TRAJECTORY-BASED HUMAN ACTIVITY RECOGNITION WITH HIERARCHICAL DIRICHLET PROCESS HIDDEN MARKOV MODELS

Qingbin Gao, Shiliang 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. Some existing approaches to this problem, e.g., hidden Markov models (HMM), have a severe limitation, namely the number of motions has to be preset. In fact, this number is difficult to define in advance in real practice. To overcome this shortcoming, we propose a new method for modeling human trajectories based on the hierarchical Dirichlet process hidden Markov models (HDP-HMM), and adopt a Gibbs sampling algorithm for model training. Using our proposed technique, the number of motions can be inferred automatically from data and is also allowed to vary among different classes of activities. Experiments on both synthetic and real data sets demonstrate the effectiveness of our approach.

Index Terms— Human activity recognition, trajectory classification, HDP-HMM, Gibbs sampler

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 [1], figure [2], motion [3], or other spatio-temporal information derived from video sequences. In this paper, we focus on recognizing human behaviors from trajectory data [4]. From daily experience we know that a human activity can be modeled by transitions among simple 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". This observation underlies the use of models with hidden states, which have the capability to capture intrinsic structures of activities.

In this paper, we propose a method for trajectory-based HAR tasks. In our method, activities are modeled by transitions of different motions and the number of motions can vary among activities. A truncated approximation of the hierarchical Dirichlet process (HDP) [5] is adopted for efficient model training. To keep the number of the hidden motions from being unnecessarily large, we further employ ideas from the sticky HDP-HMM [6]. Parameters of our model are estimated by a Gibbs sampler. The final classifier for HAR tasks is given by maximizing the log-likelihood of a test trajectory. We examine our method on both synthetic and real data sets and compare its performance against other state-of-the-art methods. Experimental results show the superiority of our method.

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

2. THE PROPOSED HUMAN ACTIVITY MODEL

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 Gaussian prior on the mean and an inverse-Wishart prior on the covariance:

$$\mu_{z_t} \sim N(\mu_0, \Sigma_0),\ \Sigma_{z_t} \sim IW(\Psi, \nu). \tag{2}$$

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 that the motion labels z_t can take on. The classical HDP-HMM has given us a traditional way to model $\boldsymbol{\pi}_k$ [5]. However, it has a severe rapid-switching problem. In particular, a standard HDP-HMM creates redundant states and rapidly switches among them. For HAR tasks, the redundant motions may cause a poor performance. To the best of our knowledge, one approach to avoiding the rapid-switching problem is the sticky

HDP-HMM [6], which extends the HDP-HMM by introducing a self-transition bias. Consequently, we model $\boldsymbol{\pi}_k$ as:

$$\begin{aligned}&\boldsymbol{\beta} = (\beta_k)_{k=1}^{L} \sim Dir(\gamma/L, ..., \gamma/L),\\ &\boldsymbol{\pi}_k \sim Dir(\alpha_0\beta_1, ..., \alpha_0\beta_L + \kappa, ..., \alpha_0\beta_L),\end{aligned} \tag{4}$$

where $Dir(\cdot)$ means a Dirichlet distribution and L is the maximum number of possible motions. α_0, κ and γ are hyperparameters and we place priors over them when we do not have strong beliefs about them, e.g., $(\alpha_0 + \kappa) \sim Gamma(a_1, b_1)$ and $\gamma \sim Gamma(a_2, b_2)$. Although the fixed truncation level reduces our model to a parametric model, it is substantially different from a classical parametric model with model selection. Actually by setting the truncation level reasonably higher than the true motion number, we will not lose any useful information. Comparing with the classical nonparametric HDP, a significant time saving can be achieved by using this technique [7].

2.1. Training with a Gibbs sampler

Let $\mathcal{T} = \{\mathcal{T}^1, ..., \mathcal{T}^{\mathcal{Y}}\}$ be all the training data, where each collection $\mathcal{T}^y = \{(\boldsymbol{x}_1, y), ..., (\boldsymbol{x}_{N_y}, y)\}$ denotes the training set of activity y. N_y is the number of training trajectories belonging to activity y. The parameters to be learned are $\boldsymbol{\Pi} = \{\boldsymbol{\pi}^1, ..., \boldsymbol{\pi}^{\mathcal{Y}}\}$ and $\boldsymbol{\Theta} = \{(\boldsymbol{\mu}, \boldsymbol{\Sigma})^1, ..., (\boldsymbol{\mu}, \boldsymbol{\Sigma})^{\mathcal{Y}}\}$, where $\boldsymbol{\pi}^y = \{\boldsymbol{\pi}_0^y, \boldsymbol{\pi}_1^y, ..., \boldsymbol{\pi}_L^y\}$ denotes the transitions of activity y and $(\boldsymbol{\mu}, \boldsymbol{\Sigma})^y = \{(\mu_1^y, \Sigma_1^y), ..., (\mu_L^y, \Sigma_L^y)\}$ denotes the emission parameters of activity y. Since the $\mathcal{Y}$ groups of parameters are learned separately in the same way, for simplicity, we will represent the targeted group-specific parameters $\boldsymbol{\pi}^y$ and $(\boldsymbol{\mu}, \boldsymbol{\Sigma})^y$ by $\boldsymbol{\pi}$ and $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, respectively.

Consider a trajectory $\boldsymbol{x} = (x_1, x_2, ..., x_T)$ with $\boldsymbol{z} = (z_1, z_2, ..., z_T)$ representing the hidden motion labels. By the Markov property, the joint posterior distribution of $\boldsymbol{z}$ is:

$$\begin{aligned}&p(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\pi}, (\boldsymbol{\mu}, \boldsymbol{\Sigma}))\\ &= p(z_1|\boldsymbol{x}, \boldsymbol{\pi}, (\boldsymbol{\mu}, \boldsymbol{\Sigma})) \prod_{t=2}^{T} p(z_t|z_{t-1}, \boldsymbol{x}, \boldsymbol{\pi}, (\boldsymbol{\mu}, \boldsymbol{\Sigma})).\end{aligned} \tag{5}$$

This implies that, we can first sample z_1, and then sample state z_t conditionally on the previous state z_{t-1} ($t = 2, \ldots, T$).

Sampling $\boldsymbol{z}$. By introducing the truncation level L, the backward message passing algorithm [8] can be used to sample each z_t. Let $m_{t,t-1}(z_{t-1})$ denote the backward message passed from z_t to z_{t-1}, which are defined as:

$$\begin{aligned}&m_{t,t-1}(z_{t-1})\\ &\propto \begin{cases}\sum_{z_t} p(z_t|\pi_{z_{t-1}}) N(x_t|\mu_{z_t}, \Sigma_{z_t}) & \\ \cdot m_{t+1,t}(z_t) & \text{if } t = 2, ..., T,\\ 1, & \text{if } t = T+1.\end{cases}\end{aligned} \tag{6}$$

Thus the conditional distribution of z_1 is:

$$p(z_1|\boldsymbol{x}, \boldsymbol{\pi}, (\boldsymbol{\mu}, \boldsymbol{\Sigma})) \propto p(z_1) N(x_1|\mu_{z_1}, \Sigma_{z_1}) m_{2,1}(z_1). \tag{7}$$

For $t = 2, ..., T$, the conditional distribution of z_t is:

$$\begin{aligned}&p(z_t|z_{t-1}, \boldsymbol{x}, \boldsymbol{\pi}, (\boldsymbol{\mu}, \boldsymbol{\Sigma}))\\ &\propto p(z_t|\pi_{z_{t-1}}) N(x_t|\mu_{z_t}, \Sigma_{z_t}) m_{t+1,t}(z_t).\end{aligned} \tag{8}$$

Sampling (μ_k, Σ_k). As mentioned in (2), we place a Gaussian prior $N(\mu_0, \Sigma_0)$ on the mean μ_k and an inverse-Wishart prior $IW(\Psi, \nu)$ on the covariance 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 and Σ_k are:

$$\Sigma_k|\mu_k \sim IW(\bar{\nu}_k \bar{\Psi}_k, \bar{\nu}_k),\ \mu_k|\Sigma_k \sim N(\bar{\mu}_k, \bar{\Sigma}_k), \tag{9}$$

where $\bar{\nu}_k = \nu + |X_k|$, $\bar{\nu}_k \bar{\Psi}_k = \nu\Psi + \sum_{x_t \in X_k} (x_t - \mu_k)(x_t - \mu_k)'$, $\bar{\Sigma}_k = (\Sigma_0^{-1} + |X_k|\Sigma_k^{-1})^{-1}$, $\bar{\mu}_k = \bar{\Sigma}_k(\Sigma_0^{-1}\mu_0 + \Sigma_k \sum_{x_t \in X_k} x_t)$.

Sampling $\boldsymbol{\pi}$. Given the priors of $\boldsymbol{\beta}$ and $\boldsymbol{\pi}$ defined by (4), the posteriors of $\boldsymbol{\beta}$ and $\boldsymbol{\pi}$ are:

$$\begin{aligned}&\boldsymbol{\beta}|\bar{\boldsymbol{m}}, \gamma \sim Dir(\gamma/L + \bar{m}_{\cdot 1}, ..., \gamma/L + \bar{m}_{\cdot L}),\\ &\boldsymbol{\pi}_k|\boldsymbol{z}, \alpha_0, \boldsymbol{\beta} \sim Dir(\alpha_0\beta_1 + n_{k1},\\ &..., \alpha_0\beta_k + \kappa + n_{kk}, ..., \alpha_0\beta_L + n_{kL}).\end{aligned} \tag{10}$$

For $k = 1, ..., L$, n_{kj} is the number of transitions from state k to state j in the current iteration. For $k = 0$, n_{0j} is the number that state j starting a trajectory, we write $\bar{n}_j = n_{0j}$ for simplicity. $\bar{m}_{jk}$ corresponds to the number of tables in restaurant j that are serving dish k in the sticky HDP-HMM, and $\bar{m}_{\cdot k} = \sum_j \bar{m}_{jk}$. In the following algorithm, we will describe the procedures to obtain $\bar{m}_{jk}$. For a detailed derivation of the sticky HDP-HMM, please follow fox et al. [6].

In general, given the training trajectories $\mathcal{T}^y$ and a previous set of the targeted parameters $\boldsymbol{\pi}^{old}$, $\boldsymbol{\beta}^{old}$, and $(\boldsymbol{\mu}^{old}, \boldsymbol{\Sigma}^{old})$, the Gibbs sampler updates them in the current iteration as follows.

- Set $\boldsymbol{\pi} = \boldsymbol{\pi}^{old}$ and $(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = (\boldsymbol{\mu}^{old}, \boldsymbol{\Sigma}^{old})$.
- Set $n_{kj} = 0$, $\bar{n}_j = 0$ and $\boldsymbol{X}_k = \emptyset$ for $(k, j) \in \{1, ..., L\}^2$.
- For $i = 1, ..., N_y$:
  - Select trajectory $\boldsymbol{x}_i$, set $T = T_{\boldsymbol{x}_i}$.
  - For $t = T, ..., 1$, compute each message as defined by (6).
  - For $t = 1, ..., T$:
    - Sample each z_t as defined by (7) or (8).
    - For a new assignment $z_t = k$, update $\bar{n}_{z_1}$ or n_{z_{t-1}, z_t}, and update the set of observations $\boldsymbol{X}_k$.
- Sample $\bar{\boldsymbol{m}}$:

– For $(j,k) \in \{1,...,L\}^2$, set $m_{jk} = 0$ and $c = 0$. For $i = 1,...,n_{jk}$, sample a temporary variable $trial \sim Bernoulli(\frac{\alpha_0\beta_k + \kappa\delta_{jk}}{c + \alpha_0\beta_k + \kappa\delta_{jk}})$. If $trial = 1$, increase m_{jk}. Increase c.

– For $j \in 1,...,K$, sample a temporary variable $\omega_j \sim Binomial(m_{jj}, \frac{\eta}{\eta + \beta_j(1-\eta)})$, where $\eta = \frac{\kappa}{\alpha_0 + \kappa}$. Set $\bar{m}_{jk}$ to $\bar{m}_{jk} = \begin{cases} m_{jk} & \text{if } j \neq k, \\ m_{jj} - \omega_j & \text{if } j = k. \end{cases}$

- Sample the global transition distribution $\boldsymbol{\beta}$ as defined by (10).
- For $k = 0,...,L$, sample the transition probabilities $\boldsymbol{\pi}_k$ as defined by (10).
- For $k = 1,...,L$, sample each emission parameters (μ_k, Σ_k) as defined by (9).
- Optionally, sample the hyperparameters α_0, κ, and γ as described in Fox et al. [6].
- Set $\boldsymbol{\pi}^{new} = \boldsymbol{\pi}$ and $(\boldsymbol{\mu}^{new}, \boldsymbol{\Sigma}^{new}) = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

2.2. Classification

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

$$y^* = \arg\max_{y \in \mathcal{Y}} \{\log p(\boldsymbol{x}|\boldsymbol{\pi}_y, (\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y))\}, \qquad (11)$$

where $\boldsymbol{\pi}_y$ and $(\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y))$ were obtained from the training procedure. The observation likelihood $p(\boldsymbol{x}|\boldsymbol{\pi}_y, (\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y))$ can be compute directly using a forward message passing [8] which we will not describe here.

3. 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.

3.1. Synthetic data

First, we concentrate on an ideal scenario which is similar to the synthetic case discussed in [1]. 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. Respectively for the two activities, training sets are generated from HMMs with transitions:

$$\mathbf{T}^A = \begin{bmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{bmatrix}, \quad \mathbf{T}^B = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}.$$

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

We run 500 iterations using the Gibbs sampler with a truncation level $L = 5$. As expected, For the both classes, the Gibbs sampler converges to the right motion numbers (which is two) after a few iterations and the numbers become stable afterwards. We also check the emission parameters and transition matrices sampled at the $500th$ iteration. For the two activities, they are respectively

$$\tilde{\mu}_1^A = [0.0200\ 0.0000]^T, \tilde{\mu}_1^B = [0.0199\ 0.0001]^T,$$
$$\tilde{\mu}_2^A = [0.0001\ 0.0200]^T, \tilde{\mu}_2^B = [0.0000\ 0.0200]^T,$$
$$\tilde{\mathbf{T}}^A = \begin{bmatrix} 0.9650 & 0.0350 \\ 0.0602 & 0.9398 \end{bmatrix}, \tilde{\mathbf{T}}^B = \begin{bmatrix} 0.5345 & 0.4655 \\ 0.5191 & 0.4809 \end{bmatrix}.$$

As we can see, the estimated emission parameters and transition matrices are very close to the true setting.

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.

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

Fig. 1. 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.

3.2. Two real-world scenes

We then consider HAR under two real-world scenes, which include a shopping center and a university campus [1]. For the shopping center scene, four classes of activities are predefined, which are "entering" (E), "leaving" (L), "passing" (P), and "browsing" (B). For the university campus scene, seven classes of activities are predefined, which are "entering" (E), "leaving" (L), "crossing park up" (CPU), "crossing

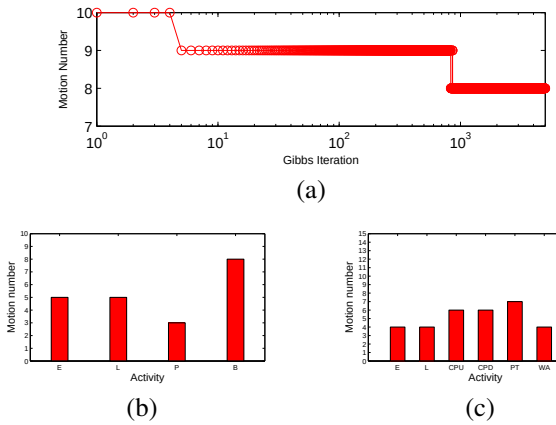

Fig. 2. Experimental results about motion numbers. (a) Updates of the motion number through the iterations with the shopping data ("browsing"). (b) Average motion numbers for the shopping center scene. (c) Average motion numbers for the campus scene.

True class	E	L	P	B
E	93.80	6.20	0	0
L	0	97.67	0	2.33
P	2.31	0	97.69	0
B	2.17	0	0	97.83

Table 1. Classification accuracies (%) of the shopping center scene obtained in our experiment.

park down" (CPD), "passing through" (PT), "walking along" (WA), and "wandering" (W). After processing, we get 53 trajectories in the shopping scene and 143 trajectories in the campus scene. Fig. 1 shows the two scenes with example trajectories.

The SD-HMM approach [1] is a recent state-of-the-art 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 run 5000 iterations on the shopping training set with the truncation level $L = 10$. Fig. 2(a) shows the updates of the motion number throughout the iterations in the case of "B". As we can see, the Gibbs sampler finally converges to a stable motion number. Due to space constraints, we do not plot results of other classes of activities, which are similar to Fig. 2(a). To evaluate the classification accuracy, we randomly select 100 sets of trained parameters between the $4000th$ and $5000th$ iteration. Fig. 2(b) shows the average motion numbers of the four classes with the 100 iterations. As we can see, "B" has the largest number. A reasonable

True class	E	L	CPU	CPD	PT	WA	W
E	97.69	0	0	0	0	2.31	0
L	1.40	97.87	0	0	0	0.73	0
CPU	0	0	97.14	0	0	0	2.86
CPD	0	0	4.59	94.68	0	0.73	0
PT	0	1.64	0	0	98.36	0	0
WA	0	0	0	4.42	0	95.58	0
W	1.17	0	0	0	0	0	98.83

Table 2. Classification accuracies (%) of the campus scene obtained in our experiment.

Approach	Shopping	Campus
HMM + AIC	16.67	8.55
HMM+BIC/MDL	5.56	7.24
SD-HMM	3.70	3.29
PROPOSED APPROACH	3.24	3.38

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

explanation is that the "B" activity is the most flexible and diverse one since people have no specific purpose when they browse in front of the shop. Similarly, we run 5000 iterations using the campus training set with the truncation level $L = 15$ and randomly select 100 sets of trained parameters between the $4000th$ and $5000th$ iteration to evaluate the performance. The average motion numbers of the seven classes with the 100 iterations are shown in Fig. 2(c). As we can see, "E" and "L" have the same number of motions, which is also true for "CPU" and "CPD". The reason may be that they are essentially similar activities though with opposite directions (see Fig. 1). Furthermore, "W" has the largest number of motions, which corresponds to the result of "B" in the previous shopping center scene. The confusion matrices for the shopping center scene and the campus scene are respectively given in Table 1 and Table 2. As we can see, our approach achieves a good performance for all classes of activities, even in the cases of complicated activities like "B" and "W".

To illustrate the general performance, we show the overall error rate of our method in Table 3. Moreover, the results of other methods from [1] are employed as a comparison. As we can see, our method either outperforms (on the shopping data) or closely matches (on the campus data) the SD-HMM.

4. CONCLUSION

In this paper, we have presented a method based on the HDP-HMM framework for modeling and recognizing human activities. We model the distributions of displacements in a trajectory as Gaussians and the temporal evolution of invisible motions as a Markov chain. By adopting an approximation and extension to the standard HDP-HMM, our method can infer an appropriate number of motions from data. 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.

5. REFERENCES

[1] J.C. Nascimento, A.T. Figueiredo, and J.S. Marques, "Trajectory classification using switched dynamical hidden Markov models," *IEEE Transactions on Image Processing*, vol. 19, no. 5, pp. 1338–1348, 2010.

[2] N. Vaswani, A.R. Chowdhury, and R. Chellappa, "'Shape activity': a continuous-state HMM for moving/deforming shapes with application to abnormal activity detection," *IEEE Transactions on Image Processing*, vol. 14, no. 10, pp. 1603–1616, 2005.

[3] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas, "Conditional models for contextual human motion recognition," in *Proceedings of the International Conference on Computer Vision*. IEEE, 2005, pp. 1808–1815.

[4] Q. Gao and S. Sun, "Trajectory-based human activity recognition using hidden conditional random fields," in *Proceedings of the International Conference on Machine Learning and Cybernetics*. IEEE, 2012, vol. 3, pp. 1091–1097.

[5] Y.W. Teh, M.I. Jordan, M.J. Beal, and D.M. Blei, "Hierarchical Dirichlet processes," *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1566–1581, 2006.

[6] 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*, vol. 5, no. 2A, pp. 1020–1056, 2011.

[7] J.V. Gael, Y. Saatci, Y.W. Teh, and Z. Ghahramani, "Beam sampling for the infinite hidden Markov model," in *Proceedings of the International Conference on Machine Learning*. IMLS, 2008, vol. 25, pp. 1088–1095.

[8] C.M. Bishop, *Pattern Recognition and Machine Learning*, Springer Science, Spring Street, New York, USA, 2006.