Human action recognition based on mixed gaussian hidden markov model

Jiawei Xu[1,*], and Qian Luo[1]

[1]Beijing Information Science and Technology University, 35 North Fourth Ring Middle Road, Yayun Village Street, Chaoyang District, Beijing, China

Abstract. Human action recognition is a challenging field in recent years. Many traditional signal processing and machine learning methods are gradually trying to be applied in this field. This paper uses a hidden Markov model based on mixed Gaussian to solve the problem of human action recognition. The model treats the observed human actions as samples which conform to the Gaussian mixture model, and each Gaussian mixture model is determined by a state variable. The training of the model is the process that obtain the model parameters through the expectation maximization algorithm. The simulation results show that the Hidden Markov Model based on the mixed Gaussian distribution can perform well in human action recognition.

1 Introduction

Due to the development of sensors, the data used for human action recognition has become more abundant [1]. People are gradually turning their attention to the field of human action recognition. Many methods have tried to solve the problem of human action recognition, such as random forest [1], variants of random forest [2], graph convolutional neural networks [3], Deep Progressive Reinforcement Learning [4], Directed Graph Neural Networks [5] and so on. Most of these methods are based on constructing action features, and use discriminative methods to complete the recognition task. Hidden Markov model establishes the joint probability distribution of hidden variables and observation variables and introduces mixed Gaussian distribution to approximate the real distribution of human movements, which can express the distribution of the data themselves better. It can perform well in human action recognition.

2 Mixed gaussian hidden markov model

Hidden Markov model is a kind of dynamic Bayesian network. It is used to process time series data. The model consists of the initial state probability vector $\boldsymbol{\pi}$, the state transition probability matrix $\mathbf{A}=\{a_{ij}\}$ $(i=1,..,m;j=1,\ldots,n)$ and the observation probability matrix $\mathbf{B}=\{b_{jk}\}$ $(j=1,..,m;k=1,\ldots,n)$. m is the number of state variables $\mathbf{Z}=\{z_1, \ldots, z_m\}$ in the model. When

[*] Corresponding author: xujiawei@bistu.edu.cn

the hidden Markov model is based on Gaussian mixture distribution, n is the number of Gaussian distribution in Gaussian mixture model determined by state variable z_j（j=1,…, m）. The process of model training is to determine the values of $\boldsymbol{\pi}$, **A**, and **B**. In view of the complexity of human actions, this paper applies Gaussian mixture model to approximate the real distribution of human action data. The Gaussian mixture model is as formula (1).

$$p(x)=\sum_{k=1}^{n}\alpha(k)N(x\,|\,u_k,\Sigma_k) \tag{1}$$

The Gaussian mixture distribution is composed of n Gaussian distributions. $\alpha(k)$ is the weight of each Gaussian distribution, where $\sum_{k=1}^{n}\alpha(k)=1$. In the mixed Gaussian hidden Markov model, each state variable z_j has one Gaussian mixture model itself, and $\alpha(k)$ is the element b_{jk} of the observation probability matrix **B**. Meanwhile, $\alpha(k)$, u_k ,and Σ_k are parameter values that need to be determined during model training.

The Hidden Markov Model assumes that human actions are controlled by a set of unobserved hidden variables, namely state variables **Z**. The state variables **Z** may be a set of postures in practice, but may be other unobservable factors. $\boldsymbol{\pi}$ and **A** determine the state time series **Z**={z_1, …, z_t}, and **B** determines the observation time series **X**={x_1, …, x_t}. The generation process of the observation sequence is to select a state variable z_t at a certain time t. Next, select a specific Gaussian mixture model according to z_t , then an observation sample x_t is generated from the selected Gaussian mixture distribution. The generation process of observation x_{t+1} at time t+1 is as follows. Firstly, z_t and the state transition probability matrix **A** determine z_{t+1}, and then x_{t+1} is generated by the Gaussian mixture distribution of z_{t+1}. The initial state is generated from the initial state probability vector $\boldsymbol{\pi}$. It can be seen that the hidden Markov model uses random sampling controlled by a hidden Markov chain to explain the generation of observation sequences.

3 Expectation maximization algorithm

In order to solve the model parameters, this paper uses the Expectation Maximization(EM) algorithm. The EM algorithm is essentially an iterative update algorithm. The iteration process is divided into E step and M step. E step is to find the expectation of the conditional probability of the hidden variable **Z** of the log-likelihood function of the complete data, as in formula (2). M step finds the parameter value θ that maximizes the expectation in E step, as in formula (3).

$$Q(\theta,\theta^{(i)})=\sum_{Z}\log P(X,Z\,|\,\theta)P(Z\,|\,X,\theta^{(i)}) \tag{2}$$

$$\theta^{(i+1)}=\arg\max_{\theta}Q(\theta,\theta^{(i)}) \tag{3}$$

$\theta^{(i)}$ represents the parameter value obtained in the i th iteration in equations (2) and (3). The value $\theta^{(i+1)}$ is the θ which is obtained by maximizing $Q(\theta,\theta^{(i)})$ based on the fixed $\theta^{(i)}$ in the conditional probability $P(Z \mid X, \theta^{(i)})$. $\theta^{(i+1)}$ is the parameter value used in the *i*+1th iteration to compute $Q(\theta, \theta^{(i+1)})$.

In the Mixed Gaussian Hidden Markov Model, the parameters needed to be solved are a_{ij}, b_{jk}, u_k and Σ_k. The update process of parameters is as follows given the initial values **π₀**, a_0, b_0, u_0 and Σ_0.

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1}\sum_{k=1}^{K} \gamma_t(i,k)} \tag{4}$$

$$b_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k)}{\sum_{t=1}^{T}\sum_{k=1}^{K} \gamma_t(j,k)} \tag{5}$$

$$u_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k) \cdot x_t}{\sum_{t=1}^{T} \gamma_t(j,k)} \tag{6}$$

$$\Sigma_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k) \cdot (x_t - u_{jk})(x_t - u_{jk})^T}{\sum_{t=1}^{T} \gamma_t(j,k)} \tag{7}$$

where $\xi_t(i,j)$ represents the probability of being in state *i* at time *t* and being in state *j* at time *t*+1 given models and observations. $\gamma_t(j,k)$ is the probability that the observation is generated by the *k* th Gaussian component in the state *j* at time *t*. These two variables can be obtained by the forward-backward algorithm. Returning to the EM algorithm, the process of using the forward-backward algorithm to obtain the statistics $\xi_t(i,j)$ and $\gamma_t(j,k)$ is equivalent to E step. The parameter update process of formulas (4) ~ (7) is M step. E step and M step iteratively update until convergence, and the final required model parameters are obtained.

4 The simulation of algorithm

The dataset used in this simulation is the MSRAction3D dataset [6]. The MSRAction3D dataset is 3 dimensional skeleton data with respect to human actions. Because the posture of an action at a certain moment is composed of 20 joint point samples, and each joint point is composed of 3 dimensional space coordinates, the sample at a certain moment of the human body motion is a 60 dimensional data. The visualization of skeleton data is shown in Figure 1. It can be seen that the human skeleton is composed of 20 joint points. The data cost is greatly reduced by the use of skeleton sequences to express human actions.

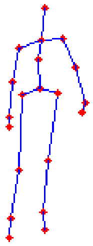

Fig. 1. The visualization of skeleton data.

This paper uses 8 actions as the training set, and the corresponding actions made by the participants in the non-training set as the test set. Actions are classified by number labels, and the selected action type labels are 2, 3, 5, 6, 10, 13, 18, and 20. For example, the action labeled 2 is shown in Figure 2. The action is “hand wave”.

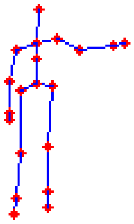

Fig. 2. The action “hand wave”.

Since it needs to be assumed that the number of hidden state and the Gaussian mixture number corresponding to the state in the mixed Gaussian hidden Markov model, this paper selects the number of states as 5 and the Gaussian mixture number as 3. The simulation trains a model for each type of action, and then substitutes the test action into each model to calculate the likelihood probability. The test action is recognized as the action corresponding to the model with the highest likelihood. Comparing the real label and the test action label, we can reach the recognition accuracy of each test action and the overall accuracy. Table 1 shows the recognition accuracy of each action, and Table 2 shows the recognition accuracy of human actions by different methods. This paper assumes that it is consistent that the number of states and the Gaussian mixture number for training all actions.

Table 1. Recognition accuracy of each action.

Action label	2	3	5	6	10	13	18	20
Accuracy (%)	83.3	33.3	36.4	63.7	100.0	100.0	100.0	66.7

Table 2. Recognition accuracy of different methods.

Method	HDP-HMM [8]	Li et al. [8]	Mixed Gaussian HMM (ours)
Accuracy (%)	75.3	74.7	75.5

It can be seen that the recognition effect of the mixed Gaussian Hidden Markov Model is quite different between different actions from Table 1. The main reason for the large difference in the recognition accuracy of each action is that the action itself has fewer features that can distinguish other actions. For example, action 5 can be easily judged as actions 2, 3, and 18 in simulation, because it has fewer unique features such that can be easily judged as other types of actions. One way to improve accuracy is to extract more new and detailed features from the action. For example, consider the skeleton as a directed graph to construct the relationship between points and edges[5], that is, the relationship between joint points and limbs. The new features may be more conducive to correctly judge the type of actions. For the overall recognition accuracy, Table 2 lists the accuracy of the three methods, showing that the model has well performance on human action recognition.

5 Conclusion

The mixed Gaussian Hidden Markov model has a good effect on human action recognition. The application of the method requires that the number of the state and Gaussian mixture need to be specified firstly. In future work, non-parametric methods can be introduced to determine the number of states and Gaussian mixture suitable for the model to improve the recognition accuracy.

References

1. Shotton J , Girshick R , Fitzgibbon A , et al. Efficient Human Pose Estimation from Single Depth Images[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2013, 35(12):2821-2840.
2. Tang D , Chang H J , Tejani A , et al. Latent Regression Forest: Structured Estimation of 3D Articulated Hand Posture[C]// 2014 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2014.
3. Sijie Yan, Yuanjun Xiong, and Dahua Lin, Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition, In AAAI, 2018.
4. Tang Y , Tian Y , Lu J , et al. Deep Progressive Reinforcement Learning for Skeleton-Based Action Recognition[C]// 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2018.
5. Shi L , Zhang Y , Cheng J , et al. Skeleton-Based Action Recognition With Directed Graph Neural Networks[C]// 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2020.
6. Li W , Zhang Z , Liu Z . Action recognition based on a bag of 3D points[C]// Computer Vision & Pattern Recognition Workshops. IEEE, 2010:9-14.
7. https://github.com/drbinliang/HMM-Action-Recognition
8. Raman N , Maybank S J . Action classification using a discriminative multilevel HDP-HMM[J]. Neurocomputing, 2015, 154(C):149-161.