HUMAN ACTIVITY RECOGNITION WITH BETA PROCESS HIDDEN MARKOV MODELS

QING-BIN GAO, SHI-LIANG SUN

Department of Computer Science and Technology, East China Normal University
500 Dongchuan Road, Shanghai 200241, P.R. China
E-MAIL: qbgao10@gmail.com, slsun@cs.ecnu.edu.cn

Abstract:

Trajectory-based human activity recognition aims at understanding human behaviors in video sequences, which is important for intelligent surveillance. Some existing approaches to this problem, e.g., the hierarchical Dirichlet process hidden Markov models (HDP-HMM), have a severe limitation, namely the motions cannot be shared among activities. To overcome this shortcoming, we propose a new method for modeling human trajectories based on the beta process hidden Markov models (BP-HMM). Using our technique, the number of features and the sharing schema can both be inferred automatically from training data. We develop an efficient Markov chain Monte Carlo algorithm for model training. Experiments on both synthetic and real data sets demonstrate the effectiveness of our approach.

Keywords:

Human activity recognition; trajectory classification; beta process; Markov chain Monte Carlo

1. Introduction

Effective human activity recognition (HAR) is crucial for the successful application of intelligent surveillance systems. The purpose of HAR is to understand what people are doing from their position, figure, motion, or other spatio-temporal information derived from video sequences. In this paper, we focus on recognizing human behaviors from trajectory data [1, 2]. From daily experience we know that a human activity can be modeled by transitions among simple motions, and moreover, multiple trajectories may share some motions. For example, in a certain shopping mall, the activity of a customer "entering the shop" may be decomposed into "moving east" first and then "moving north", and the activity of a customer "leaving the shop" may be decomposed into "moving south" first and then "moving east". In this simple case, motion "moving east" is shared while motion "moving north" and motion "moving south" are class-specific.

In this paper, we propose a method for trajectory-based HAR tasks. We first try to discover a set of latent motions, including the shared motions and the unique motions, in specific trajectories. Then we model the trajectories by transitions of different motions. A Markov chain Monte Carlo (MCMC) algorithm is developed for efficient model training. The final classifier for HAR tasks is given by maximizing the log-likelihood of a test trajectory. Our work is inspired by the BP-HMM [3], which is also concerned with feature sharing. However, there are important distinctions. First, main dynamic system in the BP-HMM is a vector autoregressive process while we use a HMM with Gaussian emissions which is more appropriate for trajectory-based HAR tasks. Second, our sampling method is partially different from the BP-HMM, which is more suitable for our model.

2. Modeling human activities with BP-HMM

Our task is to map a sequential trajectory $\boldsymbol{x}$ to a single activity label y. Formally, let $\boldsymbol{x} = (x_1, x_2, ..., x_T)$ be a specific trajectory where $x_t \in R^2$ denotes the displacement of a person from time $t-1$ to time t. Note that the two components of vector x_t respectively correspond to vertical and horizontal displacement. z_t denotes the invisible motion label of x_t. In our model each x_t is a draw from a Gaussian distribution with unknown mean and covariance:

$$x_t \sim N(\mu_{z_t}, \Sigma_{z_t}). \tag{1}$$

We place a conjugate normal-inverse-Wishart prior, which we denote by $NIW(\cdot)$, on the Gaussian:

$$(\mu_{z_t}, \Sigma_{z_t}) \sim NIW(\zeta, \vartheta, \nu, \Delta). \tag{2}$$

As in usual HMMs, we model the sequence of motion labels as a Markov chain:

$$z_t \mid z_{t-1} \sim Multinomial(\boldsymbol{\pi}_{z_{t-1}}), \tag{3}$$

where $\boldsymbol{\pi}_{z_{t-1}}$ is the transition probability vector of state z_{t-1}.

Let $k = 1, 2, ...$ denote the distinct values the motion labels z_t can take on. The classical HDP-HMM [4] and its sticky extension [2, 5] have given us ways to model $\boldsymbol{\pi}_k$. However, the motions will not be shared following these methods. In order to share motions among trajectories, we represent the set of motions available to each trajectory with a binary feature vector. In particular, we denote the binary feature vector of the currently considered trajectory $\boldsymbol{x}$ by $\boldsymbol{f} = [f_1, f_2, ...]$, with $f_k = 1$ indicating that motion k is included in trajectory $\boldsymbol{x}$ for some values $t \in \{1, ..., T\}$. Thus we can interpret how the motions are shared by discovering the feature vectors. For example, if we find $f_k = \bar{f}_k = 1$ for trajectory $\boldsymbol{x}$ and trajectory $\bar{\boldsymbol{x}}$, we can conclude that the both trajectories exhibit motion k. In this paper, we generate the feature vectors by the beta process (BP) [6]:

$$B \sim BP(\beta, \gamma B_0), \ B = \sum_{k=1}^{\infty} \omega_k \delta_{\theta_k}. \tag{4}$$

Here, we denote the motion parameters (μ_k, Σ_k) as θ_k for simplicity. B_0 is the base measure according to which θ_k are distributed, $\theta_k \sim B_0$. Hyperparameter γ controls the number of available motions in a trajectory. Hyperparameter β controls how often motions are shared among trajectories. ω_k are inclusion weights which lie in the interval $(0, 1)$, $\omega_k \in (0, 1)$. Given the above definitions, the feature vector is determined by:

$$f_k \sim Bernoulli(\omega_k). \tag{5}$$

In order to obtain the transition distribution $\boldsymbol{\pi}_k$, we first sample a set of transition weights $\boldsymbol{\eta}_k$ by:

$$\begin{aligned} &\bar{\boldsymbol{\eta}}_k \sim Dirichlet(..., n_{kj} + \alpha + \kappa\delta_{k,j}, ...), \\ &C_k \sim Gamma(K\alpha + \kappa, 1), \\ &\boldsymbol{\eta}_k = C_k \bar{\boldsymbol{\eta}}_k. \end{aligned} \tag{6}$$

Here, auxiliary variables $\bar{\boldsymbol{\eta}}_k$ and C_k are introduced in order to get $\boldsymbol{\eta}_k$. α and κ are hyperparameters. K is the number of all available motions. $\delta_{k,j}$ indicates the Kronecker delta function. n_{kj} is the number of transitions from k to j in currently considered trajectory.

Given $\boldsymbol{f}$ and $\boldsymbol{\eta}_k$, the trajectory-specific transition distributions $\boldsymbol{\pi}_k$ can be given as:

$$\boldsymbol{\pi}_k = \frac{\boldsymbol{\eta}_k \bigotimes \boldsymbol{f}}{\sum_j f_j \eta_{kj}}, \tag{7}$$

where $\bigotimes$ denotes the Hadamard vector product [7].

3. Training with a MCMC sampling method

Let $\mathcal{X} = \{(\boldsymbol{x}^{(1)}, y_1), (\boldsymbol{x}^{(2)}, y_2), ..., (\boldsymbol{x}^{(N)}, y_N)\}$ be all of our training data, where each $(\boldsymbol{x}^{(i)}, y_i)$ denotes a specific labeled trajectory. $\boldsymbol{x}^{(i)}$ is the ith sequence of observations and y_i is the corresponding activity labels represented by a set of constants. Assume we have $\mathcal{Y}$ activities, then $y_i \in \{1, 2, ..., \mathcal{Y}\}$. Since the motions are globally shared while the transitions are trajectory-specific, the parameters to be learned are $\boldsymbol{\Pi} = \{\boldsymbol{\pi}^{(1)}, \boldsymbol{\pi}^{(2)}, ..., \boldsymbol{\pi}^{(N)}\}$ and $\boldsymbol{\Theta} = \{(\mu_1, \Sigma_1), (\mu_2, \Sigma_2), ..., (\mu_K, \Sigma_K)\}$, where each $\boldsymbol{\pi}^{(i)} = \{\boldsymbol{\pi}_1^{(i)}, \boldsymbol{\pi}_2^{(i)}, ..., \boldsymbol{\pi}_K^{(i)}\}$ denotes the transitions of the ith training trajectory and each (μ_k, Σ_k) denotes the parameters of motion k. In order to learn the parameters, we develop an efficient MCMC sampling technique which is shown as follows.

Sampling feature vectors $\boldsymbol{f}^{(i)}$. We first try to obtain the feature vectors, because posterior updates to other parameters are greatly simplified if we have the motion assignments $\boldsymbol{f}^{(i)}$. Sampling each trajectory's motions requires separate updates to motions shared with some other trajectories and motions unique to itself. For shared motions, we accept or reject them according to the Metropolis-Hastings rule. For unique motions, we define a reversible pair of birth and death moves which add or delete motions to a single trajectory. These follow directly from the theory of feature sampling for the beta process autoregressive hidden Markov model (BP-AR-HMM), for details of which the interested reader is referred to [3].

Sampling motion sequences $\boldsymbol{z}^{(i)}$. After sampling the motion assignments, our training trajectories can be considered as a collection of finite HMMs with Gaussian emissions. Consider a trajectory $\boldsymbol{x}^{(i)} = (x_1^{(i)}, x_2^{(i)}, ..., x_{T_i}^{(i)})$ with $\boldsymbol{z}^{(i)} = (z_1^{(i)}, z_2^{(i)}, ..., z_{T_i}^{(i)})$ representing the hidden motion labels. By the Markov property, the joint posterior distribution of $\boldsymbol{z}^{(i)}$ is:

$$\begin{aligned} &p(\boldsymbol{z}^{(i)} | \boldsymbol{x}^{(i)}, \boldsymbol{\pi}^{(i)}, \boldsymbol{\Theta}) \\ &= p(z_1^{(i)} | \boldsymbol{x}^{(i)}, \boldsymbol{\pi}^{(i)}, \boldsymbol{\Theta}) \prod_{t=2}^{T_i} p(z_t^{(i)} | z_{t-1}^{(i)}, \boldsymbol{x}^{(i)}, \boldsymbol{\pi}^{(i)}, \boldsymbol{\Theta}). \end{aligned}$$

This implies that we can adopt a backward message-passing forward sampling scheme. In particular, we first sample $z_1^{(i)}$, and then sample state $z_t^{(i)}$ conditionally on the previous state $z_{t-1}^{(i)}$ $(t = 2, \ldots, T_i)$. Let $m_{t,t-1}(z_{t-1}^{(i)})$ denote the backward message passed from $z_t^{(i)}$ to $z_{t-1}^{(i)}$, which are defined as:

$$m_{t,t-1}(z_{t-1}^{(i)}) \propto \begin{cases} \sum_{z_t^{(i)}} p(z_t^{(i)}|\pi_{z_{t-1}^{(i)}})N(x_t^{(i)}|\mu_{z_t^{(i)}}, \Sigma_{z_t^{(i)}}) & \\ \cdot m_{t+1,t}(z_t^{(i)}), & t = 2, ..., T_i; \\ 1, & t = T_i + 1. \end{cases}$$

Thus the conditional distribution of $z_1^{(i)}$ is:

$$\begin{aligned} & p(z_1^{(i)}|\boldsymbol{x}^{(i)}, \boldsymbol{\pi}^{(i)}, \boldsymbol{\Theta}) \\ & \propto p(z_1^{(i)})N(x_1^{(i)}|\mu_{z_1^{(i)}}, \Sigma_{z_1^{(i)}})m_{2,1}(z_1^{(i)}). \end{aligned} \tag{8}$$

For $t = 2, ..., T$, the conditional distribution of $z_t^{(i)}$ is:

$$\begin{aligned} & p(z_t^{(i)}|z_{t-1}^{(i)}, \boldsymbol{x}^{(i)}, \boldsymbol{\pi}^{(i)}, \boldsymbol{\Theta}) \\ & \propto p(z_t^{(i)}|\pi_{z_{t-1}^{(i)}})N(x_t^{(i)}|\mu_{z_t^{(i)}}, \Sigma_{z_t^{(i)}})m_{t+1,t}(z_t^{(i)}). \end{aligned} \tag{9}$$

Sampling transition distributions $\boldsymbol{\pi}^{(i)}$. As described in Section 2, we sample $\boldsymbol{\pi}^{(i)}$ by:

$$\begin{aligned} \bar{\boldsymbol{\eta}}_k^{(i)} &\sim Dirichlet(..., n_{kj}^{(i)} + \alpha + \kappa\delta_{k,j}, ...), \\ C_k^{(i)} &\sim Gamma(K\alpha + \kappa, 1), \\ \boldsymbol{\eta}_k^{(i)} &= C_k^{(i)}\bar{\boldsymbol{\eta}}_k^{(i)}, \\ \boldsymbol{\pi}_k^{(i)} &= \frac{\boldsymbol{\eta}_k^{(i)} \bigotimes \boldsymbol{f}^{(i)}}{\sum_j f_j^{(i)}\eta_{kj}^{(i)}}. \end{aligned} \tag{10}$$

Sampling emission parameters (μ_k, Σ_k). As mentioned in (2), we place a $NIW(\zeta, \vartheta, \nu, \Delta)$ prior on (μ_k, Σ_k). For a specific iteration of the sampler, let $\boldsymbol{X}_k$ denote the set of observations with the same hidden motion, i.e., $\boldsymbol{X}_k = \{x_t|z_t = k\}$. Thus the posterior distributions of (μ_k, Σ_k) are [5]:

$$(\bar{\mu}_k, \bar{\Sigma}_k) \sim NIW(\bar{\zeta}_k, \bar{\vartheta}_k, \bar{\nu}_k, \bar{\Delta}_k), \tag{11}$$

where

$$\begin{aligned} \bar{\nu}_k &= \nu + |X_k|, \\ \bar{\nu}_k\bar{\Delta}_k &= \nu\Delta + \sum_{\boldsymbol{x_t}\in\boldsymbol{X}_k} \boldsymbol{x_t}\boldsymbol{x_t'} + \zeta\vartheta\vartheta' - \bar{\zeta}_k\bar{\vartheta}_k\bar{\vartheta}_k', \\ \bar{\zeta}_k\bar{\vartheta}_k &= \zeta\vartheta + \sum_{\boldsymbol{x_t}\in\boldsymbol{X}_k} \boldsymbol{x_t}, \\ \bar{\zeta}_k &= \zeta + |X_k|. \end{aligned}$$

Optionally, we place priors over the hyperparameters α, κ, γ and β when we do not have strong beliefs about them, e.g., $\alpha \sim Gamma(a_\alpha, b_\alpha)$, $\kappa \sim Gamma(a_\kappa, b_\kappa)$, $\gamma \sim Gamma(a_\gamma, b_\gamma)$ and $\beta \sim Gamma(a_\beta, b_\beta)$. The hyperparameters can be sampled as described in [7].

4. Classification

Before classification, we perform an additional procedure on the trajectory-specific transitions $\{\boldsymbol{\pi}^{(1)}, \boldsymbol{\pi}^{(2)}, ..., \boldsymbol{\pi}^{(N)}\}$, in order to get the expected class-specific transitions $\{\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, ..., \boldsymbol{\pi}_{\mathcal{Y}}\}$:

$$\boldsymbol{\pi}_y = \frac{\sum_{i|y_i=y} \boldsymbol{\pi}^{(i)}}{N_y},$$

where N_y is the number of training trajectories which belong to class y.

For testing, given a new trajectory $\boldsymbol{x}$, we classify it into activity $y^* \in \{1, ..., \mathcal{Y}\}$ by maximizing the log-likelihood:

$$y^* = \arg\max_{y\in\mathcal{Y}}\{\log p(\boldsymbol{x}|\boldsymbol{\pi}_y, \boldsymbol{\Theta}\}, \tag{12}$$

The observation likelihood $p(\boldsymbol{x}|\boldsymbol{\pi}_y, \boldsymbol{\Theta})$ can be compute directly using a forward message passing [8] which we will not describe here.

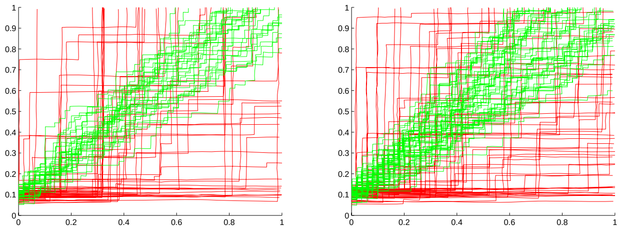
5. Experiments

We test the performance of our model on both synthetic and real data. The synthetic data have two classes of simple activities, which aims at demonstrating the capability of our approach to recover the true model. Experimental results on data from real-world scenes include comparisons with state-of-the-art methods.

5.1. Synthetic data

First, we concentrate on an ideal scenario which is similar to the synthetic case discussed in [9]. We consider two different classes of activities both of which are made up of two different motions: moving horizontally and moving vertically. The mean of horizontal displacements is $\mu_1 = [0.02\ 0]^T$ and the mean of vertical displacements is $\mu_2 = [0\ 0.02]^T$. The corresponding covariances are $\Sigma_1 = \Sigma_2 = 10^{-3}\mathbf{I}$. The difference between the two classes resides on the transitions, where one class has a low probability of switching between two different motions while the other has an identical probability of switching between any two motions. Examples of the two classes of activities are shown in Figure 1, which are depicted in red and green. Respectively for the red and green activities, training sets are generated from HMMs with transitions:

$$\mathbf{T}^{red} = \begin{bmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{bmatrix}, \qquad \mathbf{T}^{green} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}.$$



(a) (b)

Figure 1. Two synthetic activities with different transition matrices: (a) training data, (b) test data.

Given the above setting, we generate 100 training trajectories and 100 test trajectories.

We run 500 iterations using the synthetic training sets with fixed BP hyperparameters $\beta = 1$, $\gamma = 2$ and HMM hyperparameters $\alpha = 1$, $\kappa = 25$. Figure 2(a) plots the updates of motion numbers through the iterations. As we can see, our sampler converges to the right motion number after a few iterations and the number becomes stable afterwards. We check the emission parameters and transition matrices sampled at the 500th iteration. Respectively for the two motions, the estimated means are $\tilde{\mu}_1 = [0.0200\ 0.0001]^T$ and $\tilde{\mu}_2 = [0.0000\ 0.0200]^T$. The estimations of the motion parameters along with the original observations are shown in Figure 2(b). Respectively for the two classes, the estimated transition matrices are:

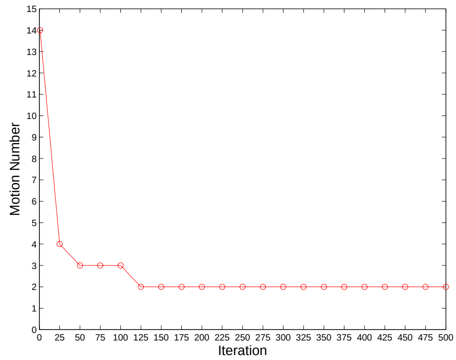
$$\tilde{\mathbf{T}}^{red} = \begin{bmatrix} 0.9530 & 0.0470 \\ 0.0455 & 0.9545 \end{bmatrix}, \quad \tilde{\mathbf{T}}^{green} = \begin{bmatrix} 0.5216 & 0.4784 \\ 0.5049 & 0.4951 \end{bmatrix}.$$

As expected, both the estimated emission parameters and transition matrices are very close to the true setting. Furthermore, the non-zero estimated transition probabilities imply that motion sharing does exhibit between the two classes of activities.

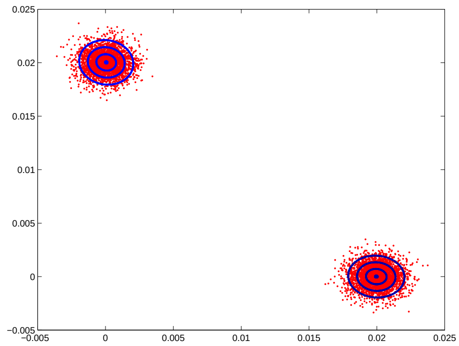
Finally, we apply the results of the 500th iteration to the test data. The classification accuracy is $\mathbf{100}\%$, showing that our model is feasible to recognize trajectories.

5.2. Two real-world scenes

We then consider HAR under two real-world scenes, which include a shopping center and a university campus [9]. For the shopping center scene, four classes of activities are predefined, which are "entering" (E), "leaving" (L), "passing" (P), and



(a)



(b)

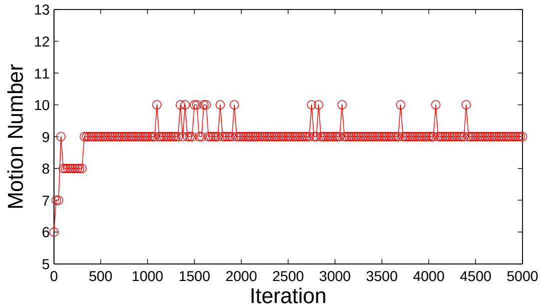
Figure 2. Experimental results with the synthetic data: (a) updates of the motion number through the iterations, (b) observations and motion parameters in the 500th iteration.

"browsing" (B). For the university campus scene, seven classes of activities are predefined, which are "entering" building (E), "leaving" building (L), "crossing park up" (CPU), "crossing park down" (CPD), "passing through" (PT), "walking along" (WA), and "wandering" (W). Figure 3 shows the two scenes with example trajectories.

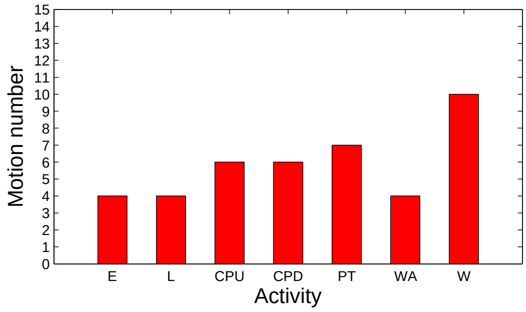
The source video sequences were obtained by surveillance cameras. Preprocessing on the video sequences included projective transformation, pedestrian tracking and position extraction [9]. However, in order to use the more meaningful displacement features, we perform an additional operation on the position data. With an original trajectory represented by $\mathbf{p} = \{p_0, p_1, ..., p_T\}$, our input trajectory will be $\boldsymbol{x} = \{x_1, x_2, ..., x_T\}$, where $x_t = p_t - p_{t-1}$ $(t = 1, ..., T)$. After the above processing, we get 53 trajectories in the shop-

(a) (b) (c) (d)

Figure 3. The two real-world scenes with example trajectories: (a) E, in the shopping center scene, (b) L, in shopping center scene, (c) CPU, in the campus scene, (d) CPD, in the campus scene.



(a)



(b)

Figure 4. Experimental results about motion numbers with the campus data: (a) updates of the motion number through the iterations, (b) class-specific motion numbers obtained by the sticky HDP-HMM.

ping scene and 143 trajectories in the campus scene.

The switched dynamical hidden Markov model (SD-HMM) approach [9] is a recent method for trajectory-based HAR. In order to assess the accuracy of our approach and perform a comparison with the SD-HMM, we consider a specific procedure for splitting the available data into training and test sets, which is totally identical with the first splitting procedure used in the SD-HMM. In particular, the training set contains three randomly picked trajectories from each class of activity and the test set contains the remaining trajectories.

We respectively run 5000 iterations on the shopping center and campus training set. We choose the campus scene, which is the more complicated one, to validate the performance of our method on motion numbers. Figure 4(a) shows the updates of the motion number through the iterations. As we can see, our sampler converges to a stable motion number, which is nine, after about 2500 iterations. The minor waves between nine and ten afterwards are reasonable. If the "birth" of a new motion happen in current iteration with no observation assigning to it, the "death" of this motion will come in the following iteration. That is why the waves occur. The class-specific motion numbers obtained by the sticky HDP-HMM [5] are shown in Figure 4(b). Comparing Figure 4(a) and Figure 4(b), we can find the total motion number obtained from our method is far less than the sum of the class-specific motion numbers, which implies that motion sharings do exist among the activities.

To evaluate the classification accuracies, we select 100 sets of trained parameters between the 2500th and 5000th iteration. We separately do the test using the selected parameters and then compute the mean accuracy. The final confusion matrices for the campus scene and the shopping center scene are respectively given in Table 1 and Table 2. As we can see, our approach achieves a good performance for all classes, even in the cases of complicated activities like "B" and "W".

In order to illustrate the general performance, we show the overall error rate of our method in Table 3. Moreover, the results of other methods from [9] are employed as a comparison. As we can see, our method either outperforms the SD-HMM in the both scenes.

TABLE 1. Classification accuracies (%) of the campus scene.

True class	E	L	CPU	CPD	PT	WA	W
E	98.94	0	0	0	0	1.06	0
L	1.13	98.87	0	0	0	0	0
CPU	0	0	98.95	0	0	0	1.05
CPD	0	0	4.78	94.38	0	0.84	0
PT	0	1.75	0	0	98.25	0	0
WA	0	0	0	0.83	0	99.17	0
W	0	0	0	0	0	9.00	91.00

TABLE 2. Classification accuracies (%) of the shopping center scene.

True class	E	L	P	B
E	98.20	1.80	0	0
L	0	97.56	0	2.44
P	2.00	0	98.00	0
B	0	8.17	0	91.83

TABLE 3. Comparison of overall error rates (%).

Approach	Shopping	Campus
HMM + AIC	16.67	8.55
HMM+BIC/MDL	5.56	7.24
SD-HMM	3.70	3.29
PROPOSED APPROACH	2.95	2.87

6. Conclusion

In this paper, we have presented a method for modeling and recognizing human activities. We model the distributions of displacements in a human trajectory as Gaussians and the temporal evolution of invisible motions as a Markov chain. Using a beta process prior, our approach discovers a set of latent motions that are shared among multiple trajectories. Parameters are learned from an efficient MCMC sampling algorithm. For recognition, a test trajectory is categorized by maximizing the log-likelihood. Experimental results have validated the good performance of our method in comparison with other methods.

Acknowledgements

Thanks to NSFC Project 61075005 and Shanghai Knowledge Service Platform Project (No. ZF1213) for funding.

References

[1] Q. Gao and S. Sun, “Trajectory-based human activity recognition using hidden conditional random fields”, Proceedings of the International Conference on Machine Learning and Cybernetics, pp. 1091-1097, 2012.

[2] Q. Gao and S. Sun, “Trajectory-based human activity recognition with hierarchical Dirichlet process hidden Markov models”, Proceedings of the 1st IEEE China Summit and International Conference on Signal and Information Processing, 2013.

[3] E. B. Fox, E. B. Sudderth, M. I. Jordan and A. S. Willsky, “Joint modeling of multiple related time series via the beta process”, arXiv preprint arXiv:1111.4226v1, 2011.

[4] Y. W. Teh, M. I. Jordan, M. J. Beal and D. M. Blei, “Hierarchical Dirichlet processes”, Journal of the American Statistical Association, pp. 1566-1581, 2006.

[5] E. B. Fox, E. B. Sudderth, M. I. Jordan and A. S. Willsky, “A sticky HDP-HMM with application to speaker diarization”, The Annals of Applied Statistics, pp. 1020-1056, 2011.

[6] R. Thibaux and M. I. Jordan, “Hierarchical beta processes and the Indian buffet process”, Proceedings of the International Conference on Artificial Intelligence and Statistics, pp. 564-571, 2007.

[7] M. C. Hughes, E. B. Fox and E. B. Sudderth, “Effective split-merge Monte Carlo methods for nonparametric models of sequential data”, Advances in Neural Information Processing Systems, pp. 1304-1312, 2012.

[8] C. M. Bishop, “Pattern Recognition and Machine Learning”, Springer Science, Spring Street, New York, USA, 2002.

[9] J. C. Nascimento, A. T. Figueiredo and J. S. Marques, “Trajectory classification using switched dynamical hidden Markov models”, IEEE Transactions on Image Processing, pp. 1338-1348, 2010.