Pattern Recognition Letters 86 (2017) 18–25

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Sequential data feature selection for human motion recognition via Markov blanket

Hongjun Zhou[a], Mingyu You[a,*], Lei Liu[b], Chao Zhuang[a]

[a] Tongji University, Caoan Road, Shanghai City and 200024, China
[b] Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences Changchun 130033, China

ARTICLE INFO

Article history:
Received 3 August 2015
Available online 13 December 2016

MSC:
41A05
41A10
65D05
65D17

Keywords:
Sequential data
Feature selection
Markov blanket

ABSTRACT

Human motion recognition is a hot topic in the field of human–machine interface research, where human motion is often represented in time sequential sensor data. This paper investigates human motion recognition based on feature-selected sequential Kinect skeleton data. We extract features from the Cartesian coordinates of human body joints for machine learning and recognition. As there are errors associated with the sensor, in addition to other uncertain factors, human motion sequential sensor data usually includes some irrelative and error features. To improve the recognition rate, an effective method is to reduce the amount of irrelative and error features from original sequential data. Feature selection methods for static situations, such as photo images, are widely used. However, very few investigations in the literature discuss this with regards to sequential data models, such as HMM (Hidden Markov Model), CRF (Conditional Random Field), DBN (Dynamic Bayesian Network), and so on. Here, we propose a novel method which combines a Markov blanket with the wrapper method for sequential data feature selection. The proposed algorithm is assessed using four sets of open human motion data and two types of learners (HMM and DBN), and the results show that it yields better recognition accuracy than traditional methods and non-feature selection models.

1. Introduction

Human motion recognition has become a major topic in the computer vision field [23] in recent years. Interest in this topic is motivated by the promise of its use in many applications, including human–robot interaction [34], content-based video indexing [35], video surveillance, and robotics [36,37], among others [1,4].

In the present study we used Kinect to capture motion data because of its powerful capabilities and low price. Kinect is the official name of the X-Box 360 somatosensory peripheral. It is comprised of a set of RGB color cameras, an infrared transmitter, an infrared CMOS camera, and a microphone array for audio input. We extracted 3D coordinates of 20 joints of the human skeleton from combined color and depth images using Kinect for Windows SDK [22].

Experimental results have shown that humans can recognize different activities by detecting only a few points of light attached to the joints of the human body. Therefore, it seems that the position, orientation, and motion of joints contain enough characteristic data for computer recognition of such activities. Furthermore, good performance can be achieved using only the spatial distribution of the joints [3]. More features such as distance, angle, velocity and angular velocity can be derived from sequences of 3D joint positions. In this way we can obtain a large feature set that includes 213 dimensions. We can also represent human motion using these large-dimension sequential data. The greater the number of features, the longer the time required to analyze these features and train the model. Large numbers of features, which include irrelative and error features, can cause a reduction in recognition accuracy [1], leading to a more complex model with lower marketing appeal. Therefore, it is necessary to select features from the sequential feature data set. Feature selection can eliminate irrelevant or redundant characteristics, and thus reduce the number of features to improve the accuracy of the model and shorten its running time. In addition, the selection of truly relevant features can simplify the model and facilitate a better understanding of the data processing. Human motion recognition thus becomes a classification problem, after representation has been applied.

Feature selection methods are roughly divided into two classes: filter and wrapper [14,15]. The wrapper method uses a predictive model to score feature subsets, while the filter method uses a

* Corresponding author.
E-mail addresses: zhouhongjun@dynavisiontek.com, myyou@tongji.edu.cn (M. You).

http://dx.doi.org/10.1016/j.patrec.2016.12.008
0167-8655/

proxy measure, instead of the error rate, to score a feature subset [29]. In our case, a straight-forward approach is to use the wrapper method. We can validate a combination set of features by carry out learning and classification phases. The method will determine an optimal subset for classification; however, the computational cost is expensive. The filter method is not needed to perform the learning and classification phases, thus it does not take as much time for selection; however, it cannot guarantee that the selected feature subset is the best for classification. Both of the above methods have been widely adapted to the static situation, but they cannot support sequential data continuously. Following the accumulation of time-slice data, we do not have an appropriate feature selection method to deal with sequential data, unless we evaluate the entire feature combination set using the wrapper method.

In this paper, we propose a novel method for feature selection based on a Markov blanket combined with the wrapper method to overcome this problem. Different to traditional static situation methods, in our approach, the feature selector has to support the sequential data continuously, and determine the subset feature combination that is best for classification. We select these features via the conditional independence relation using a Markov blanket—determined from all of the sequential feature data following time-slice accumulation and validate whether the subset feature is optimal/minimal using the wrapper method.

Richer structures such as Hidden Markov Models (HMMs) [29], Conditional Random Field (CRF) [24–26] and the Dynamic Bayesian network (DBN) [27] have been explored for use in sequential data representation, such as human motion recognition. This paper applied the HMMs and DBN learner with four sets of open human motion data to validate the proposed method. The contribution of the paper has the following points:

1. We propose a feature selection algorithm based on a Markov blanket combined with the wrapper method, which is named as the SFA-MB (**S**equential data **F**eature selection **A**lgorithm based on **M**arkov **B**lanket) algorithm. It solves the problem of sequential data feature selection. The algorithm not only considers conditional independence testing between the features via the Markov blanket, but also deals with the recognition accuracy of the learner by using the wrapper method.
2. If we use all the time-slice features in the sequential data to build a large Bayesian network, we can simply select the best feature set by a Markov blanket. However, the selected feature from each time slice is different, and determining how to fuse the selected feature based on a natural approach is a challenge. In this paper we solve the problem, and prove that the method is reasonable in both academic and experimental ways.
3. We improve the recognition accuracy based on our proposed algorithm using HMMs and DBNs learners. We validate its effectiveness through four sets of open human motion or gesture data, such as the MSR Action3D data set [22], ChaLearn Gesture data [28], and so on.

2. Related works

Human actions can usually be represented by sequential data patterns. The *k*-nearest neighbors method uses the distance between the representation of an observed motion and a training set. The most common label among the *k*-closest training sequences is chosen as the classification criterion [4]. Support vector machines (SVMs) learn a hyperplane in the feature space that is described by a weighted combination of support vectors. Schuldt et al. [5] used a local SVM approach. Lv & Nevatia [11] combined SVMs with local representations of fixed lengths. The K-Nearest Neighbors (K-NN) and SVM approaches are static models representing human motion, so we use HMMs to represent human motion in a more natural way.

Relevance vector machines (RVMs) are probabilistic variants of SVMs. Oikonomopoulos et al. [7] used RVMs for motion recognition by measuring the dynamic time warping (DTW) distance between two sequences. This approach takes into account the distance between corresponding frames. Veeraraghavan et al. [13] used DTW for sequences of normalized shape features in a nonparametric model. Yao & Zhu [9] introduced dynamic space–time warping, in which, in addition to the temporal dimension, sequences are also aligned according to image position and scale.

HMMs use hidden states that correspond to different phases in the performance of a motion. The use of graphical models for human motion recognition is not a new idea. Yamato et al. [29] clustered grid-based silhouette mesh features to form a compact codebook of observations and then trained HMMs for the recognition of different tennis strokes. Lu & Little [15] used a hybrid HMM, in which one process denotes the closet shape–motion template and the other models the position, velocity, and scale of the person in the image. Lv & Nevatia [11] used 3D joint locations, but constructed a large number of action HMMs, each of which used a subset of all joints. This resulted in a large number of weak classifiers. They used AdaBoost to form the final classifier [6,8,10].

Conditional random fields (CRFs) use a plurality of overlapping features. Sminchisescu et al. [12] used a linear chain CRF in which the state dependency was of the first order. In the present study, we extracted features from skeleton data for representation using HMMs.

In this paper, we use the richer graphical structure of HMM and DBN to represent human motion sequential data, and select effective features from the original feature set to improve the recognition accuracy.

The feature set of the above graphical models for learning is manually selected by an expert. Few studies have focused on feature selection in HMMs. The wrapper method has very high computation costs and is not feasible for large data sets. Here we focus on a modification of the filter method, a Markov blanket [16,17]. The Markov blanket of a target variable contains a minimal set of variables for which all other variables are conditionally independent of the target variable. However, Bayesian networks [19,20] are not directly related to time, so DBNs are then used to model temporal processes. The traditional Markov blanket method [16,17] cannot deal with sequential data models such as HMMs and DBNs. Most Markov blanket methods represent features and target variables as a static Bayesian network, for which it is difficult to find the Markov blanket of a target variable for HMM and DBN learning. Pei et al. [18] proposed a segmental booting method for HMM feature selection.

The basic idea of our approach is to translate the sequential feature data into a static Bayesian network, and select the optimal feature set using a Markov blanket based on the static Bayesian network. However, the problem is then determining the target variable of the Markov blanket. We set human motion labels as the target variable, and calculate the conditional independence between the target variable and an irrelative feature candidate based on the Markov blanket candidate. Different to the traditional method, we do not gather all of the time slice data to determine the Markov blanket. We find that the sequential data can be classified just by the part of the data, and it is unnecessary to classify the sequential data using all of the time-slice data. As such, we accumulated the sequential data following time-slice addition, and built a Bayesian network and selected the Markov blanket once a new time-slice dataset was added. The selected feature was evaluated using the wrapper method, and the feature set of the entire time-slice dataset that had the best score was used as the final, optimal feature set.

3. Feature representation

A distinguishing feature vector that adequately describes a motion is critical for motion recognition systems. Here we extracted 3D coordinates for 20 human joints from color and depth images from Kinect for Windows SDK. These 20 joints constituted a diagram of the human skeleton. The Cartesian coordinates, the Euclidean distance between the midpoint of hip and each joint, velocity, angle, and angular velocity were chosen as motion features.

Assuming that the 3D coordinates of the hip midpoint at time t are $p_{0t}(x_{0t}, y_{0t}, z_{0t})$ and the 3D coordinates of joint i at time t are $p_{it}(x_{it}, y_{it}, z_{it})$, then the Euclidean distance between the midpoint of the hip and joint i is $d_{oi} = \sqrt{(x_{it} - x_{0t})^2 + (y_{it} - y_{0t})^2 + (z_{it} - z_{0t})^2}$ for $i = 0, 1, 2, \ldots, 19$ joints apart from the midpoint of the hip.

The torso angle was also chosen as a motion feature after we found that distance features alone do not yield good results. The right and left elbow angles were also added to the motion features. We assumed that the 3D coordinates at time t were $p_{2t}(x_{2t}, y_{2t}, z_{2t})$ for the left shoulder, $p_{9t}(x_{9t}, y_{9t}, z_{9t})$ for the left elbow, and $p_{11t}(x_{11t}, y_{11t}, z_{11t})$ for the left wrist. The left arm vector is $v_{29} = ((x_{2t} - x_{9t})(y_{2t} - y_{9t})(z_{2t} - z_{9t}))$ and $v_{119} = ((x_{11t} - x_{9t})(y_{11t} - y_{9t})(z_{11t} - z_{9t}))$.

Assuming that the angle of the left elbow is Θ, then $\cos\Theta = (v_{29} \times v_{119}) \setminus (|v_{29}| \times |v_{119}|)$. The right elbow angle is the same as the left. The velocity and angular velocity considers the difference between two close frames. In this way we can obtain a 213-dimensional feature vector, which is too large to teach the HMM for motion recognition. To improve the recognition accuracy, we need to reduce the number of dimensions of the feature space.

4. Feature selection

A feature selection [29] algorithm can be considered as a search technique for sorting the feature subsets, along with an evaluation measure which scores the different feature subsets. The simplest algorithm is an exhaustive search, which tests each possible feature combination to find the one that minimizes the error rate, but it is computationally intractable.

Most of the research regarding these methods have focused on static data patterns, for example, a static image. However, numerous data patterns need sequential data to be properly represented, for example, sound data, human action and so on. Of course, we can use the wrapper method and a searching policy to find a feature subset that holds a local minimum score. However, as discussed above, this comes at a large computational cost. Moreover, current filter methods cannot support sequential data. Different to static data patterns, following the accumulation of a time slice, a sentential dataset will grow continuously. As shown in Fig. 1, a set of sequential data includes T sets of data vectors $D_t (t \in T)$, and D_t include an N-dimensional vector $\{S(t, 0), S(t, 1), \ldots, S(t, n)\}$. In general, the data in every time slice has the same dimension and attributes. For example, in our research, a data set of each time slice has 213-dimensional features, but each human action has a different number within the time slice. In other words, the same people who perform the same action always generate different sequential data. The dimension of the data should be same, 213, but the number of time slices (we call it time length) is different. Thus, very few literature reports regarding the filter method for feature selection can support this kind of data pattern with an appropriate model.

4.1. Markov blanket for feature selection

A Markov blanket is a filter method for feature selection that does not need learning and classification phases, so it is less expensive in time and computation costs. The Markov blanket of a target feature contains a minimum set of features for which all other features are conditionally independent of the target variable. Our aim was to experimentally determine how Markov blanket predictors could improve the classification accuracy of sequential data. A Bayesian network (BN) [16,20] is a directed acyclic graph in which nodes represent variables of a subject of interest, and arcs between the nodes describe causal relationships among the variables. A BN is a compact representation of a joint probability distribution of domain variables. A Markov blanket is a key concept of conditional independence in graphic models of BNs. The Markov blanket of a target feature *Tar* is the set consisting of the parents of *T*, the children of *Tar*, and features common to the children of *Tar*. Given its Markov blanket, a feature is conditionally independent of other features [20]. As shown for the BN in Fig. 2, variables denoted in blue color (nodes a, b, c, d, and e) constitute the Markov blanket for target variable *Tar*.

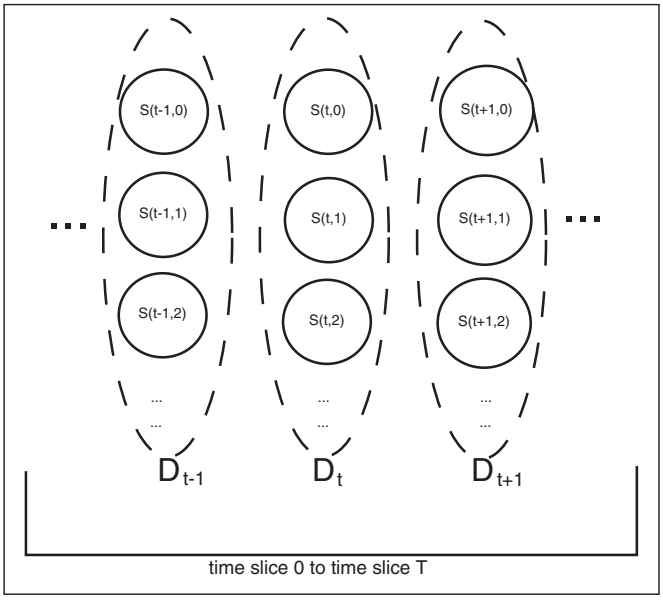

Fig. 1. A example of sequential data.

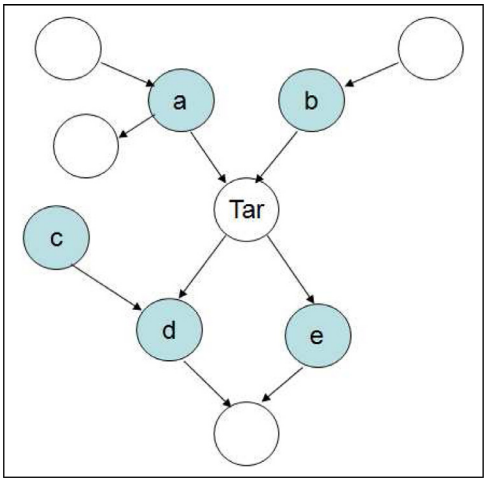

Fig. 2. Example of a BN illustrating the Markov blanket for target variable *Tar*.

4.2. 2TBN-MB

However, a BN always represents a static situation, for example a photo image. In other words, if we do not consider the temporal attribute of the data, it is easy to represent sequential data in a

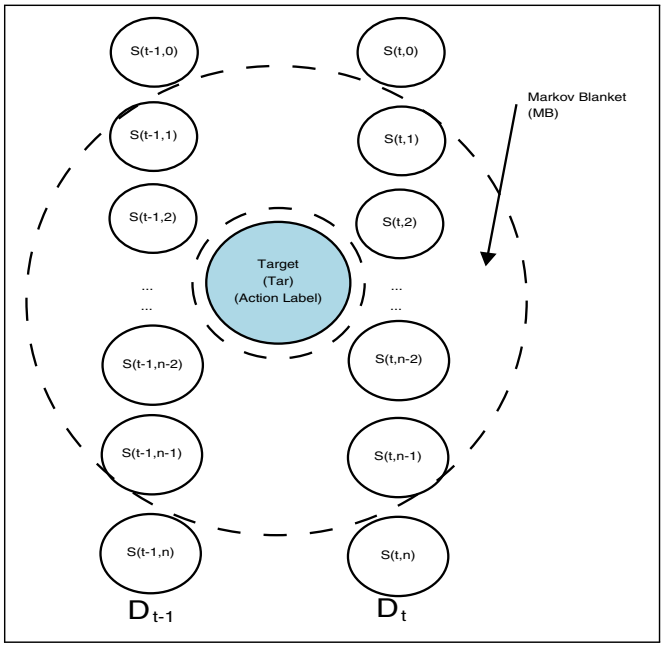

Fig. 3. Markov blanket searching via the Two-Timeslice structure of the sequential data.

BN. In this research, our application is human motion recognition. However, the sensor data of human motion is spatio-temporal, and we cannot represent the spatio-temporal sensor data using a static BN. A DBN is a more natural way to represent human motion than a static BN in this case. Since in a DBN, we always assume that at any point in time, t, the value of a variable can be calculated from the internal regressors and the immediate prior value (time $t-1$); as such we often call a DBN as a Two-Timeslice BN (2TBN) [20]. A straight-forward idea is using the 2*TBN* to teach the Markov blanket for feature selection of the spatio-temporal data.

As show as Fig. 3 and Algorithm 1, we call the set of data, $\{D_{t-1}, D_t\}$, to be Two-Timeslice data, and the human motion label is set to be the target (*Tar*) of the Markov blanket. Based on the HITON algorithm [30], the Markov blanket should be determined from the spatio-temporal human motion data. In Fig. 3, the region between two dish cycles indicates a Markov blanket. The feature set of the Markov blanket is the feature selection result. Using the selected feature, we perform the classification experiments. Unfortunately, when using this method we cannot obtain a good result (see Section 5). Since human action data is spatio-temporal data, only a few cases can be represented just by one set of Two-Timeslice data. Thus, selected features from the Two-Timeslice data based Markov blanket cannot capture complete information regarding a human action. We discuss this in detail in the following section (Section 5).

Algorithm 1 Sequential data feature selection lagorithm via a Markov lbanket based on Two-Timeslice adta (2TBN-MB).

Input:

- Target (Tar)→ Human motion label;
- The number of time slices T;
- Dataset in each time slice D_1, D_2, ...,D_t,...,D_T;
- Whole_Data = $D_1 \bigcup D_2$ ...$\bigcup D_T$.

MB_TSFS Algorithm:

1. **Initilization:**Segment D into a Two-Timeslice data structure $\{D_{t-1}, D_t\}$;
2. Set Tar to be the target of Markov blanket searching;
3. Markov blanket *MB* searching using the HITON algorithm.

Output: *MB* to be the selected feature set.

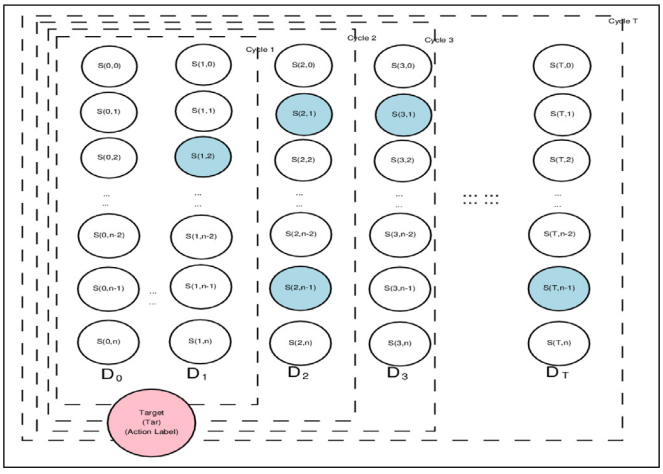

Fig. 4. An illustration of the SFA-MB procedure.

4.3. SFA-MB

4.3.1. Algorithm description

To overcome this problem, we propose the SFA-MB method. As shown in Fig. 4, we set the human motion label to be the target (Tar), the number of time slices as T, and one dataset of time slice data to be D_t. Then the entire sequential data are indicated by the variable $D = \{D_0, D_1, \ldots, D_t, \ldots D_T\}$. At time t, we use the dataset $D_{0,\ldots,t}$ and the target *Tar* to search the Markov blanket MB_t based on the HITON algorithm [30]. Using the selected feature set MB_t, we perform a wrapper procedure. Different learners should use a different wrapper method, for example, if we employ the HMM, the wrapper procedure involves HMM parameter learning and classification phases based on selected features via the Markov blanket MB_t. Finally, we leave the selected feature set, which holds the maximum classification rate. The detailed process is shown in Algorithm 2.

With regard to the selected features of the SFA-MB, we find that an alleged selected feature may be different in each time slice. For example, in Fig. 4, the blue nodes indicate the selected feature of the SFA-MB. However, although features $S_{(2,\,1)}$, $S_{(3,\,1)}$ are selected in time slice 2 and 3, the same features in the other time slices have not been chosen. However, in the sequential data $S_{(.,\,1)}$ are gained by the same data source in each time slice. For example, the position of the right hand. Therefore, we cannot only pick out certain features from certain time slices to be the selected feature. In the SFA-MB, if a certain feature in a certain time slice is selected, we have to leave all of the features in every time slice. In the above example, we selected the feature $S_{(2,\,1)}$, $S_{(3,\,1)}$, then the features $S_{(.,\,1)}$ are therefore seen as the selected result. Thus, in the selected features of the SFA-MB, we left some redundant features, such as $S_{(.,1)} - S_{(2,1)} - S_{(3,1)}$. An outstanding question is then, will these redundant features affect the recognition rate? We will discuss this in the next section.

The wrapper function in Algorithm 2 is same as Reference [30]. In Ref. [30], the author has not given a concrete algorithm about the wrapper function. For the different learner, it may have different algorithm. For example, in the case of the HMMs learner, the wrapper function should have two steps: 1)Training. Based on the selected feature set, EM algorithm will be performed for the HMMs parameters estimation. 2)Evaluation. Based on the learned param-

Algorithm 2 Sequential data feature selection algorithm via a Markov blanket (SFA-MB).

Input:

- Target (Tar)
- The number of time slices T;
- Data set in each time slice $D_1, D_2, \ldots, D_t, \ldots, D_T$;
- The entire _Data = $D_1 \bigcup D_2 \ldots \bigcup D_T$.

SFA_MB Algorithm:

1. **Assign:**Idea_feature=∅; Idea_t=0;
   Temp_Data=∅;Max_accuracy=0;
   Selected_Features=∅;
2. **FOR** t = 1 to T
3. Temp_Data =Temp_Data $\bigcup D_t$;
4. MB(Tar) = FIND-MB(Temp_Data,Tar);
5. accuracy_rate = Wrapper(Whole_Data,MB(Tar));
6. **IF** accuracy_rate > Max_accuracy **THEN**
7. Max_accuracy = accuracy_rate ;
8. Idea_t = t ;Idea_feature =MB(Tar);
9. **END IF**
10. **END FOR**
11. **FOR** k = 1 to SizeOf(Idea_feature)
12. ***fea*** = Idea_feature[k]
13. Find out all of the ***fea*** features at every time slices, we denote them by ***fea_slice*** (see Section 4.3.1).
14. Push ***fea_slice*** into Selected_Features.
15. **END FOR**
16. **RETURN** Idea_t ; Selected_Features.

FIND-MB(Data D**,Target** Tar**)**
"Return the Markov blanket of Tar through the HITON algorithm 30 based on the dataset D"

1. **V** = All of the features (nodes) in the dataset D
2. node_set = The node set directly linked to the target Tar (parents and children of node Tar) returned by HITON_PC 30
3. candidate_set = parents and children of the node belong to the node_set
4. Temp_MB = node_set $\bigcup$ candidate_set
5. $\forall$ N $\in$ Temp_MB and $\forall$ P $\in$ node_set
   IF $\nexists$ R $\subseteq$ {P} $\bigcup$ **V** - {Tar,N} $\Rightarrow \perp$(Tar,N | R)
   THEN N is still retained in Temp_MB
   ELSE remove N from Temp_MB
6. **RETURN** Temp_MB

Wrapper(Data D**,Feature Set** MB**)**
"Return the corrected rate of classification through the feature set MB based on the data set D"
Output: Idea_t ; Idea_feature.

eters and selected features, we can assess the recognition rate for the selected feature set.

4.3.2. Theoretical proof

In [16], the author provides the definitions of ***conditionally independent*** and ***Markov blanket***.

Definition 1. Two variables are said to be conditionally independent given a set of variables X if, for any assignment of values a, b, and x to the variables A, B, and X, respectively, $P(A = a|X = x; B = b) = P(A = a|X = x)$. That is, B gives us no information about A beyond what is already in X. X is can also be considered as the Markov blanket of A.

In our case, $O_{mb}, \overline{O}, O, Tar$ are defined as a Markov blanket, the reduced feature set, all of the features, and the target variable (here it is the human motion label), respectively. Based on Definition 1, we can write $P(Tar|O_{mb}; \overline{O}) = P(Tar|O_{mb})$. In Bayesian network, the nodes can be divided into three partitions, one is target node, another is Markov blanket which is denoted as O_{mb}, and another is denoted as $\overline{O}$. And the whole feature set is denoted by O, so we also can define the $O_{mb} = O - \overline{O}$. We assume $O_{redu} \subset \overline{O}$. Because $\overline{O}$ and Tar are conditionally independent given the Markov blanket O_{mb}, the Eq. (1) is established.

$$P(Tar|O_{mb}) = P(Tar|O_{mb}; O_{redu}) \tag{1}$$

Eq. (1) explains that, although the Markov blanket includes some redundant features, but it does not affect the classification result $P(Tar|O_{mb})$. This is a very interesting yet contradictory corollary. If feature selection cannot improve the classification result, the research topic of feature selection should be insignificant and absurd. In fact, feature selection does not only remove redundant variables, but also removes error and/or noise features. Different to redundant features, error and noise features will affect the classification result. A large number of experimental results show that the selected features can indeed improve classification results. In the following section we will show some similar conclusions.

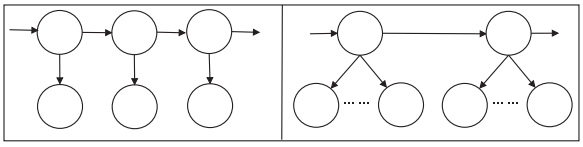

Fig. 5. The models used in our experiments: HMMs (left) and DBN (right).

5. Experiments

5.1. Computing platform and experiment data

In our experiments, the computer has one Intel Core i5 CPU and 4GB memory. The HMM inference/learning and Markov blanket learning are implemented by Matlab.

We applied our method to two types of learners, HMMs and DBN, using four sets of open data: the MSR Action3D data set [21,31], the MSRDailyActivity3D [21,32], the Online RGBD Action Dataset [33], and the Chalearn dataset [28]. The structure of the HMMs and DBN is shown in Fig. 5. Since all of the open dataset include original data and label data file, the test dataset is annotated manually.

We split the datasets (*MSR Action3D, MSRDailyActivity3D, Online RGBD Action*) into two parts: training data and test data. The size of the test data is about two times that of the training data. Since ChaLearn LAP 2014 Dataset has been divided into "training data" and "test data" by dataset provider, we have not split it afresh.

5.1.1. MSR Action3D data set

The MSR Action3D dataset consists of skeleton data obtained from a depth sensor similar to the Microsoft Kinect at a frequency of 15 Hz. The set of actions includes *high arm wave, a horizontal arm wave, hammer, hand catch, forward punch, high throw, draw X, draw tick, draw circle, hand clap, two-hand wave, side boxing, bend, forward kick, side kick, jogging, tennis swing, tennis serve, golf swing, pick up and throw*. Each action has 20–30 action samples, and 20–30 time slices.

5.1.2. MSRDailyActivity3D

MSRDailyActivity3D consists of 16 actions by 10 people. The actions include *drink, eat, read book, call cellphone, write on a paper, use laptop, use vacuum cleaner, cheer up, sit still, toss paper, play game, lie down on sofa, walk, play guitar, stand up, sit down*. Each action has about 20 samples.

Table 1
Recognition results without feature selection.

Data set	Average accuracy HMMs	Average accuracy DBN
MSR Action3D	0.8393	0.2076
MSRDailyActivity3D	0.7728	0.1243
Online RGBD Action	0.9027	0.2351
Chalearn LAP 2014	0.9464	0.3271

Table 2
Experimental results of 2TBN-MB algorithm.

Data set	Average accuracy (HMMs)	Average accuracy (DBN)
MSR Action3D	0.8337	0.4057
MSRDailyActivity3D	0.8115	0.4262
Online RGBD Action	0.8148	0.3721
Chalearn LAP 2014	0.6549	0.3357

Table 3
Experimental results of SFA-MB algorithm.

Data set	Average accuracy (HMMs)	Average accuracy (DBN)
MSR Action3D	0.9180	0.4235
MSRDailyActivity3D	0.9417	0.4583
Online RGBD Action	0.9795	0.3921
Chalearn LAP 2014	0.9523	0.4163

5.1.3. Online RGBD Action Dataset

The Online RGBD Action Dataset includes 7 actions, *drinking, eating, using laptop, reading cellphone, making phone call, reading book, using remote*. Each action includes 46 samples and 199–256 time slices.

5.1.4. Chalearn LAP 2014 dataset

The competition organizer provided three datasets: "training data", "validation data" and "test data". Each data set consists of hundreds of files, and each file contains approximately one-minute gesture data, which include skeleton data and RGBD video data. In the gesture data, there are 20 type of gestures and each sample data includes 16–26 time slices. In our experiments, we only used the "training data" for learning, and testing the gesture recognition. The actions includes *Go away, Come here, Perfect! Crafty, No fun! What do you want? They get together, Are you crazy? What have you done? There is no interest to me, OK, What would you do?, Enough already! You want to take, No good any more, I'm hungry, That was a long time ago, It's very delicious! They have agreed, I'm sick*.

5.2. Recognition by selected features

In this section, two methods, 2TBN-MB and SFA-MB, were employed to perform the training and recognition using the above four sets of open data. Each experiment was carried out according to the 10-fold cross-validation scheme.

5.2.1. 2TBN-MB and SFA-MB

Table 1 shows the recognition results of the four open datasets without the feature-selection process. Tables 2 and 3 show the recognition results of the four open datasets using HMMs and DBN based on the 2TBN-MB and SFA-MB algorithms, respectively. Comparing these to the results in Table 1, we find that the SFA-MB feature selection method improves the recognition rate of HMMs and DBN significantly. Since the structure of the DBN is designed simply, compared to HMMs, the results of the DBN are worse. In this research, we do not focus on the DBN recognition. We also find that the results of SFA-MB are significantly better than 2TBN-MB, which is attributed to the Two-Timeslice structure, which does not have capabilities for capturing information of the spatio-temporal human action data for classification (see Section 4.2).

5.2.2. Comparing to traditional method for feature selection

In contrast, we also employed several traditional wrapper feature selection methods to the MSR Action3D data set based on the HMMs learner.

PCA. We applied PCA (Principal Components Analysis) to reduce the feature dimension of each time slice. This reduced the feature dimension by almost 90%. Using the compressed features, the computational time for HMM recognition was 50% shorter, but the recognition accuracy was largely unchanged. The HMM recognition accuracy using compressed features was 82.23%, which is very similar to the original accuracy of 81.85%. Thus, PCA significantly reduced the computation cost without affecting the classification performance. Based on MSR Action3D data set, we have selected 21 features by PCA. The principal components from 1th to 21th account for or "variance explain" 95% of the overall variability.

GA and SFS. Fig. 4 shows the HMMs recognition accuracy of MSR when used on the Action3D Dataset using the GA and SFS methods. For GA, we set the crossover probability to 0.8, the mutation probability to 0.03 and the population size to 50 for 100 iterations. The final convergence population yielded a recognition accuracy of 47.58%. SFS obtained 85.02% recognition accuracy, which is better than that of the GA. SFS is not terminated until all of the search and calculation is over, and the feature subset which has the best score will be the best one. All of data (all time slice data) is used in SFS and GA feature selection procedure.

Contrasting the traditional methods, PCA, GA (Genetic Algorithm) and SFS (Sequential Forward Selection), when considering recognition accuracy or computational cost, SFA-MB yields better results than all the other methods. In order to further validate the performance of our method, we added some experiments to compare the PCA, GA and SFS using the open dataset: MSRDailyActivity3D, Online RGBD Action, and Chalearn LAP 2014. Table 4 shows the experiment results of the comparison.

In Algorithms 1 and 2, we have used a threshold for Markov blanket decision in HITON [30] algorithm. Conditional independence core between the variable data is calculated by HITON, and the core will be used to select which variable related to the target variable closely. So, the value of threshold will be effect the result of Markov blanket. In our system, the threshold is selected manually, the optimization of the threshold has not be discussed. It will be a challenge of our future work.

Definition 1 and the discussion of Section 4.3.2 illustrate that the redundant variables can not affect the classification results, but the error and the noise features will greatly effect the classification accuracy theoretically. As show as Table 4, Algorithm SFA-MB substantially improved the classification accuracy in open dataset *MSR Action3D, MSRDailyActivity3D, Online RGBD Action*, but a slight improvement has been occurred in dataset *Chalearn LAP 2014*. By investigating the source of dataset, we found the error and the noise data of *Chalearn LAP 2014* is significantly less than the others. In the dataset, we define the obvious error value to be the error and the noise data, for example, all zero pattern. The error and the noise data investigation is carried out only by human manually, we have not built a system to find the error and the noise data automatically. In the future, automatic error/noise data discovery should be a challenge work. And quantitative studying of the relationship between the classification results and the error/noise/redundant data should also be a meaningful subject in future.

6. Conclusions

In this paper, we proposed a Markov blanket based feature selection method for human motion recognition. The method

Table 4
HMMs recognition accuracy when applied to four datasets based on various feature selection methods.

Data set	Feature selection methods	Recognition rate	Feature selection time (seconds)	Feature number
MSR Action3D	No feature selection	0.8393	0	213
	PCA	0.8223	4035	21
	GA	0.4758	19653	38
	SFS	0.8502	14582	5
	2TBN-MB	0.8337	6543	31
	SFA-MB	*0.9180*	7875	27
MSRDailyActivity3D	No feature selection	0.7728	0	213
	PCA	0.7983	32035	23
	GA	0.3281	119653	67
	SFS	0.8129	104582	42
	2TBN-MB	0.8115	63543	28
	SFA-MB	*0.9417*	91331	43
Online RGBD Action	No feature selection	0.9027	0	213
	PCA	0.8567	8053	37
	GA	0.6129	29433	102
	SFS	0.8502	25842	74
	2TBN-MB	0.8148	13932	42
	SFA-MB	*0.9795*	23543	19
Chalearn LAP 2014	No feature selection	0.9464	0	213
	PCA	0.8982	90155	68
	GA	0.7712	396537	45
	SFS	0.8502	242853	97
	2TBN-MB	0.6549	134532	45
	SFA-MB	*0.9523*	200608	34

effectively solved the feature selection problem of sequential data. Comparing to the traditional wrapper method, SFA-MB not only improved recognition accuracy, but also reduced computational cost. We proved the rationality and efficiency of the SFA-MB via theoretical and experimental evidence. Four sets of open data were validated using the SFA-MB, and the experiment results showed that it effectively improved recognition performance.

Acknowledgments

This work is supported by National Natural Science Foundation of China (No. 51475334).

References

[1] P. Turaga, R. Chellappa, V.S. Subrahmanian, O. Udrea, Machine recognition of human activities: a survey, Circuits Syst. Video Technol., IEEE Trans. 18 (11) (2008) 1473–1488.
[2] J. Yamato, J. Ohya, K. Ishii, Recognizing human action in time-sequential images using hidden markov model, in: Proceedings of the 1992 IEEE International Conference on Computer Vision and Pattern Recognition, 1992, pp. 379–385.
[3] A.A. Chaaraoui, J.R. Padilla-Lopez, P. Climent-Perez, F. Florez-Revuelta, Evolutionary joint selection to improve human action recognition with RGB-d devices, Expert Syst. Appl. 41 (3) (2014) 786–794.
[4] R. Poppe, A survey on vision-based human action recognition, Image Vis. Comput. 28 (6) (2010) 976–990.
[5] C. Schuldt, I. Laptev, B. Caputo, Recognizing human actions: A local SVM approach, in: Proceedings of the 17th International Conference on Pattern Recognition, 2004, pp. 32–36.
[6] I. Laptev, B. Caputo, C. Schuldt, T. Lindeberg, Local velocity-adapted motion events for spatio-temporal recognition, Comput. Vis. Image Understanding 108 (3) (2007) 207–229.
[7] A. Oikonomopoulos, I. Patras, M. Pantic, Spatiotemporal salient points for visual recognition of human actions, Syst. Man Cybern. Part B IEEE Trans. 36 (3) (2005) 710–719.
[8] A. Veeraraghavan, A.K. Roy-Chowdhury, R. Chellappa, Matching shape sequences in video with applications in human movement analysis, Pattern Anal. Mach. Intell. IEEE Trans. 27 (12) (2005) 1896–1909.
[9] B. Yao, S.-C. Zhu, Learning deformable action templates from cluttered videos, in: Proceedings of the IEEE 12th International Conference on Computer Vision, 2009, pp. 1507–1514.
[10] W.-L. Lu, J.J. Little, Simultaneous tracking and action recognition using the PCA-HOG descriptor, in: Proceedings of the 3rd Canadian Conference on Computer and Robot Vision, 2006. 6–6
[11] F. Lv, R. Nevatia, Recognition and segmentation of 3-d human action using HMM and multi-class adaboost, Lect. Notes Comput. Sci. 3954 (2006) 359–372.
[12] C. Sminchisescu, A. Kanaujia, D. Metaxas, Conditional models for contextual human motion recognition, Comput. Vis. Image Understanding 104 (2) (2006) 210–220.
[13] M. Muller, T. Roder, Motion templates for automatic classification and retrieval of motion capture data, in: Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2006, pp. 137–146.
[14] H. Almuallim, T.G. Dietterich, Learning with many irrelevant features, in: Proceedings of the 9th National Conference on Artificial Intelligence, AAAI Press, 1991, pp. 547–552.
[15] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artif. Intell. 97 (1997) 273–324.
[16] D. Koller, M. Sahami, Toward optimal feature selection, in: Proceedings of the 13th International Conference on Machine Learning, Morgan Kaufmann, 1996, pp. 284–292.
[17] S. Yaramakala, D. Margaritis, Speculative markov blanket discovery for optimal feature selection, in: Proceedings of the 5th IEEE International Conference on Data Mining, IEEE Computer Society, 2005, pp. 809–812. Washington, DC
[18] Y. Pei, I. Essa, T. Starner, J.M. Rehg, Discriminative feature selection for hidden markov models using segmental boosting, in: Proceedings of the 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, 2008.
[19] F.V. Jensen, An Introduction to Bayesian Networks, Springer, New York, 1996.
[20] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann Publishers, San Francisco, 1988.
[21] MSR Action Recognition Datasets and Codes, http://research.microsoft.com/en-us/um/people/zliu/ActionRecoRsrc/default.htm.
[22] http://msdn.microsoft.com/zh-tw/hh367958.
[23] S. Ma, L. Sigal, S. Sclaroff, Space-time tree ensemble for action recognition, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.
[24] Y. Wang, G. Mori, Learning a discriminative hidden part model for human action recognition, NIPS, volume 2, 2008.
[25] Y. Wang, G. Mori, Hidden part models for human action recognition: probabilistic versus max margin, TPAMI 33 (7) (2011) 1310–1323. 2, 3
[26] D. Weinland, E. Boyer, R. Ronfard, Action recognition from arbitrary views using 3d exemplars, ICCV, 2007.
[27] H.-I. Suk, B.-K. Sin, S.-W. Lee, Recognizing hand gestures using dynamic bayesian network, in: IEEE International Conference on Automatic Face & Gesture Recognition, 2008.
[28] ChaLearn Looking at People, http://gesture.chalearn.org/.
[29] I. Guyon, A. Elisseeff, An Introduction to Variable and Feature Selection, in: JMLR, volume 3.
[30] C.F. Aliferis, I. Tsamardinos, A.R. Statnikov, HITON: A novel markov blanket algorithm for optimal variable selection, American Medical Informatics Association Annual Symposium, 2003.
[31] W. Li, Z. Zhang, Z. Liu, Action recognition based on a bag of 3d points, IEEE International Workshop on CVPR for Human Communicative Behavior Analysis (in conjunction with CVPR2010), 2010. San Francisco, CA, June

[32] J. Wang, Z. Liu, Y. Wu, J. Yuan, Mining actionlet ensemble for action recognition with depth cameras, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2012),Providence, Rhode Island, June 16–21, 2012.
[33] G. Yu, Z. Liu, J. Yuan, Discriminative orderlet mining for real-time recognition of human-object interaction, ACCV, 2014.
[34] G. Yusuke, T. Wataru, N. Yoshihiko, Gesture recognition using hybrid generative-discriminative approach with fisher vector, in: IEEE International Conference on Robotics and Automation, 2015.
[35] B. Fernando, E. Gavves, J. Oramas, A. Ghodrati, T. Tuytelaars, Modeling video evolution for action recognition, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015), 2015.
[36] T. Inamura, Y. Nakamura, I. Toshima, Embodied symbol emergence based on mimesis theory, Int. J. Rob. Res. 23 (4) (2004) 363–377.
[37] D. Kulic, W. Takano, Y. Nakamura, Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden markov chains, Int. J. Rob. Res. 27 (7) (2008) 761–784.