Ambiguity Detection by Fusion and Conformity: A Spectral Clustering Approach

Fatih Porikli
Mitsubishi Electric Research Laboratories
Cambridge, MA, 02139, USA
fatih@merl.com

Abstract

Event detection requires interpretation of the "semantically meaningful" object actions. To achieve this task, the gap between the numerical features of objects and the symbolic description of the meaningful activities needs to be bridged. We develop an ambiguity detection framework that has two significant advantages over past work. First, we introduce a fusion method for a set of time-wise and object-wise features including not only the trajectory coordinates but also the histograms and HMM based representations of object's speed, orientation, location, size, and aspect ratio. This fusion method enable detection of events that cannot be detected with the existing trajectory features reported so far. Second, we improve existing spectral clustering algorithms by automatically estimating the optimal number of clusters. Furthermore, we determine the conformity of the objects within the given data space. We compute a separate HMM for each object using a time-series that is composed of the mixture of its features. Then, we construct an aggregated affinity matrix from the pair-wise similarity scores of objects using the HMM's. We apply eigenvector decomposition and obtain object clusters. We show that the number of eigenvectors used in the decomposition is proportional to the optimal number of clusters. We examine the affinity matrix to determine the deviance of objects from common assemblages within the space. Our simulations reveal that the proposed detection methods accurately discover both usual and unusual events.

1 Motivation

Although many algorithms exist for unsupervised classification of patterns into clusters, most of these methods require the data space X consists of 'identical length' data points (feature vectors) $x_i = (x_{i1}, ..., x_{iN})$ where N is the dimension of the data space, i.e. $X : \mathcal{R}^N$. Such algorithms include the ordinary implementations of decision trees, neural nets, Bayesian classifiers, ML-estimators, support vector machines, Gaussian mixture models, k-means, and hierarchical approaches, self-organizing maps, etc [2].

On the contrary, not all classification problems can be formulated into a data space that contains only constant length feature vectors. For instance, in a surveillance setting, the trajectories of people may have quite different temporal and spatial lengths. One way to adapt variable length data for ordinary classification, is to normalize the length of the feature vectors. However, such a temporal normalization of the feature vector length causes severe degradation and aliasing problems.

Thus, we project our variable length features into a model based space in which we can compute pair-wise affinities. Note that we are not "parameterizing" the features onto a uniform space, but we convert the affinity computation problem into a process of pair-wise comparisons using stochastic models capable of capturing the temporal and other aspects of the input features. We choose Hidden Markov Model (HMM), which captures the probabilistic transition properties of sequential data, as a model machine. Since we end up having the pair-wise correspondences, we apply spectral clustering.

One fundamental question of automatic cluster discovery is how to estimate the number of "meaningful" clusters in the data. We seek answers to this question by analyzing the fitness of the clusters obtained after a recursive eigenvector decomposition of the affinity matrix using increasing number of ordered eigenvectors. In [5], we introduced a conformity score to detect unusual events. Here, we extend the conformity score by evaluating the variance of affinity both inside (intra) and outside (inter) of the obtained clusters, and we monitor the evolution of this score to achieve the minimum variance. The optimum number of clusters is observed as the eigenvector decomposition that gives the minimum intra and inter variance. In this work, we concentrate on the outputs that produced by object tracking, which are coordinate sequences with an associated histogram and tensor-based data. We give a flow diagram in Fig. 1.

Next, we explain HMM's and fusion. In the following sections, we discuss affinity matrix, eigenvector decomposition, estimating the number of clusters, and finding unusual

KIMAS 2005, April 18-21, 2005, Waltham, MA, USA.
0-7803-9013-X/05/$20.00 © 2005 IEEE.

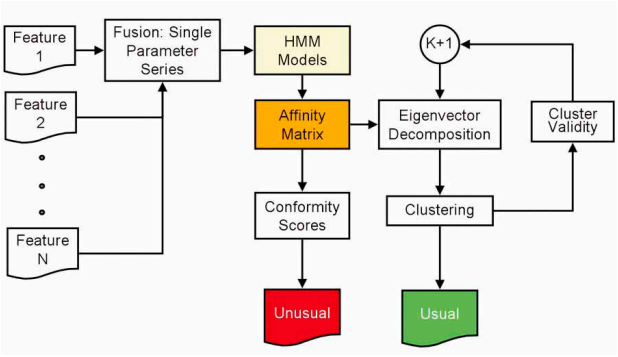

Figure 1: Tracking features are restructured into a time-series and one HMM model is fitted for each sequence. The affinity matrix is calculated by comparing the HMM's of the given objects.

events. Then, we give details of the clustering and present simulation results.

2 Model Machines: HMM

An HMM is a probabilistic model composed of a number of interconnected states, each of which emits an observable output. A discrete hidden Markov model is defined by a set of states and an alphabet of output symbols [6]. 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 as follows: at each time step the source is in one state, and after emitting an output symbol according to the emission distribution of the current state, the source jumps to a next state according to the transition distribution of its current state. 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.

An K-state $\{S_1, S_2, ..., S_K\}$, HMM is represented by:

1. A set of prior probabilities $\pi = \{\pi_i\}$ where $\pi_i = P(q_1 = S_i)$,$1 \leq i \leq K$.
2. A set of state transition probabilities $H = \{h_{ij}\}$, where $h_{ij} = P(q_{t+1} = S_j | q_t = S_i)$,$1 \leq i, j \leq K$.
3. A set of output distributions $B = \{b_{ij}\}$, where $b_{ij}(y) = P(O_{t+1} = y | q_t = S_i, q_{t+1} = j)$,$1 \leq i, j \leq K$.

where q_t and O_t are the state and observation respectively at time t. It is common to denote the an M-mixture of HMM's by (H_m, B_m, π_m), $1 \leq m \leq M$. For an HMM, algorithms exist for: 1) computing the probability of observing a sequence, given a model, 2) finding the state sequence that maximizes the probability of the given sequence, when the model is known (the Viterbi algorithm), 3) inducing the HMM that maximizes (locally) the probability of the given sequence (the BaumWelch algorithm, an expectationmaximization algorithm). The problem of estimating the correct number of nodes is a difficult one: a full Bayesian solution for obtaining the posterior probability on M, requires a complex integration over the HMM parameter space, as well as knowledge about the priors on the mixture parameters and about the priors on M itself. Often this integration cannot be solved in closed form, and Monte-Carlo methods and other approximation methods are used to evaluate it.

For each sequence, we fit a HMM (a discrete model in case the sequence components are labels, a continuous model in case the components are real numbers that reflect certain proportional properties, e.g. magnitude, coordinate, etc). The number of states K, number of models M, and the HMM topology (left-to-right) are assigned same for each sequence. A set of parameters f_i corresponds the HMM variables s_i ($f_i = (H_m, B_m, \pi_m)_i$) as described above. This set of parameters, which consists of the state transition, observation, and prior matrix, enable us to compute distances between a pair of objects.

3 Fusion

A trajectory is a time series of coordinates representing the motion path of an object over the duration, i.e. number of frames that object exists. Certain object properties are dynamic and change their values from frame to frame during the tracking process, e.g. the speed of an object. Instead of depending only on the instantaneous values, using normalized histograms as features enables to capture the history of these dynamic features. A histogram in fact corresponds to the probability density distribution, thus it sustains statistical characteristics such as mean, variance and other higher order moments. There are several scalar features that describe an object. In spite of its simplicity, the duration is one of the distinctive features. A general orientation descriptor records the direction of the object between its first and last appearance. Other dynamic properties, such as orientation, aspect ratio, slant (angle between a reference axis and the main diagonal of object), size, instantaneous speed, location, and color, are approximated by histograms. The color properties may be represented by a conventional histogram or by a few number of dominant colors with an additional computational cost.

It is apparent that running separate clustering algorithms on each of these features would be computationally expensive, susceptible to errors due to insufficiently informative features, and also it will bring another problem of the explaining and merging of these separate clustering results.

Thus, we restructure the various features of an object into an aggregated time-series form. We combine sequences, histograms, label data, and scalars into a single series of tensors, in which a tensor represents the current coordinate, the current orientation, current color histogram, current shape, and etc. In other words, at each time instant on the trajectory, we have not only the coordinate information but also the associated color, label, etc. This combined series will be in fact a sequence of vectors describing object's current properties. Thus, when we fit a HMM for an object, the model will contain the dynamics of all the above features.

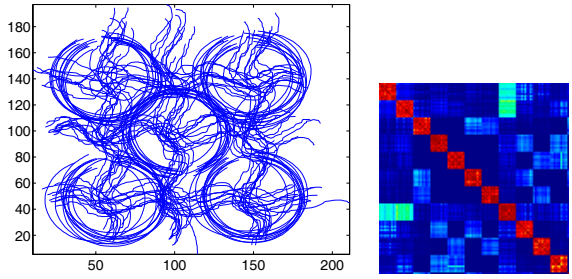

Figure 2: Set of trajectories and the corresponding affinity matrix.

4 Affinity Matrix and Clustering

Now, we can compute our affinity matrix. Given two sequences s_i and s_j, and two corresponding models f_i and f_j, we compute a likelihood score that indicates the fitness of the given sequences to the models. There are two term to evaluate self-fitness and cross-fitness; self-fitness is the likelihood of sequences are generated by their corresponding models, i.e. sequence s_i is generated by model f_i and sequence s_j is generated by f_j. In an ideal representation, these likelihood should have maximum value, which means that a model can generate only one sequence, and each sequence perfectly fits into its own model. The second term evaluates the cross-likelihood. We compute the likelihood of each sequence is generated by the other model, i.e. sequence s_i is generated by model f_j and sequence s_j is generated by f_i. In case the input sequences are similar, the cross-likelihood will have a higher value. However, if the sequences are different, then their models will not be same, and the cross-likelihood will be small. We put this notion into the following formulation as:

$$\begin{aligned} d(s_i, s_j) \quad = \quad & |L(s_i|f_i) + L(s_j|f_j) \\ & -L(s_i|f_j) - L(s_j|f_i)| \end{aligned} \tag{1}$$

where $L(\cdot)$ is the likelihood. Then, the elements a_{ij} of the affinity matrix A are equal to

$$a_{ij} = e^{-d(s_i,s_j)/2\sigma^2} \tag{2}$$

and σ^2 is a normalizing factor. The affinity matrix have values close to 1 if the corresponding sequences fit well to each other's models, and close to 0 otherwise. Note that similarity matrix $A \in \mathcal{R}^{n\times n}$ is a real semi-positive symmetric matrix, thus $A^T = A$. We give a sample affinity matrix for 110 random length trajectories in Fig. 2. As visible, the described HMM likelihood accurately captures the affinities between the trajectories.

Although spectral clustering [1], [7], [4], [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. Here we show that the number of eigenvectors is proportional to the number of clusters.

Let $V \equiv [\mathbf{v}_1 \ \mathbf{v}_2 \ .. \ \mathbf{v}_M]$ be a matrix formed by the columns of the eigenvectors. Let D be a diagonal matrix $diag[\lambda_1, .., \lambda_M]$. Lets also assume eigenvalues are $\lambda_1 \geq \lambda_2 \geq ...\lambda_M$. Then the generalized eigenvalue problem is

$$(A-I)V = [A\mathbf{v}_1 \ .. \ A\mathbf{v}_M] = [\lambda_1\mathbf{v}_1 \ .. \ \lambda_M\mathbf{v}_M]D = VD \tag{3}$$

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 $A = VDV^T$.

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

$$P_k = \begin{bmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_M \end{bmatrix} = \begin{bmatrix} v_{11} & \cdots & v_{1k} & 0 & \cdots \\ \vdots & & & & \vdots \\ v_{M1} & \cdots & v_{Mk} & 0 & \cdots \end{bmatrix} \tag{4}$$

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 $C_k = P_kP_k^T$. For a given P_k, the value of p_{ij} indicates the degree of similarity between the object i and object j. Values close to one correspond to a match whereas negative values and values close to zero suggest that objects 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

$$w_{ij} = \begin{cases} 1 & c_{ij} \geq \epsilon \\ 0 & c_{ij} < \epsilon \end{cases} \tag{5}$$

where $\epsilon \approx 0.5$. The clustering is then becomes grouping the objects 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

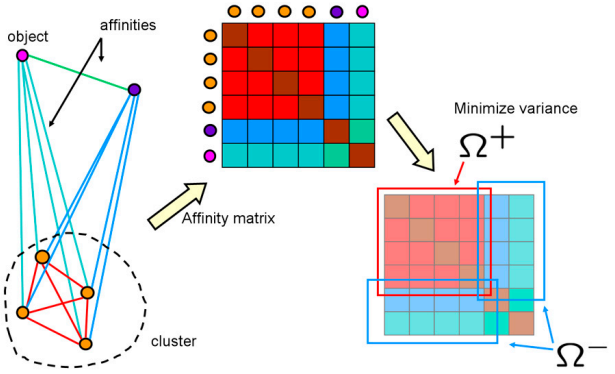

Figure 3: Validity measures the variance of affinity intra and inter clusters.

$\max \mathbf{v}^T A\mathbf{v}$ 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 $(I - A)\mathbf{v} = \lambda I\mathbf{v}$. 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. Keep in mind that a threshold operation will perform best if the separation is high. To this end, if the distribution of values have only two distinct classes then a balanced threshold passing through the center will divide the points into two separate clusters. 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 $\mathbf{v}_1 \perp \mathbf{v}_2$. Thus, as a base rule, the number of largest eigenvalues (in absolute value) to span subspace is one less than the number of clusters.

As opposed to using only the largest or first and second largest eigenvectors (also the generalized second minimum which is the ratio of the first and the second depending the definition of affinity), the correct number of eigenvectors should be selected with respect to the target cluster number. Using only one or two does fail for multiple clusters scenarios.

We obtained projections that gives us the maximum separation but we did not determine the degree of separation i.e. maximum and minimum values of projected values on the basis vectors. For convenience, we normalize the projections i.e. the *rows* of current projection matrix (P_k) as $\mathbf{p}\mathbf{p}^T = 1$ and then compute the correlation $P_k P_k^T$. Correlation will make rows that their projections are similar to get values close to 1 (equal values will give exactly 1), and dissimilar values to 0. By maximizing the separation (distance) between the points in different clusters on an orthonormal basis, we pushed for the orthogonality of points depending their clusters; $\mathbf{p}_i\mathbf{p}_j^T \approx 1$ if they are in the same cluster, and $\mathbf{p}_i\mathbf{p}_j^T \approx 0$ if they are not.

As a summary, the clustering process requires

1. Computation of A,
2. Extraction of eigenvectors v_k for $k = 1, .., k^*$,
3. Decomposition $C_k = P_k P_k^T$ and W_k for $k = 1, .., k^*$,
4. Thresholding the decomposed matrix,
5. Combining objects that have affinity higher than the threshold in the same clusters,
6. Computation of validity score α_k.

The maximum possible cluster number k^* does not affect the determination of the fittest cluster; it is only an upper limit on the number of eigenvectors in decomposition.

After each clustering stage, we compute a validity score α_k as

$$\alpha_k = \frac{1}{M_k}\sum_{i=1}^{M_k}(var(\Omega_i^+) + var(\Omega_i^-)) \qquad (6)$$

where Ω_i is a cluster of objects such that Ω_i^+ is the affinity values of the objects within this cluster, thus $var(\Omega_i^+)$ is the variance of these values. Ω_i^- is the variance of affinity values between the objects such that one object within the cluster and the other is outside. M_k is number of clusters as illustrated in Fig. 3. The validity score gets lower values for the better fits. By evaluating the minima of this score we determine the correct cluster number automatically. Thus, we answer the basic question of clustering; "what should be the total cluster number?"

5. Detection of Unusual Events

Using the affinity matrix, conformity scores of the objects are computed. The conformity score of an object is the sum of the corresponding row (or column) of the affinity matrix. The object that has the minimum score corresponds to most different, thus most unusual event.

One distinct advantage of the conformity score is that it does not assert unusuality in case all events are similar as illustrated in Fig. 5. In this example, we generated points on a circle, and then located four additional points inside the circle. Three of these points are closer each other than the fourth point. A human will easily interpret that all the points along the circle are in one group and the three points inside circle constitutes another smaller group, and the remaining single point is not similar to any other points in the data space. That is what we want to accomplish after unusual event detection. When we computed the conformity scores, we observed that the points on the circle have almost

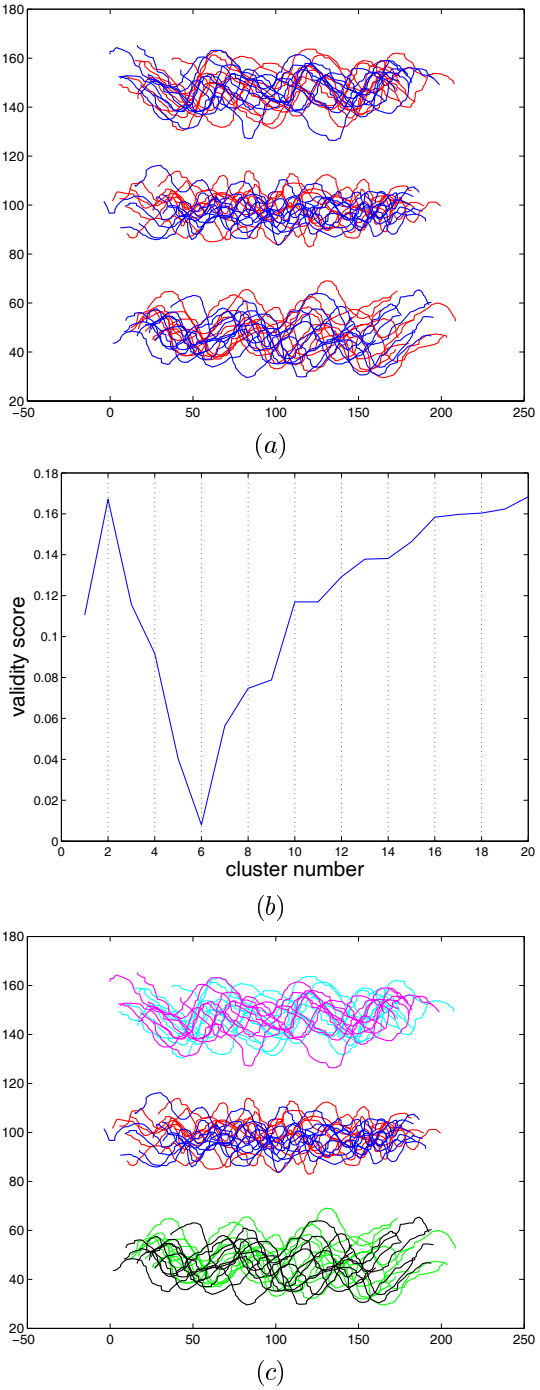

Figure 4: Fusion: (a) Set of input trajectories. In this example set, objects have a color feature in addition to the trajectory, i.e. half of the objects are red and others are blue. (b) Validity scores. The minimum is obtained at the cluster number 6. (c) Corresponding clustering results. As visible, the additional feature is successfully fused within the trajectory information, and the algorithm accurately estimated the correct number of cluster.

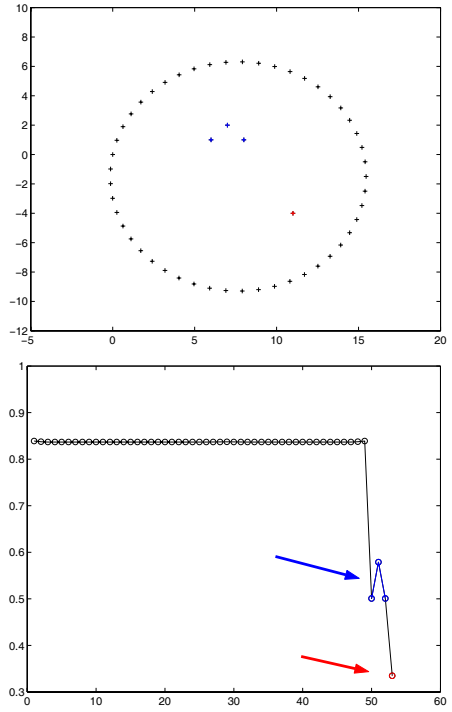

Figure 5: **Top:** Set of objects as points in 2D. **Bottom:** Corresponding conformity scores. Note that, as an object becomes more distinctive, its conformity score drops accordingly.

identical scores. For the smaller group of three points, the conformity dropped. Similar to human cognition, the single point had the smallest score, thus it is obtained as the most unusual.

We also conducted another test that two clusters become increasingly separated from each other as shown in Fig. 7. We observed that, as the distance between the main cluster and smaller cluster increases, the conformity of the points belong to the smaller cluster decreases. When the cluster were together, the conformity scores were similar. These examples show that the proposed method can accurately distinguish cases that there are a measurable unusuality from the cases that nothing significant occurs as opposed to the conventional approaches. The conformity score effectively determines the degree of the ambiguity as well as it distinguishes the unusual event from the ordinary.

Figure 6 shows results for clustering of 111 trajectories. There are 11 similar clusters that each consists of 10 trajectory. The remaining 111^{th} trajectory has a different path, and it is not similar to the rest. The validity score has accurately estimated the optimal number of clusters as 12. The clusters are color coded in Fig. 6-c. As visible, all trajectories are successfully identified within their appropriate clus-

ters. We also computed the conformity scores. The red arrow in the conformity graph points the most distinct trajectory, which is also painted red in the clustering results. Note that, since most trajectories passes through the center of the graph, the trajectories marked as green have higher correlation with the rest of the trajectories, and the proposed method gave higher conformity scores as expected.

6 Summary

In conclusion, the main contributions of this paper are:

- We proposed a method to compare the variable length sequences using the HMM parameter space
- We showed that the number of largest eigenvalues (in absolute value) to span subspace is proportional to the number of clusters.
- We used the above result as a quality assessment criterion for cluster fit.
- We defined a measure to discover the unusual objects.

References

[1] G.L. Scott and H. C. Longuet-Higgins, "Feature grouping by relocalisation of eigenvectors of the proxmity matrix" *In Proc. British Machine Vision Conference*, 103-108, 1990.

[2] A. K. Jain , M. N. Murty , P. J. Flynn, "Data clustering: a review", *ACM Computing Surveys (CSUR)*, 31(3), 264-323, 1999.

[3] M. Meila and J. Shi, "Learning segmentation by Random Walks", *Proc. of Advances in Neural Information Processing Systems*, 2000.

[4] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm", *Proc. of Neural Information Processing Systems*, 2001.

[5] F. Porikli, T. Haga, "Event detection by eigenvector decomposition using object and frame features", *In Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshop*, 2004.

[6] L. Rabiner. "A tutorial on hidden markov models and selected applications in speech recognition", *Proceedings of IEEE*, 77(2), 257285, 1989.

[7] Y. Weiss, "Segmentation using eigenvectors: a unifying view", *Proceedings IEEE International Conference on Computer Vision*, 975-982, 1999.

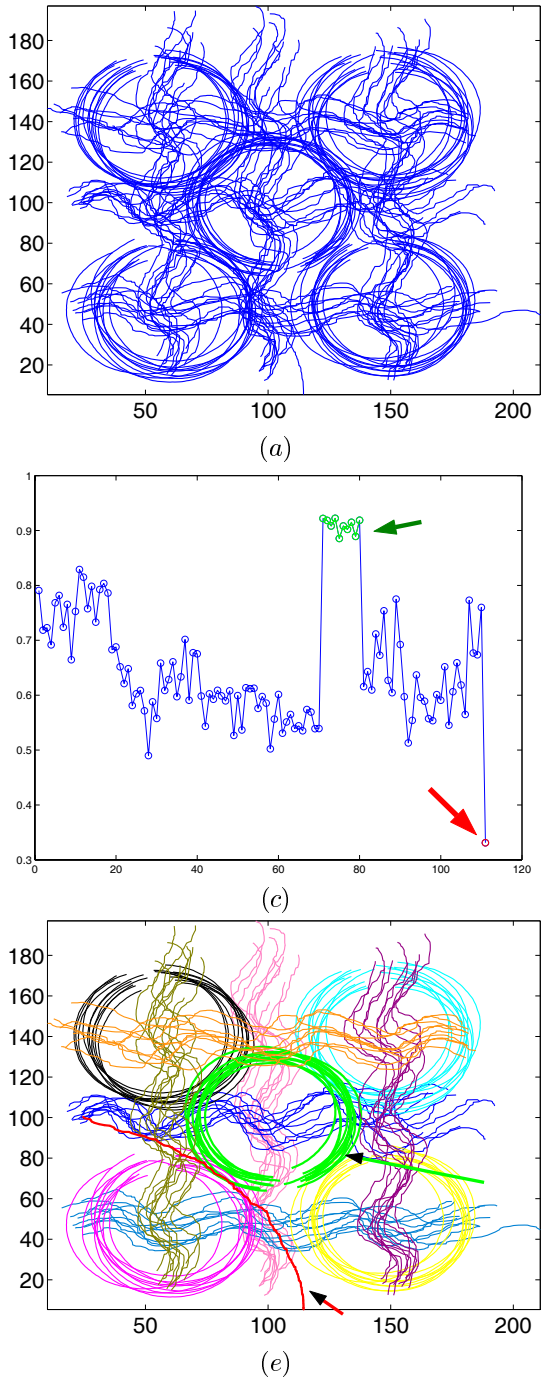

Figure 6: (a) Set of trajectories. (b) Corresponding conformity scores. (c) Result of automatic clustering. Red arrow in the conformity graph shows the most distinct trajectory, which is also pointed by an arrow in the clustering result. Green arrows both indicate the most common trajectories. Note that, since most trajectories passes through the center on the graph, the trajectories marked as green are more commonality with the rest, and the proposed method gave higher conformity scores as expected.

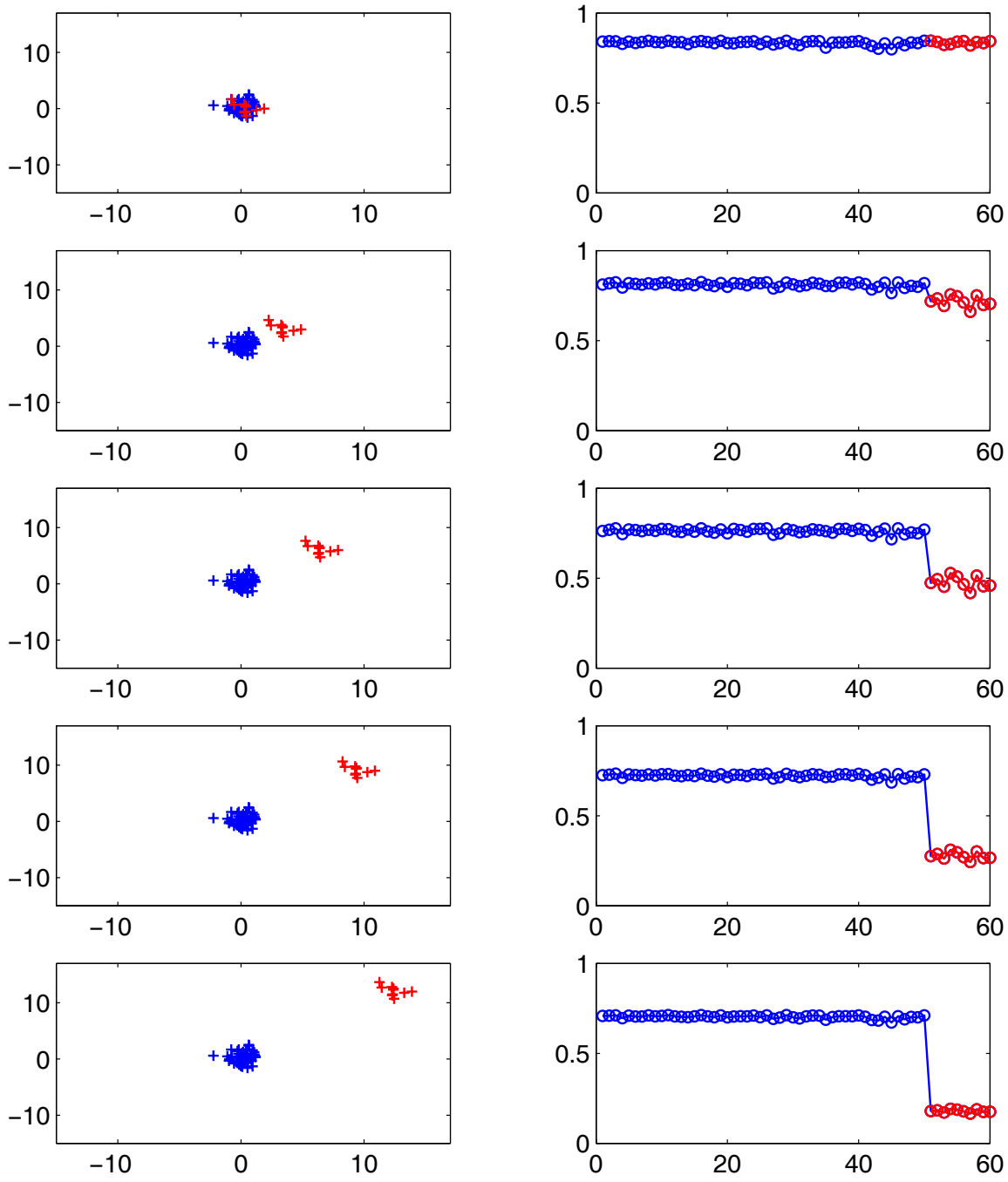

Figure 7: **Left column:** Objects. **Right column:** Conformity scores. As the clusters become different the conformity scores of the objects changes. Note that, in case all objects are similar as in the top example, our method successfully recognize that there is no unusual objects as opposed to always ordering objects. It is evident that, the separation is accurately reflected to the conformity scores.