skipping and the Ascending stairs have a tendency for the legs lift, but only 1% of the skipping ropes fall in the D state. Therefore, the D state accurately describes the state of only the leg lift.

The action of Descending stairs, Rope jumping has a high ratio in the D state. At the same time, we also found that Nordic walking also has some proportion in the D state. According to the description of these actions in the dataset. These actions have a noticeable downward movement of the leg. For example, when going down the stairs, the action of the single leg is an alternating movement from rest to leg fall to rest. Each landing in the Rope jumping action contains an obvious leg drop movement. Similarly, the Nordic walking is a combination of running and jumping. Therefore, the D state accurately divides the actions with the leg falling.

The state sequence obtained by DBSE could be analyzed in a more direct way. We map each subsequence to its corresponding 100 sample points so that we get the sample point data set with the status label. We draw the density curves of the sampling points of the three sensors in each state and observe the statistical distribution characteristics of the four types of states.

The density curve clearly distinguishes the four states. The corresponding density curve of the first sensor in Fig. 4(a). It shows that the standard deviation of the data is very small. The acceleration of the action is small, corresponding to the slow action state. In Fig. 4(b), compared with Fig. 4(a), the profile of the density curve becomes fatter. It means that the standard deviation is relatively large. The fluctuation of the corresponding acceleration sensor is very significant. It is a state of strenuous action. The peaks of the three curves in Fig. 4(d) are lower than in Fig. 4(c), indicating that the acceleration of the leg fall is greater than the acceleration of the leg lift.

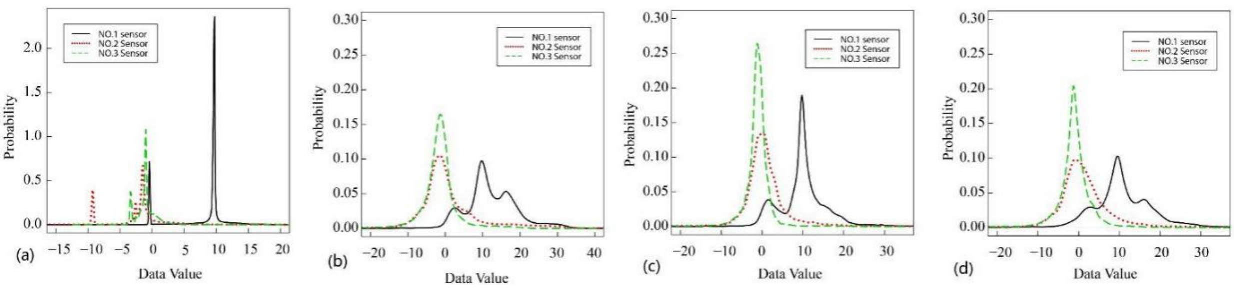


Fig. 4. (color online) The four images depict the density distribution curve for each dimension of the three-dimensional sensor in four states. The x-axis is the value of the data and the y-axis is the probability of the event occurring. The area enclosed by the probability density curve is 1. We judge the four states by the shape of the density curve. The higher the curve, the slower the movement and vice versa

4 DBSE, TICC and FCM

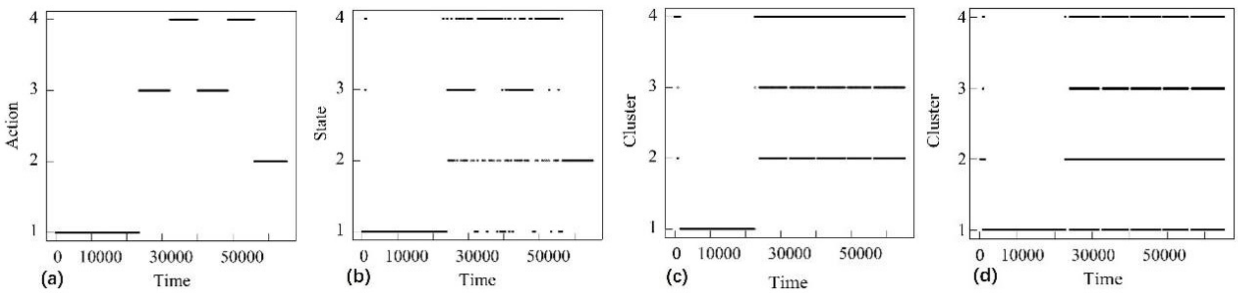
DBSE, TICC and FCM are capable to process multivariate time series. The DBSE algorithm only needs standardization and Centralization of the data during data preprocessing, and the TICC algorithm needs to process the missing values and outliers of the data. Because there are many types of data missing, such as missing completely at random (MCAR), Missing at random (MAR), and Missing not at random (MNAR) [31], it is difficult for TICC to be used directly on data sets with missing values. When selecting features that can describe the time series. The DBSE needs to cut the time series into subsequences for a fixed length of time. TICC needs to set up a sliding window to calculate each sample point and its surrounding points. The principle of DBSE is based on statistical distribution, and TICC is the Markov network [32]. The clustering method of the DBSE algorithm uses the k-means. The TICC is EM iteration [33]. DBSE is quicker and easier. The FCM (Fuzzy C-means) is a new method based on sampling point clustering, which is different from the above two methods. We select the same data set to compare the three algorithms.

4.1 Description of Dataset

Four actions are selected to form the data set: 1. Lying in the slow action state; 2. Running in the strenuous action state; 3. Descending stairs with the legs falling; 4. Ascending stairs with the leg lifting. The data set we constructed contains four distinct leg actions and four different leg states, so we chose 4 for the optimal number of clusters. The length of the DBSE subsequence is set to 100 sample points. The sliding window in TICC also sets 100 sampling points. Lambda regularization parameter is 0.11. Beta parameter controls the smoothness of the output is 600. Maximum number of iterations of the TICC algorithm is 100.

4.2 Analysis of Result

Through a simple comparison between Fig. 5(b), Fig. 5(c), Fig. 5(a) and Fig. 5(d), it is revealed that the sequence of actions of the data set can be better restored from the state sequence, although compared with the original action sequence (Fig. 5(a)). Some locations in the figures have discrete values, but through the analysis of chapter 3.3, our state is closer to the real action situation. In other words, the DBSE generated state sequence more accurately depicts the action than the action sequence (The action sequence is generated by the label, and the action performed by the subject may not be standard). The TICC algorithm (Fig. 5(c)) only successfully distinguishes the lying action. It obviously does not have a good distinguishing ability for a slightly complicated action. The FCM (Fuzzy C-means) is the most widely used and successful among many fuzzy clustering algorithms (Fig. 5(d)). It obtains the membership of each sample point to all cluster centers by optimizing the objective function.



(a) is an action sequence generated from the action label, the x-axis is time, the y-axis is the action label

(b) is the state sequence generated by the DBSE, the x-axis is time, and the y-axis is state

(c) is a clustering result graph generated by the TICC algorithm. The x-axis is the time. The y-axis is the class label

(d) is a clustering result graph generated by the FCM algorithm. The x-axis is the time. The y-axis is the class label

Fig. 5.

We present the above results in the form of a table. The Table 3 records the accuracy of algorithms. The accuracy is based on the Fig. 5(a). The ratio of the number of sampling points falling into a certain period in a certain state to the total sampling points in that state. DBSE works best with time subsequence as input. TICC is a time period obtained from the sampling point as the center, that is, the subsequence is obtained from the point. FCM is point as input, and the effect is the worst of the three methods.

Table 3. Accuracy of the three algorithms (%)

	DBSE	TICC	FCM
Lying	97.86	89.98	92.50
Ascending Stairs	55.38	35.74	4.66
Descending Stairs	72.38	40.02	8.94
Running	93.83	37.13	41.31

5 Discussion

The state information obtained by DBSE can be used to further identify the abnormal state, fault detection, human-computer interaction. For example, we can obtain the proportion of the points falling into the different states to simply identify the action. We can think the action is more likely to a Nordic walking rather than a flat road run if an action presented huge strenuous action states (state B) and huge leg fall state (state C). We can think it is more likely to the Descending stairs instead of the Nordic walking if leg falling state with almost equivalent and small amount of slow action state (state A) and strenuous action state (state B). Although both have huge leg fall (state C).

6 Conclusions

In this paper, the DBSE algorithm is introduced. It can extract states from multivariate time series of actions. The algorithm based on subsequence. It uses statistical features (mean μ, variance δ^2, and skewness *sk*) to represent them and distance-based clustering to extract states. By comparing with other algorithms, we found the algorithm has excellent performance.

Compared with the sliding window state extraction method (TICC) and the sampling point processing method (FCM), the DBSE performance is better than most algorithms on extracting multivariate time series state. It is closer to real action when using the same human action dataset. The algorithm requires less calculated performance. It has a better tolerance for missing values and noise points. The statistical and distance-based clustering design makes the algorithm more suitable for high-frequency sampling data.

The algorithm needs the sampling points in the subsequence as close to the normal distribution as possible. The algorithm has some aspects that need to be further studied. In the experiments, we set the subsequence with the length of 100 sampling points, divide the time series into every second. However, the setting may not the DBSE optimal value. The experiments should select an optimum value from different lengths of the various datasets. The paper uses the k-means clustering method because it is simple and fast. But may not an optimal solution. More clustering algorithms will solve the problem in the future. There are many parameters of statistical distribution. It is limited to the three parameters in the paper. The parameters will be tested to find a better feature vector. In addition, the experiment is based on human behavior data, but according to the algorithm principle (Chapter 2), the algorithm can be used in more fields. Finally, the real world is complex and changeable. The algorithm does not yet have the self-adapt ability. We believed that this method can be applied to a wider range of fields.

Acknowledgments

This work is supported by National Key R&D Program of China (No. 2018YFF0214704). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

References

[1] E. Omidvar, Z.N. Tahroodi, Evaluation and prediction of meteorological drought conditions using time-series and genetic programming models, Journal of Earth System Science 128(3)(2019) 73.

[2] A.T. Walden, L. Zhuang, Constructing brain connectivity group graphs from EEG time series, Journal of Applied Statistics 46(6)(2019) 1107-1128.

[3] K. Li, R. Habre, H. Deng, Applying multivariate segmentation methods to human activity recognition from wearable sensors' data, JMIR mHealth and uHealth 7(2)(2019) e11201.

[4] J. Liang, Y. Zhou, Clustering multivariate time series from large sensor networks, in: Proc. 2017 International Conference on Computer Systems, Electronics and Control (ICCSEC), 2017.

[5] J. Wu, L. Yao, B. Liu, An overview on feature-based classification algorithms for multivariate time series, in: Proc. 2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), 2018.

[6] D. Hallac, S. Vare, S. Boyd, J. Leskovec, Toeplitz inverse covariance-based clustering of multivariate time series data, in: Proc. 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017.

[7] S.E.R. Júnior, G.L. de Oliveira Serra, Evolving methodology in unobservable components space for multivariate time series forecasting, in: Proc. 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2018.

[8] H. Kremer, S. Gunnemann, T. Seidl, Detecting climate change in multivariate time series data by novel clustering and cluster tracing techniques, in: Proc. 2010 IEEE International Conference on Data Mining Workshops, 2010.

[9] D. Ruta, L. Cen, E. Damiani, Fast summarization and anonymization of multivariate big time series, in: Proc. 2015 IEEE International Conference on Big Data (Big Data), 2015.

[10] J. Parkka, M. Ermes, P. Korpipaa, J. Mantyjarvi, J. Peltola, I.Korhonen, Activity classification using realistic data from wearable sensors, IEEE Transactions on Information Technology in Biomedicine 10(1)(2006) 119-128.

[11] X. Long, P. Fonseca, J. Foussier, R. Haakma, R.M. Aarts, Sleep and wake classification with actigraphy and respiratory effort using dynamic warping, IEEE Journal of Biomedical and Health Informatics 18(4)(2013) 1272-1284.

[12] S. Aghabozorgi, M.R. Saybani, T.Y. Wah, Incremental clustering of time-series by fuzzy clustering, Journal of Information Science and Engineering 28(4)(2012) 671-688.

[13] T. Afanasieva, N. Yarushkina, I. Sibirev, Time series clustering using numerical and fuzzy representations, in: Proc. 2017 Joint 17th World Congress of International Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems (IFSA-SCIS), 2017.

[14] L. Si, Z. Wang, C. Tan, X. Liu, Vibration-based signal analysis for shearer cutting status recognition based on local mean decomposition and fuzzy C-means clustering, Applied Sciences 7(2)(2017) 164.

[15] P. D'Urso, E.A. Maharaj, Wavelets-based clustering of multivariate time series, Fuzzy Sets and Systems 193(2012) 33-61.

[16] S. Aghabozorgi, T.Y. Wah, Clustering of large time series datasets, Intelligent Data Analysis 18(5)(2014) 793-817.

[17] S. Aghabozorgi, T. Ying Wah, T. Herawan, H.A. Jalab, M.A. Shaygan, A. Jalali, A hybrid algorithm for clustering of time series data based on affinity search technique, The Scientific World Journal (2014) 56294. DOI: 10.1155/2014/562194.

[18] X. Wang, A. Wirth, L. Wang, Structure-based statistical features and multivariate time series clustering, in: Proc. Seventh IEEE International Conference on Data Mining (ICDM 2007), 2007.

[19] D. Wang, Y. Long, Z. Xiao, Z. Xiang, W. Chen, A temporal self-organizing neural network for adaptive sub-sequence clustering and case studies, in: Proc. 2016 International Conference on Computer, Information and Telecommunication Systems (CITS), 2016.

[20] M. Varsta, J. Heikkonen, J. Lampinen, J.D.R. Millán, Temporal kohonen map and the recurrent self-organizing map: analytical and experimental comparison, Neural Processing Letters 13(3)(2001) 237-251.

[21] T. Voegtlin, Recursive self-organizing maps, Neural Networks 15(8-9)(2002) 979-991.

[22] C.-L. Liu, W.-H. Hsaio, Y.-C. Tu, Time series classification with multivariate convolutional neural network, IEEE Transactions on Industrial Electronics 66(6)(2019) 4788-4797.

[23] L. Zhou, G. Du, D. Tao, H. Chen, J. Cheng, L.Gong, Clustering multivariate time series data via multi-nonnegative matrix factorization in multi-relational networks, IEEE Access 6(2018) 74747-74761.

[24] N.J. Farin, N. Mansoor, S. Momen, I. Mobin, N. Mohammed, Sequence classification: a regression based generalization of two-stage clustering, in: Proc. 2016 International Workshop on Computational Intelligence (IWCI), 2016.

[25] T. Nakashima, G. Schaefer, Y. Kuroda, M.A.R. Ahad, Performance evaluation of a two-stage clustering technique for time-series data, in: Proc. 2016 5th International Conference on Informatics, Electronics and Vision (ICIEV), 2016.

[26] B. Wang, Y. Lei, N. Li, A hybrid prognostics approach for estimating remaining useful life of rolling element bearings. IEEE Transactions on Reliability (2018). DOI: 10.1109/TR.2018.2882682.

[27] J. Peng, Y. Xia, A cutting algorithm for the minimum sum-of-squared error clustering, in: Proc. 2005 SIAM International Conference on Data Mining, 2005.

[28] S. Aranganayagi, K. Thangavel, Clustering categorical data using silhouette coefficient as a relocating measure, in: Proc. International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007), 2007.

[29] A. Reiss, D. Stricker, Introducing a new benchmarked dataset for activity monitoring, in: Proc. 16th IEEE International Symposium on Wearable Computers (ISWC), 2012.

[30] A. Reiss, D. Stricker, Creating and benchmarking a new dataset for physical activity monitoring, in: Proc. 5th Workshop on Affect and Behaviour Related Assistance (ABRA), 2012.

[31] K.G. Moons, R.A. Donders, T. Stijnen, F.E. Harrell Jr., Using the outcome for imputation of missing predictor values was preferred, Journal of Clinical Epidemiology 59(10)(2006) 1092-1101.

[32] H. Rue, L. Held, Gaussian Markov Random Fields: Theory and Applications, CRC Press, New York, 2005.

[33] C.B. Do, S. Batzoglou, What is the expectation maximization algorithm? Nature Biotechnology 26(8)(2008) 897-899.