entropy

MDPI

Article

Variational Beta Process Hidden Markov Models with Shared Hidden States for Trajectory Recognition

Jing Zhao, Yi Zhang, Shiliang Sun * and Haiwei Dai

School of Computer Science and Technology, East China Normal University, Shanghai 200062, China; jzhao@cs.ecnu.edu.cn (J.Z.); 51194506051@stu.ecnu.edu.cn (Y.Z.); 51131201017@stu.ecnu.edu.cn (H.D.)
* Correspondence: slsun@cs.ecnu.edu.cn

Abstract: Hidden Markov model (HMM) is a vital model for trajectory recognition. As the number of hidden states in HMM is important and hard to be determined, many nonparametric methods like hierarchical Dirichlet process HMMs and Beta process HMMs (BP-HMMs) have been proposed to determine it automatically. Among these methods, the sampled BP-HMM models the shared information among different classes, which has been proved to be effective in several trajectory recognition scenes. However, the existing BP-HMM maintains a state transition probability matrix for each trajectory, which is inconvenient for classification. Furthermore, the approximate inference of the BP-HMM is based on sampling methods, which usually takes a long time to converge. To develop an efficient nonparametric sequential model that can capture cross-class shared information for trajectory recognition, we propose a novel variational BP-HMM model, in which the hidden states can be shared among different classes and each class chooses its own hidden states and maintains a unified transition probability matrix. In addition, we derive a variational inference method for the proposed model, which is more efficient than sampling-based methods. Experimental results on a synthetic dataset and two real-world datasets show that compared with the sampled BP-HMM and other related models, the variational BP-HMM has better performance in trajectory recognition.

Keywords: hidden Markov models; variational inference; trajectory recognition; Beta process

check for updates

Citation: Zhao, J.; Zhang, Y.; Sun, S.; Dai, H. Variational Beta Process Hidden Markov Models with Shared Hidden States for Trajectory Recognition. *Entropy* **2021**, *23*, 1290. https://doi.org/10.3390/e23101290

Academic Editor: Udo Von Toussaint

Received: 9 August 2021
Accepted: 28 September 2021
Published: 30 September 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Trajectory recognition is important and meaningful in many practical applications, such as human activities recognition [1], speech recognition [2], handwritten character recognition [3] and navigation task with mobile robot [4]. In most practical applications, the trajectory is affected by the hidden features corresponding to each point. The hidden Markov model (HMM) [2], hierarchical conditional random field (HCRF) [5,6] and the HMM-based models, such as the hierarchical Dirichlet process hidden Markov model (HDP-HMM) [7], the Beta process hidden Markov model (BP-HMM) [8–10] and the Gaussian mixture model hidden Markov model (GMM-HMM) [2] are used to model sequential data and identify their classes [11–14].

The HMM is a popular model which has been applied widely in human activity recognition [1,15], speech recognition [2,16] and remote target tracking [2,17]. Besides, the HMM is becoming a more significant part as a building block of smart cities and Industry 4.0 [18,19] and implemented in extensive applications such as driving behaviors prediction [20] and the inernet of thing (IoT) signature anomalies [21]. One drawback of the HMM is having to ensure in advance the number of hidden states that need to be selected or cross-validated. To address this problem, several methods based on model selection are employed, such as BIC [22] or some Bayesian non-parameter prior like the BP [23] and the HDP [24]. Besides, directly using the original HMM for classification has another disadvantage, in which each HMM is trained for one class separately and thus information from different classes cannot be shared. It is worth mentioning that the sampled BP-HMM

proposed by Fox et al. [9] can not only learn the number of hidden features automatically but also obtain the sharing features between different classes, which has been proved to be meaningful for human activity trajectory recognition. The sampled BP-HMM learns the shared states among different classes by jointly modeling all trajectories together, in which a hidden state indicator for one trajectory with a BP prior is introduced and thus a state transition matrix for each trajectory is maintained. When used for classification, the sampled BP-HMM calculates the class-specific transition matrix by averaging the transition matrices of the trajectories from the corresponding class. However, from the perspective of performance or efficiency, if the sampled BP-HMM [1,7] is used for classification, there is still a lot of room for improvement.

From the perspective of performance, the classification procedure in the sampled BP-HMM [1] is too rough to make full use of the trained model, in which the state transition matrix for each class is calculated by averaging the transition matrixes of all the trajectories. Obviously, this will lead to the loss of information, especially when the training set has some ambiguous trajectories. For instance, a "running" class has some "jogging" trajectories. One naive method to solve it is to select the K best HMMs for each class. However, it will cost plenty of time to select representatives for each class. In order to take account of both performance and efficiency, we change the way of modeling data in BP-HMMs. Differently from those versions of BP-HMMs [1,8–10,25], in variational BP-HMMs, an HMM is created for each class instead of for each trajectory.

From the perspective of efficiency, the existing approximate inference for the BP-HMM is based on sampling methods [1,9] which often converge slowly. This drawback of the sampled BP-HMM [1] is inconvenient to practical applications. To provide a faster convergence rate than sampling methods, we develop variational inference for the BP-HMM. If the variational lower bound is unchanged or almost unchanged, the iteration will stop. To be amenable to the variational method, we use the stick-breaking construction of the BP [26] instead of the Indian buffet process (IBP) construction [27] in the sampled BP-HMM.

In this paper, we propose a variational BP-HMM for trajectory recognition, in which the way of the data modeling and the inference method are novel compared with the previous sampled BP-HMM. On the one hand, the new method of modeling trajectories enables the model to obtain better classification performance. Specifically, the hidden state can be optionally shared, and the class-specific state indicator is more suitable for classification than the trajectory-specific state indicator in the sampled BP-HMM. The transition matrix is actually learned from the data instead of averaging all the trajectory-specific transitions. On the other hand, the derived variational inference of the BP-HMM makes the model more efficient. In particular, we use the two-parameter BP as the prior of the class-specific state indicator, which is more flexible than the one-parameter Indian buffet process in the sampled BP-HMM. We apply our model to the navigation task of mobile robots and human activity trajectory recognition. Experimental results on the synthetic and real-world data show that the proposed variational BP-HMM with sharing hidden states has advantages to trajectory recognition.

The remainder of this paper is organized as follows. Section 2 gives an overview of the BP and HMM. In Section 3, we review the model assumption of the sampled BP-HMM. In Section 4, we present the proposed variational BP-HMM including the model setting and its variational inference procedure. Experimental results on both synthetic and real-world datasets are reported in Section 5. Finally, Section 6 gives the conclusion and future research directions.

2. Preliminary Knowledge

In order to explain the variational BP-HMM more clearly, the key related backgrounds including BP and HMM will be introduced in the following sub-sections.

2.1. Beta Process

The BP is defined by Hjort [28] for applications in survival analysis. It is a significant application as a non-parametric prior for latent factor models [23,26], and used as a non-parameter prior for selecting the hidden state set of the HMM [8,9,25]. At the beginning, the BP is defined on the positive real line $(\mathbb{R}^+)$ then extended to more general spaces Ω (e.g., $\mathbb{R}$).

A BP, $B \sim \mathrm{BP}(\alpha, B_0)$, is a positive Lévy process. Here, α is the concentration parameter and B_0 is a fixed measure on Ω. Let $\gamma = B_0(\Omega)$. The $\mathrm{BP}(\alpha, B_0)$ is formulated as

$$\begin{aligned} B_K &= \sum_{k=1}^{\infty} \pi_k \delta_{\omega_k}, \\ \omega_{i_j} &\overset{i.i.d.}{\sim} \frac{1}{\gamma} B_0, \end{aligned} \tag{1}$$

where $\{\omega\}$ are atoms in B. If B_0 is continuous, the Lévy measure of the BP is expressed as

$$\nu(d\omega, d\pi) = \alpha(\omega)\pi^{-1}(1-\pi)^{c(\omega)-1} d\pi B_0(d\omega). \tag{2}$$

If B_0 is discrete, in the form of $B_0 = \sum_k q_k \omega_k$, the atoms in B and B_0 have the same location. It can be represented as follows

$$\begin{aligned} B_K &= \sum_{k=1}^{K} \pi_k \delta_{\omega_k}, \\ \pi_k &\overset{i.i.d.}{\sim} \mathrm{Beta}\Big(\frac{\alpha\gamma}{K}, \alpha\Big(1-\frac{\gamma}{K}\Big)\Big), \\ \omega_k &\overset{i.i.d.}{\sim} \frac{1}{\gamma} B_0. \end{aligned} \tag{3}$$

As $K \to \infty$ and $H_K \to \infty$, B represents a BP [29].

The BP is conjugate to a class of Bernoulli process, denoted by $\mathrm{BeP}(B)$. For example, we define a Bernoulli process $F \sim \mathrm{BeP}(B)$. In this article, we focus on the discrete Bernoulli process in the form of $B = \sum_k \pi_k \delta_{\omega_k}$, and then the Bernoulli process can be expressed as $F = \sum_k b_k \delta_{\omega_k}$, where $B \in [0, 1]$, b_k is the independent Bernoulli variable with the probability π_k. If B is a BP, then

$$\begin{aligned} B &\sim \mathrm{BP}(\alpha, B_0), \\ F &\sim \mathrm{BeP}(B), \end{aligned} \tag{4}$$

is called the Beta-Bernoulli process.

Similarly to Dirichlet process which has two principle methods for drawing samples, (1) the Chinese restaurant process [30], (2) the stick-breaking process [31], the BP generates samples using the Indian buffet process (IBP) [23] and the stick-breaking process [29].

The original IBP can be seen as a special case of the general BP, i.e., an IBP is a one-parameter BP. Similarly to the Chinese restaurant process, the IBP is described in the view of customers choosing dishes. It is also employed to construct two-parameter BPs but with some details changed. Specifically, the procedure for constructing $\mathrm{BP}(\alpha, B_0), \gamma = B_0(\Omega)$ is as follows:

1. The first customer takes the first $\mathrm{Poisson}(\gamma)$ dishes.
2. The nth customer then takes dishes that have been previously sampled with probability $\frac{m_k}{\alpha+n-1}$, where m_k is the number of people who have already sampled the dish k. He also takes $\mathrm{Poisson}\big(\frac{\alpha\gamma}{\alpha+n-1}\big)$ new dishes.

The BP has been shown as a de Finetti mixing distribution underlying the Indian buffet process, and an algorithm has been presented to generate the BP [23].

The stick-breaking process of the BP, $B \sim \mathrm{BP}(\alpha, B_0)$, is provided by Paisley et al. [29]. It is formulated as follows.

$$\begin{aligned} B &= \sum_{i=1}^{\infty} \sum_{j=1}^{C_i} V_{i_j}^{(i)} \prod_{l}^{i-1} \left(1 - V_{i_j}^{(l)}\right) \delta_{\omega_{i_j}}, \\ C_i &\overset{i.i.d.}{\sim} \mathrm{Poisson}(\gamma), \\ V_{i_j}^{(l)} &\overset{i.i.d.}{\sim} \mathrm{Beta}(1, \alpha), \\ \omega_{i_j} &\overset{i.i.d.}{\sim} \frac{1}{\gamma} B_0. \end{aligned} \tag{5}$$

It is clearly shown from the above equations that in every round (indexed by i), C_i atoms have been selected, the weights of them follow an i-times stick-breaking process in which each breaking has the $\mathrm{Beta}(1, a)$ probability and C_i is drawn from $\mathrm{Poisson}(\gamma)$.

2.2. Hidden Markov Models

The HMM [2] is a state space model where each sequence uses a Markov chain of discrete latent variables, with each observation conditioned on the state of the corresponding latent variables. Obviously, they are appropriate to model the data varying over time, and the data can be considered to be generated by the process that switches between different phases or states at different time-points. The HMM has been proved as a valuable tool in human activity recognition, speech recognition and many other popular areas [32].

Suppose that the trajectory observation $X = \{x_1, \ldots, x_N\}$ is an $N \times d$ matrix and $Z = \{z_1, \ldots, z_N\}$ is a N dimensional latent variable vector which has a value set Ω_1 with size K. The joint distribution of X and Z is expressed as

$$\begin{aligned} p(X, Z \mid \theta) = & p(z_1|\boldsymbol{\pi}) \left\{ \prod_{t=2}^{T} p(z_t|z_{t-1}, \Pi) \right\} \\ & \prod_{t=1}^{T} p(x_t|z_t, \boldsymbol{\phi}), \end{aligned} \tag{6}$$

where $\theta = \{\boldsymbol{\pi}_0, \boldsymbol{\pi}_k, \boldsymbol{\phi}\}$, and A is a $K \times K$ matrix with $\pi_{jk} = p(z_{t+1} = k|z_t = j)$, $t = \{1, \ldots, T-1\}, i, j \in \Omega_1$ with $\sum_k \pi_{jk} = 1$, and π_{0k} is a K dimensional vector with $\pi_{0k} = p(z_1 = k), k \in \Omega_1$ with $\sum_k \pi_{0k} = 1$. Furthermore, in the nonparametric version of HMM, the matrix Π can be assumed to obey a Dirichlet distribution, i.e., $\boldsymbol{\pi}_j \sim \mathrm{Dir}(\alpha_1, \alpha_2, \ldots, \alpha_K)$ where $\sum \alpha_k = 1$. The probabilistic graphical model is represented in Figure 1.

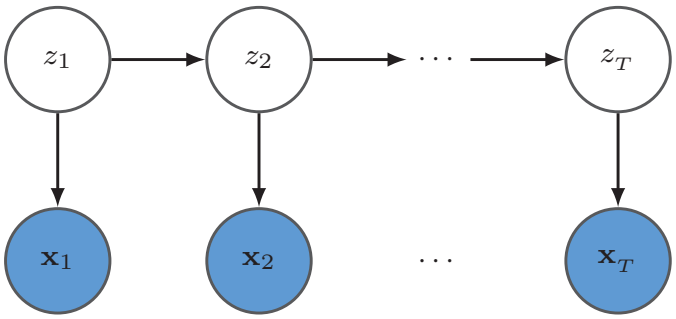


Figure 1. The probabilistic graphical model for an HMM where $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$ represents an observation sequence and $Z = \{z_1, z_2, \ldots, z_T\}$ represents the corresponding hidden state sequence.

If $\mathbf{x}_t$ is discrete with value set Ω_2 in the size of D, ϕ is a $K \times D$ matrix with element $\phi_{i_j} = p(\mathbf{x}_t = j|z_t = i), i \in \Omega_1, j \in \Omega_2$. Π and ϕ are named respectively as the transition matrix and emission matrix. If $\mathbf{x}_t$ is continuous, the emission matrix will be replaced by the emission distribution, where ϕ is often defined as a distribution like Gaussian

distribution $p(\mathbf{x}_t|z_t = k) = \mathcal{N}(\mathbf{x}_t|\boldsymbol{\mu}_k, \Sigma_k), k \in \Omega_1$. In the fully Bayesian framework, μ_k, Σ_k can be regarded as random variables with distribution like normal inverse Wishart or Gaussian with Gamma distribution.

Marginal likelihood is often used to evaluate how an HMM is fit for the trajectories. Therefore, the HMM is usually trained by maximizing the marginal likelihood over the training trajectories. Baum–Welch (BW) algorithm, as an EM method is a famous algorithm for learning parameters of HMMs. The parameters include the transition matrix, the initial state distribution and the emission matrix (distribution's parameters). In the BW algorithm, the forward-backward algorithm is employed to calculate the marginal probability. It should be noted that since the BW algorithm can only find the local optimum, multiple initializations are usually used to obtain better solutions. Given the learned parameters, the most likely state sequence corresponding to a trajectory is required in many practical applications. Viterbi algorithm is an effective method to obtain the most probable state sequence.

HMMs are a kind of generative model; they model the distribution of all the observed data. In trajectory classification tasks, such as activity trajectory recognition, different HMMs are used to model different classes of trajectories separately. After training these HMMs, the parameters in different HMMs are used to evaluate the newly come trajectory to find the most probable class. Specifically, to model large multiple trajectories from different classes, a separate HMM is defined for each class of trajectories, where θ^c represents its parameters. Given the trained HMMs, the class label y^* of a new test trajectory $\mathbf{x}^*$ is determined according to

$$y^* = \arg\max_c \ln p(\mathbf{x}^*|\theta^c), \tag{7}$$

where $p(\mathbf{x}^*|\theta^c)$ can be calculated using the forward-backward algorithm.

3. The Sampled BP-HMM

The sampled BP-HMM [9] is proposed to discover the available hidden states and the sharing patterns among different classes. It jointly models multiple trajectories and learns a state transition matrix for each trajectory. The sampled BP-HMM is successfully applied to trajectory recognition tasks, such as human activity trajectory recognition [1,10].

The sampled BP-HMM uses HMMs to model all the trajectories from all the classes and uses the BP as the prior of the indicator variables with each one corresponding to one trajectory. Suppose $X = \{X^{(1)}, X^{(2)}, \ldots, X^{(n)}, \ldots, X^{(N)}\}, N \in \mathbb{N}^+$ where $X^{(n)}$ is the nth trajectory. Each trajectory is modeled by an HMM. These HMMs share a global hidden feature set Ω with the size of ∞. The sampled BP-HMM uses a hidden state selection matrix F with the size of $N \times \infty$ to indicate the available states for each trajectory, i.e., $f_{nk} = \{0, 1\}$ indicators whether the nth HMM owns the kth state. The prior of the transition matrix $\Pi^{(n)}$ for each trajectory is related to F, The transition matrix of the nth HMM is

$$\boldsymbol{\pi}_j^{(n)} \sim Dir([r, r, \ldots, r + \kappa, r, \ldots] \odot \mathbf{f}_n), j > 1, \tag{8}$$

and the initial state probability vector $\boldsymbol{\pi}_0^{(n)}$ is also related to F,

$$\boldsymbol{\pi}_0^{(n)} \sim Dir([r, r, \ldots, r, r, \ldots] \odot \mathbf{f}_n). \tag{9}$$

Similarly to the standard HMM, the latent variable $\mathbf{z}^{(n)}$ is a discrete sequence with

$$z_1^{(n)} \sim \boldsymbol{\pi}_0^{(n)}, z_{t+1}^{(n)}|z_t^{(n)} \sim \boldsymbol{\pi}_{z_t^{(n)}}^{(n)}, t = 1 \ldots T, \tag{10}$$

and the emission distribution of the nth HMM is

$$\begin{aligned} &X_t^{(n)}|z_t^{(n)} \sim \mathcal{N}\left(\boldsymbol{\mu}_{z_t^n}, \Sigma_{z_t^n}\right), \\ &(\boldsymbol{\mu}_k, \Sigma_k) \sim \mathrm{NIW}(u_0, \lambda_0, \Phi_0, \upsilon_0). \end{aligned} \tag{11}$$

In order to build a non-parameter model, the hidden states selection matrix F is constructed by a BP-BeP.

$$\begin{aligned} B &\sim \mathrm{BP}(\alpha, B_0), \\ f_i|B &\sim \mathrm{BeP}(B). \end{aligned} \tag{12}$$

From the perspective of the characteristic of BPs, we can find that the greater the concentration parameter α, the sparser the hidden state selection matrix F, and greater γ will lead to more hidden features.

Given the above model assumptions, the sampled BP-HMM uses the Gibbs sampling method to train the model and uses the gradient based method to learn the parameters. With the state transition matrix for each trajectory being learned, the average state transition matrix for each class can be calculated by the mean operation. The new test trajectories are classified according to their likelihood probabilities conditional on each class.

4. The Proposed Variational BP-HMM

In this section, we will introduce the proposed variational BP-HMM which has more reasonable assumptions and more efficient inference procedure than the sampled BP-HMM. We first describe key points of our model and present our stick-breaking representation for the BP which allows for variational inference. Then we give the joint distribution of the proposed BP-HMM and the variational inference for the BP-HMM.

4.1. BP-HMM with the Shared Hidden State Space and Class Specific Indicators

As introduced above, the existing sampled BP-HMM can jointly learn the trajectories from different classes by sharing a same hidden state space. It can also automatically determine the available states and the corresponding transition matrices for one trajectory by the introducing state selection matrix F. However, in the sampled BP-HMM, the state transition matrix and initial probabilities are trajectory-specific, and it is not appropriate to perform mean operation on these transition matrices and probabilities to obtain a average matrix and probabilities for each class.

In order to model trajectories from different classes more reasonably, we introduce a shared hidden state space and class-specific indicators. We define a state selection vector $\mathbf{f}_c$ for each class which are used to distinguish the differences between classes and define state initial probabilities $\boldsymbol{\pi}_0$ and transition matrix $\boldsymbol{\pi}_j$ for each class which are used to capture the commonness with one class. The transition matrix of the cth class from state j is

$$\boldsymbol{\pi}_j^{(c)} \sim Dir([r, r, \ldots, r+\kappa, r, \ldots] \odot \mathbf{f}_c), j > 0, \tag{13}$$

and the initial state probability vector $\boldsymbol{\pi}_0^{(c)}$ is also related to F,

$$\boldsymbol{\pi}_0^{(c)} \sim Dir([r, r, \ldots, r, r, \ldots] \odot \mathbf{f}_c). \tag{14}$$

Similarly to the standard HMM, the latent variable $\mathbf{z}^{(n)}$ for the nth trajectory is a discrete sequence with

$$z_1^{(n)} \sim \boldsymbol{\pi}_0^{(y_n)}, \quad z_{t+1}^{(n)}|z_t^{(n)} \sim \boldsymbol{\pi}_{z_t^{(n)}}^{(y_n)}, \quad t = 1, \ldots, T. \tag{15}$$

where y_n denotes the class of nth trajectory.

From the way of modeling, the proposed new version of the BP-HMM is different from the sampled BP-HMM [1] which learns an HMM for each trajectory, and it is also different from the traditional HMMs which learn an HMM for each class separately. The proposed BP-HMM can use all the sequences from different classes to jointly train a whole BP-HMM with each HMM corresponding to one class. Therefore, the proposed BP-HMM can better model the trajectories from multiple classes and can further make better classification.

4.2. A Simpler Representation for Beta Process

Besides the model assumption, the proposed variational BP-HMM has different representation of the BP. As introduced in Section 2, the IBP construction of the BP describes the process by conditional distributions. This kind of representation is only suitable for sampling methods which are similar to the Chinese restaurant construction of DPs. Therefore, different from the sampled BP-HMM which uses the IBP construction for the BP to lend it to a Gibbs sampler, we use the stick-breaking construction for the BP to adapt to variational inference. There is some work in constructing stick-breaking representation of BPs for variational inference. The stick-breaking construction is used for the IBP which is closely related to the BP and can be seen as a one-parameter BP [26]. The two-parameter BP is also constructed through stick-breaking processes to server for variational inference [29]. Recently, a simpler representation of the two-parameter BP based on stick-breaking construction is developed to make simpler variational inference [33]. In order to approximate posterior inference to the BP with variational Bayesian method more easily, we refer to the simpler representation of the BP [33]. Let d_k mark the round in which the kth atom appears. That is,

$$d_k = 1 + \sum_{i=1}^{\infty} \delta\left(\sum_{j=1}^{i} C_j < k\right). \tag{16}$$

Note $\delta(\cdot)$ is a binary indicator and it equals to 1 if the formula is true. Using the latent indicators, the representation of B in (6) is simplified as

$$B = \sum_{k=1}^{\infty} V_{k_{d_k}} \prod_{l=1}^{d_k-1} (1 - V_{k_l}) \delta_{\omega_k}, \tag{17}$$

with ω and V drawn as before.

Let $T_k = -\sum_{l<d_k} \ln(1 - V_{k_l})$. Since each individual term $-\ln(1 - V_{k_l}) \overset{iid}{\sim} \text{Exponential}(\alpha)$, it follows that $T_k \overset{iid}{\sim} \text{Gamma}(d_k - 1, \alpha)$. This gives the following representations of the BP,

$$\begin{aligned} &B = \sum_{k=1}^{\infty} V_k e^{-T_k} \delta_{\omega_k}, \\ &V_k \overset{i.i.d.}{\sim} \text{Beta}(1, \alpha), \\ &T_k \sim \text{Gamma}(d_k - 1, \alpha), \\ &\sum_{k=1}^{\infty} 1(d_k = r) \overset{i.i.d.}{\sim} \text{Poisson}(\gamma), r \in \mathbb{N}^+, \\ &\omega_k \overset{i.i.d.}{\sim} \frac{B_0}{\gamma}. \end{aligned} \tag{18}$$

Here we should notice that each d_k does not have a distribution, but the cardinality of $\{d_k = r\}$ is drawn by Poisson(γ). In addition, $T_k = 0$ with probability one when $d_k = 1$. In this BP, the atom $\omega_k = \{\boldsymbol{\mu}_k, \Sigma_k\}$ and Gamma priors with hyper-parameters $\{a_1, a_2\}$, $\{b_1, b_2\}$ are given to α and γ:

$$\begin{aligned} &\alpha \sim \text{Gamma}(a_1, a_2), \\ &\gamma \sim \text{Gamma}(b_1, b_2). \end{aligned} \tag{19}$$

4.3. Joint Distribution of the Proposed BP-HMM

Assume that the total class number is C and the trajectory number is N. Let X represent the data, $W = \{\alpha, \gamma, \{\boldsymbol{\mu}_k, \Sigma_k\}, \{d_k\}, \{V_k\}, \{T_k\}, \{f_{ck}\}, \{\boldsymbol{\pi}_k^{(c)}\}, Z\}$ represents the set of all latent variables in the model, including θ which is the set of all the hyper-parameters, and Y is the set of all the class labels.

The probabilistic graphical model is shown in Figure 2, where its joint likelihood is

$$p(X, W|\theta) = p(X|W, \theta) \times p(W|\theta). \tag{20}$$

The likelihood $p(X|W, \theta)$ is defined as a multi-normal distribution by

$$p(X|W, \theta) = \prod_{n=1}^{N} \prod_{t=1}^{T} \prod_{k=1}^{K} \mathcal{N}(\mathbf{x}_t^{(n)}|\boldsymbol{\mu}_k, \Sigma_k)^{\delta(z_t=k)}. \tag{21}$$

The prior distribution of the parameter W and detailed setup are expressed in Appendix A.

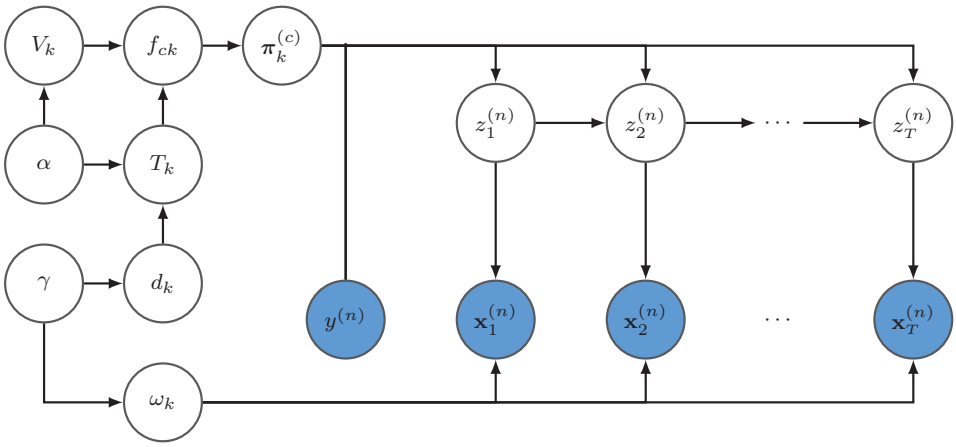


Figure 2. This is the probabilistic graphical model of the proposed variational BP-HMM. $X^{(n)} = \{\mathbf{x}_1^{(n)}, \mathbf{x}_2^{(n)}, \ldots, \mathbf{x}_T^{(n)}\}$ is the nth observed trajectory, $\mathbf{z}^{(n)} = \{z_1^{(n)}, z_2^{(n)}, \ldots, z_T^{(n)}\}$ is the hidden state sequence of the nth trajectory, and $y^{(n)}$ is the class label of the nth trajectory which indicates choosing the state transition probabilities from the class it belongs to. In this graphical model, we omit the hyper-parameters.

4.4. Variational Inference for the Proposed BP-HMM

We use a factorized variational distribution over all the latent variables to approximate the intractable posterior $p(W|X, \theta)$. Two truncations are set in the inference: one is truncation of the number of hidden states at K and the other is the truncation of the round number at R. Specifically, we assume the variational distribution as

$$\begin{aligned} Q = & q(\alpha)q(\gamma) \prod_{k=1}^{K} \Big\{ q(\boldsymbol{\mu}_k, \Sigma_k) q(d_k) q(V_k) q(T_k) \\ & \times \prod_{c=1}^{C} q(f_{ck}) q(\boldsymbol{\pi}_k^{(c)}) \Big\} \prod_{c=1}^{C} q(\boldsymbol{\pi}_0^{(c)}) \prod_{n=1}^{N} q(\boldsymbol{z}^{(n)}), \end{aligned} \tag{22}$$

where

$$
\begin{aligned}
&q(\alpha) = \text{Gamma}(\alpha|k_1, k_2),\\
&q(\gamma) = \text{Gamma}(\gamma|\tau_1, \tau_2),\\
&q(\boldsymbol{\mu}_k, \Sigma_k) = \text{NIW}(\boldsymbol{\mu}_k, \Sigma_k|u_k, \lambda_k, \Phi_k, \upsilon_k),\\
&q(d_k) = \text{MultiNomial}(d_k|\boldsymbol{\varphi}_k),\\
&q(V_k) = \text{Beta}\left(V_k|\tau_{k_1}, \tau_{k_2}\right),\\
&q(T_k) = \text{Gamma}\left(T_k|u'_k, v'_k\right),\\
&q(f_{ck}) = \text{Bernoulli}(f_{ck}|v_{ck}),\\
&q\left(\boldsymbol{\pi}_k^{(c)}\right) = \text{Dir}\left(\boldsymbol{\pi}_k^{(c)}|{r'_{k1}}^{(c)}, {r'_{k2}}^{(c)}, \ldots, {r'_{kK}}^{(c)}\right),\\
&q\left(\boldsymbol{z}^{(n)}|y_n\right) = \prod_{t=1}^{T}\prod_{k_1=1}^{K}\prod_{k_2=1}^{K} {a_{k_1k_2}^{(y_n)}}^{\delta\left(z_t^{(n)}=k_1, z_{t+1}^{(n)}=k_2\right)}\\
&\times \prod_{k=1}^{K} {a_{0k}^{(y_n)}}^{\delta\left(z_0^{(n)}=k\right)} \prod_{t=1}^{T}\prod_{k=1}^{K} {b_{tk}^{(y_n)}}^{\delta\left(Z_t^{(n)}=k\right)}.
\end{aligned}
$$

It is obvious that V_k and T_k do not have conjugate posterior. Thus the distributions are selected for better accuracy and more convenience. Here a^*_{0k} is an estimation of the probability of the initial state distribution, $a^*_{j_1j_2}$, where $j_1 > 0$ and $j_2 > 0$ is an estimation of the probability of transition from state j_1 to j_2 and b^*t_j is an estimation of the emission probability density given the system in state j at time point t. In order to simplify our representation, we do not use sub-index. Here $a_i = \{a_{ij}\}$, $j = 1, \ldots, K$. Let ϕ be the set of variational parameters. We expand the lower bound as $\mathcal{L}(X, \phi) = \mathbb{E}_Q(\ln P(X, W|\theta)) - \mathbb{E}_Q[\ln Q]$ which is expressed in detail in Appendix B.

4.5. Parameter Update

In the framework of variational mean field approximation, the parameters of some variational distributions can be analytically solved using

$$\ln q(\mathbf{w}_j) = \mathbb{E}_{q(W \neq \mathbf{w}_j)}[\ln p(X, W|\theta)] + \text{const.} \tag{23}$$

However, in some cases that the prior distribution and posterior distribution over one latent variable are not conjugate, the variational distribution over this variable cannot have an analytical solution. The parameters of this variational distribution should be optimized through gradient based methods with the variational lower bound being the objective.

In our model, the variational distributions $q(\alpha)$, $q(\gamma)$, $q(\boldsymbol{\mu}_k, \Sigma_k)$, $q(d_k)$, $q(\boldsymbol{\pi}_k)$, $q(Z)$ have a closed form solution, and we can get their parameter update formulas according to (23). While the variational distributions $q(V_k)$, $q(T_k)$, $q(f_{ck})$ cannot be analytically solved, we can update their parameters by corresponding gradients. Next, we give the way of calculating variational distributions and show the procedure for training the variational BP-HMM in Algorithm 1. The detailed parameters update formulas or the gradients with respect to the parameters are presented in Appendix C.

Algorithm 1 Variational Inference for the Proposed BP-HMM

Initialize θ and ϕ.
Given R and threshold and Initialize RunTime = 0;
while $|\mathcal{L} - \mathcal{L}_{\mathbf{old}}| <$ threshold or RunTime $<$ R **do**
  $\mathcal{L}_{\mathbf{old}} = \mathcal{L}$
  for each trajectory n **do**
    Update $q(\mathbf{z}^{(n)})$
    Calculate $q(z_t^{(n)} = k)$ and $q(z_t^{(n)} = k_1, z_{t+1}^{(n)} = k_2)$
  end for
  for each class c **do**
    Update each $q(\boldsymbol{\pi}_k^{(c)}), k = 0, \ldots, K$
    Update each $q(f_{ck}), k = 1, \ldots, K$
  end for
  for each $k = 1, \ldots, K$ **do**
    Update $q(\boldsymbol{\mu}_k, \Sigma_k)$, $q(d_k)$, $q(T_k)$, $q(V_k)$
  end for
  Update $q(\alpha)$, $q(\gamma)$
  Calculate $\mathcal{L}$
end while

4.5.1. Calculation for $q(\alpha)$, $q(\gamma)$, $q(\boldsymbol{\mu}_k, \Sigma_k)$, $q(d_k)$, $q(\boldsymbol{\pi}_k^{(c)})$, $q(Z)$

$$\begin{aligned}
\ln q(\alpha) &= \mathbb{E}_q[\ln p(\alpha) + \sum_{k=1}^{K} \ln p(V_k|\alpha) + \ln p(T_k|d_k, \alpha)], \\
\ln q(\gamma) &= \mathbb{E}_q[\ln p(\gamma) + \sum_{k=1}^{K} \ln p(d_k|\gamma)], \\
\ln q(\boldsymbol{\mu}_k, \Sigma_k) &= \mathbb{E}_q[\ln p(\boldsymbol{\mu}_k, \Sigma_k|\theta) \\
&\quad + \sum_{n=1}^{N} \sum_{t=1}^{T} p(\mathbf{x}^{(n)}|\mathbf{z}^{(n)}, \boldsymbol{\mu}_k, \Sigma_k)], \\
\ln q(d_k) &= \mathbb{E}_q[\ln p(d|\gamma) + \ln p(T_k|d_k, \alpha) \\
&\quad + \sum_{c=1}^{C} \ln p(f_{ck}|V_k, T_k, d_k)], \\
\ln q(\boldsymbol{\pi}_k^{(c)}) &= \mathbb{E}_q[\ln p(\boldsymbol{\pi}_k^{(c)}|f_{ck}, r, \kappa) \\
&\quad + \sum_{n=1}^{N} \sum_{t=1}^{T-1} \delta(y_n = c) \ln p(z_{t+1}^{(n)}|\boldsymbol{\pi}_k^{(c)}, z_t^{(n)} = k)], \\
\ln q(\mathbf{z}^{(n)}) &= \mathbb{E}_q[\ln p(\mathbf{x}^{(n)}|\mathbf{z}^{(n)}, \boldsymbol{\mu}_k, \Sigma_k) \\
&\quad + \sum_{n=1}^{N} \ln p(\mathbf{z}^{(n)}|\Pi^{(y_n)})],
\end{aligned}$$

4.5.2. Optimization for $q(V_k)$, $q(T_k)$, $q(f_{ck})$

The variational parameters of $q(V_k)$, $q(T_k)$, $q(f_{ck})$ include $\{\tau_{k_1}, \tau_{k_2}\}$, $\{u'_k, v'_k\}$, $\{v_{ck}\}$. They are updated by the gradient based method where the gradients of the lower bound $\mathcal{L}$ with respect to these parameters should be calculated.

4.5.3. Remarks

Note that when updating the BP parameters, we should calculate the expectation as

$$\begin{aligned}\mathbb{E}_q[\ln p(f_{ck}|V_k, T_k)] &= v_{ck}\mathbb{E}_q[\ln V_k e^{-T_k}] \\ &\quad +(1-v_{ck})\mathbb{E}_q[1-\ln V_k e^{-T_k}],\end{aligned}$$

of which the second term is intractable. We refer the work in [33] to use a Taylor expansion to $\mathbb{E}_q[\ln(1-V_k e^{-T_k})]$ about the point one,

$$\mathbb{E}_q[\ln(1-V_k e^{-T_k})] = -\sum_{m=1}^{M}\frac{1}{m}(V_k e^{-T_k})^m. \tag{24}$$

For clarity, we define each term $\frac{1}{m}\mathbb{E}\left[\left(V_k e^{-T_k}\right)^m\right]$ in the Taylor expansion using the notation $\triangle_k(m)$ as

$$\begin{aligned}\triangle_k(m) &= \frac{1}{m}\frac{\Gamma(\tau_{k_1}+\tau_{k_2})}{\Gamma(\tau_{k_1}+\tau_{k_2}+m)}\frac{\Gamma(\tau_{k_1}+m)}{\Gamma(\tau_{k_1})}\left(\frac{v'_k}{v'_k+m}\right)^{u'_k} \\ &= \prod_{i=1}^{m}\frac{\tau_{k_1}+i-1}{\tau_{k_1}+\tau_{k_2}+i-1}\left(\frac{v'_k}{v'_k+m}\right)^{u'_k},\end{aligned}$$

and define $\triangle_k(\cdot) = \sum_{m=1}^{M}\triangle_k(m)$. Therefore, $\mathbb{E}_q[\ln(1-V_k e^{-T_k})] = -\triangle_k(\cdot)$.

4.6. Classification

Our model is applicable to trajectory recognition like human activity trajectory recognition. We use the proposed variational BP-HMM to model all the training data from different classes, with each HMM corresponding to a class. Given the learned model with the hyperparameters and variational parameters $\{\theta, \phi\}$, a new test trajectory $\mathbf{x}^*$ can be classified according to its marginal likelihood $p(\mathbf{x}^*|\theta,\phi)$. Denote y^* as the label of the test trajectory; the classification criteria can be expressed as

$$\begin{aligned}y^* &= \arg\max_c \ln p(\mathbf{x}^*|\{\mathbf{a}_k^{*(c)}, u_k, \lambda_k, v_k, \Phi_k\}, \mathbf{a}_0^{*(c)}), \\ &= \arg\max_c \ln\Bigg(\int p(\mathbf{x}^*|\{\boldsymbol{\mu}_k, \Sigma_k\}, \mathbf{z})p(\mathbf{z}|\{\mathbf{a}_k^{*(c)}\}) \\ &\qquad \prod_{k=1}^{K} p(\boldsymbol{\mu}_k, \Sigma_k|u_k, \lambda_k, v_k, \Phi_k)d\mathbf{z}d\boldsymbol{\mu}_k d\Sigma_k\Bigg),\end{aligned} \tag{25}$$

where $a_{jk}^{*(c)}$ is an estimate of the probability of transition from state j to k in the cth class. The likelihood can be calculated through the forward-backward algorithm.

This classification mechanism is more reasonable than the method in [1], as the transition matrix is actually learned.

5. Experiment

To demonstrate the effectiveness of our model on trajectory recognition, we conduct experiments on one synthetic dataset and two real-world datasets; the detailed data statistics are illustrated in Table 1 and the following subsections. We compare our model with HCRF, LSTM [34], HMM-BIC and the sampled BP-HMM. In particular, in HCRF, the number of hidden states is set to 15 and the size of the window is set to 0. In LSTM, we use a recurrent neural network with one hidden layer as its architecture. In HMM-BIC, the state number is selected from the range $[1, 20]$. In the sampled BP-HMM, the hyperparameters are set according to Sun et al. [1]. In the variational BP-HMM, the hyperparameters $\{a_1, a_2, b_1, b_2, r, \kappa\}$ are randomly initialized and selected by maximizing the variational

lower bound, and the emission hyperparameters are initialized with k-means. Particularly, the state truncation parameters in variational BP-HMM are set according to specific datasets, e.g., $K = 7$ for the synthetic data and $K = 20$ for the two real-world data. All experiments are repeated ten times with different training and test division methods, and the average classification accuracy with the standard deviation is reported.

Table 1. Data statistics for the CCP, HATR and WFNT datasets and corresponding classes.

Datasets	#Train Trajectories	#Classes (Descriptions)
CCP	20 (5/class)	4 (Normal, Cyclic, IT, DT)
HATR	300 (50/class)	6 (PTSS, PTES, GA, CPH, WFB, WTS)
WFNT	40 (10/class)	4 (F, L, R, B)

5.1. Synthetic Data

The synthetic data called control chart patterns (CCP) have some quantifiable similarities. They contain four pattern types which can be downloaded from the UCI machine learning repository. The CCP are trajectories that show the level of a machine parameter plotted against time. 400 trajectories are artificially generated by the following four equations [35]:

1. Normal pattern (Normal): $y(t) = m + rs$,
 where $m = 3, s = 2$ and $0 < r < 1$.
2. Cyclic pattern (Cyclic): $y(t) = m + rs + a\text{Sin}(2\pi t/T)$,
 where $0 < a, T < 15$.
3. Increasing trend (IT): $y(t) = m + rs + gt$,
 where $0.2 < g < 0.5$.
4. Decreasing trend (DT): $y(t) = m + rs - gt$,
 where $0.2 < g < 0.5$.

Figure 3 shows the generated synthetic data. In this experiment, 20 trajectories are used for training with 5 trajectories for each class. The classification results are presented in Table 2. The results are obtained through five-fold cross-validation. In order to illustrate that the sharing patterns have been learned by our method, the Hinton diagrams of the variational parameter V are given in Figure 4, where the occurrence probabilities of the hidden states are presented by the sizes of the blocks. For example, we can find that IT and DT share the 4th, 5th, 6th features.

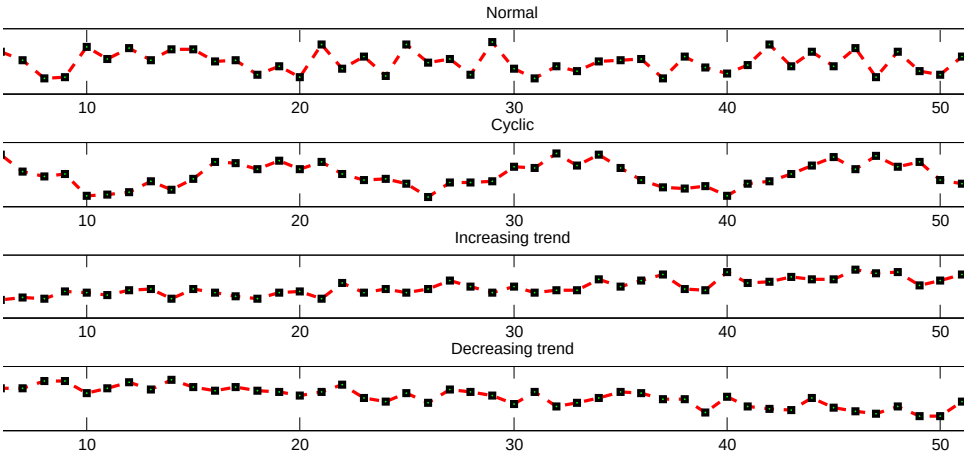


Figure 3. Examples of control chart patterns.

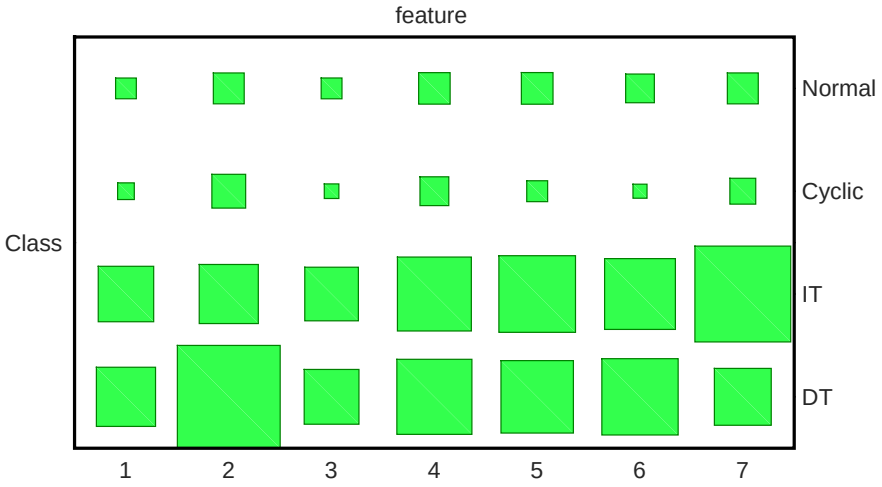


Figure 4. Selection results of hidden states for four classes on control chart patterns; these four classes are normal pattern (Normal), cyclic pattern (Cyclic), increasing trend (IT) and decreasing trend (DT). The occurrence probabilities $q(f_{ck})$ of the hidden states are presented by the sizes of the green blocks. The large size of the green blocks represents high occurrence probability of hidden states.

Table 2. Comparisons of the classification accuracy for the proposed method VBP-HMM versus HCRF, LSTM, HMM-BIC and the sampled BP-HMM in CCP.

	Classification Accuracy				
Approach	**HCRF**	**LSTM**	**HMM-BIC**	**SBP-HMM**	**VBP-HMM**
CCP	0.88 ± 0.03	0.95 ± 0.02	0.97 ± 0.01	0.96 ± 0.02	1.00 ± 0.00

We compare our method with HCRF, LSTM, HMM-BIC and the sampled BP-HMM. As we can see from Table 2, our method outperforms all the other methods.

In this experiment, the sharing patterns contribute to improving the performance. Since an HMM is created for each class of trajectories in our proposed method instead of each trajectory in the sampled BP-HMM, our method has better performance than the sampled BP-HMM.

5.2. Human Activity Trajectory Recognition

Human activity trajectory recognition (HATR) [36] is important in many applications such as health care. In our human activity trajectory recognition experiment, parking lot data are collected from the video [1]. We use the data tagged manually [1], which has 300 trajectories with 50 trajectories for each class. Six classes are defined, which are "passing through south street" (PTSS), "passing through east street" (PTES), "going around" (GA), "crossing park horizontally" (CPH), "wandering in front of building" (WFB) and "walking in top street" (WTS). As seen from [1], the sampled BP-HMM is the best method among the methods including HCRF, LSTM, HMM-BIC and the sampled BP-HMM in HATR. Here we use the same training and test data to compare the variational BP-HMM with the sampled BP-HMM. Table 3 shows the comparisons of the classification accuracy for the proposed method VBP-HMM versus HCRF, LSTM, HMM-BIC and the sampled-BP-HMMs in HATR. The results are obtained through five-fold cross-validation. As can be seen from Table 3, the accuracy of our method is 0.96, while the accuracy of the sampled BP-HMM is 0.91 [1]. The detailed confusion matrix for our method is given in Table 4. The state sharing patterns learned by variational BP-HMM are displayed with the Hinton diagrams in Figure 5, in which GA and CPH, as well as GA and WTS, are more likely to share states. The good performance verifies the superiority of modeling an HMM for each class. Moreover, we take some examples of the correct classification and misclassification

results for visualization as in Figures 6 and 7. As illustrated in Figure 7, the misclassified trajectories often contain some deceptive subpatterns such as the trajectory of CPH in subfigure (d) containing a back turn and a left turn like the GA class.

Table 3. Comparisons of the classification accuracy for the proposed method VBP-HMM versus HCRF, LSTM, HMM-BIC and the sampled BP-HMM in HATR.

	Classification Accuracy				
Approach	**HCRF**	**LSTM**	**HMM-BIC**	**SBP-HMM**	**VBP-HMM**
HATR	0.68 ± 0.03	0.75 ± 0.03	0.95 ± 0.02	0.91 ± 0.02	0.96 ± 0.02

Table 4. Classification accuracy for human activity trajectory recognition.

	Classification Accuracy					
Predicted Class	**PTSS**	**PTES**	**GA**	**CPH**	**WFB**	**WTS**
PTSS	1.00	0.00	0.00	0.00	0.00	0.00
PTES	0.00	1.00	0.00	0.00	0.00	0.00
GA	0.00	0.00	0.97	0.00	0.00	0.00
CPH	0.00	0.00	0.03	0.96	0.00	0.00
WFB	0.00	0.00	0.00	0.03	1.00	0.23
WTS	0.00	0.00	0.00	0.01	0.00	0.77

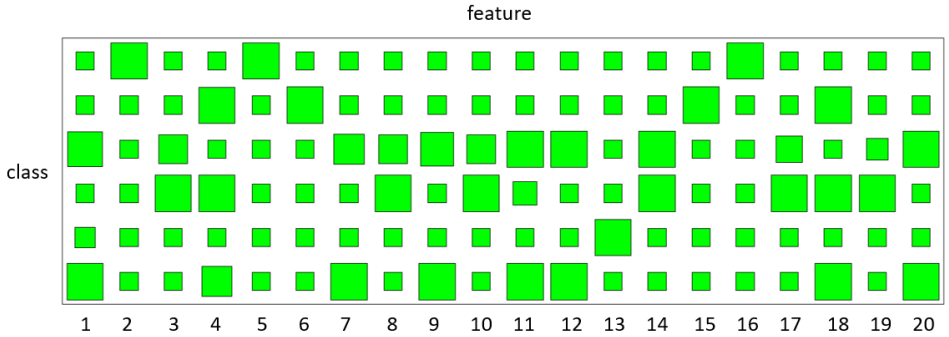
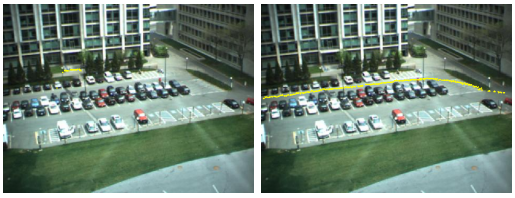


Figure 5. Selection results of hidden states for different classes on human activity trajectory recognition. The occurrence probabilities $q(f_{ck})$ of the hidden states are presented by the sizes of the green blocks. The large size of the green blocks represents high occurrence probability of hidden states.

5.3. Wall-Following Navigation Task

We perform the Wall-Following navigation task (WFNT) in which data are collected from the sensors on the mobile robot SCITOS-G5 [4]. We think that this task is a trajectory with historical data, and two ultrasound sensors datasets are selected, because the cost is as low as possible in civil applications with acceptable accuracy. There are 187 trajectories in the data and four classes need to be recognized, which are "front distance" (F), "left distance" (L), "right distance" (R) and "back distance" (B). We randomly select 40 training trajectories with 10 for each class. The confusion matrix of classification is shown in Table 5 and the state sharing patterns learned by variational BP-HMM are displayed with the Hinton diagrams in Figure 8, where R and F, as well as R and B, have a small number of shared states.

(**a**) Right: PTSS (**b**) Right: PTES (**c**) Right: GA (**d**) Right: CPH



(**e**) Right: WFB (**f**) Right: WTS

Figure 6. Correct classification results of HATR dataset for the classes: PTSS, PTES, GA, CPH, WFB, WTS, respectively.

(**a**) Wrong: CPH→GA (**b**) Wrong: WFB→WTS (**c**) Wrong: WTS→CPH

Figure 7. Misclassification results of HATR dataset for the three classes: CPH, WFB and WTS which are misclassified to GA, WTS and CPH, respectively.

Table 5. The confusion matrix of the Wall-Following navigation recognition.

	Classification Accuracy			
Predicted Class	**F**	**L**	**R**	**B**
F	0.95	0.18	0.00	0.00
L	0.02	0.73	0.00	0.00
R	0.03	0.09	0.95	0.0
B	0.00	0.00	0.05	1.00

The comparison of the classification accuracy for our method VBP-HMM versus HCRF, LSTM, HMM-BIC and the sampled BP-HMM is shown in Table 6. The results are obtained by five-fold cross-validation. It is obvious that our method is much better than the sampled BP-HMM, because we create an HMM for each class of trajectories rather than create an HMM for each trajectory. Although the sharing patterns are not obvious in this experiment, our method has better performance than the other methods. As we have analyzed, sharing patterns among different classes will be learned automatically by our model, which helps to localize precisely the difference of different classes. When there is no sharing pattern among classes, the advantage will be weakened.

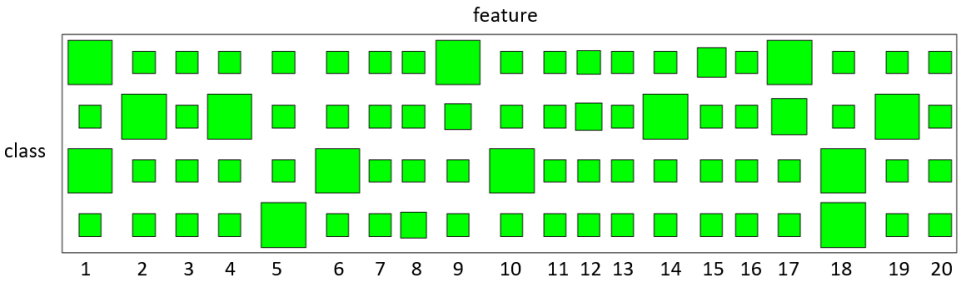


Figure 8. Selection results of hidden states for different classes on the Wall-Following navigation recognition. The occurrence probabilities $q(f_{ck})$ of the hidden states are presented by the sizes of the green blocks. The large size of the green blocks represents high occurrence probability of hidden states.

Table 6. Comparisons of the classification accuracy for the proposed method VBP-HMM versus HCRF, LSTM, HMM-BIC and the sampled BP-HMM in WFNT.

	Classification Accuracy				
Approach	**HCRF**	**LSTM**	**HMM-BIC**	**SBP-HMM**	**VBP-HMM**
WFNT	0.80 ± 0.03	0.73 ± 0.08	0.86 ± 0.04	0.85 ± 0.02	0.89 ± 0.01

5.4. Performance Analysis

In our experiments, the results show that the proposed variational BP-HMM has a great improvement compared to the sampled BP-HMM which uses average transition over trajectories from each class. We analyze the advantages of variational BP-HMM for the following reasons. Due to the small amount of training data in our experiment, the performance of LSTM is not satisfactory. HMM-BIC finds an optimal state number through model selection but it cannot make use of the shared information among classes, and its performance is the second-best overall. Although the sample BP-HMM can share hidden states among classes, it does not make correct use of the shared information in classification and thus does not gain better results. Our proposed variational BP-HMM constructs a mechanism to learn shared hidden states by introducing state indicator variables and maintains class-specific state transition matrices which are very helpful for classification tasks.

Moreover, we give the total cost time of the variational BP-HMM, HMM-BIC, LSTM, HCRF and the sampled BP-HMM in Table 7, where we can see the variational BP-HMM performs much more efficiently than the sampled BP-HMM. This is attributed to the efficiency of the variational methods. Although the sampled BP-HMM and the variational BP-HMM have similar time complexity, due to the sampling operation, the cost time of the sampled BP-HMM is usually several times that of the variational BP-HMM. In other words, the variational BP-HMM converges much faster than the sampled BP-HMM. Besides, compared with HMM-BIC, it only takes about twice the time to achieve significant performance improvements. Above all, we can conclude that the proposed variational BP-HMM is an effective and efficient method for trajectory recognition.

Table 7. Comparisons of total time cost for the proposed method VBP-HMM versus HCRF, LSTM, HMM-BIC and the sampled-BP-HMM in experiments.

	Total Time Cost (s)				
Approach	**HCRF**	**LSTM**	**HMM-BIC**	**SBP-HMM**	**VBP-HMM**
CCP	196	6	54	2151	117
HATR	118	13	93	2521	205
WFNT	1005	8	115	1819	312

6. Conclusions

In this paper, we have proposed a novel variational BP-HMM for modeling and recognizing trajectories. The proposed variational BP-HMM has shared hidden state space which is used to capture the commonality of the cross-category data and class-specific indicators which are used to distinguish the data from different classes. As a result, in the variational BP-HMM, multiple HMMs are used to model multiple classes of trajectories among which a hidden state space is shared.

The more reasonable assumptions of the proposed model make it more suitable for jointly modeling trajectories over all classes and further making trajectory recognition. Experimental results both on synthetic and real-world data have verified that the proposed variational BP-HMM can find the feature sharing patterns among different classes, which helps to better model trajectories and further improve the classification performance. Moreover, compared with the sampled BP-HMM, the derived variational inference for the proposed BP-HMM can reduce the time cost of the training procedure. The experimental time records also show the efficiency of the proposed variational BP-HMM.

Author Contributions: Conceptualization, J.Z. and S.S.; methodology, J.Z. and S.S.; Software, J.Z., Y.Z. and H.D.; formal analysis, J.Z. and Y.Z.; writing—original draft preparation, J.Z.; writing—review and editing, J.Z., Y.Z. and S.S.; supervision, J.Z. and S.S.; Visualization, Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the NSFC Projects 62006078 and 62076096, the Shanghai Municipal Project 20511100900, Shanghai Knowledge Service Platform Project ZF1213, the Shanghai Chenguang Program under Grant 19CG25, the Open Research Fund of KLATASDS-MOE and the Fundamental Research Funds for the Central Universities.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available in a publicly accessible repository. The data presented in this study are openly available in UCI Machine Learning Repository at http://archive.ics.uci.edu/ml/index.php, accessed on 27 September 2021, reference number [4,35,36].

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. The Prior Distribution of the Parameter W

Denote $\sum_{k=1}^{\infty}\delta(d_k=r)$ as d. The prior distribution of the parameter W is expressed as

$$\begin{aligned} p(W|\theta) &= p(\alpha)p(\gamma)p(d|\gamma) \\ &\times \prod_{k=1}^{\infty}\Big\{p(V_k|\alpha)p(T_k|d_k,\theta)p(\boldsymbol{\mu}_k,\Sigma_k|\theta) \\ &\qquad \times \prod_{c=1}^{C} p\left(f_{c_k}|V_k,T_k,d_k\right)p(\boldsymbol{\pi}_k^{(c)}|\boldsymbol{f}_c,\theta)\Big\} \\ &\times \prod_{c=1}^{C} p(\boldsymbol{\pi}_0^{(c)}|\boldsymbol{f}_c,\theta) \\ &\times \prod_{n=1}^{N}\Big\{\prod_{t=1}^{T}\prod_{k_1=1}^{\infty}\prod_{k_2=1}^{\infty}\left(\pi_{k_1k_2}^{(y_n)}\right)^{\delta\left(z_t^{(n)}=k_1,z_{t+1}^{(n)}=k_2\right)} \\ &\qquad \times \prod_{k=1}^{\infty}\left(\pi_{0k}^{(y_n)}\right)^{\delta\left(z_0^{(n)}=k\right)}\Big\}. \end{aligned} \tag{A1}$$

We use

$$p\left(f_{c_k}|V_k,T_k,d_k\right) = p\left(f_{c_k}|V_k\right)^{\delta(d_k=1)}p\left(f_{c_k}|V_k,T_k\right)^{\delta(d_k>1)}$$

to account for the class in which an atom appears. Two terms $p(T_k|d_k,\alpha)$ and $p(d|\gamma)$ are given by Paisley et al. [33],

$$p(T_k|d_k,\alpha) = \frac{\alpha^{\nu_k(\delta)}}{\prod_{r\geqslant 2}\Gamma(r-1)^{1(d_k=r)}} T_k^{\nu_k(2)} e^{-\alpha T_k \delta(d_k>1)},$$

where $\nu_k(s) = \sum\limits_{r\geqslant 2}(r-s)e^{-\alpha T_k \delta(d_k=r)}$, and

$$p(d|\gamma) = \prod_{r=1}^{\infty} \frac{\gamma \sum\limits_k \delta(d_k=r)}{\sum\limits_k \delta(d_k=r)!} e^{-\gamma\left(\sum\limits_{r'=r}^{\infty}\sum\limits_{k=1}^{\infty}\delta\left(\delta\left(d_k=r'\right)>0\right)\right)}.$$

In $p(T_k|d_k,\alpha)$, the indicator d_k is used for selecting the Gamma prior parameters of T_k. Moreover, the term $p(T_k|d_k,\alpha)$ in (21) will be removed if $d_k = 1$. The binary indicator $\delta\left(\sum_{r'=r}^{\infty}\sum_{k=1}^{\infty}\delta(\delta(d_k=r')>0)\right)$ in $p(d|\gamma)$ means that at least one of the K indexed atoms occur in round r or the round after r.

Appendix B. The Lower Bound $\mathcal{L}(X,\phi)$

We expand the lower bound as $\mathcal{L}(X,\phi) = \mathbb{E}_Q(\ln P(X,W|\theta)) - \mathbb{E}_Q[\ln Q]$ which is expressed as

$$\begin{aligned}
&\mathcal{L}(X,\phi)\\
&= \sum_{n=1}^{N}\Bigg\{\sum_{t=1}^{T}\sum_{k=1}^{K}\mathbb{E}_q[\delta(z_t^{(n)}=k)\ln p(X_t|\boldsymbol{\mu}_k,\Sigma_k,\theta)]\\
&\quad+\sum_{t=1}^{T-1}\sum_{k_1=1}^{K}\sum_{k_2=1}^{K}\mathbb{E}_q[\delta(z_t^{(n)}=k_1,z_{t+1}^{(n)}=k_2)\ln(\boldsymbol{\pi}_k^{(y_n)})]\\
&\quad+\sum_{k=1}^{K}\mathbb{E}_q\left[\delta(z_0^{(n)}=k)\ln\left(\pi_{0k}^{(y_n)}\right)\right]\Bigg\}\\
&+\sum_{c=1}^{C}\Bigg\{\sum_{k=0}^{K}\mathbb{E}_q\left[\ln(p(\boldsymbol{\pi}_k^{(c)}|\boldsymbol{f}_c,\theta))\right]\\
&\quad+\sum_{k=1}^{K}\mathbb{E}_q[\delta(d_k=1)\ln p(f_{ck}|V_k)]\\
&\quad+\sum_{k=1}^{K}\mathbb{E}_q[\delta(d_k>1)\ln p(f_{ck}|V_k,T_k)]\Bigg\}\\
&+\sum_{k=1}^{K}\Bigg\{\mathbb{E}_q\ln p(\boldsymbol{\mu}_k,\Sigma_k|\theta)+\mathbb{E}_q[\ln p(T_k|\alpha,d_k)]\\
&+\mathbb{E}_q[\ln p(V_k|\alpha)]\Bigg\}\\
&+\sum_{r=1}^{\infty}\mathbb{E}_q[\ln p(\Sigma_k\delta(d_k=r)|\gamma)]\\
&+\mathbb{E}_q[\ln p(\alpha)]+\mathbb{E}_q[\ln p(\gamma)]\\
&-\mathbb{E}_Q[\ln Q_{-T}]-\sum_{k=1}^{K}\varphi_k(r>1)\mathbb{E}_q[\ln q(T_k)].
\end{aligned}$$

Note that we multiply the entropy of T_k, $\mathbb{E}_q(T_k)\ln q(T_k)$, by the variational probability $\varphi_k(r>1)$ as done in [33] for keeping the entropy of T_k from blowing up when $\varphi_k(1)\to 1$, where $\varphi_k(r>1) = \sum_{r>1}\varphi_{kr} = \mathbb{E}_q[\delta(d_k>1)]$.

Appendix C. Coordinate Update for the Key Distributions

Appendix C.1. Coordinate Update for $q(f_{ck})$

Since the Dirichlet distribution of π_k, f_{ck} cannot be obtained by analysis directly. The gradient ascent algorithm is used for updating $f_{ck}, c \in \{1, \ldots, C\}$. The derivative of $\mathcal{L}$ with respect to v_{ck} is

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial v_{ck}} = \sum_{j=0}^{K} \Bigg\{ & (r^{(c)} - 1)\left(\Psi\left(r_{jk}^{\prime(c)}\right) - \Psi\Big(\sum_{t=1}^{K} r_{jt}^{\prime(c)}\Big)\right)\delta(j \neq k) \\
& + \left(r^{(c)} + \kappa - 1\right)\left(\Psi\left(r_{jk}^{\prime(c)}\right) - \Psi\Big(\sum_{k=1}^{K} r_{jk}^{\prime(c)}\Big)\right)\delta(j = k) \\
& + \Psi\left(\sum_{t=1}^{K} v_{ct} r^{(c)}\delta(t \neq j) + v_{cj}\left(r^{(c)} + \kappa\right)\delta(t = j)\right) \\
& \times \left(\left(r^{(c)} + \kappa\right)\delta(j = k) + r^{(c)}\delta(j \neq k)\right) \\
& - \Psi\left(v_{ck} r^{(c)}\right) r^{(c)}\delta(j \neq k) \\
& - \Psi\left(v_{ck}\left(r^{(c)} + \kappa\right)\right)(r^{(c)} + \kappa)\delta(j = k)\Bigg\} \\
& + \varphi_k(1)\left(\Psi(\tau_{k_1}) - \Psi(\tau_{k_2})\right) \\
& + \varphi_k(r > 1)\left(\Psi(\tau_{k_1}) - \Psi(\tau_{k_1} + \tau_{k_2}) - \frac{u_k'}{v_k'} + \triangle_k(\cdot)\right) \\
& + \ln v_{ck} - \ln(1 - v_{ck}).
\end{aligned}$$

Appendix C.2. Coordinate Update for $q(d_k)$

The update for each φ_k is given below for $r = 1, \ldots, R$. Let

$$\begin{aligned}
\rho(r) = & (r-1)(\Psi(k_1) - \ln k_2) - \ln\Gamma(r-1) \\
& + (r-2)\left(\Psi(u_k') - \ln v_k'\right).
\end{aligned}$$

If $r = 1$,

$$\varphi_k(1) \propto \exp\left\{ n_{0k}(\Psi(\tau_{k_2}) - \Psi(\tau_{k_1} + \tau_{k_2})) - \xi \sum_{i \neq k} \varphi_i(1) \right\}.$$

If $r > 2$,

$$\begin{aligned}
\varphi_k(r) \propto \exp\Bigg\{ & n_{1k}\left(\Psi(\tau_{k_1}) - \Psi(\tau_{k_1} + \tau_{k_2}) - \frac{u_k'}{v_k'}\right) \\
& - n_{0k}\Delta_k(\cdot) + \mathrm{H}[q(T_k)] + \rho(r) \\
& - \xi \sum_{i \neq k} \varphi_k(r) - \frac{\tau_1}{\tau_2} \sum_{j=2}^{r} \prod_{k' \neq k} \sum_{r'=1}^{j-1} \varphi_{k'}(r') \Bigg\}.
\end{aligned}$$

where $n_{1k} = \sum_{c=1}^{C} v_{ck}$, $n_{0k} = C - n_{1k}$ and $\xi = \sum_{x}^{\infty} x^{-2}\ln(x) \approx 0.9375$.

Appendix C.3. Coordinate Update for $q(V_k)$

We use the gradient ascent algorithm to jointly update (τ_{k_1}, τ_{k_2}) by the gradients of the lower bound with respect to (τ_{k_1}, τ_{k_2}). Let $\lambda_1 = -n_{0_k}\varphi_k(1) - n_{1_k} - \frac{k_1}{k_2} - 1 + \tau_{k_1} + \tau_{k_2}$, $\lambda_2 = -n_{0_k}\varphi_k(r>1)$, $\lambda_3 = n_{1_k} + 1 - \tau_{k_1}$ and $\lambda_4 = n_{0_k}\varphi_k(1) + \frac{k_1}{k_2} - \tau_{k_2}$. The derivatives are

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \tau_{k_1}} =& \lambda_3 \Psi'(\tau_{k_1}) + \lambda_1 \Psi'(\tau_{k_1} + \tau_{k_2}) + \lambda_2 \frac{\partial \Delta_k(\cdot)}{\partial \tau_{k_1}},\\ \frac{\partial \mathcal{L}}{\partial \tau_{k_2}} =& \lambda_4 \Psi'(\tau_{k_2}) + \lambda_1 \Psi'(\tau_{k_1} + \tau_{k_2}) + \lambda_2 \frac{\partial \Delta_k(\cdot)}{\partial \tau_{k_2}}.\end{aligned}$$

Since $\Psi(x)$ can be expanded as $\Psi(x) = -\gamma + \sum_{k=1}^{\infty}\left(\frac{1}{k} + \frac{1}{x+k}\right)$ and its derivative is $\Psi'(x) = \sum_{k=1}^{\infty}(x+k-1)^{-2}$, we can get

$$\begin{aligned}\frac{\partial \Delta_k(\cdot)}{\partial \tau_{k_1}} = \sum_{m=1}^{M}\Bigg\{ \frac{1}{m}\left(\frac{v'_k}{v'_k + m}\right)^{u'_k} \prod_{i=1}^{m} \frac{\tau_{k_1} + i - 1}{\tau_{k_1} + \tau_{k_2} + i - 1} \\ \times \{\Psi(\tau_{k_1} + \tau_{k_2} + m) + \Psi(\tau_{k_1}) \\ - \Psi(\tau_{k_1} + \tau_{k_2}) - \Psi(\tau_{k_1} + m)\}\Bigg\},\end{aligned}$$

and

$$\begin{aligned}\frac{\partial \Delta_k(\cdot)}{\partial \tau_{k_2}} = \sum_{m=1}^{M}\Bigg\{ \frac{1}{m}\left(\frac{v'_k}{v'_k + m}\right)^{u'_k} \prod_{i=1}^{m} \frac{\tau_{k_1} + i - 1}{\tau_{k_1} + \tau_{k_2} + i - 1} \\ \times \left(\Psi(\tau_{k_1} + \tau_{k_2} + m) - \Psi(\tau_{k_1} + \tau_{k_2})\right)\Bigg\}.\end{aligned}$$

Appendix C.4. Coordinate Update for $q(T_k)$

We use the gradient ascent algorithm to jointly update (u'_k, v'_k) by the gradients of the lower bound with respect to (u'_k, v'_k). The derivatives are

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial u'_k} =& \Psi'(u'_k) \sum_{r>1}(r-2)\varphi_k(r) \\ &+ \varphi_k(r>1)\left(1 - \frac{n_{1_k} + \frac{k_1}{k_2}}{v'_k}\right. \\ &\left. - n_{0_k}\frac{\partial \Delta_k(\cdot)}{\partial u'_k} + (1-u'_k)\Psi'(u'_k)\right), \\ \frac{\partial \mathcal{L}}{\partial v'_k} =& -\frac{1}{v'_k}\sum_{r>1}(r-2)\varphi_k(r) + \varphi_k(r>1) \\ &\times \left(\frac{u'_k}{{v'_k}^2}\left(n_{1_k} + \frac{k_1}{k_2}\right) - n_{0_k}\frac{\partial \Delta_k(\cdot)}{\partial v'_k} - \frac{1}{v'_k}\right),\end{aligned} \tag{A2}$$

where

$$\begin{aligned}\frac{\partial \Delta_k(\cdot)}{\partial u'_k} &= \sum_{m=1}^{M}\Bigg\{\frac{1}{m}\prod_{i=1}^{m}\frac{\tau_{k_1}+i-1}{\tau_{k_1}+\tau_{k_2}+i-1} \\ &\quad\times\left(\frac{v'_k}{v'_k+m}\right)^{u'_k}\left(\ln\left(v'_k\right)-\ln\left(v'_k+m\right)\right)\Bigg\}, \\ \frac{\partial \Delta_k(\cdot)}{\partial v'_k} &= \sum_{m=1}^{M}\{\frac{1}{m}\prod_{i=1}^{m}\frac{\tau_{k_1}+i-1}{\tau_{k_1}+\tau_{k_2}+i-1} \\ &\quad\times u'_k\left(\frac{v'_k}{v'_k+m}\right)^{u'_k-1}\frac{m}{\left({v'_k}^2+m\right)^2}.\end{aligned} \tag{A3}$$

Appendix C.5. Coordinate Update for $q(\alpha)$

The update formulae for (k_1, k_2) are

$$\begin{aligned}k_1 &= K+\sum_{k=1}^{K}\sum_{r>1}^{R}(r-1)\varphi_k(r)+a_1, \\ k_2 &= -\sum_{k=1}^{K}\mathbb{E}[\ln(1-V_k)]+\sum_{k=1}^{K}E[T_k]\varphi_k(r>1)+a_2.\end{aligned} \tag{A4}$$

It is shown that $\varphi_k(1)$ has nothing to do with the update of α.

Appendix C.6. Coordinate Update for $q(\gamma)$

The update formulae for (τ_1, τ_2) are

$$\begin{aligned}\tau_1 &= K+b_1, \\ \tau_2 &= \sum_{r=1}^{R}\{1-\prod_{k=1}^{K}\sum_{r'=1}^{r-1}\varphi_k\left(r'\right)\}+b_2.\end{aligned} \tag{A5}$$

It can be seen from above that τ_1 does not change with iterations while τ_2 depends on $\varphi_k(r')$.

Appendix C.7. Coordinate Update for $q(\boldsymbol{\mu}_k, \Sigma_k)$

The variational parameter update of $q(\boldsymbol{\mu}_k, \Sigma_k)$ is analytical and the update formulae are

$$
\begin{aligned}
\boldsymbol{u}_k =& \frac{\sum_n^N \sum_t^T q\left(z_t^{(n)}=k\right)\mathbf{x}_t^{(n)} + \lambda_0 \boldsymbol{u}_0}{\lambda_0 + \sum_n^N \sum_t^T q\left(z_t^{(n)}=k\right)}, \\
\lambda_k =& \lambda_0 + \sum_{n=1}^{N}\sum_{t=1}^{T} q\left(z_t^{(n)}=k\right), \\
v_k =& v_0 + \sum_{n=1}^{N}\sum_{t=1}^{T} q\left(z_t^{(n)}=k\right), \\
\Phi_k =& \lambda_0 \boldsymbol{u}_0^\top \boldsymbol{u}_0 + \sum_{n=1}^{N}\sum_{t=1}^{T} q\left(z_t^{(n)}=k\right)\mathbf{x}_t^{(n)^\top}\mathbf{x}_t^{(n)} \\
&+ \Phi_0 - \frac{1}{\lambda_0 + \sum_n^N \sum_t^T q\left(z_t^{(n)}=k\right)} \\
&\times \left(\sum_n^N\sum_t^T q\left(z_t^{(n)}=k\right)\mathbf{x}_t^{(n)} + \lambda_0\boldsymbol{u}_0\right)^\top \\
&\times \left(\sum_n^N\sum_t^T q\left(z_t^{(n)}=k\right)\mathbf{x}_t^{(n)} + \lambda_0\boldsymbol{u}_0\right).
\end{aligned}
$$

Appendix C.8. Coordinate Update for $q\left(\boldsymbol{\pi}_j^{(c)}\right)$

In order to update $\boldsymbol{\pi}_j^{(c)}$, two cases, $j > 0$ and $j = 0$, should be analyzed. For $j > 0$, the logarithmic distribution of $\boldsymbol{\pi}_j^{(c)}$ is updated as

$$
\begin{aligned}
\ln q\left(\boldsymbol{\pi}_j^{(c)}\right) = \mathbb{E}_q[&\ln\big(p\left(\pi_j|f_n, r, \kappa\right) \\
&\times \prod_{n=1}^{N}\prod_{t=1}^{T-1}\delta(y_n=c)P(z_{t+1}^{(n)}|\pi, z_t^{(n)})],
\end{aligned}
$$

where

$$
p\left(z_{t+1}^{(n)}|\boldsymbol{\pi}_k^{(y_n)}, z_t^{(n)}=j\right) = \prod_{k=1}^{K} {\pi_{jk}^{(y_n)}}^{\delta\left(z_t^{(n)}=j, z_{t+1}^{(n)}=k\right)},
$$

and

$$
p\left(\boldsymbol{\pi}_j^{(c)}|f_{ck}, r^{(c)}, \kappa\right) = \frac{1}{B}\prod_{k\neq j}^{K} {\pi_{jk}^{(c)}}^{(r^{(c)}f_{ck}-1)} {\pi_{jj}^{(c)}}^{((r^{(c)}+\kappa)f_{ck}-1)}.
$$

Here B is the normalizing constant of the Dirichlet distribution $q\left(\boldsymbol{\pi}_j^{(c)}\right)$. We can get

$$
\begin{aligned}
\ln q\left(\boldsymbol{\pi}_j^{(c)}\right) = &\sum_{k\neq j}^{K}\Bigg\{\sum_{(n=1)}^{N}\sum_{t=1}^{T-1}\delta(y_n=c)q\left(z_t^{(n)}=j, z_{t+1}^{(n)}=k\right) \\
&+ rv_{ck} - 1\Bigg\}\ln\left(\pi_{jk}^{(c)}\right) + \Bigg((r+\kappa)v_{ck} - 1 \\
&+ \sum_{n=1}^{N}\sum_{t}^{T-1}\delta(y_n=c)q\left(z_t^{(n)}=j, z_{t+1}^{(n)}=j\right)\Bigg)\ln\left(\pi_{jj}^{(c)}\right).
\end{aligned}
$$

Thus the parameters $\mathbf{r}_j^{\prime(c)}$ for the Dirichlet distribution $q\left(\boldsymbol{\pi}_j^{(c)}\right)$ are formulated as

$$r_{jk}^{\prime(c)} = \sum_{n=1}^{N}\sum_{t=1}^{T-1}\delta(y_n = c)q\left(z_t^{(n)} = j, z_{t+1}^{(n)} = k\right) + r^{(c)}v_{ck},$$
$$\text{with } k \neq j,$$

and

$$r_{jj}^{\prime(c)} = \sum_{n=1}^{N}\sum_{t=1}^{T-1}\delta(y_n = c)q\left(z_t^{(n)} = j, z_{t+1}^{(n)} = j\right) + \left(r^{(c)} + \kappa\right)v_{cj}.$$

For $j = 0$, $\boldsymbol{\pi}_j^{(c)}$ is the prior probability of the hidden states. We can obtain

$$r_{0k} = \sum_{n=1}^{N}\delta(y_n = c)q\left(z_1^{(n)} = k\right) + r^{(c)}v_{ck}.$$

Appendix C.9. Coordinate Update for $q(Z)$

For each class of trajectories,

$$\begin{aligned}
a_{jk}^{*\,(c)} &= \exp\left(\mathbb{E}_q \ln p\left(\pi_{jk}^{(c)}\right)\right)\\
&= \exp\left(\Psi\left(r_{jk}^{\prime(c)}\right) - \Psi\left(\sum_{i=1}^{K} r_{ji}^{\prime(c)}\right)\right),\\
b_{tj}^{*\,(n)} &= \exp\left(\mathbb{E}_q \ln p\left(\mathbf{x}_t^{(n)} | \boldsymbol{\mu}_j, \Sigma_j\right)\right)\\
&= \exp\Bigg\{ -\frac{p}{2}\ln(2\pi) - \frac{1}{2}\mathbb{E}_q \ln\left(|\Sigma_j|\right)\\
&\quad - \frac{1}{2}\left(\frac{d}{\lambda_k} + \left(\mathbf{x}_t^{(n)} - \mathbf{u}_j\right)^{\top}(v_j - p - 1)\Phi_j^{-1}\left(\mathbf{x}_t^{(n)} - \mathbf{u}_j\right)\right)\Bigg\},
\end{aligned}$$

where p is the dimension of $\mathbf{x}_t^{(n)}$ and

$$\mathbb{E}_q \ln\left(|\Sigma_j|\right) = -\sum_{i=1}^{p}\Psi\left(\frac{v_j + 1 - i}{2}\right) - p\ln(2) + \ln|\Phi_j|.$$

From the above, we need the marginal probabilities $q\left(z_t^{(n)} = j\right)$ and $q\left(z_t^{(n)} = j, z_{t+1}^{(n)} = k\right)$. The detailed calculations are as follows. Both of them can be calculated by the forward-backward algorithm. The forward procedure is

$$\begin{aligned}
&\iota_k^t = P(\mathbf{x}_1 = \mathbf{x}_1, \mathbf{x}_2 = \mathbf{x}_2, \ldots, \mathbf{x}_t = \mathbf{x}_t, z_t k | W, \Theta),\\
&\iota_k^1 = a_{0j}^* b_{tk}^*,\\
&\iota_j^{t+1} = b_{t+1j}^* \sum_{k=1}^{K} \iota_k^t a_{kj}^*.
\end{aligned}$$

The backward procedure is

$$\begin{aligned}
&\beta_k(t) = P(\mathbf{x}_{t+1} = \mathbf{x}_{t+1}, \ldots, \mathbf{x}_T = \mathbf{x}_T | z_t = k, W, \Theta),\\
&\beta_k(T) = 1,\\
&\beta_k(T) = \sum_{j=1}^{K} \beta_j(t+1) a_{kj}^* b_{t+1j}^*.
\end{aligned}$$

Thus, the expressions of the posterior distributions are

$$q(z_t = j) = \frac{\iota_j(t)\beta_j(t)}{\sum_{j=1}^{K} \iota_j(t)\beta_j(t)},$$
$$q(z_t = j, z_{t+1} = k) = \frac{\iota_j(t)a^*_{jk}\beta_j(t)b^*_{t+1k}}{\sum_{k=1}^{K} \iota_k(t)\beta_k(t)}.$$

Now we can update the variational parameters in the BP-HMM according to the above equations. We judge the convergence of this update according to the change of the lower bound.

References

1. Sun, S.; Zhao, J.; Gao, Q. Modeling and recognizing human trajectories with beta process hidden Markov models. *Pattern Recognit.* **2015**, *48*, 2407–2417. [CrossRef]
2. Rabiner, L.R. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* **1989**, *77*, 257–286. [CrossRef]
3. Braiek, E.; Aouina, N.; Abid, S.; Cheriet, M. Handwritten characters recognition based on SKCS-polyline and hidden Markov model (HMM). In Proceedings of the International Symposium on Control, Communications and Signal Processing, Hammamet, Tunisia, 21–24 March 2004; pp. 447–450.
4. Freire, A.L.; Barreto, G.A.; Veloso, M.; Varela, A.T. Short-term memory mechanisms in neural network learning of robot navigation tasks: A case study. In Proceedings of the Latin American Robotics Symposium, Valparaiso, Chile, 29–30 October 2009; pp. 1–6.
5. Gao, Q.B.; Sun, S.L. Trajectory-based human activity recognition using hidden conditional random fields. In Proceedings of the International Conference on Machine Learning and Cybernetics, Xi'an, China, 15–17 July 2012; Volume 3, pp. 1091–1097.
6. Bousmalis, K.; Zafeiriou, S.; Morency, L.P.; Pantic, M. Infinite hidden conditional random fields for human behavior analysis. *IEEE Trans. Neural Netw. Learn. Syst.* **2012**, *24*, 170–177. [CrossRef] [PubMed]
7. Gao, Q.; Sun, S. Trajectory-based human activity recognition with hierarchical Dirichlet process hidden Markov models. In Proceedings of the International Conference on Signal and Information Processing, Beijing, China, 6–10 July 2013; pp. 456–460.
8. Fox, E.B.; Hughes, M.C.; Sudderth, E.B.; Jordan, M.I. Joint modeling of multiple time series via the beta process with application to motion capture segmentation. *Ann. Appl. Stat.* **2014**, *8*, 1281–1313. [CrossRef]
9. Fox, E.; Jordan, M.I.; Sudderth, E.B.; Willsky, A.S. Sharing features among dynamical systems with beta processes. *Adv. Neural Inf. Process. Syst.* **2009**, *22*, 549–557.
10. Gao, Q.B.; Sun, S.L. Human activity recognition with beta process hidden Markov models. In Proceedings of the International Conference on Machine Learning and Cybernetics, Tianjin, China, 14–17 July 2013; Volume 2, pp. 549–554.
11. Gao, Y.; Villecco, F.; Li, M.; Song, W. Multi-scale permutation entropy based on improved LMD and HMM for rolling bearing diagnosis. *Entropy* **2017**, *19*, 176. [CrossRef]
12. Filippatos, A.; Langkamp, A.; Kostka, P.; Gude, M. A sequence-based damage identification method for composite rotors by applying the Kullback–Leibler divergence, a two-sample Kolmogorov–Smirnov test and a statistical hidden Markov model. *Entropy* **2019**, *21*, 690. [CrossRef] [PubMed]
13. Granada, I.; Crespo, P.M.; Garcia-Frías, J. Combining the Burrows-Wheeler transform and RCM-LDGM codes for the transmission of sources with memory at high spectral efficiencies. *Entropy* **2019**, *21*, 378. [CrossRef] [PubMed]
14. Li, C.; Pourtaherian, A.; van Onzenoort, L.; Ten, W.E.T.a.; de With, P.H.N. Infant facial expression analysis: Towards a real-time video monitoring system using R-CNN and HMM. *IEEE J. Biomed. Health Inform.* **2021**, *25*, 1429–1440. [CrossRef] [PubMed]
15. Li, J.; Todorovic, S. Action shuffle alternating learning for unsupervised action segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, virtual meeting, 19–25 June 2021; pp. 12628–12636.
16. Zhou, W.; Michel, W.; Irie, K.; Kitza, M.; Schlüter, R.; Ney, H. The rwth ASR system for ted-lium release 2: Improving hybrid HMM with specaugment. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Barcelona, Spain, 4–8 May 2020; pp. 7839–7843.
17. Zhu, Y.; Yan, Y.; Komogortsev, O. Hierarchical HMM for eye movement classification. In *European Conference on Computer Vision Workshops*; Springer: Cham, Switzerland, 2020; pp. 544–554.
18. Lom, M.; Pribyl, O.; Svitek, M. Industry 4.0 as a part of smart cities. In Proceedings of the 2016 Smart Cities Symposium Prague (SCSP), Prague, Czech Republic, 26–27 May 2016; pp. 1–6.
19. Castellanos, H.G.; Varela, J.A.E.; Zezzatti, A.O. Mobile Device Application to Detect Dangerous Movements in Industrial Processes Through Intelligence Trough Ergonomic Analysis Using Virtual Reality. In *The International Conference on Artificial Intelligence and Computer Vision*; Springer: Cham, Switzerland, 2021; pp. 202–217.
20. Deng, Q.; Söffker, D. Improved driving behaviors prediction based on fuzzy logic-hidden markov model (fl-hmm). In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; pp. 2003–2008.
21. Fouad, M.A.; Abdel-Hamid, A.T. On Detecting IoT Power Signature Anomalies using Hidden Markov Model (HMM). In Proceedings of the 2019 31st International Conference on Microelectronics (ICM), Cairo, Egypt, 15–18 December 2019; pp. 108–112.

22. Nascimento, J.C.; Figueiredo, M.A.; Marques, J.S. Trajectory classification using switched dynamical hidden Markov models. *IEEE Trans. Image Process.* **2009**, *19*, 1338–1348. [CrossRef] [PubMed]
23. Thibaux, R.; Jordan, M.I. Hierarchical beta processes and the Indian buffet process. In Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics, San Juan, Puerto Rico, 21–24 March 2007; pp. 564–571.
24. Teh, Y.W.; Jordan, M.I.; Beal, M.J.; Blei, D.M. Sharing clusters among related groups: Hierarchical Dirichlet processes. *Adv. Neural Inf. Process. Syst.* **2004**, *17*, 1385–1392.
25. Hughes, M.C.; Fox, E.; Sudderth, E.B. Effective split-merge monte carlo methods for nonparametric models of sequential data. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1295–1303.
26. Teh, Y.W.; Grür, D.; Ghahramani, Z. Stick-breaking construction for the Indian buffet process. In Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics, San Juan, Puerto Rico, 21–24 March 2007; pp. 556–563.
27. Griffiths, T.L.; Ghahramani, Z. Infinite latent feature models and the Indian buffet process. *Adv. Neural Inf. Process. Syst.* **2005**, *18*, 475–482.
28. Hjort, N.L. Nonparametric Bayes estimators based on beta processes in models for life history data. *Ann. Stat.* **1990**, *18*, 1259–1294. [CrossRef]
29. Paisley, J.W.; Zaas, A.K.; Woods, C.W.; Ginsburg, G.S.; Carin, L. A stick-breaking construction of the beta process. In Proceedings of the 27th International Conference on Machine Learning (ICML 2010), Haifa, Israel, 21–24 June 2010; pp. 847–854.
30. Ferguson, T.S. A Bayesian analysis of some nonparametric problems. *Ann. Stat.* **1973**, *1*, 209–230. [CrossRef]
31. Selhuraman, J. A constructive definition of the Dirichlet prior. *Statist. Sin.* **1994**, *2*, 639–650.
32. Cao, Y.; Li, Y.; Coleman, S.; Belatreche, A.; McGinnity, T.M. Adaptive hidden Markov model with anomaly states for price manipulation detection. *IEEE Trans. Neural Netw. Learn. Syst.* **2014**, *26*, 318–330. [CrossRef] [PubMed]
33. Paisley, J.W.; Carin, L.; Blei, D.M. Variational Inference for Stick-Breaking Beta Process Priors. In Proceedings of the 28th International Conference on Machine Learning, Bellevue, WA, USA, 28 June–2 July 2011; pp. 889–896.
34. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
35. Alcock, R.J.; Manolopoulos, Y. Time-series similarity queries employing a feature-based approach. In Proceedings of the 7th Hellenic Conference on Informatics, Ioannina, Greece, 26–29 August 1999; pp. 27–29.
36. Ziaeefard, M.; Bergevin, R. Semantic human activity recognition: A literature review. *Pattern Recognit.* **2015**, *48*, 2329–2345. [CrossRef]