Dual Sticky Hierarchical Dirichlet Process Hidden Markov Model and Its Application to Natural Language Description of Motions

Weiming Hu, Guodong Tian, Yongxin Kang, and Chunfeng Yuan

(National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190)

{wmhu, guodong.tian, yongxin.kang, cfyuan}@nlpr.ia.ac.cn

Stephen Maybank

(Department of Computer Science and Information Systems, Birkbeck College, Malet Street, London WC1E 7HX)

sjmaybank@dcs.bbk.ac.uk

Abstract: In this paper, a new nonparametric Bayesian model called the dual sticky hierarchical Dirichlet process hidden Markov model (HDP-HMM) is proposed for mining activities from a collection of time series data such as trajectories. All the time series data are clustered. Each cluster of time series data, corresponding to a motion pattern, is modeled by an HMM. Our model postulates a set of HMMs that share a common set of states (topics in an analogy with topic models for document processing), but have unique transition distributions. The number of HMMs and the number of topics are both automatically determined. The sticky prior avoids redundant states and makes our HDP-HMM more effective to model multimodal observations. For the application to motion trajectory modeling, topics correspond to motion activities. The learnt topics are clustered into atomic activities which are assigned predicates. We propose a Bayesian inference method to decompose a given trajectory into a sequence of atomic activities. The sources and sinks in the scene are learnt by clustering endpoints (origins and destinations) of trajectories. The semantic motion regions are learnt using the points in trajectories. On combining the learnt sources and sinks, semantic motion regions, and the learnt sequence of atomic activities, the action represented by the trajectory can be described in natural language in as automatic a way as possible. The effectiveness of our dual sticky HDP-HMM is validated on several trajectory datasets. The effectiveness of the natural language descriptions for motions is demonstrated on the vehicle trajectories extracted from a traffic scene.

Index terms: HDP-HMM, Sticky prior, Motion pattern learning, Natural language description

1. Introduction

One of the most challenging problems in computer vision is to understand and semantically interpret object motions in dynamic image sequences. For instance, a visual surveillance system should be able to interpret what is happening in the dynamic scene. The result of the interpretation may be an abstract or symbolic model, or natural language, etc. In particular, the automatic generation of natural language descriptions of motions and actions in videos is the ultimate goal of computer vision. Some efforts have been made to assess how far we are from this goal. Remagnino et al. [33] proposed a visual event interpretation system to describe the actions of pedestrians and vehicles in a traffic scene. An agent-orientated Bayesian network produces natural language annotations for events. Kollnig et al. [39] characterized trajectories of vehicle motions using verbs and verb phrases. Lou et al. [34] semantically interpreted vehicle and pedestrian actions for visual traffic surveillance. The trajectories were analyzed to generate natural language descriptions of object motions. Kojima et al. [35] described human activities in video images based on a concept hierarchy of actions. By associating concepts with extracted semantic features, appropriate syntactic components such as verbs, objects, etc. were included in natural language sentences. The limitation of the existing methods for generating natural language descriptions of object motions is that they need a large number of manually defined rules. The degree of automation is significantly limited. While there is active research [36, 37, 38] on generating natural language descriptions of videos using object detection and text mining, etc, the research on generating natural language descriptions by understanding object motions has halted for more than ten years because of difficulties in the automatic generation of natural language descriptions.

In this paper, we mine activities from time series data, discover typical activities and their semantic structure, and generate natural language descriptions of motions with as little supervision as possible. On one hand, object trajectories are important cues for characterizing activities, as they contain rich information such as origins, destinations and motion directions. One the other hand, Bayesian topic models for activities or atomic activities can be constructed in a natural way. Therefore, we combine trajectory analysis with Bayesian models to jointly mine primitive activities and their intrinsic temporal structures. A semantic description of each action is composed, and then transformed into natural language.

1.1. Related work

For the context of our work, we briefly review trajectory analysis and Bayesian topic models.

1.1.1. Trajectory analysis

The tasks required for trajectory analysis [52, 53, 54] include trajectory similarity measure, clustering, and spatiotemporal dynamics modeling.

There exist several methods for measuring similarities between trajectories. These include Euclidean distance [7], dynamic time warping (DTW)-based distance [8], principal component analysis (PCA)-based Euclidean distance [22], Hausdorff distance [31, 40], longest common subsequence (LCSS) distance [30], edit distance on real sequences (EDR) [41, 43], distance based on discrete Fourier transform (DFT) coefficients [26], curve fitting parameters-based distance [42], tensor compression representation-based distance [23], DNA sequence matching-based distance [44], and 4D motion histogram-based distance [45]. The Euclidean distance and the PCA-based Euclidean distance are simple to compute and appropriate for trajectories with simple shapes, but they are not robust enough to noisy trajectories and/or to trajectories with complex shapes. The Hausdorff distance uses the distance between points to define a distance between trajectories. This avoids matching the points in different trajectories. However, the Hausdorff distance does not fully utilize the sequence information in trajectories. The DTW distance is based on point correspondences between trajectories. A single point in one trajectory can correspond to multiple points in another trajectory. The time complexity of DTW is high, and it is not robust on noisy trajectories, because every point in a trajectory must have at least one corresponding point in the other trajectory. The LCSS distance is similar to the DTW distance, except that outlying points may be omitted from the correspondence between trajectories. The LCSS requires angle information and the setting of several thresholds. The EDR may lose information when the edit operation is extended from sequences of discrete symbols to sequences of real values. The DFT-based distance and the curve fitting-based distance are able to remove some noise and reduce the dimension of the space of trajectories. However, the DFT and the curve fitting may omit useful information in a trajectory. Overall, when the trajectories are long, distance measures that involve data

compression work effectively. When trajectories have complex dynamics, distances based on time warping have outstanding advantages.

Based on the estimated distance between trajectories, trajectories can be clustered by methods such as *K*-means, spectral clustering, self-organizing mapping (SOM) [26], and hierarchical Dirichlet process (HDP) [13, 40]. A cluster of trajectories often represents a route along which objects move repeatedly and frequently or a sequence of actions carried out repeatedly in the scene. The performances of different methods for clustering trajectories are compared experimentally in [9] and [10]. It was reported that spectral clustering usually yields the most accurate results.

One key task in trajectory modeling is to estimate the spatiotemporal dynamics in each cluster of trajectories. These dynamics form a motion pattern. Saleemi et al. [46] estimated the probability density function for the spatiotemporal features of object locations and transition times using the kernel density method. Hu et al. [7] modeled a trajectory pattern using a chain of Gaussian distributions. Morris and Trivedi [28] developed hidden Markov models (HMMs), which can update parameters online, to model spatiotemporal motion features associated with different routes. Bashir et al. [47] represented a trajectory as a sequence of principal component coefficients and modeled each cluster of trajectories using HMMs. Nguyen et al. [48] used a hierarchical hidden Markov model to represent the complex activities in trajectories. Veeraraghavan and Papanikolopoulos [49] learned sequences of spatiotemporal activities which are represented by the trajectories and modeled by stochastic context-free grammars. The above methods used HMMs to model the obtained clusters of trajectories, rather than used HMMs to cluster trajectories. Moreover, the above methods lack semantically meaningful descriptions of activities.

1.1.2. Bayesian topic models

Bayesian models add priors to the distributions for generating the hidden variables and observations in ordinary probabilistic models. As a result, Bayesian models can represent prior knowledge of data. Prior distributions, chosen appropriately, can restrain the model to avoid over-fitting. The Bayesian topic models used in document analysis are also appropriate for automatically generating semantic descriptions of object motions.

Latent Dirichlet allocation (LDA) [11] is a classical Bayesian topic model. Niebles et al. [12] applied LDA to unsupervised learning of human actions. Wang et al. [50, 51] extended LDA to the LDA mixture model which was applied to learn motion patterns. The LDA's limitations are that the number of clusters has to be known a priori and sequential dependencies in data are not modeled.

The Dirichlet process (DP)-based Bayesian non-parametrical models effectively estimate the number of clusters. For instance, Wang et al. [13, 50] proposed the dual hierarchical Dirichlet process (Dual-HDP) model in which the topics were modeled in a hierarchy and the number of topics in each layer was automatically determined. The Dual-HDP model does not include the sequential correlations between words in a document. To solve this problem, Kuettel et al. [14] proposed the dependent Dirichlet process hidden Markov model (DDP-HMM) to model temporal dependencies with an arbitrary number of HMMs. However, the DDP-HMM lacks the ability to cluster documents and to share topics between different documents. Fox et al. [15] proposed a sticky hierarchical Dirichlet process hidden Markov model (HDP-HMM), which is robust to noise and accurate in learning state sequences. However, this model cannot simultaneously determine the number of clusters of documents and the number of topics.

1.2. Our work

With the aim of handling the above limitations in Bayesian topic models, we propose a new nonparametric Bayesian model, the dual sticky HDP-HMM [55]. We further apply it to trajectories to learn motion patterns and generate natural language descriptions of motions.

Given a set of object trajectories, we segment each trajectory into several sub-trajectories in order to characterize the time varying information in the trajectories. Each sub-trajectory is represented by a feature vector. We cluster all the sub-trajectories in the set of trajectories. Each cluster of sub-trajectories corresponds to an activity, such as "going ahead" and "turning to left", and forms a visual word represented by a feature vector. These visual words form a word codebook. A trajectory contains a sequence of activities and is represented as a word sequence treated as a document. By representing trajectories in this way, word-document style analysis can be carried out on the trajectory set. A cluster of similar activities forms an atomic activity corresponding to a visual topic which consists of a set of visual words. After word-document style analysis is carried out on the trajectory set, the topics in the documents are found. Table 1 compares the terminologies of trajectory modeling and the topic model.

Table 1: Comparison of the terminologies of trajectory modeling and the topic model

Trajectory modeling	Topic models
A trajectory	A document
A sub-trajectory cluster/An activity	A word
A set of sub-trajectory clusters	A word codebook
An atomic activity	A topic

This paper develops a new word-document style analysis model, namely the dual sticky HDP-HMM. Our model is able to simultaneously cluster documents, find topics in documents, and model the sequential correlations in documents, without knowing in advance the number of clusters of documents and the number of topics. Each HMM corresponds to a cluster of documents. All the HMMs share a common set of topics. A HMM has a state transition matrix which models sequential correlations between words. The transition matrices of the HMMs are regularized by a sticky prior, which makes the model more robust to the variations in the observations of each state. As a result, complex noisy time series with multiple observations at each time can be effectively handled. We develop a Gibbs sampler to learn the dual sticky HDP-HMM. We propose a Bayesian inference mechanism to evaluate the likelihood, predict cluster memberships, and estimate the topics in a given document.

By applying the dual sticky HDP-HMM to the trajectory set, trajectory patterns are distinguished, and their intrinsic temporal structures are revealed. In the learnt HDP-HMM, an HMM corresponds to a type of action, which is composed of atomic activities represented by topics. The sequence of topics for a given trajectory is predicted by Bayesian inference. The learnt topics are generalized to higher level atomic activities. Predicates are assigned to the generalized atomic activities. We learn sources and sinks in the scene by clustering the endpoints of trajectories. We learn semantic regions in the scene by clustering points in trajectories. Then, an action is jointly determined by a source, a sink and a motion process in which the object moves through some semantic regions in a way defined by one or more generalized atomic activities. A natural language description of the action is composed to answer the questions "Where does the object come from and get to?" and "How does the object go from the origin to the destination?".

The remainder of the paper is organized as follows: Section 2 describes the trajectory representation method. Section 3 introduces the HDP and HDP-HMM. Section 4 proposes our dual sticky HDP-HMM. Section 5 applies our dual HDP-HMM to learn trajectory patterns. Section 6 presents our method for generating semantic description of motions. Section 7 demonstrates the experimental results. Section 8 summarizes the paper.

2. Trajectory Representation

We consider two types of trajectory: ordinary trajectories which are obtained by linking 2D points and generalized trajectories in which each point consists of multiple observations such as multiple points of interest in a frame in a video. We represent ordinary trajectories and generalized trajectories using a model originally devised for word documents.

2.1. Ordinary trajectories

We segment each ordinary trajectory into several

sub-trajectories which have simpler shapes and also have semantic meanings. As general trajectory segmentation approaches [22, 23] based on the variation in curvature are sensitive to noise, we use the spectral clustering-based method [24] to segment trajectories. The position coordinates and the frame number of each point in a trajectory are used as features input to the clustering method. Points which are consecutive in time and spatially close are likely to be clustered into the same sub-trajectory. The sub-trajectories obtained in this way mostly approximate to straight short lines or to simple curves. In both cases there are semantic meanings, i.e., "going ahead" and "turning". In order to avoid the specification in advance of the number of segments, we propose an improved spectral clustering method which replaces the *K*-means clustering in the last step of the spectral clustering with a non-parametric adaptive mean-shift clustering algorithm [25]. As a result, the improved spectral clustering is able to automatically identify the number of segments in a trajectory. As spectral clustering tends to group points into balanced clusters, the sub-trajectories belonging to the same trajectory have comparable numbers of points. The above trajectory segmentation compresses the trajectory data and makes it feasible to produce semantic descriptions of the motions.

After the trajectory segmentation, a feature vector is extracted for each sub-trajectory. According to [26], the DFT-coefficient feature vector for a trajectory is more robust than the original point-based feature vector and the polynomial fitting-based feature vector. According to [27], the PCA features, which are obtained by carrying out PCA on the coordinates of sequential points in trajectories, are suitable for trajectories with simple shapes, which the segmented sub-trajectories in this application usually have. So, we concatenate the DFT-coefficients [26] and the PCA features [22] on each sub-trajectory to produce a feature vector. Then, a trajectory is represented by a sequence of feature vectors.

Using the vector quantization technique, all the sub-trajectories are clustered to obtain V cluster centers $\{b_i\}_{i=1}^{V}$. Any feature vector in the feature space can be represented by the center to which the feature vector is closest. In this way, we construct a codebook $B=\{b_i\}_{i=1}^{V}$. Each center b_i $(1\le i\le V)$ in B is called a visual word. As a result, a trajectory is transformed into a visual word sequence $b_{I_1}, b_{I_2}, ..., b_{I_i}, ..., b_{I_T}$, where $I_i \in \{1,2,...,V\}$ is an indicator variable which assigns the i-th sub-trajectory into the corresponding word, and T is the number of sub-trajectories in the trajectory. In this way, trajectories are represented in the word-document style, and a topic model, the dual sticky HDP-HMM, can be applied to the trajectories for motion analysis. In particular, the multinomial distribution which is conjugate with the Dirichlet distribution can be used to construct the observation model.

2.2. Generalized trajectories

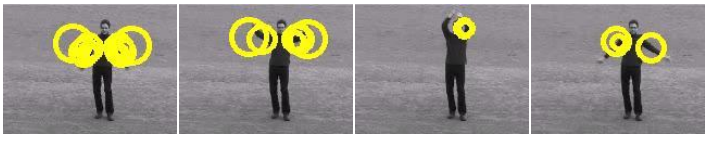


Fig. 1. Points of interest detected in four frames in a video.

We define generalized trajectories as the ones that have multiple observations at each time. In a video, a sequence of multiple points of interest in each frame forms a generalized trajectory. Fig. 1 shows the points of interest in four frames in a video of hand waving, where each circle represents an extracted point of interest. A feature vector is extracted for each point of interest. The feature vectors for all the extracted points of interest are clustered to form a codebook, and each point of interest is encoded by a visual word. As shown in Fig. 2, each frame contains a set of multiple visual words, and a video is represented by a sequence of sets of words, where the numbers of words in different sets may be unequal. The encoded sequences are used to learn a dual sticky HDP-HMM. As there are multi-observations at each time (frame), the existing HDP-HMM methods which just use HMM states without mixture states are unable to directly handle them, but our dual sticky HDP-HMM is specifically designed for multi-observations.

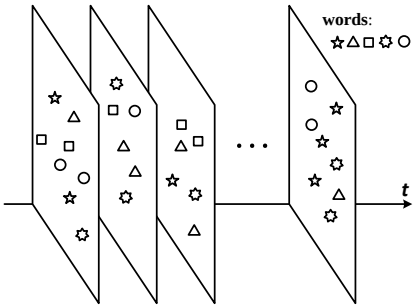


Fig. 2. A video is represented as a generalized trajectory with multi-observations at each frame.

3. HDP and HDP-HMM

We briefly summarize the Dirichlet distribution, the Dirichlet process, the DP mixture model, the HDP, the HDP-HMM, and the sticky HDP-HMM [1, 2, 4, 5, 15].

3.1. Dirichlet distribution

Given a K-dimensional parameter vector $\mathbf{a}=[a_1,a_2,...,a_K]$, the probability density function of the Dirichlet distribution of a K-dimensional random vector $\boldsymbol{\pi}=[\pi_1,\pi_2,...,\pi_K]$ $(0\le\pi\le 1, \sum_{k=1}^{K}\pi_k=1)$ is defined as:

$$p(\boldsymbol{\pi}\mid\mathbf{a})=\frac{\Gamma(\sum_{k=1}^{K}a_k)}{\prod_{k=1}^{K}\Gamma(a_k)}\prod_{k=1}^{K}\pi_k^{a_k-1} \tag{1}$$

where $\Gamma()$ is the Gamma function (See Appendix A which is available online). It is clear that in (1) $\boldsymbol{\pi}$ is a discrete probability distribution for which the probability that a realization z of a random variable Z takes the value of k is π_k. The Dirichlet distribution with a parameter vector $\mathbf{a}$ is denoted as Dir($\mathbf{a}$). When K=2, Dir($\mathbf{a}$) reduces to the Beta distribution Beta(a_1, a_2). The discrete distribution Dis($\boldsymbol{\pi}$) with parameter vector $\boldsymbol{\pi}$ is sampled N times to yield N samples $\mathbf{z}_{1:N}$. Let n_k be the number of the samples which take value k. The random variable $\mathbf{n}=[n_1,n_2,...,n_K]$ has the multinomial distribution parameterized by N and $\boldsymbol{\pi}$. Its probability mass function is:

$$p(\mathbf{n}\mid N,\boldsymbol{\pi})=\frac{N!}{\prod_{k=1}^{K}n_k!}\prod_{k=1}^{K}\pi_k^{n_k}\ . \tag{2}$$

Given N samples $\mathbf{z}_{1:N}$, the posterior probability of $\boldsymbol{\pi}$ is inferred as:

$$p(\boldsymbol{\pi}\mid\mathbf{z}_{1:N},\mathbf{a})\propto p(\boldsymbol{\pi}\mid\mathbf{a})\,p(\mathbf{n}\mid N,\boldsymbol{\pi})\propto\prod_{k=1}^{K}\pi_k^{a_k+n_k-1}\ . \tag{3}$$

The posterior distribution of $\boldsymbol{\pi}$ is a Dirichlet distribution parameterized by a_1+n_1 ,…, a_k+n_k ,…, a_K+n_K . The distributions $p(\boldsymbol{\pi}\mid\mathbf{a})$ and $p(\boldsymbol{\pi}\mid z_{1:N},\mathbf{a})$ are conjugate. The mean $\mathbf{E}(\boldsymbol{\pi}\mid\mathbf{z}_{1:N})$ of $p(\boldsymbol{\pi}\mid\mathbf{z}_{1:N},\mathbf{a})$ is given by $E(\pi_k)=(a_k+n_k)/(a_0+N)$, where $a_0=\sum_{k=1}^{K}a_k$. The distribution of z_{N+1} given $\mathbf{z}_{1:N}$ and $\mathbf{a}$ is inferred as:

$$p(z_{N+1}=k\mid\mathbf{z}_{1:N},\mathbf{a})=\int\pi_k\,p(\boldsymbol{\pi}\mid\mathbf{z}_{1:N},\mathbf{a})d\boldsymbol{\pi}=\mathbf{E}[\pi_k\mid\mathbf{z}_{1:N}]=\frac{a_k+n_k}{a_0+N}\ . \tag{4}$$

It is used to predict the distribution of future samples.

3.2. Dirichlet process

Let Θ be a measurable parameter space for the clusters. Let H be a probability measure over Θ. Let $\{T_1,T_2,...,T_K\}$ be a disjoint partition of Θ: $\bigcup_{k=1}^{K}T_k=\Theta$ and $\forall k\ne l$, $T_k\cap T_l=\phi$. A random probability distribution G on Θ obeys a Dirichlet process DP(α, H) [1, 2, 3], if the probability measures of G over the parts $\{T_1,T_2,...,T_K\}$ obey a Dirichlet distribution:

$$(G(T_1),G(T_2),...,G(T_K))\sim\mathrm{Dir}(\alpha H(T_1),\alpha H(T_2),...,\alpha H(T_K)) \tag{5}$$

where $\alpha>0$ is the concentration parameter and H is a base distribution on Θ. The base distribution H corresponds to the expectation or mean of G, and α indicates the average deviation of samples away from H.

Typically, a draw G from DP(α, H) is obtained by the stick-breaking construction [16]. An infinite sequence $\{\pi'_k\}_{k=1}^{\infty}$ of positive real values is generated by $\pi'_k \mid \alpha \sim \text{Beta}(1,\alpha)$, $k=1,2,...,\infty$. A second infinite sequence $\{\pi_k\}_{k=1}^{\infty}$ of positive real values is generated by:

$$\pi_k = \pi'_k \prod_{l=1}^{k-1}(1-\pi'_l), \quad k=1,2,...,\infty. \tag{6}$$

A parameter vector sequence $\{\boldsymbol{\theta}_k\}_{k=1}^{\infty}$ is sampled from the base distribution H on Θ : $\boldsymbol{\theta}_k \sim H$, $k=1,2,...,\infty$. The stick-breaking representation of G is the discrete distribution consisting of $\boldsymbol{\pi}$ and $\{\boldsymbol{\theta}_k\}_{k=1}^{\infty}$:

$$G(\boldsymbol{\theta}) = \sum_{k=1}^{\infty} \pi_k \delta(\boldsymbol{\theta}, \boldsymbol{\theta}_k) \tag{7}$$

where δ is the indicator function:

$$\delta(\boldsymbol{\theta}, \boldsymbol{\theta}_k) = \begin{cases} 1 & if\ \boldsymbol{\theta} = \boldsymbol{\theta}_k \\ 0 & otherwise \end{cases}. \tag{8}$$

The construction of $\boldsymbol{\pi}$ is usually denoted by $\boldsymbol{\pi} \sim \text{GEM}(\alpha)$, where GEM is the abbreviation of three researchers' names.

It is assumed that there are N samples drawn from G. They take K ($K < N$) different values $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K$, i.e., they form K clusters. This clustering process can be described using the Chinese restaurant process (CRP). Customers enter a Chinese restaurant and sit at a number of tables. The first customer sits at any table in the restaurant. When there are N customers seated at K tables, let n_k be the number of customers at table k. The N+1-th customer selects the k-th table with probability $n_k/(N+\alpha)$ and selects a new table with probability $\alpha/(N+\alpha)$. The customers seated at table k share the same dish $\boldsymbol{\theta}_k$, which defines a cluster.

3.3. DP mixture model

Fig. 3 shows the graphical model of the basic mixture model with K components. The model is associated with weights $\boldsymbol{\pi} = \{\pi_k\}_{k=1}^{K}$ and parameter vectors $\{\boldsymbol{\theta}_k\}_{k=1}^{K}$ for the components. Each component k is associated with an observation distribution $\text{F}(\boldsymbol{\theta}_k)$. The process of generating an observation y_i include sampling from $\boldsymbol{\pi}$ to generate a cluster label z_i and then generating y_i according to the distribution $\text{F}(\boldsymbol{\theta}_{z_i})$. It can be represented by:

$$\begin{aligned} z_i \mid \boldsymbol{\pi} &\sim \text{Dis}(\boldsymbol{\pi}) \\ y_i \mid \{\boldsymbol{\theta}_k\}_{k=1}^{K}, z_i &\sim \text{F}(\boldsymbol{\theta}_{z_i}). \end{aligned} \tag{9}$$

One limitation of the basic mixture model is that the parameter K should be known.

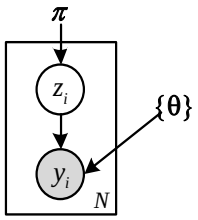


Fig. 3. The graphical model of the basic mixture model.

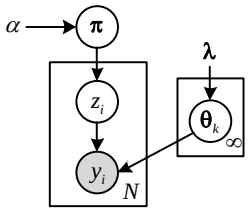


Fig. 4. The stick-breaking graphical model for the DP mixture model.

The DP mixture model can be adapted to include an estimate of K. A DP prior DP(α, H) is imposed on the distribution represented by $\boldsymbol{\pi}$ and $\{\boldsymbol{\theta}\}$. The generation process of the DP mixture model is represented by:

$$\begin{aligned} &\boldsymbol{\pi} \mid \alpha \sim \text{GEM}(\alpha) \\ &\boldsymbol{\theta}_k \mid \boldsymbol{\lambda} \sim H(\boldsymbol{\lambda}) && k=1,2,...,\infty \\ &z_i \mid \boldsymbol{\pi} \sim \text{Dis}(\boldsymbol{\pi}) && i=1,2,...,N \\ &y_i \mid \{\boldsymbol{\theta}_k\}_{k=1}^{\infty}, z_i \sim \text{F}(\boldsymbol{\theta}_{z_i}) && i=1,2,...,N \end{aligned} \tag{10}$$

where $\boldsymbol{\lambda}$ is the parameter vector of the base distribution H. The corresponding stick-breaking graphical model is shown in Fig. 4. The number K of clusters is naturally determined by the number of different labels.

The DP mixture model can model words in a document by treating the words as observations and topics (states) $\{\boldsymbol{\theta}_k\}$ as components. However, it cannot model the words in a set of documents to take account of the sharing of the topics between different documents.

3.4. HDP

The HDP [2, 4, 5] is an extension of the DP mixture model using two levels of DPs. At the first level, a global distribution G_0 is drawn from DP(γ, H) with concentration parameter γ and base distribution H. At the second level, a Dirichlet process $\text{DP}(\alpha, G_0)$, which uses G_0 as the base distribution, independently generates M distributions $\{G_j(\boldsymbol{\theta})\}_{j=1}^{M}$: $G_j(\boldsymbol{\theta}) = \sum_{k=1}^{\infty} \pi_{jk} \delta(\boldsymbol{\theta}, \boldsymbol{\theta}_k)$. Each G_j generates a document $\mathbf{y}_j = \{y_{ji}\}_{i=1}^{N_j}$, where N_j is the number of words in document j and i indexes a word. The topics $\{\boldsymbol{\theta}_k\}$ are shared among the M documents $\{\mathbf{y}_j\}_{j=1}^{M}$. Fig. 5 shows the graphical model of the HDP. Let $\boldsymbol{\pi}_j = \{\pi_{jk}\}_{k=1}^{\infty}$. It follows that $\boldsymbol{\pi}_j \sim \text{DP}(\alpha, \boldsymbol{\beta})$, where $\boldsymbol{\beta}$ is a discrete distribution. Then, $\boldsymbol{\pi}_j$ can be obtained by the following stick-breaking construction:

$$\pi'_{jk} \sim \text{Beta}\left(\alpha\beta_k, \alpha\left(1 - \sum_{l=1}^{k}\beta_l\right)\right), \quad k=1,2,...,\infty \tag{11}$$

$$\pi_{jk} = \pi'_{jk} \prod_{l=1}^{k-1}(1-\pi'_{jl}) \quad k=1,2,...,\infty. \tag{12}$$

Then, the HDP is described by:

$$\begin{aligned} &\boldsymbol{\beta} \mid \gamma \sim \text{GEM}(\gamma) \\ &\boldsymbol{\theta}_k \mid \boldsymbol{\lambda} \sim H(\boldsymbol{\lambda}) && k=1,2,...,\infty \\ &\boldsymbol{\pi}_j \mid \alpha, \boldsymbol{\beta} \sim \text{DP}(\alpha, \boldsymbol{\beta}) && j=1,2,...,M \\ &z_{ji} \sim \boldsymbol{\pi}_j && j=1,2,...,M,\ i=1,2,...,N_j \\ &y_{ji} \sim \text{F}(\boldsymbol{\theta}_{z_{ji}}) && j=1,2,...,M,\ i=1,2,...,N_j. \end{aligned} \tag{13}$$

The HDP can be described as a Chinese Restaurant Franchise (CRF) process [4]. Each distribution generated from a Dirichlet distribution corresponds to a restaurant. The restaurants form a franchise and share the same menu. A distribution G_j generated from $\text{DP}(\alpha, G_0)$ corresponds to a low level restaurant. The customers in the restaurant sit at tables according to the Chinese restaurant process. The distribution G_0 generated from DP(γ, H) corresponds to the top level restaurant in which the customers are the tables in the low level restaurants and they sit also according to the Chinese restaurant process.

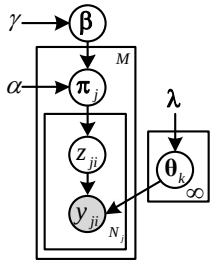


Fig. 5. The graphical model of the HDP

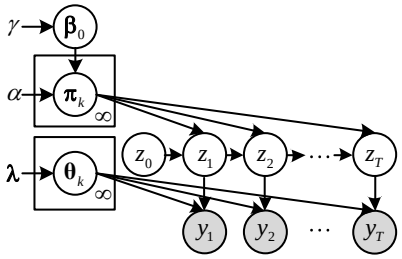


Fig. 6. The graphical model for the HDP-HMM.

3.5. HDP-HMM

The HDP-HMM [4] is an infinite state HMM with an HDP prior. A sequence of observations is represented by $(y_1, y_2, ..., y_T)$, and the sequence of their corresponding states is represented by $(z_1, z_2, ..., z_T)$. The states $(z_1, z_2, \ldots, z_T)$ are linked through a state transition matrix. Each observation y_t is sampled independently of the other observations conditional on z_t. An HMM is a dynamic system consisting of multiple mixture models. Each state k corresponds to a mixture model whose coefficients are the elements of the k-th row ($\boldsymbol{\pi}_k$) in the state transition matrix. The distribution of the row indexed by the current state z_t is used to generate the next state z_{t+1}. The observation y_{t+1} is sampled from the mixture component (topic) $\boldsymbol{\theta}_{z_{t+1}}$. The stick-breaking interpretation of the HDP-HMM is shown in Fig. 6. It is described by:

$$\begin{aligned} &\boldsymbol{\beta}_0 \sim \text{GEM}(\gamma) \\ &\boldsymbol{\theta}_k \mid \boldsymbol{\lambda} \sim H(\boldsymbol{\lambda}) && k=1,2,...,\infty \\ &\boldsymbol{\pi}_k \mid \alpha, \boldsymbol{\beta}_0 \sim \text{DP}(\alpha, \boldsymbol{\beta}_0) && k=0,1,...,\infty \\ &z_t \mid \{\boldsymbol{\pi}_k\}_{k=1}^{\infty}, z_{t-1} \sim \text{Dis}(\boldsymbol{\pi}_{z_{t-1}}) && t=1,2,...,T \\ &y_t \mid \{\boldsymbol{\theta}_k\}_{k=1}^{\infty}, z_t \sim \text{F}(\boldsymbol{\theta}_{z_t}) && t=1,2,...,T. \end{aligned} \tag{14}$$

It follows that all the mixture models in the HMM share the same topics generated by DP(γ, $\boldsymbol{\beta}_0$). The traditional HMM needs a known number of states, but the HDP-HMM can automatically deduce an appropriate number of states.

The HDP-HMM has the following limitations:

- Redundant states occur often, and the learnt state sequences tend to have fast switching between states.
- Each state only corresponds to a single modal distribution. Complex multimodal data may not be effectively modeled.

3.6. Sticky HDP-HMM

Fox et al. [15] proposed the sticky HDP-HMM to handle the above limitations in the HDP-HMM by adding a prior to augment the probability of self-transition of states. Fig. 7 shows the graphical model of the sticky HDP-HMM. It is seen that the only difference between the sticky HDP-HMM and the HDP-HMM is that a parameter κ is added into the sticky HDP-HMM. The generation of $\boldsymbol{\pi}_k$ is replaced by:

$$\boldsymbol{\pi}_k \sim \mathrm{DP}\left(\alpha+\kappa, \frac{\alpha\boldsymbol{\beta}_0+\kappa\boldsymbol{\delta}_k}{\alpha+\kappa}\right) \tag{15}$$

where $\kappa > 0$ causes an increase in the prior probability of self-transition, and $\boldsymbol{\delta}_k$ is an infinite dimensional vector whose k-th entry is 1 and all the other entries are 0. Using κ increases the probability of state k in the base distribution for generating $\boldsymbol{\pi}_k$, and then increases the probability of self-transition, making each state sticky. As a result, fast switching between states, as well as redundant states, are suppressed.

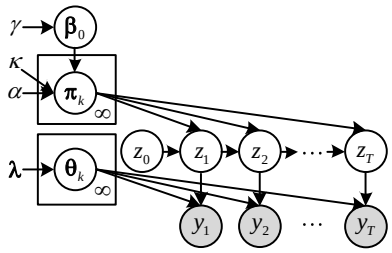


Fig. 7. The graphical model of the sticky HDP-HMM.

As shown in Fig. 8, for the sticky HDP-HMM, the observation model can be upgraded to a multimodal distribution with a DP prior. The observation model of each state is mixture models. The mixture weight vector $\boldsymbol{\psi}$ is generated by stick breaking construction: $\boldsymbol{\psi} \sim \mathrm{GEM}(\sigma)$. Given state z_t generated at time t, the label s_t of the mixture model component is generated from $\boldsymbol{\psi}_{z_t}$: $s_t \sim \boldsymbol{\psi}_{z_t}$. The observation y_t is generated by the mixture component jointly determined by z_t and s_t: $y_t \sim \mathrm{F}(\boldsymbol{\theta}_{z_t,s_t})$. An appropriate number of mixture components for each state can be automatically determined. Complex multimodal observations are hierarchically modeled: the global structure of the data is modelled by the HMM states and the local information in the data is modeled by the component labels.

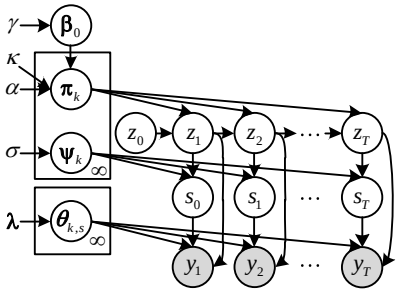


Fig. 8. The sticky HDP-HMM with multimodal observation models.

The limitation of the sticky HDP-HMM is that it is unable to cluster documents.

4. Dual Sticky HDP-HMM

To solve the limitation in the sticky HDP-HMM, we propose a new topic model, named a dual sticky HDP-HMM. It has the following properties:

- It is able to capture the temporal correlations between words in each document.
- It is able to automatically identify topics.
- It is able to cluster documents, and each document cluster is modeled by a HMM.
- All the HMMs share the same set of topics.
- A sticky prior is used to suppress fast switching between states and redundant states, making the model more robust to the variations in observations belonging to the same state.
- It is able to hierarchically construct observation models for multimodal data using both the HMM states and the component labels.
- It is able to model multi-observations for each HMM state, which can be used to directly handle the encoded generalized trajectories.
- The DPs are used to adaptively determine the number of document clusters, the number of topics, and the number of mixture components for each state.

In the following, we first propose the generative process of the dual sticky HDP-HMM, then infer the Gibbs sampling process for the dual sticky HDP-HMM, and finally present the Bayesian inference method for a given sample.

4.1. Generative process

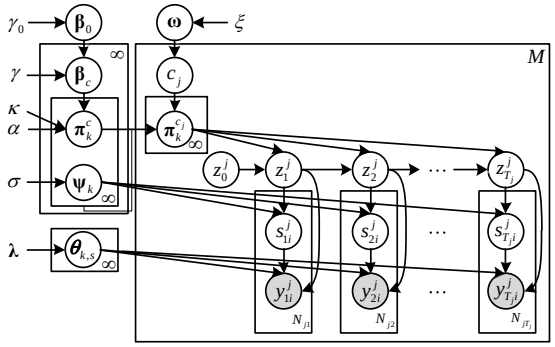


Fig. 9. The stick-breaking graphical model of the dual sticky HDP-HMM.

Fig. 9 shows the stick-breaking graphical model of the dual sticky HDP-HMM. Its generation process is outlined as follows:

Step 1: A random global distribution G_0 for topics (i.e., atomic activities) is generated from a Dirichlet process $\mathrm{DP}(\gamma_0, H)$ using the stick-breaking process:

$$G_0(\boldsymbol{\theta}) = \sum_{k=1}^{\infty} \beta_{0k}\delta(\boldsymbol{\theta}, \boldsymbol{\theta}_k) \tag{16}$$

where $\boldsymbol{\theta}_k$ denotes topic k. Each topic k corresponds to a mixture model: $\boldsymbol{\theta}_k = \{\boldsymbol{\theta}_{k,s}\}_{s=1}^{\infty}$. A component $\boldsymbol{\theta}_{k,s}$ of the mixture model is a discrete distribution over a word codebook (i.e., the set of sub-trajectory clusters/activities): $\boldsymbol{\theta}_{k,s} = \{p_{k,s}^w\}_{w=1}^V$, where $p_{k,s}^w$ is the probability that word (i.e., sub-trajectory cluster/activity) w occurs in the s-th component, and V is the size of the word codebook (i.e., the number of sub-trajectory clusters/ activities). The discrete distribution parameter vector $\boldsymbol{\theta}_{k,s}$ is drawn from the Dirichlet distribution $\mathrm{Dir}(\boldsymbol{\lambda})$, where λ is the parameter vector of the Dirichlet distribution H. Let $\boldsymbol{\beta}_0 = \{\beta_{0,k}\}_{k=1}^{\infty}$. Then,

$$\begin{aligned} &\boldsymbol{\beta}_0 \mid \gamma_0 \sim \mathrm{GEM}(\gamma_0) \\ &\boldsymbol{\theta}_{k,s} \mid \boldsymbol{\lambda} \sim \mathrm{Dir}(\boldsymbol{\lambda}), \end{aligned} \quad k = 1,2,...,\infty,\ s = 1,2,...,\infty. \tag{17}$$

Step 2: For each cluster c of documents (i.e., trajectories), a random distribution G_c of topics (i.e., atomic activities) is sampled from the Dirichlet process $\mathrm{DP}(\gamma, G_0)$:

$$G_c(\boldsymbol{\theta}) = \sum_{k=1}^{\infty} \beta_{c,k}\delta(\boldsymbol{\theta}, \boldsymbol{\theta}_k), \quad c = 1,2,...,\infty. \tag{18}$$

The sequence $\{\theta\}$ of topics (i.e., atomic activities)

is shared with the base distribution G_0. The probability sequence $\boldsymbol{\beta}_c = \{\beta_{c,k}\}_{k=1}^{\infty}$ is generated by the Dirichlet distribution with the base distribution $\boldsymbol{\beta}_0$:

$$\boldsymbol{\beta}_c \mid \gamma, \boldsymbol{\beta}_0 \sim \mathrm{DP}(\gamma, \boldsymbol{\beta}_0), \ c = 1, 2, ..., \infty. \quad (19)$$

Step 3: For any topic $\boldsymbol{\theta}'$ in the topic parameter space Θ, a topic transition distribution $G_{\boldsymbol{\theta}'}^c$ in document cluster c is generated from the Dirichlet process $\mathrm{DP}(\alpha, G_c)$ which uses G_c as the base distribution:

$$G_{\boldsymbol{\theta}'}^c(\boldsymbol{\theta}) = \sum_{k=1}^{\infty} \pi_{\boldsymbol{\theta}'k}^c \delta(\boldsymbol{\theta}, \boldsymbol{\theta}_k), \boldsymbol{\theta}' \in \Theta, c = 1, 2, ..., \infty \quad (20)$$

where the topic sequence $\{\boldsymbol{\theta}\}$ is shared with G_0 and G_c. Besides the topics $\boldsymbol{\theta}_k \in \{\boldsymbol{\theta}\}$ whose transition distributions are $G_{\boldsymbol{\theta}_k}^c$, we select an initial topic $\bar{\boldsymbol{\theta}}_0 \notin \{\boldsymbol{\theta}\}$, and define the initial topic transition distribution $G_{\bar{\boldsymbol{\theta}}_0}^c$. We denote the probability sequence $\{\pi_{\boldsymbol{\theta}_k k'}^c\}_{k'=1}^{\infty}$ of $G_{\boldsymbol{\theta}_k}^c$ as $\boldsymbol{\pi}_k^c (k \geq 1)$, and denote the probability sequence $\{\pi_{\bar{\boldsymbol{\theta}}_0 k'}^c\}_{k'=1}^{\infty}$ of $G_{\bar{\boldsymbol{\theta}}_0}^c$ as $\boldsymbol{\pi}_0^c$. They are generated from a Dirichlet process with a sticky prior:

$$\boldsymbol{\pi}_k^c \mid \alpha, \kappa, \boldsymbol{\beta}_c \sim \mathrm{DP}\left(\alpha + \kappa, \frac{\alpha \boldsymbol{\beta}_c + \kappa \boldsymbol{\delta}_k}{\alpha + \kappa}\right), \quad (21)$$
$$c = 1, 2, ..., \infty, \ k = 0, 1, 2, ..., \infty,$$

where κ is defined as in (15). Let $\boldsymbol{\Pi}^c = \{\boldsymbol{\pi}_k^c\}_{k=0}^{\infty}$. It is the state transition matrix for document cluster c.

Step 4: The weight vector $\boldsymbol{\psi}_k$ of the multinomial mixture for topic k is generated using the stick-breaking construction: $\boldsymbol{\psi}_k \mid \sigma \sim \mathrm{GEM}(\sigma)$.

Step 5: The prior distribution $\boldsymbol{\omega} = \{\omega_c\}_{c=1}^{\infty}$ of cluster labels of documents (i.e., trajectories) is generated by a stick breaking construction: $\boldsymbol{\omega} \mid \xi \sim \mathrm{GEM}(\xi)$, where $\xi > 0$ is a concentration parameter and ω_c is the probability of generating a document belonging to cluster c.

Step 6: A cluster label c_j for each document j is generated from the distribution $\boldsymbol{\omega}$: $c_j \mid \boldsymbol{\omega} \sim \mathrm{Dis}(\boldsymbol{\omega})$, $j = 1, 2, ..., M$, where M is the number of documents.

Step 7: The topic parameter vector $\bar{\boldsymbol{\theta}}_0^j$ for any document j at time 0 is fixed to $\bar{\boldsymbol{\theta}}_0$. The topic sequence $\{\bar{\boldsymbol{\theta}}_t^j\}_{t=1}^{T_j}$ for document j is generated from the topic transition matrix of document cluster c_j in the ascending order of time t (a time corresponds to the serial number of a point in a trajectory): $\bar{\boldsymbol{\theta}}_t^j \sim G_{\bar{\boldsymbol{\theta}}_{t-1}^j}^{c_j}$, $t = 1, 2, ..., T_j$, where T_j is the length of document j. Namely, for any document j, its topic at time 0 is labeled as 0. The topic label sequence $z^j = \{z_t^j\}_{t=1}^{T_j}$ for document j is generated from the state transition matrix $\boldsymbol{\Pi}^{c_j}$ for document cluster c_j in the ascending order of time t: $z_t^j \mid \{\boldsymbol{\Pi}^c\}_{c=1}^{\infty}, c_j, z_{t-1} \sim \mathrm{Dir}(\boldsymbol{\pi}_{z_{t-1}}^{c_j})$, $t = 1, 2, ..., T_j$. The topic at each time t is taken from the topic sequence $\{\boldsymbol{\theta}\}$ according to the topic label z_t^j: $\bar{\boldsymbol{\theta}}_t^j = \boldsymbol{\theta}_{z_t^j}$.

Step 8: The component label $s(i)_t^j$ of the i-th observation at time t for document j (corresponding to the i-th point of interest in frame t in video j modeled as a generalized trajectory) is drawn from the mixture distribution $\psi_{z_t^j}$ of the topic z_t^j for document j at time t: $s(i)_t^j \mid \{\boldsymbol{\psi}_k\}_{k=1}^{\infty}, z_t^j \sim \mathrm{Dis}(\boldsymbol{\psi}_{z_t^j})$, $i = 1, 2, ..., N_{j,t}$, where $N_{j,t}$ is the number of observations at time t for document j. This indicates that at each time t, $N_{j,t}$ observations can be generated for document j.

Step 9: The i-th word $y(i)_t^j$ at time t in document j is generated according to the discrete distribution jointly indicated by z_t^j and $s(i)_t^j$: $y_t^j(i) \mid \{\{\boldsymbol{\theta}_{k,s}\}_{s=1}^{\infty}\}_{k=1}^{\infty}, z_t^j, s(i)_t^j \sim \mathrm{Dis}(\boldsymbol{\theta}_{z_t^j, s(i)_t^j})$, $t = 1, 2, ..., T_j$.

It is noted that each document is generated by one of the HMMs. This ensures that clustering of documents takes place. Complex multimodal data are modeled hierarchically using the HMM states and the mixture states. Complex trajectories with local noise can be more accurately modeled than if more HMM states were used without the mixture states. The sticky prior which can avoid redundant states makes modeling multimodal observations more effective. The mixture component labels for each state ensure that the model can handle the generalized trajectories introduced in Section 2.2.

4.2. The learning method

The learning process for the dual sticky HDP-HMM is the reverse of its generative process. The task of learning is, given the documents $\{\mathbf{y}^j = \{y_t^j\}\}_{j=1}^{M}$, to infer the number C of document clusters, the number K of topics, the cluster label c_j and the hidden state sequence $\mathbf{z}^j = \{z_t^j\}$ for each document j, the topic sequence $\{\bar{\boldsymbol{\theta}}\}$, the global topic prior $\boldsymbol{\beta}_0$, the topic prior $\boldsymbol{\beta}_c$ for each document cluster c, and the state transition matrix $\{\boldsymbol{\pi}_k^c\}_{k=0}^{K}$ for each cluster c. We develop a Gibbs sampler by using an efficient truncated approximation of the DP [15]. The numbers of document clusters, topics, and components in an observation mixture for each state are limited by large numbers L_c, L_z, and L_s, respectively. We iteratively sample $\{\mathbf{z}^j\}$, $\{\mathbf{s}^j\}$, $\boldsymbol{\beta}_0$, $\{\boldsymbol{\beta}_c\}$, $\{\boldsymbol{\pi}_k^c\}$, $\{\boldsymbol{\psi}_k\}$, $\{c_j\}$, $\boldsymbol{\omega}$, and $\{\boldsymbol{\theta}_{k,s}\}$.

4.2.1. Sampling $\{\mathbf{z}^j, \mathbf{s}^j\}$

The stick-breaking construction process is truncated and then approximated by a Dirichlet distribution [20]. The following equations hold:

$$\begin{aligned} &\mathrm{GEM}_{L_z}(\gamma_0) \triangleq \mathrm{Dir}(\gamma_0 / L_z, \gamma_0 / L_z, ..., \gamma_0 / L_z) \\ &\mathrm{DP}_{L_z}(\gamma, \boldsymbol{\beta}_0) \triangleq \mathrm{Dir}(\gamma\beta_{0,1}, \gamma\beta_{0,2}, ..., \gamma\beta_{0,L_z}) \\ &\mathrm{DP}_{L_z}(\alpha, \boldsymbol{\beta}_c) \triangleq \mathrm{Dir}(\alpha\beta_{c,1}, \alpha\beta_{c,2}, ..., \alpha\beta_{c,L_z}) \end{aligned} \quad . \quad (22)$$

When $L_z \to \infty$, the approximation converges to the initial Dirichlet process [4]. When the learning is complete, the topics for which there are no samples are removed. In this way, an appropriate number of topics is obtained.

Application of the multiplication formula to the joint posterior probability of the hidden state sequence $\mathbf{z}^j$ for document j yields:

$$p(\mathbf{z}^j \mid \mathbf{y}^j, \{\Pi^c\}, c_j, \{\boldsymbol{\theta}\}, \{\boldsymbol{\psi}_k\}) = \prod_{t=1}^{T_j} p(z_t^j \mid z_{1:t-1}^j, \mathbf{y}^j, \Pi^{c_j}, \{\boldsymbol{\theta}\}, \{\boldsymbol{\psi}_k\}) \quad (23)$$

where c_j is substituted into $\{\Pi^c\}$. The conditional probability rule and the total probability theorem yield the equation:

$$p(z_t^j \mid z_{1:t-1}^j, \mathbf{y}^j, \boldsymbol{\Pi}^{c_j}, \{\boldsymbol{\theta}\}, \{\boldsymbol{\psi}_k\}) = \frac{p(z_{1:t}^j, \mathbf{y}^j, \boldsymbol{\Pi}^{c_j}, \{\boldsymbol{\theta}\}, \{\boldsymbol{\psi}_k\})}{\sum_{z_t=1}^{L_z} p(z_{1:t}^j, \mathbf{y}^j, \boldsymbol{\Pi}^{c_j}, \{\boldsymbol{\theta}\}, \{\boldsymbol{\psi}_k\})} \quad . \quad (24)$$

Let $\vartheta_t^j(z_t^j) = p(y_{t+1:T} \mid z_t, \boldsymbol{\pi}_{z_{t-1}^j}^{c_j}, \{\boldsymbol{\psi}_k\}, \{\boldsymbol{\theta}\})$. It is computed recursively using the backward algorithm [21] in HMM:

$$\vartheta_t^j(k) \propto \begin{cases} \sum_{k'=1}^{L_z} \left(\pi_{kk'}^{c_j} \left(\prod_{i=1}^{N_{j,t}} \sum_{s=1}^{L_s} p(s_{t,i}^j \mid \boldsymbol{\psi}_{z_t^j}) p(y_{t,i}^j \mid \boldsymbol{\theta}_{z_t^j, s_{t,i}^j}) \right) \vartheta_{t+1}^j(k') \right) & t < T_j \\ 1 & t = T_j \end{cases} \quad (25)$$

where $k = 1, 2, ..., L_z$ and $\pi_{kk'}^{c_j}$ is the probability of transitioning from state k to state k' for document cluster c_j. It represents the backward message transferred from time t+1 to time t in document j. According to the mathematical derivation in Appendix B which is available online, sequence $\mathbf{z}^j$ is obtained by recursively sampling z_t^j using the conditional distribution:

$$\begin{aligned} &p(z_t^j \mid z_{t-1}^j, \mathbf{y}^j, \boldsymbol{\Pi}^{c_j}, \{\boldsymbol{\theta}\}, \{\boldsymbol{\psi}_k\}) \\ &\propto p(z_t^j \mid \boldsymbol{\pi}_{z_{t-1}}^{c_j}) \left(\prod_{i=1}^{N_{j,t}} \sum_{s=1}^{L_s} p(s_{t,i}^j \mid \boldsymbol{\psi}_{z_t^j}) p(y_{t,i}^j \mid \boldsymbol{\theta}_{z_t^j, s_{t,i}^j}) \right) \vartheta_t^j(z_t^j). \end{aligned} \quad (26)$$

Namely, z_1^j is sampled from $p(z_1^j \mid z_0^j, \mathbf{y}^j, \Pi^{c_j}, \{\boldsymbol{\theta}\}, \{\boldsymbol{\psi}_k\})$, z_2^j is sampled from $p(z_2^j \mid z_1^j, \mathbf{y}^j, \Pi^{c_j}, \{\boldsymbol{\theta}\}, \{\boldsymbol{\psi}_k\})$, and so forth, until $z_{T_j}^j$ is sampled.

After z_t^j is sampled, according to Bayes' rule and Fig. 9, the mixture component label $s_{t,i}^j$ is sampled by:

$$p(s_{t,i}^j \mid \{\boldsymbol{\psi}_k\}, \boldsymbol{\theta}, y_{t,i}^j, z_t^j) \propto p(s_{t,i}^j \mid \boldsymbol{\psi}_{z_t^j}) p(y_{t,i}^j \mid \boldsymbol{\theta}_{z_t^j, s_{t,i}^j}) \quad (27)$$

where $p(s_{t,i}^j \mid \boldsymbol{\psi}_{z_t^j})$ is the prior probability and $p(y_{t,i}^j \mid \boldsymbol{\theta}_{z_t^j, s_{t,i}^j})$ is the conditional likelihood.

4.2.2. Sampling $\boldsymbol{\beta}_0$, $\{\boldsymbol{\beta}_c\}$, and $\{\boldsymbol{\pi}_k^c\}$

As stated in Section 3.4, the HDP can be represented by the Chinese Restaurant Franchise process. A distribution generated from a Dirichlet process corresponds to a restaurant. The Chinese Restaurant Franchise process is introduced to obtain the formulae for sampling $\boldsymbol{\beta}_0$, $\{\boldsymbol{\beta}_c\}$, and $\{\boldsymbol{\pi}_k^c\}$.

For sampling $\{\boldsymbol{\pi}_k^c\}$, we derive the posterior sampling formula $\boldsymbol{\pi}_k^c | \{\mathbf{z}^j\}, \{c_j\}, \boldsymbol{\beta}_c, \alpha$. For document j, let $n_{k,l}^j$ be the number of transitions from state k to state l. In the restaurant $R_{\pi_k^c}$ corresponding to the distribution π_k^c, the number $\bar{n}_{k,l}^c$ of the customers who have dish l is $\bar{n}_{k,l}^c = \sum_{j|c_j=c} n_{k,l}^j$, i.e., $\bar{n}_{k,l}^c$ can be obtained by counting the number of the transitions from state k to state l in the hidden state sequence for document cluster c. According to (3) and (22), the posterior sampling formula for $\{\boldsymbol{\pi}_k^c\}$ is:

$$\boldsymbol{\pi}_k^c | \{\mathbf{z}^j\}, \{c_j\}, \boldsymbol{\beta}_c, \alpha \sim \mathrm{Dir}(\alpha\beta_{c,1} + \bar{n}_{k,1}^c, ..., \alpha\beta_{c,k} + \kappa + \bar{n}_{k,k}^c, ..., \alpha\beta_{c,L_z} + \bar{n}_{k,L_z}^c) \quad (28)$$

where the sticky parameter κ is included in the k-th term on the right side of the formula.

Sampling $\{\boldsymbol{\beta}_c\}$ depends on the number of the customers who have dish l in the restaurant $R_{\boldsymbol{\beta}_c}$ corresponding to the distribution $\boldsymbol{\beta}_c$. As the same dish can be put on different dining tables in low level and middle level restaurants, the number $\breve{m}_{k,l}^c$ of the dining tables on which dish l is put in the restaurants $R_{\pi_k^c}$ is unknown. We sample $\{\breve{m}_{k,l}^c\}$ using a specifically designed Chinese Restaurant process [15]. The posterior sampling formula for $\{\boldsymbol{\beta}_c\}$ is inferred by conjugate updates. The details of the mathematical derivation are given in Appendix B which is available online.

Sampling $\boldsymbol{\beta}_0$ depends on the number $\tilde{m}_l^c$ of the dining tables on which dish l is put in the restaurant $R_{\boldsymbol{\beta}_c}$. It is unknown. We design a specific Chinese Restaurant processes [15] to sample $\{\tilde{m}_l^c\}$. The posterior sampling formula for $\boldsymbol{\beta}_0$ is inferred by conjugate updates. For details of the mathematical derivation, see Appendix B.

4.2.3. Sampling $\{c_j\}$

Sampling $\{c_j\}$ has the approximation: $\mathrm{GEM}_{L_c}(\xi) \triangleq \mathrm{Dir}(\xi/L_c, \xi/L_c, ..., \xi/L_c)$ where, as aforementioned, L_c is the upper limit of the number of document clusters. According to Bayes' rule and Fig. 9, the marginal posterior distribution of c_j is given by:

$$\begin{aligned} &p(c_j = c \mid c^{-j}, \mathbf{z}^j, \{\boldsymbol{\pi}_k^c\}, \xi) \\ &\propto p(c_j = c \mid c^{-j}, \xi) p(\mathbf{z}^j \mid \{\boldsymbol{\pi}_k^c\}, c_j = c) \end{aligned}, \quad c = 1, 2, ..., L_c \quad (29)$$

where c^{-j} is the set of the cluster labels of all the documents excluding document j. According to (4), the following formula holds:

$$p(c_j = c \mid c^{-j}, \xi) \propto \frac{\xi / L_c + \hat{n}_c^{-j}}{\xi + M - 1} \quad (30)$$

where $\hat{n}_c^{-j}$ is the number of the documents in cluster c excluding document j. According to the multiplication formula, the following equation holds:

$$\begin{aligned} &p(\mathbf{z}^j \mid \{\boldsymbol{\pi}_k^c\}_k, c_j = c) = \prod_{t=1}^{T_j} p(z_t^j \mid z_{t-1}^j, \{\boldsymbol{\pi}_k^c\}) \\ &= \prod_{t=1}^{T_j} \pi_{z_{t-1}^j, z_t^j}^c = \prod_{k=1}^{L_c} \prod_{l=1}^{L_c} (\pi_{k,l}^c)^{n_{k,l}^j}. \end{aligned} \quad (31)$$

Substitution of (30) and (31) into (29) yields:

$$p(c_j = c \mid c^{-j}, \mathbf{z}^j, \{\boldsymbol{\pi}_k^c\}, \xi) \propto \frac{\xi / L_c + \hat{n}_c^{-j}}{\xi + M - 1} \prod_{k=1}^{L_z} \prod_{l=1}^{L_z} (\pi_{k,l}^c)^{n_{k,l}^j}. \quad (32)$$

4.2.4. Sampling $\boldsymbol{\psi}_k$, $\boldsymbol{\omega}$, and $\boldsymbol{\theta}_{k,s}$

The posterior probability formulae $\boldsymbol{\psi}_k | \{\mathbf{z}^j\}, \{\mathbf{s}^j\}, \sigma$, $\boldsymbol{\omega} | \{c_j\}$, and $\boldsymbol{\theta}_{k,s} | \{\mathbf{z}^j\}, \{\mathbf{s}^j\}, \{\mathbf{y}^j\}$ for sampling $\boldsymbol{\psi}_k$, $\boldsymbol{\omega}$, and $\boldsymbol{\theta}_{k,s}$ respectively are derived by conjugate updates. Details of the mathematical derivation are given in Appendix B.

4.2.5. The Gibbs sampling procedure

In the Gibbs sampling procedure, we iteratively sample the variables $\{\mathbf{z}^j\}$, $\{\mathbf{s}^j\}$, $\boldsymbol{\beta}_0$, $\{\boldsymbol{\beta}_c\}$, $\{\boldsymbol{\pi}_k^c\}$, $\{\boldsymbol{\psi}_k\}$, $\{c_j\}$, $\boldsymbol{\omega}$, and $\{\boldsymbol{\theta}_{k,s}\}$ using the above posterior conditional probability formulae. When a variable is sampled, the other variables take the values of their most recent estimates. Maximum a posteriori estimation (MAP) is used to obtain the most probable value of the variable. The Gibbs sampler converges when stable values of the variables are obtained. In practice, we empirically determine the number of the iterations required to obtain convergence.

4.2.6. Determination of the numbers of clusters

After the learning is complete, there are a number of null clusters of documents, null topics, and null mixture components for each hidden state, with which no samples are associated. These null clusters are removed. Then, the numbers of the remaining clusters of documents, topics, and mixture components for each state are just the determined numbers. The closer the initial values of these numbers are to the true numbers, the more efficient the convergence.

4.3. Bayesian inference

Given the learnt dual sticky HDP-HMM, for a given document $\mathbf{y}$ we infer its likelihood for a cluster, its cluster membership, and its latent topic sequence.

4.3.1. Likelihood estimation

Given the mode parameters $\prod^c$ and $\{\boldsymbol{\theta}\}$, the likelihood $l_c(\mathbf{y}) = p(\mathbf{y} \mid \boldsymbol{\Pi}^c, \{\boldsymbol{\theta}\})$ of a given document $\mathbf{y}$ for document cluster c can be estimated efficiently using the back-forward algorithm [21] in HMM:

$$\begin{aligned} &l_c(\mathbf{y}) = p(\mathbf{y} \mid \Pi^c, \{\boldsymbol{\theta}\}) \\ &\propto \sum_{k'=1}^{L_z} \left(\pi_{kk'}^c \left(\prod_{i=1}^{N_{j,1}} \sum_{s=1}^{L_y} p(s_{1,i}^j \mid \boldsymbol{\psi}_{z_1^j}) p(y_{1,i}^j \mid \boldsymbol{\theta}_{z_1^j, s_{1,i}^j}) \right) \vartheta_1^j(k') \right) \end{aligned} \quad (33)$$

where $\vartheta_1^j(k')$ is recursively estimated using (25).

The likelihood can be used to detect anomalous documents. We look for the document cluster $c*$ that has the maximum likelihood with document $\mathbf{y}$:

$$c* = \arg\max_c l_c(\mathbf{y}). \quad (34)$$

If $l_c(\mathbf{y})$ is less than a threshold Ω_{c*}, then $\mathbf{y}$ is treated as anomalous. We calculate the likelihood $l_{c*}(\mathbf{y}^j)$ of each document $\mathbf{y}^j$ in the cluster $c*$ and take the minimum as the threshold Ω_{c*}:

$$\Omega_{c*} = \min_{j|c_j=c*} l_{c*}(\mathbf{y}^j). \quad (35)$$

4.3.2. Cluster prediction

According to Bayes' rule and Fig. 9, the posterior distribution of the cluster label $c_{\mathbf{y}}$ of a given document $\mathbf{y}$ is represented by:

$$p(c_{\mathbf{y}} = c \mid \mathbf{y}, \{c_j\}_{j=1}^M, \{\prod^c\}, \boldsymbol{\theta}, \xi) \propto p(c_{\mathbf{y}} = c \mid \{c_j\}_{j=1}^M, \xi) p(\mathbf{y} \mid \prod^c, \{\boldsymbol{\theta}\}) \quad (36)$$

According to (4), $p(c_{\mathbf{y}} = c \mid \{c_j\}_{j=1}^M, \xi) \propto (\xi / L_c + \hat{n}_c) / (\xi + M)$, where $\hat{n}_c$ is the number of the documents in the learnt cluster c. Then, (36) is transformed to

$$p(c_{\mathbf{y}} = c \mid \mathbf{y}, \{c_j\}_{j=1}^M, \{\prod^c\}, \boldsymbol{\theta}, \xi) \propto \frac{\xi / L_c + \hat{n}_c}{\xi + M} l_c(\mathbf{y}). \quad (37)$$

We predict the cluster label $\hat{c}$ of $\mathbf{y}$ by:

$$\hat{c} = \arg\max_c p(c_{\mathbf{y}} = c \mid \mathbf{y}, \{c_j\}_{j=1}^M, \{\prod^c\}, \{\boldsymbol{\theta}\}, \xi). \quad (38)$$

4.3.3. Topic estimation

Given the predicted cluster label $\hat{c}$ of $\mathbf{y}$, we estimate the topic sequence $\mathbf{z}$ of $\mathbf{y}$ conditional on $\prod^c$ and $\{\boldsymbol{\theta}\}$ using the Viterbi algorithm [21] in HMM. A variable $\varphi_t(k)$ is defined as follows:

$$\varphi_t(k) = \arg\max_{\mathbf{z}_{1:t-1}} p(\mathbf{y}_{1:t}, \mathbf{z}_{1:t-1}, z_t = k \mid \prod^{\hat{c}}, \{\boldsymbol{\theta}\}). \quad (39)$$

It represents the maximum probability for a possible part state sequence $\mathbf{z}_{1:t-1}$ when part document $\mathbf{y}_{1:t}$ is observed and $z_t = k$. The values of $\{\varphi_t(k)\}_{t=1}^T$ are computed recursively [21] by:

$$\varphi_t(k)=\begin{cases}\max\limits_{k'}\left(\varphi_{t-1}(k')\pi^{\hat{c}}_{k',k}\right)\prod\limits_{i=1}^{N_t}\sum\limits_{s=1}^{L_s}p(s_{t,i}\mid\psi_{z_t^i})\,p(y_{t,i}\mid\boldsymbol{\theta}_{z_t,s_{t,i}}) & t>1\\ \pi_{0k}\prod\limits_{i=1}^{N_1}\sum\limits_{s=1}^{L_s}p(s_{1,i}\mid\psi_{z_t})\,p(y_{1,i}\mid\boldsymbol{\theta}_{z_1,s_{1,i}}) & t=1\end{cases} \quad (40)$$

where $k=1,2,...,K$. Another variable $\eta_t(k)$ is defined as follows:

$$\eta_t(k)=\arg\max_{k'}(\varphi_{t-1}(k')),\ \ t=2,3,...,T. \quad (41)$$

It represents the most possible state at time *t*-1, when the state is *k* at time *t*. The states at different times are estimated backward from time *T* until time 1 as follows:

$$\hat{z}_t=\begin{cases}\eta_{t+1}(\hat{z}_{t+1}) & t<T\\ \arg\max\limits_{k'}(\eta_T(k')) & t=T.\end{cases} \quad (42)$$

5. Learning Trajectory Patterns

Trajectories are typical time series data. We apply the proposed dual sticky HDP-HMM to trajectory modeling. The trajectory samples with the word-document style representation described in Section 2 are input to learn the dual sticky HDP-HMM. The dual sticky HDP-HMM with a single observation at each time is used to model ordinary trajectories. The dual sticky HDP-HMM with multi-observations at each time is used to model generalized trajectories.

After the dual sticky HDP-HMM is learnt, several trajectory clusters are found, and each motion pattern is modeled as a sequence of atomic activities, where each state is considered as an atomic activity. Each $\boldsymbol{\beta}_c$ is a discrete distribution on the labels of atomic activities, and corresponds to a cluster of actions consisting of atomic activities. All the actions share the same atomic activities. For each action cluster, a transition matrix is obtained. The sticky prior makes the model more robust to the variations in a given atomic activity. Each atomic activity is a distribution for the words corresponding to the observations of the atomic activity and modeled by a mixture of multinomials. A component of an atomic activity corresponds to a discrete distribution on the codebook.

6. Semantic Description of Motions

We define a trajectory-based motion pattern as a set of motions that have the same origin, motion process, motion modes, and destination. A semantic description of trajectory-based object motions is obtained using the following four stages:

- learning sources and sinks in the scene by clustering the endpoints of trajectories to find the areas in which objects usually appear or disappear,
- learning the regions of interest, in which objects usually move,
- learning semantic atomic activities in the scene using the dual sticky HDP-HMM,
- generation of natural language description for object motions based on the latent topic sequences predicted by Bayesian inference.

6.1. Source and sink modeling

We automatically determine the sources and sinks that objects enter to or exit from by clustering the starting points and the terminal points of trajectories, respectively. The sources and sinks are modeled as GMMs, in which each Gaussian component corresponds to a source or sink. The center of the source or sink corresponds to the mean of the Gaussian component. Its shape and extension are reflected in the covariance matrix. How often objects enter or exit through the source or sink is indicated by the weight of the component in the GMM. In order to automatically determine the number of Gaussians from data, we use the DP as the nonparametric prior on the parameters of GMMs. The model is called the DP Gaussian mixture model (DP-GMM).

6.1.1. DP-GMM

The graphical model of the DP-GMM is shown in Fig. 4, where the parameter vector $\boldsymbol{\pi}=(\pi_k)_{k=1}^{\infty}$ yields the weights of the components in the GMM, and $\boldsymbol{\theta}_k$ is the *k*-th component's distribution parameter containing the mean vector $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$ of a Gaussian distribution. Let $\mathcal{W}^{-1}$ be the inverse-Wishart (IW) distribution:

$$\mathcal{W}^{-1}(\boldsymbol{\Sigma}\mid\upsilon,\boldsymbol{\Delta})=p(\boldsymbol{\Sigma}\mid\upsilon,\boldsymbol{\Delta})\propto|\boldsymbol{\Sigma}|^{-\left(\frac{\upsilon+d+1}{2}\right)}\exp\left\{-\frac{1}{2}\mathrm{trace}\left(\upsilon\boldsymbol{\Delta}\boldsymbol{\Sigma}^{-1}\right)\right\}, \quad (43)$$

where $\boldsymbol{\Delta}$ is a $d\times d$ positive definite matrix and υ is a scalar value representing the number of degrees of freedom [29]. The base probability measure is chosen as the normal-inverse-Wishart (NIW) distribution which is defined as:

$$p(\boldsymbol{\mu},\boldsymbol{\Sigma}\mid e,\boldsymbol{\mu}_0,\upsilon,\boldsymbol{\Delta})=\mathcal{N}(\boldsymbol{\mu}\mid\boldsymbol{\mu}_0,\boldsymbol{\Delta}/e)\mathcal{W}^{-1}(\boldsymbol{\Sigma}\mid\upsilon,\boldsymbol{\Delta}), \quad (44)$$

where $\mathcal{N}$ is the normal distribution, $\boldsymbol{\mu}_0$ is the prior mean for $\boldsymbol{\mu}$, and *e* is a measure of the degree of belief in the prior $\boldsymbol{\mu}_0$. The NIW distribution is the conjugate prior of the multivariate normal distribution.

6.1.2. Inference

Traditionally the parameters of DP mixture models are inferred by Chinese restaurant process (CRP) based-Gibbs samplers. This kind of sampler runs slowly because the statistical parameters have to be updated every time in order to sample an observation's component label. Referring to the work of Ishwaran and Zarepour [56], Ishwaran and James [57], Kivinen et al. [58] and Fox et al. [17] on truncated approximation to a DP, we develop a simple blocked Gibbs sampler for the inference of the DP-GMM. A large number *L* is chosen as an upper bound of the number of components. The DP-GMM is approximated by the truncated process:

$$\begin{aligned}&\boldsymbol{\pi}\mid\alpha\sim\mathrm{Dir}(\alpha/L,...,\alpha/L)\\ &z_i\mid\boldsymbol{\pi}\sim\mathrm{Dis}(\boldsymbol{\pi}) && i=1,2,...,N\\ &\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\mid e,\boldsymbol{\mu}_0,\upsilon,\boldsymbol{\Sigma}\sim\mathrm{NIW}(e,\boldsymbol{\mu}_0,\upsilon,\boldsymbol{\Sigma}) && k=1,2,...,L\\ &\mathbf{y}_i\mid\{\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\}_{k=1}^{L},z_i\sim\mathcal{N}(\boldsymbol{\mu}_{z_i},\boldsymbol{\Sigma}_{z_i}) && i=1,2,...,N.\end{aligned} \quad (45)$$

When $L\to\infty$, this finite mixture model converges to the DP-GMM. At the end of the inference stage, the components without samples are removed. The blocked Gibbs sampler iteratively samples $\mathbf{z}_{1:N}$, $\boldsymbol{\mu}_{1:L}$, $\boldsymbol{\Sigma}_{1:N}$, and $\boldsymbol{\pi}_{1:L}$ until convergence.

According to Bayes' rule, the posterior probability distribution for sampling the component label z_i of the *i*-th sample is derived as:

$$\begin{aligned}&p(z_i=k\mid\boldsymbol{\pi},\boldsymbol{\theta},y_i)\propto p(z_i=k\mid\boldsymbol{\pi},\boldsymbol{\theta})\,p(y_i\mid z_i=k,\boldsymbol{\pi},\boldsymbol{\theta},)\\ &=\pi_k\mathcal{N}(\mathbf{y}_i\mid\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k),\quad k=1,...,L\end{aligned}. \quad (46)$$

Due to the conjugacy between the NIW and the normal distribution, the posterior distribution of the parameters $\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k$ of the *k*-th component is also a NIW: $\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\sim\mathrm{NIW}(\bar{e}_k,\bar{\boldsymbol{\mu}}_k,\bar{\upsilon}_k,\bar{\boldsymbol{\Delta}}_k)$. The parameters $\bar{e}_k,\bar{\boldsymbol{\mu}}_k,\bar{\upsilon}_k,\bar{\boldsymbol{\Delta}}_k$ of this NIW are updated by:

$$\begin{aligned}&\bar{e}_k=e+\textstyle\sum_i\delta(z_i,k)\\ &\bar{\boldsymbol{\mu}}_k=\frac{e\boldsymbol{\mu}_0+\sum_{i\mid z_i=k}\mathbf{y}_i}{\bar{\kappa}_k}\\ &\bar{\upsilon}_k=\upsilon+\textstyle\sum_i\delta(z_i,k)\\ &\bar{\boldsymbol{\Delta}}_k=\frac{\upsilon\boldsymbol{\Delta}+\sum_{i\mid z_i=k}\mathbf{y}_i\mathbf{y}_i^{T}+e\boldsymbol{\mu}_0\boldsymbol{\mu}_0^{T}-\bar{e}_k\bar{\boldsymbol{\mu}}_k\bar{\boldsymbol{\mu}}_k^{T}}{\bar{\upsilon}_k}.\end{aligned} \quad (47)$$

The posterior distribution is an updated NIW distribution in the same form as $H(\boldsymbol{\lambda})$. The weight vector $\boldsymbol{\pi}$ is sampled by:

$$\boldsymbol{\pi}\sim\mathrm{Dir}(\alpha/L+n_1,\alpha/L+n_2,...,\alpha/L+n_L) \quad (48)$$

where n_k is the number of samples assigned to the *k*-th component.

6.1.3. Model validation

Because of detection and tracking errors, some trajectories may have noisy endpoints that yield incorrect sources or sinks. The components corresponding to the incorrect sources and sinks usually have low weights and large variances. They are discarded using the following criterion [28]: the component *k* is judged to be an outlier if $\log\pi_k-0.5\log|\boldsymbol{\Sigma}_k|<\varepsilon$, where ε is a predefined threshold.

6.2. Learning motion regions

We cluster points in all the trajectories to learn the

regions of interest in which objects usually move. For a point in a trajectory, we extract its position coordinates, angle for representing the direction of motion velocity, delta of the motion direction in contrast with the previous point, and frame number as its features. After the one dimensional median filtering is carried out on each feature, the five features are weighted to form a five dimensional vector for the point. The DP-GMM is applied to the feature vectors of all the points in order to cluster them. From the obtained clusters, we compute the densities of points in the clusters and select the clusters in which points are denser around the centers. We draw the envelope curve of the points in each selected cluster using the convex envelope method. These envelope curves indicate the regions of interest in the scene, forming a semantic map of the scene.

6.3. Learning of semantic atomic activities

Each topic learned by the dual sticky HDP-HMM corresponds to an atomic activity. But topics are located at particular positions in the image and it is difficult for users to assign predicates to a large number of topics. Therefore, we further generalize the topics to obtain position-free atomic activities which are determined by the object motion directions. We describe each topic using a predicate describing the generalized atomic activities and an adverb describing the current position of the topic.

The generalized atomic activities are learned by clustering the directions of all the topics. We represent the direction features for each topic using the pyramid histogram of gradients (PHoG) extracted from the sub-trajectories belonging to the topic. The PHoG is invariant under spatial translations. For each sub-trajectory $\{x_t, y_t\}_{t=1:T}$, we compute the sequence of oriented gradients $G=\{g_t\}_{t=1:T-1}$, where $g_t = \arctan(x_{t+1}-x_t, y_{t+1}-y_t)$. We divide G into two parts: $G_1=\{g_t\}_{t=1:T/2}$ and $G_2=\{g_t\}_{t=T/2:T-1}$. From G_1, G_2, and G, we compute three histograms h_1, h_2 and h, respectively. The feature vector for this sub-trajectory is [h_1, h_2, 0.5h]. The feature vectors for all the sub-trajectories belonging to a given topic are averaged to produce a single vector which is used to represent this topic. The χ^2 distance is used to measure the similarity between any two topics. We use the DP-GMM to cluster topics into the generalized atomic activities.

The number of the generalized atomic activities is much less than the number of topics. This makes it feasible for users to assign manually a predicate, such as "turn right", to a generalized atomic activity. In particular, in a traffic scene there are two types of generalized atomic activities for vehicles: one is that the vehicles move straight in a fixed direction, and the other is that the vehicles turn from one direction to another, as illustrated in Fig. 10. Furthermore, the PHoG is more discriminative than the traditional HoG. As illustrated in Fig. 10 (b), the HOGs computed from the two oriented sub-trajectory curves are similar, but the PHoGs for them are quite different. The generalized atomic activities are the primitives that compose actions and have easily understandable semantic meanings suitable to form descriptions of actions in natural language.

(a) (b)

Fig. 10. Two types of generalized atomic activities: (a) moving straight; (b) turning

6.4. Estimating semantic meanings of speed

Semantic meanings of speed (i.e. "normal speed", "high speed", and "low speed") can be acquired statistically. The mean μ_s and covariance σ_s of values of speed in the sample trajectories are calculated. "Normal speed" lies between $\mu_s-\sigma_s$ and $\mu_s+\sigma_s$. "Low speed" is lower than $\mu_s-\sigma_s$. "High speed" is higher than $\mu_s+\sigma_s$. So, the keywords for the speed information can be automatically assigned to each topic.

6.5. Natural language description for motions

A trajectory-based action is a sequence of generalized atomic activities. An action is distinguished by the combination of the source, the sink, the path of the movement, and the moving process. The source and the sink have clear semantic meanings. We need to describe the temporal structure of the action. Using the method in Section 4.3.3, we estimate the topic sequence of the action. We describe a topic using the predicate describing the generalized atomic activity corresponding to the topic and the adverb describing where the topic exists in the image. For example, a topic is described as "move ahead in Region A" or "turn left in Region B". Then, an action is described using a combination of sentences. First, a sentence describes where the object enters the scene. Second, topics estimated based on the learnt dual sticky HDP-HMM are described by sentences. Third, a sentence describes where the object disappears from the scene. The semantic description is not output for every topic. The output is only activated when a generalized atomic activity different from the most recent generalized atomic activity occurs, or the object is entering a new region, or the semantic information about the speed is changed.

Generating natural language descriptions needs grammar rules. We introduce simple grammars for generating natural language descriptions, as in most traffic surveillance scenarios, it is usually only necessary to answer the questions like "Who does what at where? And How?" The grammar rules are listed as follows:

-- sentence = sub-sentence + punctuation;
-- sub-sentence = subject + predicate [+ object] [+ adverb];
-- subject = [modifier+] noun;
-- predicate = [pre-modifier +] verb/verb phrase [+ post- modifier];
-- object = [modifier +] noun.

The items in square brackets are optional. We integrate the semantic map of the scene with the predicates for the generalized automatic activities to map trajectory-based actions to natural language. The noun and verb/verb phrase are chosen using the extracted information. The modifier for the noun mainly includes the target's size, color etc. The modifier for the verb/verb phrase mainly includes information about speed etc. The adverb usually includes the "where" information. For instance, the descriptions can be "a red vehicle enters the scene from Source A", and "a red vehicle turns right in Region B".

The description has uncertainty due to noise in the trajectories and the uncertainty of the learning model. Based on (37), we estimate the description uncertainty using the following normalized cluster prediction probability:

$$\frac{p(c_y=\hat{c}\,|\,\mathbf{y},\{c_j\}_{j=1}^{M},\{\prod^{\hat{c}}\},\boldsymbol{\theta},\xi)}{\sum_{c} p(c_y=c\,|\,\mathbf{y},\{c_j\}_{j=1}^{M},\{\prod^{c}\},\boldsymbol{\theta},\xi)}. \tag{49}$$

We can output the uncertainty by a sentence such as "The probability for the above descriptions is about 61%". If an anomaly is detected by (34), the following sentence can be output: "This action may be an anomaly".

7. Experimental Results

To demonstrate the claimed contributions of the proposed method, we evaluated, in succession, the correctness of the Gibbs sampler, the learning performance of the dual sticky HDP-HMM, and the ability to generate natural language descriptions of motions.

7.1. Correctness of Gibbs sampling

To validate our Gibbs sampler, we randomly generated two classes of sequential documents from our model according to two different transition matrices. Each class consists of three sequential documents and the size (length) of each sequential document is 500. The state labels of each document are determined by the initial states and the state transition matrix. The observation model is a discrete distribution for which the size of the vocabulary is 20. We removed the class labels and hidden state information from the generated sequential samples. Then, these samples were input to our Gibbs sampler to train the dual sticky HDP-HMM. The estimated hidden states and class labels

were compared with the true states and the true classes. The normalized Hamming distance was used to evaluate their dissimilarities. Fig. 11 shows the changes in the Hamming distance for the six documents when the iteration number is increased to 200. Detailed results are shown in Appendix C which is available online. It is seen that after only a few iterations the state sequences show a stable convergence, starting from randomly initialized state sequences. Fig. 12 shows the changes in the normalized Hamming distance of the class labels of the documents, when the number of iterations is increased. It is seen that the correct class labels are obtained after 45 iterations. It is shown that our Gibbs sampler mixes well and that the model correctly uncovers the structure from the samples.

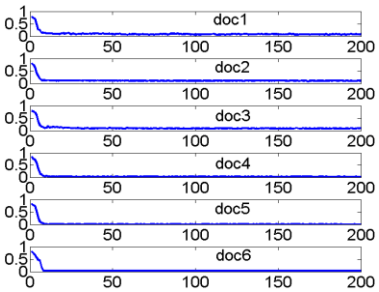


Fig. 11. The state Hamming distances for the six documents.

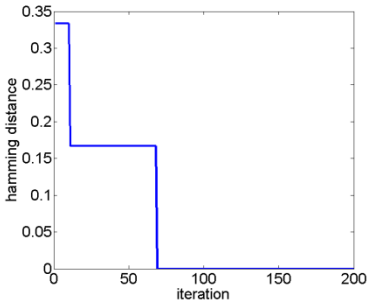


Fig. 12. The normalized Hamming distance between the ground truth class labels and the estimated class labels of the documents.

7.2. Learning performance of dual sticky HDP-HMM

The following two criteria were used to evaluate the learning performance:

- **Clustering accuracy (CA):** Let C be the number of the learnt clusters, let $\hat{n}_i$ be the number of the trajectories in the i-th learnt cluster, and let $\hat{m}_i$ be the number of the trajectories which have the same ground truth cluster label and occupy the highest proportion in the i-th learnt cluster. The clustering accuracy (CA) is defined as:

$$\frac{1}{C}\sum_{i=1}^{C}\frac{\hat{m}_i}{\hat{n}_i}. \tag{50}$$

- **Correct clustering rate (CCR):** The CCR is based on a one-to-one mapping between the labels of the clusters and the ground truth labels. For the i-th pair of labels in the mapping, let c_i^{out} be the label of the cluster, and let c_i^{truth} be the corresponding ground truth label. Let $D_{c_i^{out}}$ be the set of samples in the cluster with label c_i^{out}. Let $D_{c_i^{truth}}$ be the set of samples with the ground truth label c_i^{truth}. It is required to find a one-to-one mapping to maximize $\sum_i |D_{c_i^{out}} \cap D_{c_i^{truth}}|$, where $|.|$ is the number of elements in the set herein. This is a typical assignment problem which can be solved using the Munkres algorithm [19]. Under the found mapping, the correct clustering rate (CCR) [9, 10] is defined as:

$$\frac{1}{M}\sum_{i=1}^{\min\left(C^{out},C^{truth}\right)}\left|D_{c_i^{out}} \cap D_{c_i^{truth}}\right| \tag{51}$$

where M is the number of samples, C^{out} is the number of clusters, and C^{truth} is the number of ground truth labels. If $C^{out} > C^{truth}$ or $C^{out} < C^{truth}$, then the CCR is reduced. Therefore, CCR tests more things than the CA.

The number of iterations was set to 3000. The hyper-parameters γ_0, γ, α, κ, σ, and ξ in the dual sticky HDP-HMM were given non-informative priors and also sampled by the Gibbs sampler. For more details, please refer to [17].

We used the following four benchmark datasets to test the learning performance of the dual sticky HDP-HMM: the hand sign dataset, the human action dataset, the synthetic trajectory dataset, and the traffic trajectory dataset.

7.2.1. The hand sign dataset

The hand sign dataset is from the Australian sign language collection. There are 35 clusters of trajectories of hand sign words. Each cluster consists of 20 trajectories of hand movements of different signers. As in the previous work [26, 30] on trajectory analysis, we only use the 2D (x, y)-coordinate features from this dataset, because the (x, y) coordinates are the fundamental features of trajectories. As shown in Appendix D, trajectories have large noise and large within-class spread.

We used wavelet decomposition combined with median filtering to smooth the trajectories. After that, the method in Section 2.1 was used to segment trajectories and represent each trajectory as a sequence of visual words. The dimension of the PCA features was 10. The size of the code book was 1000. The sequences of visual words were used to learn a dual sticky HDP-HMM. The initial numbers of topics, trajectory clusters, and components for each state were set to 100, 50, and 10, respectively.

Table 2. The clustering results for the different methods on the hand sign trajectory dataset

Methods		CCR(%)	CA(%)	Estimated number of clusters
K-means	with ground truth number	18.19	39.31	×
	with number by our method	17.98	34.87	
Spectral clustering	ground truth number	22.93	32.13	×
	with number by our method	20.77	32.95	
Dual-HDP-HMM		17.13	31.87	37
Dual sticky HDP-HMM		21.43	35.03	34

Table 2 compares our dual sticky HDP-HMM with the K-means method, the spectral clustering method, and a dual HDP-HMM which is obtained by removing the sticky prior and multimodal observation modeling from the dual sticky HDP-HMM. The K-means method measures trajectory similarities using the Euclidean distance between trajectories [7], while the spectral clustering method uses the DTW distance [8]. For the K-means and spectral clustering methods, we show both the results when the number of clusters was set to the ground truth number of clusters and to the number of clusters obtained by our dual sticky HDP-HMM. From the table, the following points are noted:

- There is inconsistency between the two measures CA and CCR. For example, the CCR for the K-means method with the ground truth number of labels is lower than the CCR for the dual sticky HDP-HMM and the CCR for the spectral clustering method. The CA for the K-means method is the highest among all of the methods.
- For each method, its CCR is lower than its CA. This is because the CCR requires a one-to-one mapping between the labels of the obtained clusters and the ground truth labels, but the CA allows each ground truth label to correspond to one or more clusters.
- Under the CCR criterion, the clustering result of the dual sticky HDP-HMM is slightly lower than the best result that the spectral clustering method with the ground truth number of clusters yields, but it is higher than the result obtained by the spectral clustering method with the number of clusters estimated by the dual sticky HDP-HMM. The result of the dual sticky HDP-HMM is even higher than that of the K-means method with the ground truth number of clusters.
- Under the CA criterion, the clustering result of the dual sticky HDP-HMM is lower than the best result yielded by the K-means method with the ground truth number of clusters, but is higher than the results obtained by the spectral clustering method with the ground truth number

of clusters and by the *K*-means method with the number of clusters estimated by the dual sticky HDP-HMM.

- The dual sticky HDP-HMM yields better results than the simplified dual HDP-HMM without the sticky prior and multimodal observation modeling. This indicates that the sticky prior and multimodal observation modeling in the dual sticky HDP-HMM enhance the robustness to large noise and large within-class differences for complex trajectories.
- Both the numbers of clusters of trajectories estimated by the dual sticky HDP-HMM and the simplified dual HDP-HMM are much less than the initial number of clusters and are close to the ground truth number of labels. This shows that the HDP-HMM yields accurate estimates of the number of clusters.

7.2.2. The human action dataset

To verify the ability of the dual sticky HDP-HMM to model multi-observations at each time, we carried out experiments on a human action dataset [6]. It contains 599 videos involving the following 6 classes of actions: boxing, clapping hands, hand waving, jogging, running, and walking. Each class includes actions made by 25 people under 4 different scenarios.

We exacted the spatiotemporal points of interest using the method in [18]. Each video is represented by a generalized trajectory. The method presented in Section 2.2 was used to model the generalized trajectories as documents with multi-observations at each time. We tested our method first on 4 subsets which were taken from 4 scenarios respectively, and then on the entire dataset. For the subsets, the size of the codebook was set to 1000, and the initial values of parameters L_z, L_s, and L_c were set to 60, 10, and 10, respectively. For the entire dataset, the size of the codebook was set to 4000, and the initial values of parameters L_z, L_s, and L_c were set to 120, 10, and 10, respectively. We compared the dual sticky HDP-HMM with the *K*-means method, the *K*-medoids method, the spectral clustering method, LDA, and Dual HDP. For the *K*-means method, the *K*-medoids method, and the spectral clustering method, a video was represented by the histogram of its visual words represented by vectors obtained by PCA dimension reduction applied to the features of the spatiotemporal points of interest. For the *K*-medoids method and the spectral clustering method, the χ^2 distance between histograms was used to measure the similarities between videos. The ground truth number of actions was used for the *K*-means method, the *K*-medoids method, the spectral clustering method, and LDA. It is noted that LDA was also used for unsupervised learning on the dataset in [12], but no clustering result was reported in [12].

Table 3. Clustering performance of different methods on the human action dataset

Datasets	Criteria	*K*-means	*K*-medoids	Spectral clustering	LDA	Dual-HDP	Dual sticky HDP-HMM
Scene 1	CCR(%)	25.93	40.67	54.67	42.67	50.67	60.67
	CA(%)	86.25	42.56	67.87	43.69	95.44	77.05
Scene 2	CCR(%)	24.67	36.67	35.33	39.33	40.00	36.67
	CA(%)	67.20	38.63	52.12	38.57	45.63	51.63
Scene 3	CCR(%)	18.12	32.21	37.58	24.83	30.20	44.97
	CA(%)	77.91	37.35	50.86	26.64	40.60	61.08
Scene 4	CCR(%)	18.00	38.67	45.33	40.67	39.33	49.33
	CA(%)	86.21	39.55	52.74	41.23	57.43	71.30
Entire data	CCR(%)	21.04	27.21	30.72	26.54	31.05	36.72
	CA(%)	62.72	28.68	38.07	29.39	65.87	57.94

Table 3 compares the clustering performances of different methods. Our method produces 8 clusters, i.e., there two redundant clusters. The results show the following points:

- Under the criterion of CCR, our dual stick HDP-HMM yields the highest results on the subsets from scenes 1, 3, and 4 and on the entire dataset. Because the CCR criterion punishes any discrepancy between the estimated number of clusters and the ground truth labels, the higher CCR of our dual sticky HDP-HMM confirms its ability to find the number of clusters.
- Under the criterion of CA, our dual stick HDP-HMM yields the second highest results on the subsets from scenes 3 and 4, and the third highest results on the subsets from scenes 1 and 2 and on the entire dataset.
- Overall, the dual sticky HDP-HMM yields the best results. This is because our method effectively models sequential dependencies between visual words. The competing methods are BoW-based and do not take account of sequential dependency. Moreover, the multimodal multi-observation distribution in our method effectively models the generalized trajectories. In practice, the existing HDP-HMMs cannot produce a model of this type.

Besides clustering, our dual sticky HDP-HMM is able to effectively learn atomic activities and their sequential dependencies. It appears that the states in the learnt dual sticky HDP-HMM correspond to semantically meaningful atomic activities. Fig. 13(a) shows a part of the learnt transition matrix for the running action from scene 1, containing four atomic activities that occur most frequently in running. Fig. 13(b) shows the corresponding state transition diagram. Two sample frames assigned to each state are shown, one above the state and one below the state. It is seen that states A and B correspond, respectively, to the activities of dropping a leg and lifting a leg during running. States C and D have the same semantic meanings as states A and B except for contrary running directions. There are large transition probabilities between states A and B and between states C and D. This fits the fact that lifting a leg and dropping a leg occur alternately in the process of running. Figs. 13(c) shows the transition matrix for the seven most frequent atomic activities in the action of hand waving. Fig. 13 (d) shows the corresponding state transition diagram, with sample frames above and below the states. It is seen that each state corresponds to a pose in the process of handing waving. The transitions between these states form a cycle. This fits the fact that the arm movements in hand waving are repetitive.

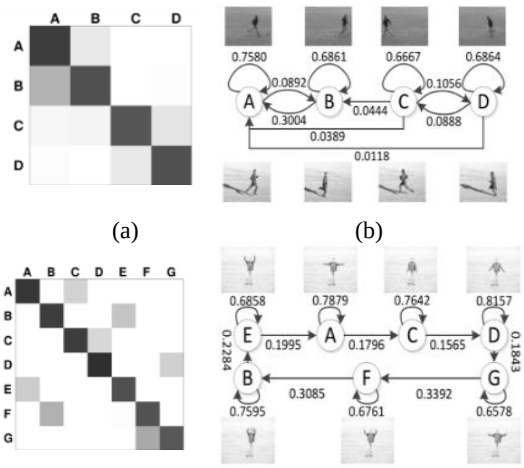


Fig. 13. Partial transition matrices and transition diagrams: (a) and (b) correspond to running; (c) and (d) correspond to hand waving.

7.2.3. The synthetic trajectory dataset

The synthetic trajectory benchmark dataset contains 2500 trajectories from 50 clusters. Each cluster consists of 50 trajectories with complex shapes. From the dataset, we randomly selected 30 subsets such that each subset contained 10 clusters. We measured the average of the values of clustering accuracies for each subset. For the HDP-HMM, the initial number of trajectory clusters was set to 20 which is much larger than the ground truth cluster number, 10.

We compared the clustering accuracy of the dual sticky HDP-HMM with those of mean shift, spectral clustering, SOM, and *K*-means, where spectral clustering and *K*-means clustering use, respectively, the DTW-based distance and the Euclidean distance as the trajectory similarity measure, and mean shift and SOM use the DFT features and PCA features to measure trajectory similarities. We set the number of trajectory clusters for each of the four competing algorithms to the number found automatically by the dual sticky HDP-HMM. The means and standard deviations of clustering accuracy are shown in Table 4, where the standard deviations of clustering accuracy assess the uncertainty for the clustering accuracies. It is seen that our dual sticky HDP-HMM yields

the highest mean and lowest standard deviation. The reason is that, compared with the non DP-based algorithms, the dual sticky HDP-HMM makes more effective use of the sequential information in trajectories.

Table 4. Clustering accuracies on the synthetic dataset

Algorithms	CA (%)	CCR (%)
Mean shift	92.11±1.92	61.99±3.98
Spectral clustering	92.87±1.83	51.95±3.71
SOM	90.83±1.95	54.13±3.36
K-means	89.52±1.86	50.03±3.99
Dual sticky HDP-HMM	94.60±1.83	63.19±3.15

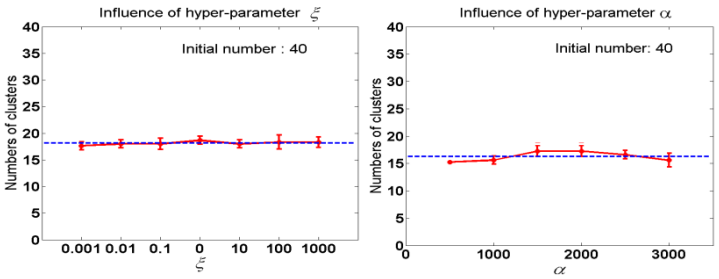
7.2.4. The traffic dataset

In the vehicle motion trajectory dataset, there are 1500 trajectories collected by tracking vehicles in a real traffic scene and labeled manually to produce 15 clusters. We also compared our dual sticky HDP-HMM with the competing algorithms on the traffic dataset. The initial number of trajectory clusters for the dual sticky HDP-HMM was set to 40 which is substantially larger than the ground truth cluster number. The means and standard deviations of clustering accuracy are shown in Table 5. It is seen that our algorithm yields more accurate results than the competing algorithms.

Table 5. Clustering accuracies on the traffic dataset

Algorithms	CA (%)	CCR (%)
Mean shift	93.27±1.03	76.86±2.93
Spectral clustering	93.56±1.23	73.93±3.88
SOM	90.73±1.87	68.93±4.05
K-means	89.80±2.01	69.31±4.85
Dual sticky HDP-HMM	94.79±0.92	90.60±2.38

Fig. 14 shows the curves of the inferred number of clusters of trajectories when the hyper-parameters ξ and α on which the inferred number mainly depends are changed. The number of clusters is represented by the mean value and the standard deviation for each value of the relevant parameter. It is seen that the change in the inferred number is not large when the values of the hyper-parameters are varied. If the values of the hyper-parameters are fixed, then the change in the inferred number is not large when different randomly sampling tests were carried out. Therefore, the inferred number of clusters is robust against variations in the hyper-parameters. Furthermore, all the learnt numbers of trajectory clusters are substantially less than the initial number of clusters. Therefore, the model learns the number of clusters.



(a) (b)

Fig. 14. The curves of the referred number of clusters on the traffic dataset with the changes in (a) the hyper-parameter ξ and (b) the hyper-parameter α.

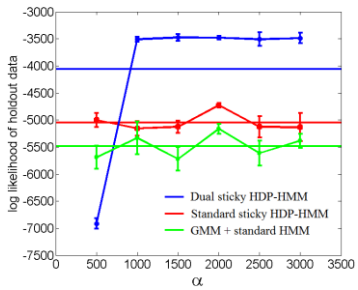


Fig. 15. The log likelihoods of the heldout data with changes in α.

The log likelihood of the held out data was used to compare the dual sticky HDP-HMM with the standard sticky HDP-HMM and a GMM and standard HMM combined model obtained by clustering the trajectories using a GMM with no temporal dependence and then modeling each cluster of trajectories using a standard HMM with the number of states chosen by cross-validation. Fifty trajectories were randomly selected as heldout data. The remaining trajectories were used for learning. Fig. 15 shows the mean values and the standard deviations of the log likelihoods of the heldout data for the three models, when different values of the hyper-parameter α are taken. It is seen that the dual sticky HDP-HMM yields the highest means and lowest standard deviations. It better models the data than the two competing models.

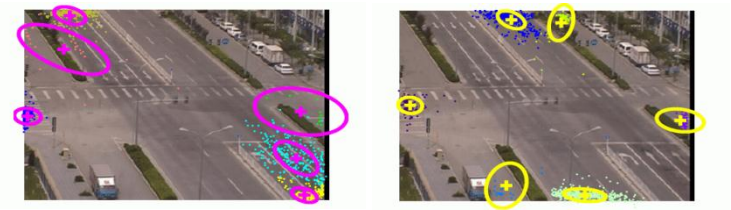
We generated state sequences from the learnt dual sticky HDP-HMM. The state sequences were transformed to trajectories by replacing each state with its corresponding trajectory segment. Some generated trajectories are shown in Fig. 16. It is seen that the trajectories generated from the posterior predictive distribution of the fit model are very similar to those produced in practice.

Fig. 16. The trajectories generated from the posterior predictive distribution of the fit model learnt from the traffic dataset.

7.3. Generation of natural language description

We generated natural language descriptions of the actions of vehicles in the traffic scene. The dataset of vehicle motion trajectories in Section 7.2.4 was used.

7.3.1. Source and sink modeling



(a) (b)

Fig. 17. The learned sources and sinks.

Fig. 17 shows the sources and sinks learned by the blocked Gibbs sampler for DP-GMM. The learnt sources are indicated by magenta ellipses and crosses, and the learnt sinks are colored by yellow. The centers of the sources and sinks are marked by crosses. The ellipses indicate the extents of the sources and sinks. The numbers of sources and sinks were correctly determined by the DP-GMM, although some endpoints were assigned to wrong clusters. In order to evaluate the efficiency of the proposed blocked inference method for DP-GMM, we compared it with the traditional Chinese restaurant process (CRP)-based Gibbs sampler using the criteria of computational time and clustering error rates [10]. The code for the CRP-based method was downloaded from [32]. The results are shown in Table 6. Our method has a lower clustering error rate than the CRP-based method, and runs much faster than the CRP-based method.

Table 6. The comparison between our method and the traditional CRP-based method for DP-GMM

	Runtime (seconds)	Clustering error rate (%)
CRP-based	700.27	3.46
Our method	27.07	3.29

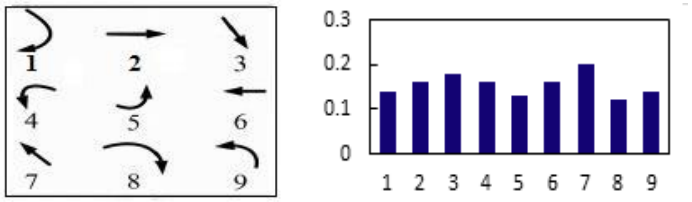
Fig. 18. Learnt motion regions.

7.3.2. Learning motion regions

The method presented in Section 6.2 was used to learn dominant motion regions in the traffic scene. As shown in Fig. 18, there are 11 dominant motion regions learnt. They automatically form a semantic map of the traffic scene.

7.3.3. Learning generalized atomic activities

The generalized atomic activities were learnt by clustering the topics. Fig. 19 (a) shows the learnt generalized atomic activities on the vehicle motion trajectory dataset. Fig. 19(b) exhibits the distribution of the generalized atomic activities. It is seen that only 9 generalized atomic activities were found from the traffic scene. Each action represented by a trajectory can be formed by combining these 9 generalized atomic activities. We manually assigned a predicate to each of the 9 generalized atomic activities.



(a) (b)

Fig. 19. The results of learning the generalized atomic activities on the traffic scene dataset: (a) displays the 9 identified generalized atomic activities; (b) displays the distribution of the generalized atomic activities.

7.3.4. Natural language description

The most typical 12 types of action in the traffic scene are shown in Fig. 20. The learnt different generalized atomic activities are marked by different colors. The tails of the trajectories are in red to indicate the motion directions of the trajectories. These actions were correctly described in natural language. For instance, an example of the first type of action was described as follows: "A red car enters the scene from Source 2 at normal speed"; "The car goes upward in Area 11 at normal speed"; "The car goes upward in Area 2 at normal speed"; "The car goes upward in Area 7 at normal speed"; "The car goes upward in Area 5 at normal speed"; "The car goes upward in Area 8 at normal speed"; "The car leaves the scene from Exit 4 at normal speed". An example of the fourth type of action is described as follows: "A black vehicle enters the scene from Source 3"; "The vehicle moves upward in Area A11"; "The vehicle moves upward in Area 2"; "The vehicle turns left in Area 10"; "The vehicle goes leftward in Area 10"; "The vehicle goes leftward in Area 4"; "The vehicle leaves the scene from Exit 7". The identical descriptions of the consecutive identical primitives were merged, using the criterion of output activation described in Section 6.5.

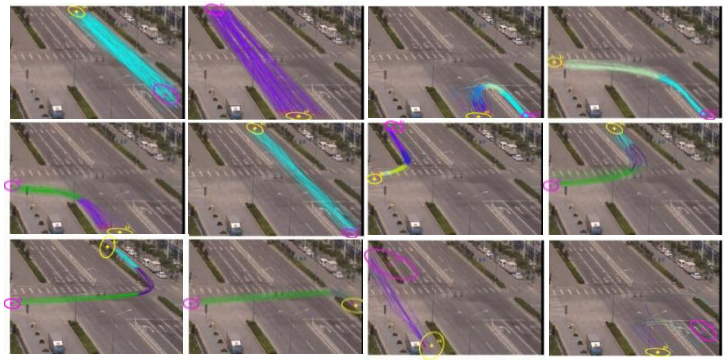


Fig. 20. 12 typical types of action in the traffic scene.

8. Conclusion

We have proposed the dual sticky HDP-HMM. Complex time series produced by a single observation or multiple observations at a time distributed in multimodal ways can be clustered and visual topics can be found. The number of clusters of documents and the number of states in HMMs have both been automatically determined. With a sticky prior on self-transitions, large variations in observations of the same state have been effectively handled. Bayesian inference has been proposed to predict the sequence of topics for a given trajectory. The learnt topics have generalized to semantically meaningful generalized atomic activities to which predicates have been assigned. The endpoints of trajectories have been clustered into sources and sinks using DP-GMM with an efficient blocked Gibbs sampler for inference. Semantic motion regions have been learnt using the points in trajectories. In this way, not only can our method discover motion patterns automatically, but it can also describe the semantic structures of trajectories using natural language. Experiments have demonstrated that the dual sticky HDP-HMM has good performance on action clustering and semantic learning. The experiments on real vehicle trajectories have demonstrated the effectiveness of our method for generating natural language descriptions for motions.

References

[1] T.S. Ferguson, "A Bayesian analysis of some nonparametric problems," *The Annals of Statistics*, vol. 1, no. 2, pp. 209-230, 1973.

[2] Y.W. Teh, "Dirichlet processes," in Encyclopedia of Machine Learning, Springer, 2010.

[3] R.M. Neal, "Markov chain sampling methods for Dirichlet process mixture models," *Journal of Computational and Graphical Statistics*, vol. 9, no. 2, pp. 249-265, 2000.

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

[5] Y.W. Teh and M.I. Jordan, "Hierarchical Bayesian nonparametric models with applications," *in Bayesian Nonparametrics: Principles and Practice*, N. Hjort, C. Holmes, P. Muller, and S. Walker, Eds. Cambridge University Press, pp. 158-207 2010.

[6] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local SVM approach," *in Proc. of International Conference on Pattern Recognition*, pp. 32-36, 2004.

[7] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and S. Maybank, "A system for learning statistical motion patterns," *IEEE Trans. on Pattern Analysis and Machine Intelligent*, vol. 28, no. 9, pp. 1450-1464, 2006.

[8] E.J. Keogh and M.J. Pazzani, "Scaling up dynamic time warping for data mining applications," *in Proc. of ACM International Conference on Knowledge Discovery and Data Mining*, pp. 285-289, 2000.

[9] Z. Zhang, K. Huang, and T. Tan, "Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes," *in Proc. of IEEE International Conference on Pattern Recognition*, pp. 1135-1138, 2006.

[10] B.T. Morris and M.M. Trivedi, "Learning trajectory patterns by clustering: experimental studies and comparative evaluation," *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 312-319, 2009.

[11] D. Blei, A. Ng, and M. Jordan, "Latent Dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, no. 5, pp. 993-1022, 2003.

[12] J.C. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatialtemporal words," *International Journal of Computer Vision*, vol. 79, no. 3, pp. 299-318, 2008.

[13] X. Wang, K.T. Ma, G.-W. Ng, W.E.L. Grimson "Trajectory analysis and semantic region modeling using nonparametric hierarchical Bayesian models," *International Journal of Computer Vision*, vol. 95, no. 3, pp. 287-312, Dec. 2011.

[14] D. Kuettel, M.D. Breitenstein, L.V. Gool, and V. Ferrari, "What's going on? discovering spatio-temporal dependencies in dynamic scenes," *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1951-1958, June 2010.

[15] E.B. Fox, E.B. Sudderth, M.I. Jordan, and A.S. Willsky, "An HDP-HMM for systems with state persistence," *in Proc. of International Conference on Machine Learning*, Helsinki, Finland, pp. 312-319, July 2008.

[16] J. Sethuraman, "A constructive definition of Dirichlet prior," *Statistica Sinica*, vol. 4, pp. 639-650, 1994.

[17] E.B. Fox, E.B. Sudderth, M.I. Jordan, and A.S. Willsky, "A sticky HDP-HMM with application to speaker diarization," *Annals of Applied Statistics*, vol. 5, no. 2A, pp. 1020-1056, 2011.

[18] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, June 2008.

[19] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the Society of Industrial and Applied Mathematics*, vol. 5, no. 1, pp. 32-38, 1957.

[20] H. Ishwaran and M. Zarepour, "Exact and approximate sum representations for the Dirichlet process," *Canadian Journal of Statistics*, vol. 30, no. 2, pp. 269-283, June 2002.

[21] L.R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257-286, Feb. 1989.

[22] F.I. Bashir, A.A. Khokhar, and D. Schonfeld, "Real-time motion trajectory-based indexing and retrieval of video sequences," *IEEE Trans. on Multimedia*, vol. 9, no. 1, pp. 58-65, Jan. 2007.

[23] X. Ma, F. Bashir, A.A. Khokhar, and D. Schonfeld, "Event analysis based on multiple interactive motion trajectories," *IEEE*

Trans. on Circuits and Systems for Video Technology, vol. 19, no. 3, pp. 397-406, March 2009.
[24] J. Sun, W. Zhang, X. Tang, and H. Shum, "Bidirectional tracking using trajectory segment analysis," *in Proc. of IEEE International Conference on Computer Vision*, vol. 1, pp. 717-724, 2005.
[25] B. Georgescu, I. Shimshoni, and P. Meer, "Mean shift based clustering in high dimensions: a texture classification example," *in Proc. of IEEE International Conference on Computer Vision*, vol. 1, pp. 456-463, Oct. 2003.
[26] A. Naftel and S. Khalid. "Motion trajectory learning in the DFT-coefficient feature space," *in Proc. of IEEE International Conference on Computer Vision Systems*, pp. 47-47, Jan. 2006.
[27] Z. Zhang, K. Huang, and T. Tan, "Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes," *in Proc. of IEEE International Conference on Pattern Recognition*, pp. 1135-1138, 2006.
[28] B.T. Morris and M.M. Trivedi, "Trajectory learning for activity understanding: unsupervised, multilevel, and long-term adaptive approach," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2287-2301, 2011.
[29] I. Alvarez, J. Niemi, and M. Simpson, "Bayesian inference for a covariance matrix," 2014, Available at: http://works.bepress.com/jarad_niemi/2/
[30] M. Vlachos, G. Kollios, and D. Gunopulos, "Discovering similar multidimensional trajectories," *in Proc. of International Conference on Data Engineering*, pp. 673-684, 2002.
[31] S. Atev, G. Miller, and N.P. Papanikolopoulos, "Clustering of vehicle trajectories," *IEEE Trans. on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 647-657, Sep. 2010.
[32] Y.W. Teh, "Nonparametric Bayesian mixture models - release1," http://www.stats.ox.ac.uk/ ~teh/software.html.
[33] P. Remagnino, T. Tan, and K. Baker, "Agent orientated annotation in model based visual surveillance," *in Proc. of IEEE International Conference on Computer Vision*, pp. 857-862, 1998.
[34] J. Lou, Q. Liu, T. Tan, and W. Hu, "Semantic interpretation of object activities in a surveillance system," *in Proc. of International Conference on Pattern Recognition*, vol. 3, pp. 777-780, 2002.
[35] A. Kojima, T. Tamura, and K. Fukunaga, "Natural language description of human activities from video images based on concept hierarchy of actions," *International Journal of Computer Vision*, vol. 50, no. 2, pp. 171-184, Nov. 2002.
[36] N. Krishnamoorthy, G. Malkarnenkar, R. Mooney, K. Saenko, and S. Guadarrama, "Generating natural-language video descriptions using text-mined knowledge," *in Proc. of AAAI Conference on Artificial Intelligence*, pp. 541-547, 2013.
[37] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko, "Translating videos to natural language using deep recurrent neural networks," *in Proc. of Conference of the North American Chapter of the Association for Computational Linguistics -- Human Language Technologies*, pp. 1494-1504, Denver, Colorado, June 2015.
[38] J. Thomason, S. Venugopalan, S. Guadarrama, K. Saenko, and R. Mooney, "Integrating language and vision to generate natural language descriptions of videos in the wild," *in Proc. of International Conference on Computational Linguistics*, pp. 1218-1227, Dublin, Ireland, August 2014.
[39] H. Kollnig, H.-H. Nagel, and M. Otte, "Association of motion verbs with vehicle movements extracted from dense optical flow fields," *in Proc. of European Conference on Computer Vision*, vol. 801, *Lecture Notes in Computer Science*, pp. 338-347, 1994.
[40] X. Wang, K. Tieu, and E. Grimson, "Learning semantic scene models by trajectory analysis," *in Proc. of European Conference on Computer Vision*, vol. 3953, Lecture Notes in Computer Science pp. 110-123, 2006.
[41] L. Chen, M.T. Ozsu, and V. Oria, "Robust and fast similarity search for moving object trajectories," *in Proc. of ACM SIGMOD International Conference on Management of Data*, pp. 491-502, 2005.
[42] J.J. Little and Z. Gu, "Video retrieval by spatial and temporal structure of trajectories," *in Proc. of SPIE Storage and Retrieval for Media Databases*, vol. 4315, pp. 545-552, 2001.
[43] J.-W. Hsieh, S.-L. Yu, and Y.-S. Chen, "Motion-based video retrieval by trajectory matching," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 396-409, March 2006.
[44] N. Piotto, N. Conci, and F.G.B. De Natale, "Syntactic matching of trajectories for ambient intelligence applications," *IEEE Trans. on Multimedia*, vol. 11, no. 7, pp. 1266-1275, Nov. 2009.
[45] C.R. Jung, L. Hennemann, and S.R. Musse, "Event detection using trajectory clustering and 4-D histograms," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 18, no. 11, pp. 1565-1575, Nov. 2008.
[46] I. Saleemi, K. Shafique, and M. Shah, "Probabilistic modeling of scene dynamics for applications in visual surveillance," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 8, pp. 1472-1485, Aug. 2009.
[47] F.I. Bashir, A.A. Khokhar, and D. Schonfeld, "Object trajectory-based activity classification and recognition using hidden Markov models," *IEEE Trans. on Image Processing*, vol.16, no. 7, pp. 1912-1919, July 2007.
[48] N.T. Nguyen, D.Q. Phung, S. Venkatesh, and H. Bui, "Learning and detecting activities from movement trajectories using the hierarchical hidden Markov model," *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 955-960, June 2005.
[49] H. Veeraraghavan and N.P. Papanikolopoulos, "Learning to recognize video-based spatiotemporal events," *IEEE Trans. on Intelligent Transportation Systems*, vol. 10, no. 4, pp. 628-638, Dec. 2009.
[50] X. Wang, X. Ma, and W.E.L. Grimson, "Unsupervised activity perception in crowded and complicated scenes using hierarchical Bayesian models," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 3, pp. 539-555, March 2009.
[51] X. Wang, K. Tieu, and E.L. Grimson, "Correspondence-free activity analysis and scene modeling in multiple camera views," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 56-71, Jan. 2010.
[52] H. Xu, Y. Zhou, W. Lin, and H. Zha, "Unsupervised trajectory clustering via adaptive multi-kernel-based shrinkage," *in Proc. of IEEE International Conference on Computer Vision*, pp. 4328-4336, 2015.
[53] W. Lin, Y. Mi, W. Wang, J. Wu, J. Wang, and Tao Mei, "A diffusion and clustering-based approach for finding coherent motions and understanding crowd scenes," *IEEE Trans. on Image Processing*, vol. 25, no. 4, pp. 1674-1687, April 2016.
[54] Z. Shao and Y. Li, "On integral invariants for effective 3-D motion trajectory matching and recognition," *IEEE Trans. on Cybernetics*, vol. 46, no. 2, pp. 511-523, Feb. 2016.
[55] G. Tian, C. Yuan, W. Hu, and Z. Cai, "Mining activities using sticky multimodal dual hierarchical Dirichlet process hidden Markov model," *in Proc. of IEEE International Conference on Image Processing*, pp. 98-102, Sep. 2013.
[56] H. Ishwaran and M. Zarepour, "Markov chain Monte Carlo in approximate Dirichlet and Beta two-parameter process hierarchical models," *Biometrika*, vol. 87, no. 2, pp. 371-390, 2000.
[57] H. Ishwaran and L.F. James, "Gibbs sampling methods for stick-breaking priors," *Journal of the American Statistical Association*, vol. 96, no. 453, pp. 161-174, March 2001.
[58] J.J. Kivinen, E.B. Sudderth, and M.I. Jordan, "Learning multiscale representations of natural scenes using Dirichlet processes," *in Proc. of IEEE International Conference on Computer Vision*, pp. 1-8, 2007.

Weiming Hu received the Ph.D. degree from the department of computer science and engineering, Zhejiang University in 1998. From April 1998 to March 2000, he was a postdoctoral research fellow with the Institute of Computer Science and Technology, Peking University. Now he is a professor in the Institute of Automation, Chinese Academy of Sciences. His research interests are in visual motion analysis, recognition of web objectionable information, and network intrusion detection.

Guodong Tian is a Ph.D. candidate in National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China. He received the B.S. degree and the M.S. degree in Information and Communication Engineering, from Beijing University of Posts and Telecommunications, Beijing, China, in 2007 and 2010, respectively. His research interests include activity recognition and motion trajectory learning.

Yongxin Kang received the B.S. degree in Automation from Northeast Forestry University, Heilongjiang, China, in 2014. He is currently working toward the M.S. degree in Pattern Recognition and Intelligent Systems, Harbin University of Science and Technology, Heilongjiang, China. His current research interests mainly focus on activity recognition and motion trajectory learning.

Chunfeng Yuan received the doctoral degree from the Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China, in 2010. She is currently an associate professor at the CASIA.. Her research interests and publications range from statistics to computer vision, including sparse representation, motion analysis, action recognition, and event detection.

Stephen Maybank received a BA in Mathematics from King's College Cambridge in 1976 and a PhD in computer science from Birkbeck college, University of London in 1988. Now he is a professor in the Department of Computer Science and Information Systems, Birkbeck College. His research interests include the geometry of multiple images, camera calibration, visual surveillance etc.