LEARNING OBJECT TRAJECTORY PATTERNS BY SPECTRAL CLUSTERING

Fatih Porikli

Mitsubishi Electric Research Laboratories
fatih@merl.com

ABSTRACT

We develop a trajectory pattern learning method that has two significant advantages over the past work. First, we represent trajectories in the HMM parameter space, thus we overcome the normalization problems of the existing methods. Second, we determine common trajectory paths by analyzing the optimal cluster number rather than using a predefined number of clusters. We compute affinity matrices and apply eigenvector decomposition to find clusters. We prove that the number of clusters governs the number of eigenvectors used to span the feature affinity space. We are thus able to automatically determine the optimal number of patterns. We show that the proposed algorithm accurately detects common paths for various camera setups.

1. INTRODUCTION

Past work on automatic detection of events using trajectory based features has mostly consisted of extraction of trajectories followed by a supervised learning based classification. There are several attempts to interpret the object activity. For example, in [1] an activity recognition method based on view-based template matching techniques is developed. In this method, action is represented by a temporal template which is a static vector-image computed from accumulative motion properties at each point of the image sequences. An action is recognized by matching this template with the templates of known actions. Davis et al. [2] represent simple periodic events (e.g., walking) by constructing dynamic models of periodic pattern of people's movements and is dependent on the robustness of tracking. The Hidden Markov Model (HMM) has also been applied to activity recognition. Starner et. al [5] use an HMM to represent a simple event and recognize this event by computing the probability that the model produce the visual observation sequence. The distributions of object trajectories are clustered in [4]. The resulting model included hundreds of clusters of object trajectories. Stauffer et al. [6] estimated a hierarchy of similar distributions of activity based using a co-occurrence clustering. Though both of these systems learned clusters corresponding to similar activity, they describe an objects entire path through the environment. Note that the above algorithms only use abstract representations of trajectories, sometimes combined with other cues such as skin color, etc. Although the extraction of trajectories and boundaries is well studied, little investigation on the secondary outputs of a tracker has been done.

In this paper, we target a more attainable goal of detecting common patterns using improved tracking features. One main difficulty of applying learning methods to the trajectory data is that the trajectories usually have different lengths, which causes the dimension of the corresponding feature vectors to be different from each other. To adapt a standard learning or data reduction technique, such as k-means, k-medoids, nearest-neighbor, PCA, ICA, etc., the number of coordinates (samples) that constitutes trajectory is often normalized to a unit length. In other words, trajectory is resampled at a different temporal scale. However, such a normalization severely disturbs the temporal properties and may cause aliasing problems.

We overcome the above difficulties by fitting an HMM model for each trajectory and we introduce parameter space representations of tracked objects. These representations enables pair-wise evaluation of the similarity between the trajectories. We adapt a spectral clustering based learning method that is also robust to increase in the dimensionality of the feature space. Our learning method is based on eigenvector decomposition [7] of the feature similarity (affinity) matrices. We improve upon the existing work by showing that the number of clusters governs the number of eigenvectors used to span the affinity space. We are, thus, able to automatically compute the optimal number of clusters using a new cluster validity score. Furthermore, the computational complexity of the proposed method is lower than the k-means in case of the trajectories all have identical lengths, which makes application of k-means possible, and the value of the length is significantly higher than the number of trajectories, which is common for tracking applications.

2. FINDING PATHS

A flow diagram of the event detection process is shown in fig. 1. First, we fit an HMM model for each trajectory using the identical model topologies. Then, we compute affinity matrices that represents the similarity of two trajectories using the HMM parameters. Finally, we decompose the affinity matrix to find the largest eigenvectors that are used to obtain the clusters of trajectories. We determine the optimal cluster number using a validity score.

2.1. Trajectories to HMM Parameters

A trajectory $T(t)$ of an object is represented as the collection of image coordinates that correspond to the center-of-masses of the object shape in the consecutive frames. We extract trajectories using a mean-shift and Gaussian mixture model based object tracker presented in [8]. Since the clustering requires such feature vectors that we can compute pair-wise distance between them, the extracted trajectories are transferred onto a parameter space. This parameter space is spanned using the HMM parameters.

An HMM is a probabilistic model composed of a number of interconnected states in a directed graph, each of which emits an observable output. Each state is characterized by two probability distributions: the transition distribution over states and the emission distribution over the output symbols. A random source described by such a model generates a sequence of output symbols. Since the activity of the source is observed indirectly, through the sequence of output symbols, and the sequence of states is not directly observable, the states are said to be hidden.

We use identical, left-to-right topology HMM's with the same number of states and same number of models in the mixture. We assign the following attributes to define an HMM; 1) a set of prior probabilities π, 2) a set of state transition probabilities B, 3) mean, variance and weights of mixture models μ, σ, w. We prefer the left-to-right topology since it can efficiently describe continuous processes such as a trajectory. We train a HMM model using the trajectory coordinates (as well as instantaneous speed and orientation) as the training data. As a result, each trajectory is assigned to a separate model. We, then, construct a vector $H = (\pi, B, \mu, \sigma, w)$ using the HMM state transfer probabilities, model means statistics and model weights, and the prior probabilities.

3. FEATURES TO PATTERNS

For each trajectory, an affinity matrix A is constructed. The elements a_{ij} of the matrix A are equal to the similarity of the corresponding trajectories i and j. The similarity is defined as $a_{ij} = e^{-d(T_i,T_j)}$. Note that matrix $A \in \mathcal{R}^{n\times n}$ is a real semi-positive symmetric matrix, thus $A^T = A$. The distance $d(i,j)$ is measured using a mutual fitness score of the models and input trajectories as

$$\begin{aligned} d(T_i, T_j) &= |L(T_i; H_i) + L(T_j; H_j) \\ &\quad -L(T_i; H_j) - L(T_j; H_i)| \end{aligned} \quad (1)$$

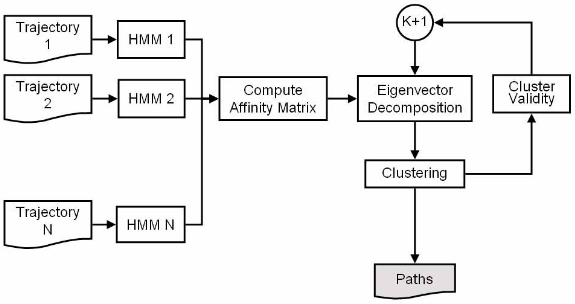

Fig. 1. Flow diagram.

The $L(T_i; \lambda_i), L(T_j; \lambda_j)$ terms indicate the likelihood of the trajectories to their own fitted model, i.e. we obtain the maximum likelihood response for the models. The cross terms $L(T_i; \lambda_j), L(T_j; \lambda_i)$ reveal the likelihood of a trajectory generated by the other trajectories model. In other words, if two trajectories are identical, the cross terms will have a maximum value, thus the distance will be equal to zero. On the other hand, if trajectories are different, their likelihood of being generated from each others model will be small, thus the distance will be high.

For affinity matrix A there are n eigenvalues λ with associated eigenvectors $\mathbf{v}$ which satisfy $A\mathbf{v} = \lambda\mathbf{v}$. Let $V \equiv [\mathbf{v}_1\ \mathbf{v}_2\ ..\ \mathbf{v}_n]$ be a matrix formed by the columns of the eigenvectors. Let D be a diagonal matrix $diag[\lambda_1, \lambda_2, .., \lambda_n]$. Lets also assume $\lambda_1 \geq \lambda_2 \geq ..\lambda_n$. Then the eigenvalue problem becomes

$$AV = [A\mathbf{v}_1\ ..\ A\mathbf{v}_n] = [\lambda_1\mathbf{v}_1\ ..\ \lambda_n\mathbf{v}_n] = VD \quad (2)$$

and $A = VDV^{-1}$. Since A is symmetric, the eigenvectors corresponding to distinct eigenvalues are real and orthogonal $VV^T = V^TV = I$, which implies $A = VDV^T$.

3.1. Eigenvector Decomposition

The decomposition of a square matrix into eigenvalues and eigenvectors is known as eigenvector decomposition. The eigenvectors are computed iteratively. The main idea behind the iterative computation is the following. Suppose we have some subspace $\mathcal{K}$ of dimension k, over which the projected matrix A has eigenvalue θ_k and a corresponding eigenvector $\mathbf{u}_k$. Let us assume that an orthogonal basis for $\mathcal{K}$ is given by the vectors $\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_k$ (already determined eigenvectors). Quite naturally the question arises how to expand the subspace in order to find a successful update for $\mathbf{u}_k$, which will become v_{k+1}. To that end we compute the defect $\mathbf{r} = A\mathbf{u}_k - \theta_k\mathbf{u}_k$. Then, we compute $\widetilde{\mathbf{z}}$ from $(D - \theta_k I)\widetilde{\mathbf{z}} = \mathbf{r}$, where D is the diagonal matrix of A as defined above. The vector $\widetilde{\mathbf{z}}$ is made orthogonal to $\mathcal{K}$, and the resulting vector is chosen as the new $\mathbf{v}_{k+1}$ by which $\mathcal{K}$ is expanded. This method find the largest eigenvalues in absolute value. The matrix $(D - \theta_k I)^{-1}$ can be viewed as a preconditioner for the vector $\mathbf{r}$. To avoid this stagnation, we concentrate on the kth approximation $\mathbf{u}_k$ of the eigenvector $\mathbf{v}$, where $\mathbf{u}_k$ is normalized $||\mathbf{u}_k|| = 1$. The residual $\mathbf{r} = A\mathbf{u}_k - \theta_k \mathbf{u}_k$ is orthogonal to $\mathbf{u}_k$ because $\theta_k = \mathbf{u}_k^T A \mathbf{u}_k$ is the value associated with $\mathbf{u}_k$. We project the eigenvalue problem $A\mathbf{v} = \lambda \mathbf{v}$ on span($\mathbf{u}_k$), and on its orthogonal complement. This leads to two coupled equations for λ and the complement $\mathbf{z}$ of $\mathbf{v}$ orthogonal to $\mathbf{u}_k$: $\lambda = u_k^T A(\mathbf{u}_k + \mathbf{z})$ and $\mathbf{z} \perp \mathbf{u}_k, \quad (I - \mathbf{u}_k \mathbf{u}_k^T)(A - \lambda I)(I - \mathbf{u}_k \mathbf{u}_k^T)\mathbf{z} = -\mathbf{r}$. Since λ is unknown, we cannot compute optimal update $\mathbf{z}$ from $\mathbf{u}_k$. However it is reasonable to replace λ by the current approximation θ_k. Thus we obtain $\mathbf{r} \perp \mathbf{u}_k, \quad (I - \mathbf{u}_k \mathbf{u}_k^T)(A - \theta_k I)(I - \mathbf{u}_k \mathbf{u}_k^T)\mathbf{z} = -\mathbf{r}$ as a good correction for $\mathbf{u}_k$. Similarly, we compute the approximate solution $\widetilde{\mathbf{z}}$ using this equation, and by making $\widetilde{\mathbf{z}}$ orthogonal to search space, we obtain $\mathbf{v}_{k+1}$. Briefly, we extract an approximate eigenvalue from the search subspace, project it, solve the projected eigenvalue problem, compute the corresponding value and residual, correct the approximate eigenvector $\mathbf{u}$, and expand the search subspace with the correction vector.

The above iterative prediction is used at the clustering stage.

3.2. Clustering

Although eigenvector based clustering [3] is addressed before in the literature, to our knowledge no one has established the relationship between the optimal clustering of the data distribution and the number of eigenvectors that should be used for spanning before. Here we show that the number of eigenvectors is proportional to the number of clusters.

Let a matrix P_k be a matrix in a subspace $\mathcal{K}$ that is spanned by the columns of V such as $P_k = [\mathbf{v}_1 \ \mathbf{v}_2 \ .. \ \mathbf{v}_k, \ 0]$ where V is the orthogonal basis satisfies $A = VDV^T$. Now, we define vectors $\mathbf{p}_n$ as the rows of the truncated matrix P_k as

$$P_k = \begin{bmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_n \end{bmatrix} = \begin{bmatrix} v_{11} & \cdots & v_{1k} & 0 & \cdots \\ \vdots & & & & \vdots \\ v_{n1} & \cdots & v_{nk} & 0 & \cdots \end{bmatrix} \tag{3}$$

We normalize each row of matrix P_k by $p_{ij} \leftarrow p_{ij} / \sqrt{\sum_j^k p_{ij}^2}$. Then a correlation matrix is computed using the normalized rows by $C_k = P_k P_k^T$. For a given P_k, the value of p_{ij} indicates the degree of similarity between the trajectory i and trajectory j. Values close to one correspond to a match whereas negative values and values close to zero suggest that trajectories are different. Let ϵ be a threshold that transfers values of matrix C_k to the binary quantized values of an association matrix W_k as

$$w_{ij} = \begin{cases} 1 & c_{ij} \geq \epsilon \\ 0 & c_{ij} < \epsilon \end{cases} \tag{4}$$

where $\epsilon \approx 0.5$. Then clustering becomes grouping the trajectories that have association values equal to one $w_{ij} = 1$.

To explain why this works, remember that eigenvectors are the solution of the classical extremal problem $\max \mathbf{v}^T A \mathbf{v}$ constrained by $\mathbf{v}^T \mathbf{v} = 1$. That is, find the linear combination of variables having the largest variance, with the restriction that the sum of the squared weights is 1. Minimizing the usual Lagrangian expression $\mathbf{v}^T A \mathbf{v} - \lambda(\mathbf{v}^T \mathbf{v} - 1)$ implies that $A\mathbf{v} = \lambda \mathbf{v}$. Thus, $\mathbf{v}$ is the eigenvector with the largest eigenvalue.

As a result, when we project the affinity matrix columns on the eigenvector $\mathbf{v}_1$ with the largest eigenvalue and span $\mathcal{K}_1$, the distribution of the a_{ij} will have the maximum variance therefore the maximum separation. With the same reasoning, the eigenvector $\mathbf{v}_2$ with the second largest eigenvalue, we will obtain the basis vector that gives the best separation after normalizing the projected space using the $\mathbf{v}_1$ since $\mathbf{v}_1 \perp \mathbf{v}_2$.

As opposed to using only the largest or first and second largest eigenvectors, the correct number of eigenvectors should be selected with respect to the target cluster number. Using only one or two eigenvectors does fail for the case of multiple clusters. After each eigenvalue computation of matrix A in the iterative algorithm, we compute a validity score α_k using the clustering results as

$$\alpha_k = \sum_c^k \frac{1}{N_c} \sum_{i,j \in Z_c} p_{ij} \tag{5}$$

where Z_c is set of trajectories included in the cluster c, N_c number of trajectories in Z_c. The validity score gets higher values for the better fits. Thus, by evaluating the local maxima of this score we determine the correct cluster number automatically. Thus, we answer the natural question of clustering; ”what should be the total cluster number?”

As a summary, the clustering for a given maximum cluster number k^* includes

1. Compute A, approximate eigenvectors using Ritz values $\lambda_k \simeq \theta_k$, find eigenvectors v_k for $k = 1, .., k^*$,
2. Find $P_k = V_k V_k^T$ and Q_k for $k = 1, .., k^*$,
3. Determine clusters and calculate α_k,
4. Compute $\alpha' = d\alpha/dk$ and find local maxima.

The maximum cluster number k^* does not affect the determination of the fittest cluster; it only limits the maximum number of possible clusters that will be searched.

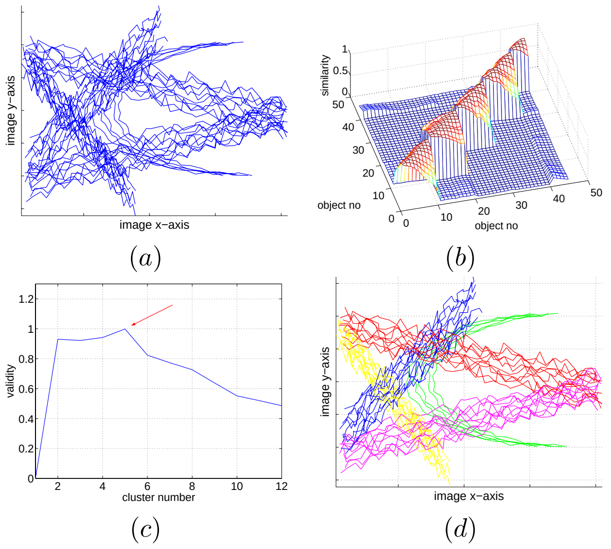

Fig. 2. (a) Set of trajectories, (b) corresponding affinity matrix, (c) validity score, (d)result of automatic clustering.

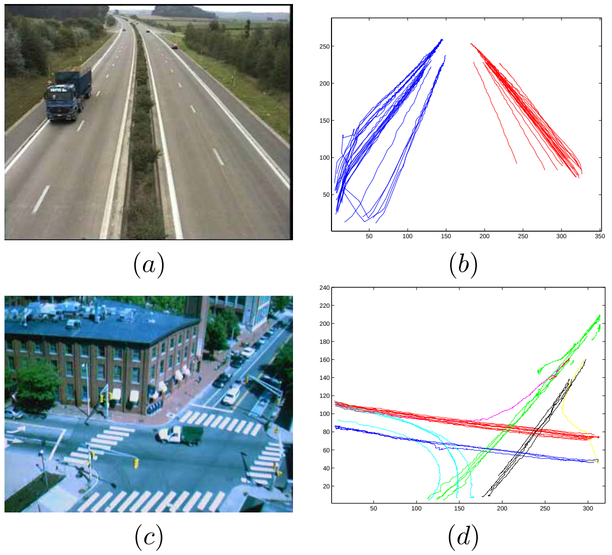

Fig. 3. First row: (a) sample image, (b) detected paths. Second row: (c) sample image, (d) detected paths.

4. EXPERIMENTS AND DISCUSSION

We simulated trajectory pattern learning by using the trajectories given in fig. 2-a. In this set, there are 5 distinct paths exist. We show the corresponding affinity matrix for the pair-wise HMM parameter distances in fig. 2-b. We obtained the optimal number of clusters using the validity score α as presented in fig. 2-c. The maximum validity score is computed for $n = 5$. The clustered trajectories after eigenvector clustering are given in fig. 2-d. As visible, the proposed method successfully found the correct clusters.

Figure 3 presents the detection results for two vehicle traffic setups; a highway surveillance scenario (fig. 3-a), and a street surveillance (fig. 3-c). The highway scenario has a total of 153 trajectories that are automatically grouped into 2 separate paths as presented in fig. 3-b, where the cluster validity score for this scenario reached its maximum value. The street scenario has a total of 39 trajectories that are accurately grouped into 7 clusters using the proposed method. For both generic and real data simulations, the proposed method accurately estimated the optimal number of patterns, and accurately clustered the trajectories into the estimated patterns.

In conclusion, the main contributions of this paper are:

- We proposed a new set of HMM based features that enable detection of patterns, which could not be detected using conventional representations.
- We showed that the number of largest eigenvalues (in absolute magnitude) to span subspace is one less than the number of clusters.
- We proposed an unsupervised clustering framework based on the above and successfully applied it to trajectory pattern detection.

5. REFERENCES

[1] J.W. Davis and A.F. Bobick, “The representation and recognition of human movement using temporal templates”, *CVPR*, 928-934, 1997.

[2] L. Davis, R. Chelappa, A. Rosenfeld, D. Harwood, I. Haritaoglu, and R. Cutler, “ Visual Surveillance and Monitoring”, *Proc. DARPA Image Understanding Workshop*, 73-76, 1998.

[3] G.L. Scott and H. C. Longuet-Higgins, “Feature grouping by relocalisation of eigenvectors of the proxmity matrix” *BMVC*, 103-108, 1990.

[4] N. Johnson and D. Hogg, “Learning the distribution of object trajectories for event recognition”, *BMVC*, 583592, 1995.

[5] T. Starner and A. Pentland, “Visual recognition of american sign language using hidden markov models”, *Proc. Int’l Workshop Automatic Face- and Gesture-Recognition*, 1995.

[6] C. Stauffer and W.E. Grimson, “Learning patterns of activity using real-time tracking”, *PAMI*, 22(8), 747-757, 2000.

[7] A. Ng, M. Jordan, and Y. Weiss, “On spectral clustering: Analysis and an algorithm”, *Proc. of Advances in Neural Information Processing Systems*, 2001.

[8] F. Porikli and O. Tuzel, ”Human body tracking by adaptive gaussian models and mean-shift analysis”, *Proc. of ICVS - PETS*, Graz, 2003.