Trajectory-Based Video Retrieval Using Dirichlet Process Mixture Models

Xi Li[†], Weiming Hu[†], Zhongfei Zhang[‡], Xiaoqin Zhang[†], Guan Luo[†]
[†]National Laboratory of Pattern Recognition, CASIA, Beijing, China
[†]{lixi, wmhu, xqzhang, gluo}@nlpr.ia.ac.cn
[‡]State University of New York, Binghamton, NY 13902, USA
[‡]zhongfei@cs.binghamton.edu

Abstract

In this paper, we present a trajectory-based video retrieval framework using Dirichlet process mixture models. The main contribution of this framework is four-fold. (1) We apply a Dirichlet process mixture model *(DPMM)* to unsupervised trajectory learning. *DPMM* is a countably infinite mixture model with its components growing by itself. (2) We employ a time-sensitive Dirichlet process mixture model *(tDPMM)* to learn trajectories' time-series characteristics. Furthermore, a novel likelihood estimation algorithm for *tDPMM* is proposed for the first time. (3) We develop a *tDPMM*-based probabilistic model matching scheme, which is empirically shown to be more error-tolerating and is able to deliver higher retrieval accuracy than the peer methods in the literature. (4) The framework has a nice scalability and adaptability in the sense that when new cluster data are presented, the framework automatically identifies the new cluster information without having to redo the training. Theoretic analysis and experimental evaluations against the state-of-the-art methods demonstrate the promise and effectiveness of the framework.

1 Introduction

For content-based video retrieval, motion information plays an important role in depicting the semantic contents of videos. In general, there are two types of motion-based video retrieval techniques: camera-based and object-based. For the camera-based approaches, camera motions, such as zooming in or out, tilting up or down, panning left or right, are used for classifying videos of different contents. Optical flow field analysis is often applied to estimating the camera motion parameters. However, videos with very similar camera motions may contain different semantic contents. For object-based video retrieval, object motion analysis and behavior understanding are the key to developing the indexing structures. Since motion trajectory is an important cue to describe motion features for a video sequence, recent work in the literature uses object motion trajectories to index motion events in videos. Therefore, our video retrieval framework takes a trajectory learning based strategy for video retrieval.

BMVC 2008 doi:10.5244/C.22.106

Object trajectories, which contain rich spatio-temporal and semantic information, are typically used for representing object motion characteristics. In this case, video retrieval is achieved by trajectory matching. Little and Gu [1] used the polynomial-based curve fitting technique to represent spatio-temporal motion information of the object trajectory. Sahouria [6] applied a wavelet transform using Harr basis to analyze object trajectory's spatio-temporal information in multiple scales. Naftel and Khalid [8] showed that the Discrete Fourier Transform (DFT) coefficient based trajectory representation performed better than polynomial-based trajectory representation in video retrieval. Bashir *et al.* [18] proposed a principal component analysis based approach to model object trajectories in a video clip. The longest common subsequence (LCS) algorithm [7] is used for measuring the similarity between two object trajectories by analyzing objects' coordinates directly. In [9], trajectories are divided into several small segments each of which is expressed by a semantic symbol. A distance measure combining the edit distance and the visual distance is exploited to determine the similarity between two trajectories. However, it is infeasible for video retrieval to match a query trajectory to all the trajectories in the database for any real systems.

More recent work on trajectory-based video retrieval focuses on trajectory learning to construct a better trajectory-based index structure. In this case, the problem of trajectory-based video retrieval is reduced to how to make trajectory learning more effective and efficient. In [8], motion trajectories are learned through self-organizing map (SOM) in the DFT-Coefficient feature space. Johnson and Hogg [2] used two competitive neural networks connected by a leaky neuron layer to model the probabilistic distribution of flow vectors and the trajectories. Compared with [2], Hu *et al.* [3] proposed a fuzzy self-organizing map (FSOM) that is much simpler in architecture as it takes the whole trajectory rather than discrete flow vectors as the input. Fu *et al.* [4] showed that spectral clustering performed well in the case of multi-cluster trajectory learning. Alon *et al.* [5] employed a finite mixture of HMM to learn motion data using the Expectation-Maximization (EM) technique. However, the aforementioned learning methods share a problem that they lack a competent criterion for estimating the "correct" number of trajectory clusters.

In this paper, we develop a trajectory-based video retrieval framework using the Dirichlet process mixture model (DPMM)[10, 11, 12, 16, 17]. In the framework, DPMM is first applied to unsupervised trajectory learning. DPMM is a countably infinite mixture model whose components can grow by itself, resulting in adaptively determining the number of trajectory clusters. Based on the number of trajectory clusters learned from DPMM, we then apply a time-sensitive DPMM (tDPMM) derived from [14] to build the index of the motion trajectories in the video database, and a probabilistic matching model for final video retrieval. Especially, a novel likelihood estimation algorithm for tDPMM is proposed for the first time. The algorithm approximates the likelihood using a collection of particles generated by Gibbs sampling. We have shown that the retrieval framework is scalable and adaptive in the sense that when new data are present there is no need to redo all the learning from the scratch.

2 Dirichlet process mixture model

2.1 Introduction to DPMM

Let $Dir(\cdot)$ denote the Dirichlet density function. If a random probability distribution G on a continuous random variable η within a probability space $\mathbb{A}$ is distributed according to the Dirichlet process ($\mathscr{DP}$) [10] parameterized by a scaling parameter α and a base measure G_0 over $\mathbb{A}$, the relation

$$(G(\eta \in A_1), G(\eta \in A_2), \ldots, G(\eta \in A_K)) \sim Dir(\alpha G_0(A_1), \alpha G_0(A_2), \ldots, \alpha G_0(A_K)) \quad (1)$$

holds true for any natural number K and K partitions $\mathbb{A}$ into $A_{1:K}$. By integrating over G, the joint distribution on a set of random variables $\eta_{1:N}$ exhibits a clustering effect; the i-th draw η_i conditioned on the previous $i-1$ draws $\eta_{1:i-1}$ is either equal to one of $\eta_{1:i-1}$ or an independent draw from G_0, which can be illustrated in the following formula:

$$p(\eta_i|\eta_{1:i-1}) \propto \alpha G_0(\eta_i) + \sum_{j=1}^{i-1} \delta(\eta_i - \eta_j) \quad (2)$$

given by the Pòlya Urn scheme [11]. As a result, $\eta_{1:N}$ are randomly partitioned by the Pòlya Urn scheme into clusters in each of which variables have the same value. The partition structure of the Pòlya Urn scheme is described as follows. Let $\eta^*_{1:L}$ denote L unique values in $\eta_{1:i-1}$; the next draw from the $\mathscr{DP}$ follows the urn distribution:

$$\eta_i = \begin{cases} \eta^*_l & \text{with probability } \frac{m_l}{i-1+\alpha} \\ \eta_{new},\ \eta_{new} \sim G_0 & \text{with probability } \frac{\alpha}{i-1+\alpha} \end{cases} \quad (3)$$

where m_l denotes the number of occurrences of η^*_l in $\eta_{1:i-1}$ for $1 \leq l \leq L$. If the random variables $\eta_{1:N}$ are exchangeable, the marginal distribution of η_i given η_{-i} is formulated as:

$$p(\eta_i|\eta_{-i}) \propto \alpha G_0(\eta_i) + \sum_{j \in -i} \delta(\eta_i - \eta_j) \quad (4)$$

where $-i$ denotes the remainder of the indices $1:N$ except i.

If the $\mathscr{DP}$ is exploited as a nonparametric prior in a hierarchical Bayesian model, we have the following Dirichlet process mixture model (DPMM):

$$Y_i \sim p(\cdot|\eta_i);\ \ \eta_i \sim G;\ \ G \sim \mathscr{DP}(\alpha, G_0). \quad (5)$$

In what follows, we give a brief introduction to Bayesian inference for DPMM. Given the exchangeable data instances $y_{1:N}$, we wish to obtain the posterior $p(\eta_{1:N}|y_{1:N})$ where $\eta_{1:N}$ denote the state random varibales associated with $y_{1:N}$. Thus, $p(\eta_{1:N}|y_{1:N})$ can be approximated by sampling from $p(\eta_i|\eta_{-i}, y_{1:N})$ iteratively using the Gibbs sampler for $1 \leq i \leq N$, and $p(\eta_i|\eta_{-i}, y_{1:N})$ can be calculated as:

$$\begin{aligned} p(\eta_i|\eta_{-i}, y_{1:N}) &\propto p(y_i|\eta_i)p(\eta_i|\eta_{-i}) \\ &\propto \alpha G_0(\eta_i)p(y_i|\eta_i) + \sum_{j \in -i} p(y_i|\eta_i)\delta(\eta_i - \eta_j) \end{aligned} \quad (6)$$

where $p(y_i|\eta_i)$ denotes the likelihood of the data.

2.2 Theoretical analysis of DPMM

Due to the intrinsic properties of the Dirichlet process, DPMM is a mixture model with a countably infinite number of components. The data associated with the same parameter value drawn from G will be grouped into a cluster by DPMM. By integrating over the latent variable G, the joint distribution on the collection of latent state (cluster) variables $\eta_{1:N}$ exhibits a clustering effect. After Bayesian inference for DPMM, we obtain the posterior $p(\eta_{1:N}|y_{1:N})$ approximated by Gibbs sampling from (6). By maximum a posteriori (MAP) estimate of the state (cluster) variables from $p(\eta_{1:N}|y_{1:N})$, the latent cluster labels associated with the data $y_{1:N}$ are obtained . The analytical proof of DPMM is given in [10, 11, 12, 16, 17].

3 Time-sensitive Dirichlet process mixture model

The following is a brief introduction to the time-sensitive Dirichlet process mixture model (tDPMM)[14]. Consider a time series of observations: $(o_1,t_1),\ldots,(o_N,t_N)$ where o_i denotes the observation associated with the time t_i for $1 \leq i \leq N$, and $t_1 < \ldots < t_N$. Let $s_{1:N}$ be the state sequence associated with $o_{1:N}$. tDPMM introduces a weight function $\omega(t,c)$ which characterizes the influence of the state c at time t given the history $s_{1:i}$ s.t. $t_i < t$. Consequently, $\omega(t,c)$ can be determined as:

$$\omega(t,c) = \sum_{\{m|t_m<t,s_m=c\}} k(t-t_m) \tag{7}$$

where $k(t) = e^{-\lambda t}$ if $t > 0$, and $k(t) = 0$ otherwise. The parameter λ in $k(t)$ is a positive constant. In tDPMM, the state transition distribution $p(s_i|s_{-i})$ derived from [14] has the following form:

$$p(s_i|s_{-i}) \propto p(s_i|s_{1:i-1}) \left(\prod_{n=i+1}^{N} p(s_n|s_{1:n-1}) \right) \tag{8}$$

in which s_{-i} denotes the remainder of $s_{1:N}$ except s_i, and $p(s_i|s_{1:i-1})$ is defined as:

$$p(s_i|s_{1:i-1}) = \begin{cases} \frac{\omega(t_i,s_i)}{\alpha+\sum_{v=1}^{V}\omega(t_i,s_v^*)} & \text{if } s_i \text{ in } s_{1:i-1} \\ \frac{\alpha}{\alpha+\sum_{v=1}^{V}\omega(t_i,s_v^*)} & \text{otherwise} \end{cases} \tag{9}$$

where V denotes the number of unique values in $s_{1:i-1}$, s_v^* denotes the v-th unique value in $s_{1:i-1}$. Notice that $p(s_i|s_{-i})$ (8) is parameterized by $\Theta = \{\alpha,\lambda\}$, i.e., $p(s_i|s_{-i}) = p(s_i|s_{-i},\Theta)$. Furthermore, $p(s_i|s_{-i},o_{1:N})$ is formulated as:

$$p(s_i|s_{-i},o_{1:N}) \propto p(s_i|s_{-i})p(o_i|o_{-i:s_{-i}=s_i}) \tag{10}$$

in which $o_{-i:s_{-i}=s_i}$ denotes the set of observations except o_i, such that their corresponding states s_{-i} are equal to s_i. Thus, $p(s_{1:N}|o_{1:N})$ is approximated by sampling from $p(s_i|s_{-i},o_{1:N})$ iteratively using the Gibbs sampler used in [14]. Besides, a stochastic EM algorithm proposed in [14] is applied to train a tDPMM for a given observation sequence. See [14] for the details of the parameter learning procedure for tDPMM.

3.1 Theoretical analysis of tDPMM

tDPMM is capable of modeling long-range interacting dependencies of the latent state variables corresponding to the observations. Its state transition distribution (8) is governed by the kernel-weighted Dirichlet process (9). Due to the intrinsic properties of the Dirichlet process, tDPMM is very suitable for modeling the time-series data with countably infinite states. After Gibbs sampling from (10), we obtain the posterior $p(s_{1:N}|o_{1:N})$. By maximum a posteriori (MAP) estimate of the state variables from $p(s_{1:N} \mid o_{1:N})$, the most probable latent state label sequence associated with the observation sequence can be obtained.

In particular, a novel Gibbs sampling based likelihood estimation algorithm for tDPMM is proposed in this paper. To the best of our knowledge, this algorithm is new in the literature. Now we are ready to discuss the determination of the likelihood probability of the retrieval model given a particular observation sequence and the tDPMM parameters $\Theta = \{\alpha,\lambda\}$ in Section 3.2.

3.2 Likelihood estimation for tDPMM

Given a particular observation sequence $O = o_{1:N}$, the likelihood conditioned on the learned model parameters $\Theta = \{\alpha,\lambda\}$ can be computed as:

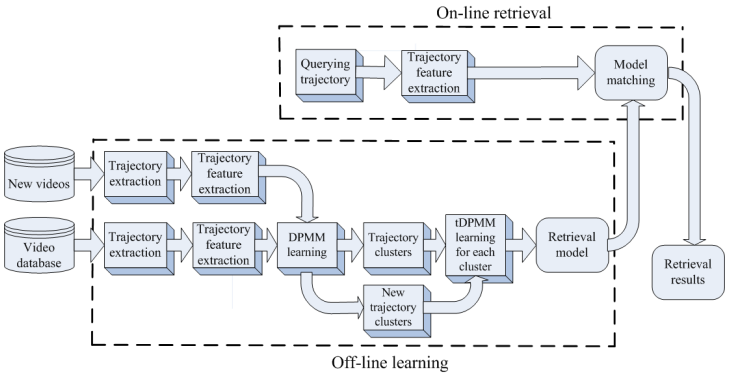


Figure 1: The architecture of the video retrieval framework.

$$p(O|\Theta) = \int p(O|S)p(S|\Theta)dS \tag{11}$$

where $S = s_{1:N}$ is the latent state sequence corresponding to $O = o_{1:N}$. However, the computational cost of (11) is expensive due to integrating over S. To simplify the computation, $p(S|\Theta)$ is approximated by sampling from $p(s_i|s_{-i}) = p(s_i|s_{-i},\Theta)$ (8) through the Gibbs sampler. In addition, we assume that $O = o_{1:N}$ are mutually independent given $S = s_{1:N}$. As a result, the likelihood $p(O|\Theta)$ can be efficiently computed as:

$$\begin{aligned} p(O|\Theta) &= E_{p(S|\Theta)}[p(O|S)] \approx \tfrac{1}{M_l}\textstyle\sum_{m=1}^{M_l} p(O|S^{(m)}) \\ &= \tfrac{1}{M_l}\textstyle\sum_{m=1}^{M_l} p(o_{1:N}|s_{1:N}^{(m)}) = \tfrac{1}{M_l}\textstyle\sum_{m=1}^{M_l}\left(\prod_{n=1}^{N} p(o_n|s_n^{(m)})\right) \end{aligned} \tag{12}$$

where $S^{(m)} = s_{1:N}^{(m)}$ denotes a collection of particles sampled from (8) at the m-th Gibbs sampling iterative step, and M_l denotes the total number of Gibbs sampling iterative steps after a sufficient burn-in period.

4 The framework for video retrieval

4.1 Overview of the framework

The video retrieval framework includes two stages: 1) the off-line learning; 2) the on-line retrieval. In the off-line learning stage, object trajectories are first extracted using an existing method [9]. Following the observation in [8], we employ DFT coefficients to represent the object trajectories. Then the extracted object trajectories are clustered into several clusters by learning the full trajectories using DPMM in the DFT-coefficient feature space. In order to precisely characterize time-varying information of a trajectory, it is necessary to segment a trajectory into several smaller units called subtrajectories. We also employ a DFT-coefficient feature vector to represent a subtrajectory. Thus, a trajectory is represented as a DFT-coefficient sequence. A tDPMM is subsequently trained for each cluster of trajectories using their corresponding DFT-coefficient sequences. The cluster information is obtained automatically from the DPMM learning. As a result, each trajectory cluster has its own tDPMM reflecting its unique spatio-temporal characteristics. The retrieval model consists of the tDPMMs of all the trajectory clusters. In the on-line retrieval stage, we take a sketch-based scheme to represent a user-specified query. Users can retrieve trajectories of any shape they expect. A probabilistic retrieval model is

developed such that the retrieved videos are ranked by their likelihoods. The architecture of the proposed framework is shown in Figure 1.

4.2 Trajectory feature extraction

Following the observation [8] that the DFT-coefficient feature is more robust than the original point-based feature, DFT coefficients are used to represent the object trajectories in our framework. In order to precisely characterize time-varying information of a trajectory, it is necessary to segment a trajectory into atomic subtrajectories. A common approach (such as [18]) to segmenting trajectories into subtrajectories is based on the variance of curve curvature. However, the curvature-based methods are sensitive to noise. Sun *et al.* [13] propose a trajectory segmentation algorithm based on spectral clustering (SC). This method assumes that the number of clusters needs to be specified in advance as it uses K-means clustering in the last step of SC. In order to tackle this problem, we propose an improved version of SC (referred here as ISC), which replaces K-means clustering with the non-parametric adaptive mean-shift clustering algorithm (referred as AMC) [19]. As a result, ISC is capable of identifying the number of subtrajectories automatically. After the trajectory segmentation, the DFT-coefficient feature [8] is extracted for each subtrajectory, leading to an E-dimensional DFT-coefficient representation so ($E = 18$ in the experiments). As a result, the trajectory is represented by a DFT-coefficient sequence $SO = (so_i)_{i=1}^{T}$, where so_i corresponds to the i-th word's DFT-coefficient representation for $1 \leq i \leq T$ (T denotes the number of subtrajectories).

4.3 Model matching and growing

After the query is represented by a feature vector in the DFT-coefficient feature space, model matching is performed via the following posterior probability distribution:

$$p(\Theta_j|R) \propto p(R|\Theta_j)p(\Theta_j) \tag{13}$$

where R denotes the querying trajectory's observation sequence associated with SO referred in Section 4.2, $\Theta_j = \{\alpha_j, \lambda_j\}$ represents the tDPMM's parameters of the j-th trajectory cluster, and $p(R|\Theta_j)$ denotes the likelihood function of the j-th trajectory cluster defined in (12). The matching results are ranked according to the posterior distribution (13).

For model growing, if new videos are added to an indexed video database based on this framework, the new trajectory clusters may be automatically determined based on the following formula derived from (6):

$$\begin{aligned} p(\eta_i|\eta_{-i}, y_{1:N+A}) &= p(\eta_i|\eta_{1:N} = W_{1:N}, \eta_{-i}^{new}, y_{1:N+A}) \\ &\propto p(y_i|\eta_i)p(\eta_i|\eta_{1:N} = W_{1:N}, \eta_{-i}^{new}) \end{aligned} \tag{14}$$

where $N+1 \leq i \leq N+A$, A denotes the number of newly added data instances, $y_{N+1:N+A}$ represents A newly added data instances, $W_{1:N}$ denotes the known state sequence associated with $y_{1:N}$, $p(y_i|\eta_i)$ denotes the likelihood of the data , η_{-i}^{new} represents the remainder of $\eta_{N+1:N+A}$ except η_i. In this way, we only need to draw $\eta_{N+1:N+A}$ from (14) iteratively using the Gibbs sampler without needing to learn the whole dataset $y_{1:N+A}$ over again. Subsequently, for each new trajectory cluster, we may train a tDPMM using SO referred in Section 4.2. The tDPMMs of the newly-generated trajectory clusters may then be added to the original retrieval model already developed under this framework in the indexed database. This model growing capability eliminates the need to redo the training

or indexing for a video database which is typical for many existing retrieval models in the literature, when the database is updated with new data, resulting in a nice scalability and adaptability of this framework.

5 Experiments

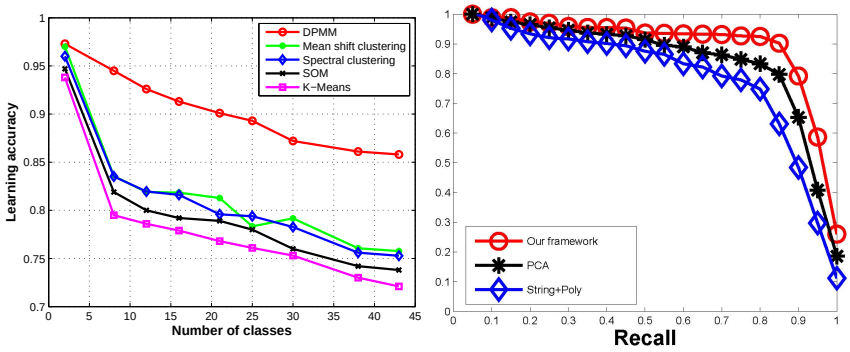
In order to evaluate the performance of the proposed framework, three datasets are used in the experiments. They are the synthetic trajectory dataset[1], the hand sign dataset from the Australian Sign Language (ASL) collection[2], and the traffic scene dataset collected for a real traffic scene. The first two datasets are labeled while the last one is not. The synthetic trajectory dataset contains 2500 trajectories from fifty clusters, each of which consists of 50 trajectories of complex shapes. For the ASL dataset, there are in total 35 clusters of hand sign words. Each cluster consists of 20 trajectories generated from different signers' hand movement. 1500 real object trajectories collected in a real traffic scene constitute the third dataset. Object trajectories are learned by DPMM in the DFT-coefficient feature space, and then grouped automatically into several clusters based on the sampling scheme (6). The scaling parameter α of DPMM is initialized as 0.2. The base measure G_0 of DPMM is assumed to be a Dirichlet distribution in the experiments. $p(\cdot|\eta_i)$ in (5) is chosen as a multinomial distribution, which is conjugate to the Dirichlet distribution G_0. To learn a tDPMM for each trajectory cluster, given the i-th trajectory cluster, the model parameters $\Theta_i = \{\alpha_i, \lambda_i\}$ of tDPMM are learned via the stochastic EM algorithm described in Section 3 in the feature subspace represented by SO referred in Section 4.2. M_l in (12) is set as 200. For the model matching in Section 4.3, each trajectory cluster is assumed to have the same prior probability.

Three experiments are conducted to demonstrate the claimed contributions of the proposed framework. The first experiment is to compare the trajectory learning accuracy of our framework with those of the other learning techniques. The second experiment is to evaluate the retrieval accuracy of our framework against two existing methods from the recent literature. The last experiment is to test the adaptive growing capability of our framework.

The first experiment aims to compare the learning accuracy of DPMM with those of four classic unsupervised learning algorithms in the literature: Mean Shift clustering [15], Spectral clustering [4], self-organizing map (SOM) [3], and K-Means. The synthetic dataset, which is the most complicated of the three datasets used in the experiments, is used to evaluate the learning accuracies of the aforementioned five algorithms. The parameter settings of the four other learning algorithms are obtained from the experiments. The learning accuracy $\mathscr{L}$ is defined as: $\mathscr{L} = \frac{1}{N}\sum_{i=1}^{N}\frac{n_i}{N_i}$, where N denotes the learned number of clusters, N_i represents the number of the samples belonging to the i-th learned cluster and n_i is the number of the samples whose true cluster labels have the highest proportion in the i-th learned cluster. The final clustering accuracies of these five algorithms are shown in Figure 2(a) . From Figure 2(a), it is clear that DPMM's learning accuracy is always higher than those of the four comparing methods. Furthermore, DPMM's learning accuracy tends to be more stable than those of the four comparing methods when the number of trajectory clusters increases.

[1] http://mmplab.eed.yzu.edu.tw/trajectory/trajectory.rar

[2] http://kdd.ics.uci.edu/databases/auslan2/auslan.data.html



(a) (b)

Figure 2: The performance evaluations of different methods. (a) Learning results for the synthetic trajectory dataset.(b) Recall-precision curves of different retrieval methods.

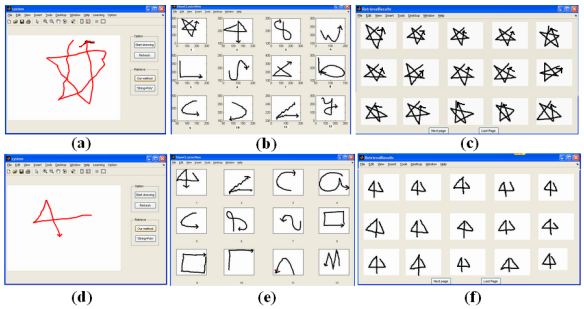


Figure 3: Retrieval results using the synthetic dataset.

In the second experiment, we aim to compare the proposed retrieval framework with two other trajectory-based video retrieval frameworks [9, 18] in the recent literature (referred here as PCA and String+Poly, respectively) using the synthetic trajectory dataset. The average recall-precision curves are shown in Figure 2(b). Clearly, the area underneath the precision-recall curve for our framework is larger than those for the two comparing methods (over 11.2% and 4.9% improvements, respectively). This fact indicates that our framework has a higher retrieval accuracy than the comparing methods.

To showcase the performance of our framework, two retrieval examples on the synthetic trajectory dataset are given in Figures 3(a)-(f). One is the full trajectory retrieval example shown in Figures 3(a)-(c), and the other is the partial trajectory retrieval example shown in Figures 3 (d)-(e). In Figure 3(a), a pentacle-like full trajectory is drawn manually as a query. The matching results are shown in Figure 3(b), where there are twelve figures ranked from left to right for each row and from up to down for each column. Each of these figures represents a trajectory cluster. We call the trajectories shown in those figures template trajectories, each of which is the one with the maximum likelihood to its own trajectory cluster tDPMM. The final retrieval results for this query are shown in Figure 3(c). In Figure 3(d), an "α"-like partial trajectory is queried. Twelve ranked trajectory clusters are returned in Figure 3(e). The final retrieval results are shown in Figure 3(f), which indicates that the framework has the capability of partial trajectory matching in response to partial querying in video retrieval due to the fact that tDPMM precisely captures the local details of a trajectory.

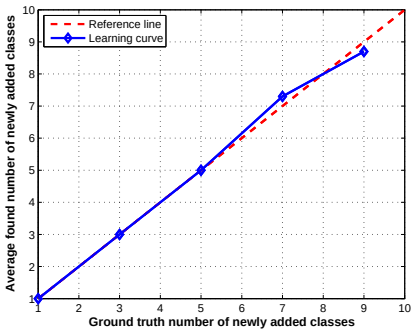


Figure 4: Learning results on model growing using the synthetic dataset.

We also give another example on the traffic scene dataset to demonstrate the promise and the power of the framework. Figure 5(a) shows a query posed to the video database of the traffic scene dataset indexed by the framework to indicate that the user intends to retrieve all the traffic video similar to the scene of a vehicle moving down and then turning right. Thus, a down-left trajectory is drawn in Figure 5(a). Figure 5(b) displays all the retrieved video shots that match what the user intends to retrieve. Note that though this example only shows a retrieval with a 2D query trajectory, this framework is valid for any shaped 3D trajectories. Since we do not have the ground truth for the real traffic scene video dataset, we are unable to report more systematic evaluations for this dataset.

The last experiment is to evaluate the performance of our framework on model growing using the synthetic dataset. In this experiment, we test how good the model growing capability of the framework is on identifying the correct number of newly added trajectory clusters given several already learned trajectory clusters. Specifically, we took five different numbers of newly added clusters (1, 3, 5, 7, and 9). In each case, we randomly selected data with that number of new clusters from the database ten different times, and then observed the discovered number of new clusters by the framework in each time. Figure 4 reports the average identified number of new clusters over the ten times for all the five cases vs. the ground truth number where the dashed line indicates the perfect match. We see that the fluctuant range of the learning curve along the perfect match line is small. In other words, the framework identifies new clusters with a very small error.

6 Conclusion

In this paper, we have proposed a trajectory-based video retrieval framework using Dirichlet process mixture models. In the framework, a Dirichlet process mixture model (DPMM) has been applied to unsupervised trajectory learning. DPMM is a countably infinite mixture probabilistic model whose components can grow by itself. Moreover, a time-sensitive Dirichlet process mixture model (tDPMM) has been used to capture the time-series characteristics of trajectories in the framework. In particular, a novel likelihood estimation algorithm for tDPMM is proposed for the first time. Furthermore, the framework has a nice scalability and adaptability in the sense that when new cluster data are presented, the framework automatically identifies the new cluster information without having to redo the training. A probabilistic model matching scheme based on tDPMM is adopted by the framework. The scheme is able to deliver higher retrieval accuracy than the peer methods in the literature. Experimental results have demonstrated the superiority of the proposed

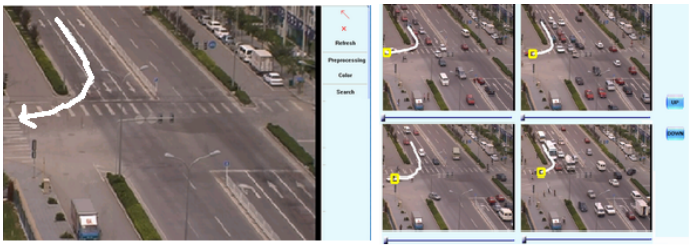


Figure 5: Retrieval results using the traffic scene dataset.

framework to the peer methods in the recent literature.

7 Acknowledgment

This work is partly supported by NSFC (Grant No. 60520120099, 60672040 and 60705003) and the National 863 High-Tech R&D Program of China (Grant No. 2006AA01Z453). Z.Z. is supported in part by NSF (IIS-0535162). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

References

[1] J. J. Little and Z. Gu, "Video Retrieval By Spatial and Temporal Structure of Trajectories," in *Proc. SPIE Storage and Retrieval for Media Databases,* Vol. 4315, pp.545-552, 2001.

[2] N. Johnson and D. Hogg, "Learning the Distribution of Object Trajectories for Event Recognition," *in Proc. British Machine Vision Conference*, 1995.

[3] W. Hu, D. Xie, T. Tan and S. Maybank, "Learning Activity Patterns Using Fuzzy Self-Organizing Neural Network," *IEEE Trans. SMC. Part B* 34(3):1618-1626,2004.

[4] Z. Fu, W. Hu and T. Tan, "Similarity Based Vehicle Trajectory Clustering and Anomaly Detection," *in Proc. ICIP'05*,Vol.2, pp.602-605, 2005.

[5] J. Alon, S. Sclaroff, G. Kollios and V. Pavlovic, "Discovering Clusters in Motion Time-series Data," *in Proc. CVPR'03,* pp:I-375 - I-381 vol.1.

[6] E. Sahouria, "Video Indexing Based on Object Motion," M.S. thesis. Dept. Elect. Eng. Comp. Sci. Univ. California, Berkeley, 1997.

[7] M. Vlachos, G. Kollios and D. Gunopulos, "Discovering Similar Multidimensional Trajectories," in *Proc. 18th Int. Conf. Data Engineering,* 202, pp.673-684, 2002.

[8] A. Naftel and S. Khalid. "Motion Trajectory Learning in the DFT-Coefficient Feature Space," *ICVS'06*, Jan. 2006.

[9] J. Hsieh, S. Yu and Y. Chen, "Motion-Based Video Retrieval by Trajectory Matching," *IEEE Trans. on Circuits and Systems for Video Technology,* Vol.16, Issue 3, pp.396 - 409, March 2006.

[10] T. Ferguson, "A Bayesian analysis of some non-parametric problems," *The Annal of Statistics*, Vol. 1, pp.209-230, 1973.

[11] D. Blackwell and J. B. MacQueen, "Ferguson distribution via Pólya urn schemes," *The Annal of Statistics*, Vol. 1, pp.353-355, 1973.

[12] D. M. Blei and M. I. Jordan, "Variational Methods for the Dirichlet Process," *ICML'04*, 2004.

[13] J. Sun, W. Zhang, X. Tang and H. Shum, "Bidirectional Tracking Using Trajectory Segment Analysis," *in Proc. ICCV'05*, Vol. 1, pp.717 - 724, 2005.

[14] X. Zhu, Z. Ghahramani and J. Lafferty, "Time-Sensitive Dirichlet Process Mixture Models," Technical Report CMU-CALD-05-104, 2005.

[15] D. Comaniciu and P. Meer, "Mean Shift:A Robust Approch Toward Feature Space Analysis," *IEEE Trans. PAMI*, Vol.24, NO.5, May 2002.

[16] R. Neal, "Markov Chain Sampling Methods for Dirichlet Process Mixture Models," *Journal of Computational and Graphical Statistics*, Vol.9, pp.249-265, 2000.

[17] C. Zhang, S. Zhu and Y. Gong, "Trend Analysis for Large Document Streams," *ICMLA'06*, pp.285-295, Dec. 2006.

[18] 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, pp.58-65, 2007.

[19] B. Georgescu, I. Shimshoni, and P. Meer, "Mean Shift Based Clustering in High Dimensions: A Texture Classification Example," in *Proc. ICCV*, Vol. 1, pp. 456-463, 2003.