Motion based behaviour learning, profiling and classification in the presence of anomalies

Shehzad Khalid *,

Department of Computer Science and Engineering, Bahria University, Islamabad, 44000, Pakistan

Abstract

Techniques for understanding video object motion activity are becoming increasingly important with the widespread adoption of CCTV surveillance systems. Motion trajectories provide rich spatiotemporal information about an object's activity. This paper presents a novel technique for clustering and classification of object trajectories using basis function approximation. In the proposed motion learning system, trajectories are treated as time series and modelled using modified DFT-based representation. A framework (Iterative HSACT-LVQ) is proposed for learning of patterns in the presence of significant number of anomalies in training data. A novel modelling technique, referred to as m-Mediods, is also proposed that models the class containing n members with m Mediods. Once the m-Mediods based model for all the classes have been learnt, the classification of new trajectories and anomaly detection can be performed by checking the closeness of said trajectory to the models of known classes. A mechanism based on agglomerative approach is proposed for anomaly detection. Our proposed techniques are validated using variety of simulated and complex real life trajectory datasets.

Key words: Object trajectory, dimensionality reduction, trajectory modelling, trajectory clustering, event mining, anomaly detection, motion recognition.
PACS:

1 Introduction

An increasing number of systems are now able to capture and store data about object motion such as those of humans and vehicles. This has acted as a spur to

* Corresponding author.
Email address: shehzad_khalid@hotmail.com (Shehzad Khalid).

the development of content-based visual data management techniques for tasks such as motion data search and retrieval, discovery and grouping of similar motion patterns, detection of anomalous behaviour, motion understanding and prediction. These techniques are essential for the development of next generation ‘actionable intelligence’ surveillance systems.

Much of the earlier research focus, in motion analysis, has been on high-level object trajectory representation schemes that are able to produce compressed forms of motion data [1][2][3][4][5][6][7]. The literature on trajectory-based motion understanding and pattern discovery is less mature but advances using Learning Vector Quantization (LVQ)[8], Self-Organising Maps (SOMs) [9][10], Hidden Markov Models (HMMs) [11][12], and fuzzy neural networks [13] have all been reported. Most of these techniques attempt to learn high-level motion behaviour patterns from sample trajectories using discrete point sequences as input to a machine learning algorithm. For realistic motion sequences, convergence of these techniques is slow and the learning phase is usually carried out offline due to the high dimensionality of the input data space.

Current systems mainly took a supervised learning approach by basing their motion analysis on a predefined classification of trajectories. Supervised approaches to motion learning are less useful in video surveillance applications since labelled training data are not usually available, or is impractical to obtain. Techniques are required that learn motion activity patterns in an unsupervised manner. Unsupervised learning of motion patterns and developing a high-accuracy activity classification system are challenging tasks. The problem gets even more complicated in the presence of anomalies. Typically, in visual surveillance, there are commonly observed motion paths and those that appear unusual or anomalous. This paper is focused on both of these issues in the context of trajectory-based learning and classification whilst identifying and filtering the anomalies. We use modified Discrete Fourier Transform (DFT-MOD) based coefficients for low dimensional feature space representation of trajectories. An iterative learning algorithm has been proposed to learn motion patterns while catering for the presence of significant number of anomalies. The proposed unsupervised learning algorithm does not require any prior information about the number of patterns present in the unclassified dataset. The paper also addresses the issue of modelling of normal patterns to be used later for classification of motion activities. A novel approach for model-based classification of trajectory patterns and anomaly detection is also presented. The proposed motion learning and classification technique is compared with other methods reported recently in literature, using simulated as well as realistic motion datasets.

The remainder of the paper is organized as follow. We review some relevant background material in section 2. Section 3 briefly describes our coefficient feature space representation of motion trajectory. In section 4, an iterative learning algorithm has been proposed to learn patterns in the presence of anomalies. Section 5 addresses the issue of modelling of normal patterns to be used later for classification of motion activities. A novel approach for model-based classification of trajectory patterns and anomaly detection is presented. Experiments have been performed to show the effectiveness of proposed system for trajectory-based learning and classification of motion patterns in the presence of anomalous motion samples. These experiments are reported in section 6. The last section summarises the paper.

2 Background and related work

Trajectory descriptors are known to be useful candidates for compressed representation of object motion in videos. Previous work has sought to represent moving object trajectories through a wide variety of direction schemes, polynomial models and other function approximations. [1][2][3][4][7][15][16][17][18]. It is surprising to find that many of these candidate time series indexing schemes have not yet been applied to the problem of motion data mining and trajectory clustering. Recent work has either used probabilistic models such as HMMs [19] or discrete point-based trajectory flow vectors (PBF) [8][9][13] as a means of learning patterns of motion activity. An agglomerative clustering algorithm based on the Longest Common Subsequence (LCSS) approach is proposed in [20][21] for grouping similar motion trajectories. The problem with PBF vector-encoded trajectory representation is the heavy computational burden making prospects for online learning of motion patterns remote.

Learning of patterns from trajectory data to extract high level information has gained interest quite recently. Earlier work rely upon labelled training data for model training [12][22][23]. Yacoob [38] and Bashir *et al.* [12][22] have presented a framework for modeling and recognition of human motion based on a trajectory segmentation scheme. Classification is performed using Gaussian Mixture Model (GMMs) and HMMs for trajectory modeling that relies on PCA-based representation of segmented object trajectories. In [39], a semantic event detection technique based on discrete HMMs is applied to snooker videos.

There exists some work on learning from unclassified training data such as [8][10][20][21][24][25][26][27][28]. Owen and Hunter [10] uses Self Organizing Feature Maps (SOFM) to learn normal trajectory patterns. While classifying trajectories, if the distance of the trajectory to its allocated class exceeds a threshold value, the trajectory is identified as anomalous. A similar approach is proposed by Hu [9] who performs learning of normal activity patterns using fuzzy SOM instead of SOFM. Zhang *et al.* [23] propose a semi-supervised model using HMMs for anomaly detection. Temporal dependencies are modelled using HMMs. The probability density function of each HMM state is assumed to be a GMM. A number of eigenspace clustering techniques have been proposed recently [34][44]. However, these approaches normally require known number of clusters. Some approaches, based on spectral clustering, attempts to approximate the number of clusters [32][33]. Affinity propagation-based approaches have also been proposed recently [41]. Affinity Propagation (AP) uses message passing mechanism between training data points to solve the k-medoid problem by finding representative exemplars within dataset with a similarity structure. However, AP requires the specification of two important parameters: preference parameter and the damping factor. It is very hard to know the value of these parameters that will yield optimal clustering results. The solution to this problem is provided by Wang *et al.* [42]. They proposed an adaptive affinity propagation method for clustering to automatically select the preference parameter to identify the correct number of clusters and finding the optimal clustering solution. However, these approaches can not cater for the presence of anomalies in training data although it is very difficult to be sure of clean training data when the trajectory dataset in unlabelled. Most of the existing unsupervised learning approaches cluster training data by defining the pairwise similarities between training samples [34][44][32][22][33][41][42]. However, calculating the pairwise affinity matrix have the computational complexity of $O(N^2)$ in both time and space where N is the number of samples in the training data.

The contribution of this paper is to present a novel mechanism for efficient and effective learning of motion patterns whilst filtering anomalous samples from training data. The proposed technique does not require any prior knowledge about the number and type of patterns hidden in datasets. A novel approach for model-based classification of trajectory patterns and anomaly detection is also presented. Clustering, classification and the detection of anomalous trajectories is carried out in the parameter space with reduced computational burden.

3 Modified DFT-based trajectory representation

This section provides a brief overview of our trajectory representation scheme based on time series representation and modified DFT (DFT-MOD). Without loss of generality, we consider the projection of a moving object O in the (x, y) image plane. O registers its location (x_i, y_i) in (x, y, t) space at each instant of time $t = t_i$. The object trajectory *T(O)* is defined by the point sequence

$$T(O) = \{(x_1, y_1, t_1), (x_2, y_2, t_2), ..., (x_n, y_n, t_n)\} \quad (1)$$

where n is the sequence length. Hence, trajectories can be treated as motion time series.

In applications to fixed-camera surveillance, it is not necessary to apply shift and scale transformations to the data before model fitting. We wish to preserve shift and scale dependence at the clustering stage. Trajectories are split into two 1-D time series in (x, t), (y, t) space. In tracking applications, observations are recorded at regular time intervals and hence we assume $t_i = i$ where i is the frame index. $T(O)$ can then be represented as two time series $X = x_i$, $Y = y_i$, $i = 1, ..., n$. We represent the trajectories using DFT-MOD based coefficient feature space representation. DFT-MOD is an extension of DFT [14]. DFT-MOD is generated by augmenting the DFT coefficients-based feature vector with some extra information regarding the length and starting location of the trajectory. These important information are not modelled correctly by DFT if we select only top few DFT coefficients which simply models the mean and trend of motion in the trajectory. All these factors may contribute to the fall-off in retrieval and classification accuracies, using simple DFT based dimensionality reduction, where starting point and duration of motion are important features for distinguishing different trajectories. Let (x_0, y_0) is the starting point and n is the length of trajectory, DFT-MOD based feature space representation of trajectory is represented as

$$\tilde{F}_{DFT-MOD} = [n, x_0, X_f, y_0, Y_f] \tag{2}$$

where X_f and Y_f are the DFT based feature space representation of x_i and y_i time series.

4 Learning of Motion Trajectories in Presence of Anomalies

In this section, we present a novel algorithm for learning patterns in the presence of anomalies in training data. The motivations of the proposed learning algorithms are to:

- develop an unsupervised learning algorithm that exploits coefficient feature sub-space and performs fast and efficient motion learning with the space and time complexity much less than $O(N^2)$ where N is the number of training samples.
- automatically identify the right number of patterns instead of requiring manual information regarding the number and types of groupings hidden in dataset.
- effectively identify the number of clusters without requiring multiple passes through the learning process for different number of cluster options.

- minimizing the adverse effects caused by the presence of anomalies in training data, on learning of normal motion patterns.

The proposed clustering mechanism is a cooperative learning algorithm that combines Learning Vector Quantization (LVQ) with Hierarchical Semi-Agglomerative ClusTering (HSACT). The architecture chosen for the LVQ consists of a single layer of input neurons connected directly to a single 1-dimensional layer of output neurons. The original LVQ structure [29] initializes the network with the number of output neurons equivalent to the number of clusters actually desired in the dataset. This type of hard clustering does not guarantee that the network will identify and distinguish all major groupings. The network may organize itself to represent variations with in one major grouping of the data by allocating more than one output neuron to that group. The situation is analogous to the problem of optimisation where it gets stuck in local minima. Similar problem is faced by k-medoid based algorithms (i.e. local minimas). One has to repeat the algorithm many times to find an acceptable solution. To increase the probability of getting the clustering right, a modified structure of LVQ to support hierarchical clustering technique is used. It is a flexible clustering technique in which LVQ is initialized with the number of output neurons that is greater than the number of groupings actually hidden in the data set. LVQ component is responsible for extracting fine groupings in trajectory dataset only once. HSACT component uses these fine clusters to generate coarse clusters and, in the process, discovering the actual number of groupings in the trajectory dataset. This is done by calculating the quality of clustering, using cluster validity index, at each iteration of HSACT algorithm. The number of clusters with the best value of validity index is taken as the number of patterns that are hidden in unclassified training data. In contrast, traditional approaches for learning and approximating the number of patterns [22][32][33] require repeating the complete clustering process for different number of clusters, resulting in significant computational overhead. The proposed clustering algorithm optimizes the criterion function that tends to minimize the within cluster variance.

Instead of having a single cycle of learning iteration, an iterative approach is taken and in each iteration, some anomalous trajectories are filtered from the training data. This is done by identifying the learned clusters with fewer cluster memberships as anomalous after each iteration. Trajectories associated to anomalous clusters are then filtered from the training data. This results in reducing the adverse affects, caused by the presence of anomalies in training data, on learning of normal motion patterns. The process continues till no trajectory is identified and filtered as anomalous. Fig. 1 depicts the iterative nature of the proposed learning mechanism to cater for the presence of anomalies in training data. High peaks in distribution curve represent training samples belonging to normal motion pattern where as low peaks represent samples which are not common and are anomalous. The learning algorithm

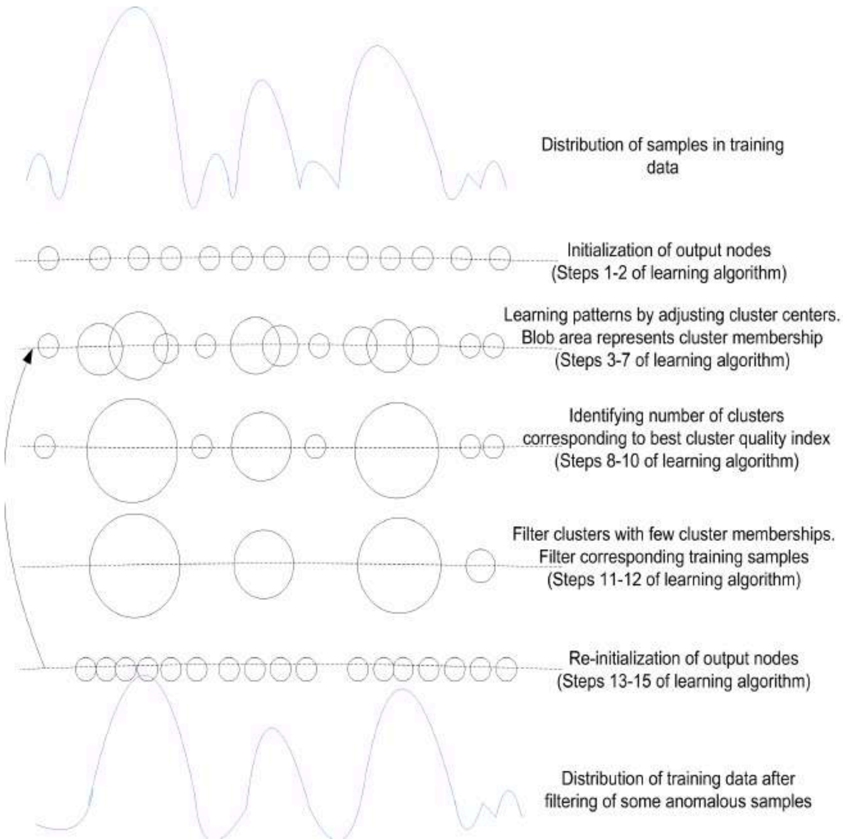


Fig. 1. Depiction of single iteration in an iterative algorithm for learning patterns in presence of anomalies

for unsupervised learning of motion patterns from corrupted training data comprises the following steps:

(1) Initialize the LVQ network with greater number of output neurons than the number of clusters we want to identify in motion trajectories using:

$$\#_{output} = \begin{cases} \xi \quad if\, \xi < 100 \\ 100 \; otherwise \end{cases} \tag{3}$$

where

$$\xi = size(DB)/4 \tag{4}$$

(2) Estimate a single multivariate Gaussian (PDF) form the training data DB as:

$$PDF = \frac{1}{\sqrt{2\pi\Sigma}} exp\left[-\frac{(X-\mu)^2}{2\Sigma}\right] \tag{5}$$

where $X \in DB$, μ is the mean and Σ is the covariance estimate associated to DB. Generate $\#_{output}$ samples from the PDF $N(\mu, \Sigma)$ and use them to initialize the weight vectors associated with each of the output neurons.

(3) Determine the winning output node k (indexed by c) such that the euclidean distance between the current input vector F and the weight vector W_k is minimum:

$$c = arg\ min\ _k \|F - W_k(t)\| \qquad \forall k \tag{6}$$

(4) Train LVQ network by adjusting the weight vector of winning output node. The weight vector of winning node c is updated using

$$W_c(t+1) = W_c(t) + \alpha(t)(F - W_c(t)) \tag{7}$$

where $\alpha(t)$ is the learning rate of LVQ and t is the training cycle index.

(5) Decrease the learning rate $\alpha(t)$ exponentially over time using:

$$\alpha(t) = 1 - e^{\frac{2(t - t_{max})}{t_{max}}} \tag{8}$$

where t_{max} is the maximum number of training iterations. t_{max} is lower bounded by the number of samples in training dataset.

(6) Repeat steps 3-5 for all the training iterations.

(7) Ignore output neurons with no training data associated to them.

(8) Calculate Cluster Validity Index (CVI) to check the quality of current state of cluster. To ignore the effect of anomalies, CVI is calculated only for those clusters with significant memberships. Criteria for clusters to be included in calculation of CVI is specified as:

$$\mathbf{\Gamma}_{valid} = \{\Gamma_i \in \mathbf{\Gamma} | |\Gamma_i| \geq \kappa\} \quad \forall i \tag{9}$$

where $\mathbf{\Gamma}$ is the set of all clusters, $|\Gamma_i|$ is the number of training samples associated to cluster Γ_i and κ is the threshold constant. For the set of valid clusters $\mathbf{\Gamma}_{valid}$, the mathematical expression of CVI is given as:

$$CVI(k) = \left(\frac{1}{k} \times \frac{E_1}{E_k} \times D_k\right) \tag{10}$$

$$E_k = \sum_{j=1}^{k} \sum_{X \in \Gamma_j \wedge \Gamma_j \in \mathbf{\Gamma}_{valid}} \|X - W_j\| \tag{11}$$

$$D_k = max_{i,j=1}^{k} \|W_i - W_j\| \tag{12}$$

where k represents number of clusters, X represents a sample training data associated to valid clusters and W_j represents weight vector associated to cluster Γ_j. In eq. (10), the factor $\frac{1}{k}$ will decrease CVI index as k is increased. On the other hand, $\frac{E_1}{E_k}$ increases CVI index as E_1 is a constant and E_k decreases with increase in k. The third factor D_k will increase with the value of k. These three factors tend to balance each other nicely. Values of k, resulting in higher values of CVI index, indicate better clustering.

(9) Identify the closest pair of cluster (i, j) (indexed by (a, b)) given by the condition

$$(a, b) = arg\,min_{(i,j)}\ [(W_i - W_j)^T (W_i - W_j)]^{\frac{1}{2}} \quad \forall i, j \wedge i \neq j \tag{13}$$

After finding the most similar pair of clusters, the two clusters are merged into one using

$$W_{ab} = \frac{mW_a + nW_b}{m + n} \tag{14}$$

where m, n are the number of sample trajectories mapped to clusters a and b respectively.

(10) Iterate through steps 8-9 till the number of clusters get equivalent to 1. Identify the number of clusters corresponding to highest CVI value.

(11) Validate the stability of clustering process. This is done by identifying and filtering the clusters with fewer cluster membership as:

$$\mathbf{\Gamma}_{anomalous} = \{\Gamma_i \in \mathbf{\Gamma} | |\Gamma_i| < \kappa\} \quad \forall i \tag{15}$$

If no cluster has been identified as anomalous, the resulting clusters are considered to be stable without having any negative effect caused by the presence of anomalies in training data. On the other hand, if some clusters have been identified as anomalous, re-initialize the LVQ network as specified in eq. (3).

(12) Identify and filter the anomalous trajectories present in the training data DB using:

$$DB_{filtered} = \{X \in DB | X \in \Gamma_i \wedge \Gamma_i \in (\mathbf{\Gamma} - \mathbf{\Gamma}_{anomalous})\} \quad \forall i \tag{16}$$

(13) Approximate Gaussian PDF of weight vectors, associated to valid clusters $\mathbf{\Gamma}_{valid}$, as:

$$PDF_{valid} = \frac{1}{\sqrt{2\pi\Sigma_{valid}}} exp\left[-\frac{(W - \mu_{valid})^2}{2\Sigma_{valid}}\right] \tag{17}$$

where W is the weight vector, μ_{valid} is the mean and Σ_{valid} is the covariance estimate associated to $\mathbf{\Gamma}_{valid}$.

(14) Re-initialize the weight vectors associated to output neurons. Let $\#_{output}$ is the number of output neurons with which the network is initialized and $\#_{valid}$ is the number of normal patterns identified in the previous learning iteration. The new network is initialized by using $\#_{valid}$ weight vectors identified in the previous learning iteration along with $(\#_{output} - \#_{valid})$ weight vectors obtained randomly from the PDF $N(\mu_{valid}, \Sigma_{valid})$ as approximated using eq. (17).

(15) Go to step 3 for learning the patterns, using the training data $DB_{filtered}$.

The space complexity of the proposed learning algorithm is $O(N)$ where N is the number of training samples. The time complexity of our algorithm is approximately $O(\omega * t_{max})$. Here, ω is the number of times required to iterate through HSACT-LVQ algorithm to filter anomalies from training data and t_{max} is the maximum number of training iterations within HSACT-LVQ algorithm. The complexity $O(\omega * t_{max})$ is much less than $O(N^2)$ for datasets with larger number of training samples.

5 Model-Based Classification and Anomaly Detection

In this section, we build on the learning of patterns outlined in previous section towards modelling of different classes representing object motion patterns. The resulting models of identified patterns can then be used to classify new unseen trajectory data as normal (i.e. belonging to one of the existing labelled classes) or anomalous (i.e. sufficiently distant from all of the known classes).

5.1 Modelling Motion Patterns

A novel mechanism is proposed for modelling various patterns that are present in motion dataset. A pattern is modelled by a set of cluster centres of mutually disjunctive sub-classes (referred to as mediods) within the pattern. The modelling mechanism is influenced by our proposed HSACT-LVQ based clustering mechanism. It has been shown in our previous work [40] that hierarchical semi-agglomerative approach using a neural network, such as HSACT-LVQ, outperforms hard clustering techniques such as k-Means. k-Means is very sensitive to the initialization of cluster centers and is normally initialized to a randomly picked sample from dataset. k-Means produce poor clustering and classification results due to poor initialization. As a result, k-Means based algorithms face the problem of local minimas. On the other hand, the proposed algorithm (HSACT-LVQ) avoids the problem of local minima by initializing itself with greater number of cluster centers then the number of groupings to be identified in the dataset. Finer clusters are then merged, based on their similarities, to generate coarse clusters representing the desired number of sub-classes (mediods). The proposed modelling technique, referred to as m-Mediods modelling, models the class containing n members with m mediods known *a-priori*. The space complexity of the proposed modelling algorithm is $O(n)$. The time complexity of our algorithm is approximately $O(t_{max})$.

The outline of proposed modelling technique is given in Fig. 2. Once the labels of trajectories from training dataset are learned, trajectories from a single class are passed as an input to the modelling algorithm. The output of the algorithm is a set of mediods used for modelling the pattern. Modelling of different

Model_Patterns($DB^{(i)}$, $\#_{mediods}$, t_{max}, $\mathbf{M}^{(i)}$){ /*
Input: Indexed training data $DB^{(i)}$ associated to pattern i.
Input: $\#_{mediods}$ is number of mediods used to model pattern i.
Input: Max number of training iterations t_{max}.
Output: Modelled pattern in the form of list of mediods ($\mathbf{M}^{(i)}$).*/

(1) Initialise SOM network with $\#_{output}$ output neurons using

$$\#_{output} = \begin{cases} \xi & if\, \xi < 100 \ \wedge \xi > (\#_{mediods} \times 2) \\ \#_{mediods} \times 2 & if \xi < (\#_{mediods} \times 2) \\ 100 & if \xi > 100 \end{cases}$$

where $\xi = size(DB^{(i)})/2$

(2) Initialise weight vectors W_i (where $1 \leq i \leq \#_{output}$) from the PDF $N(\mu, \Sigma)$ estimated from training samples in $DB^{(i)}$.
(3) Initialise t to 0.
(4) while ($t < t_{max}$){
(5) for (each feature vector F in $DB^{(i)}$){
(6) Calculate Euclidean distance between F and weight vector W_k for each output neuron.
(7) Select output neuron c with minimum distance value.
(8) Update weight vectors for output neuron c using

$$W_c(t+1) = W_c(t) + \alpha(t)(F - W_c(t))$$

(9) Decrease learning rate

$$\alpha(t) = 1 - e^{t - t_{max}}$$

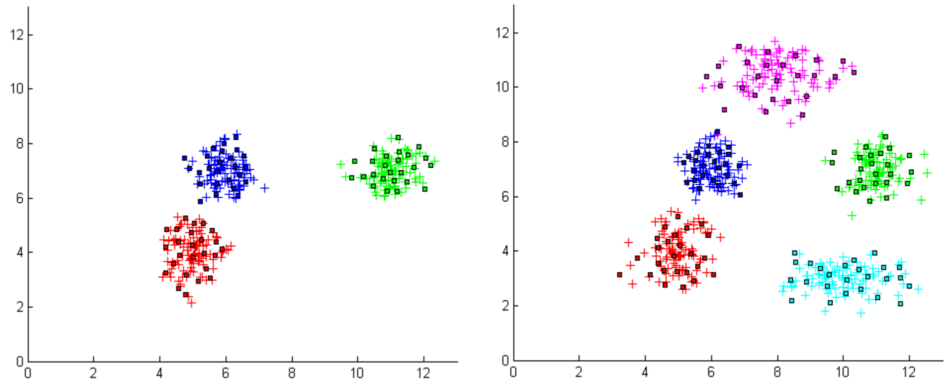
(10) if $t++ == t_{max}$
(11) Break the loop.
}// end for loop
}// end while loop
(12) Ignore clusters with zero cluster membership.
(13) while($\#_{output} > \#_{mediods}$){
(14) Select most similar output neurons.
(15) Merge similar output neurons using weighted means mechanism

$$W_{ab} = \frac{mW_a + nW_b}{m+n}$$

(16) $\#_{output} = \#_{output} - 1$
} // end of while
(17) Append weight vector W_k to $\mathbf{M}^{(i)}$.
}// end for algorithm

Fig. 2. Proposed algorithm for modelling pattern using m-Mediods

patterns, using proposed approach, is demonstrated in Fig. 3. In order to visualise the modelling process, modelling of patterns is done using simulated SIM_3 and SIM_5 datasets. In Fig. 3, each point represents an instance from the dataset. Instances belonging to the same class are represented with same colour. Squares super-imposed on each group of samples represent the mediods obtained using m-Mediods modelling algorithm to model the patterns.



(a) (b)

Fig. 3. Mediods-based modelling of patterns in (a) SIM_3 dataset (b) SIM_5 dataset

After modelling the pattern c, the distance array $\mathbf{D}^{(c)}$ corresponding to model $\mathbf{M}^{(c)}$ is pre-computed, to be used later for anomaly detection, as follows:

(1) Identify the closest pair of mediods (i, j) (indexed by (p, q)) from $\mathbf{M}^{(c)}$ as follows:

$$(p, q) = arg\ min_{(i,j)} Dist(M_i, M_j) \qquad \forall i, j \wedge i \neq j \tag{18}$$

where $Dist(M_i, M_j)$ is the euclidean distance function.

(2) Populate the distance array for the current number of mediods using

$$\mathbf{D}_l^{(c)} = (p, q, Dist(M_p, M_q)) \tag{19}$$

where l is the current number of mediods.

(3) Merge the most similar pair of mediods using

$$M_{pq} = \frac{mM_p + nM_q}{m + n} \tag{20}$$

where m, n are the number of sample trajectories mapped to mediods p and q respectively.

(4) Iterate through steps 1-3 till the number of mediods gets equivalent to 1.

5.2 Trajectory Classification and Anomaly Detection

Once the m-Mediods based model for all the classes have been learnt, the classification of new trajectories is performed by checking the closeness of said trajectory to the models of different classes. For this purpose, the trajectory is posed as a query to the entire set of mediods ($\mathbf{M}$) belonging to different classes. Identification of k Nearest Mediods (k-NM) to unseen trajectory can be specified as

$$k\text{-NM }(Q, \mathbf{M}, k) {=} \{R \in \mathbf{M} | \forall R \in C, S \in \mathbf{M} - C, \\ Dist(\mathrm{Q},R) \leq Dist(Q, S) \wedge |R| = k\} \tag{21}$$

where Q is DFT-MOD based feature vector representation of unseen trajectory to be classified and R is the set of k closest mediods. A previously unseen trajectory Q is assigned to the same class, indexed by c, to which the majority of k nearest mediods belong.

After identifying the closest activity pattern (c), it is checked to see if the unseen data is reasonably close to the closest activity pattern or not. A novel mechanism based on agglomerative approach has been proposed for anomaly detection. The description of anomaly detection algorithm is specified as follows:

(1) Initialize index l with the number of mediods (m) used to model a pattern.
(2) Identify the closest pair of mediods and their corresponding distance, for the current number of mediods l, using $\mathbf{D}^{(c)}$ as:

$$(p, q, d_{pq}) = \mathbf{D}_l^{(c)} \tag{22}$$

where d_{pq} contains the distance between mediods indexed by p and q.
(3) Identify the mediod, from $\mathbf{M}^{(c)}$, which is closest to the test sample Q. The closest mediod, indexed by r, is identified using:

$$r = arg\ min_k\ Dist(Q, M_k) \qquad \forall k \tag{23}$$

(4) Test trajectory Q is considered to be a valid member of class c if:

$$Dist(Q, M_r) \leq d_{pq} \tag{24}$$

(5) If the condition specified in eq. (24) is not satisfied, decrement the index l by 1.
(6) Merge the pair of mediods, indexed by (p, q), using

$$M_{pq} = \frac{mM_p + nM_q}{m + n} \tag{25}$$

where m, n are the number of sample trajectories mapped to mediods p and q respectively.
(7) Iterate steps 2-6 till l gets equivalent to the significance parameter τ. If the test trajectory Q has yet not been identified as a valid member of class c, it is considered to be an outlier and deemed anomalous.

The significance parameter τ determines the sensitivity of proposed anomaly detection algorithm to anomalies. Lower value of τ results in acceptance of more unusual data instances as normal members of one of the known classes and *vice versa*. Values of significance parameter τ lies in the range $1 \leq \tau < m$.

6 Experimental Results

We now present some results to demonstrate the effectiveness of the proposed clustering, classification and anomaly detection techniques in the coefficient feature space.

6.1 Experimental Datasets

Experiments are conducted on five different synthetic and real life motion trajectory datasets. These include CAV-CORR, LAB, ASL, SIM_3 and SIM_5 datasets. The characteristics of these datasets are summarized in Table 1.

6.2 Experiment 1: Learning Motion Patterns in Presence of Anomalies

The purpose of this experiment is to demonstrate the effectiveness of Iterative HSACT-LVQ algorithm for learning of patterns while catering for the presence of anomalies in unclassified training data. The experiment has been conducted on real life CAV-CORR dataset that contains anomalous trajectories within the dataset itself. CAV-CORR dataset is also suffering from the problem of perspective effects due to the presence of depth in CAV-CORR scene. To avoid this problem, we use ground plane homography to map image coordinates to ground plane for CAV-CORR dataset. The calibration is available from [30]. The homography matrix is estimated using the method outlined in [31]. Trajectories are modelled using DFT-MOD based coefficient feature vectors. The network is trained for $t_{max} = 5000$ number of iterations. The clusters with fewer cluster memberships are identified as anomalous and are filtered. We assume $\kappa = 0.05 \times |DB|$ in eq. (9) where $|DB|$ is the total number of samples in training data. If some clusters are identified as anomalous, training samples associated to such clusters are removed from the training dataset. The learning process is repeated again, but now using the filtered training data. This process continues till clustering process gets stable and no cluster is identified as anomalous.

The clustering results obtained, by applying the Iterative HSACT-LVQ methodology on CAV-CORR dataset, are shown in Fig. 4. The red trajectory in each

Dataset	Description	# of trajectories	Extraction method	Labelled (Y/N)
SIM_3/ SIM_5	Simulated datasets comprising of two dimensional coordinates generated from Gaussian distributions to form 3 or 5 clusters.	arbitrary	Simulation.	Y
LAB	Realistic dataset generated in the laboratory controlled environment for testing purposes. Trajectories can be categorised into 4 classes.	152	Tracking moving object and storing motion coordinates.	Y
CAV-CORR	A manually annotated video sequences of moving people from corridor view in a shopping centre. Object tracking coordinates are generated using interactive program and stored in XML files.	126	Parsing XML files containing motion coordinates.	N
ASL	Trajectories of right hand of signers as different words are signed. Dataset consists of signs for 95 different word classes with 70 samples per word.	6650	Extracting (x, y) coordinates of the mass of right hand from files containing complete sign information.	Y

Table 1
Overview of datasets used for experimental evaluation

class represents the trajectory that is closest to the class mean. Qualitatively, similar motion trajectory patterns appear to have been grouped together quite successfully. Trajectories that are filtered out as anomalous during learning process are shown in Fig. 5. Each anomalous trajectory is represented by separate colour. It is clear from Fig. 5 that anomalous trajectories are dissimilar from normal motion patterns as shown in Fig. 4.

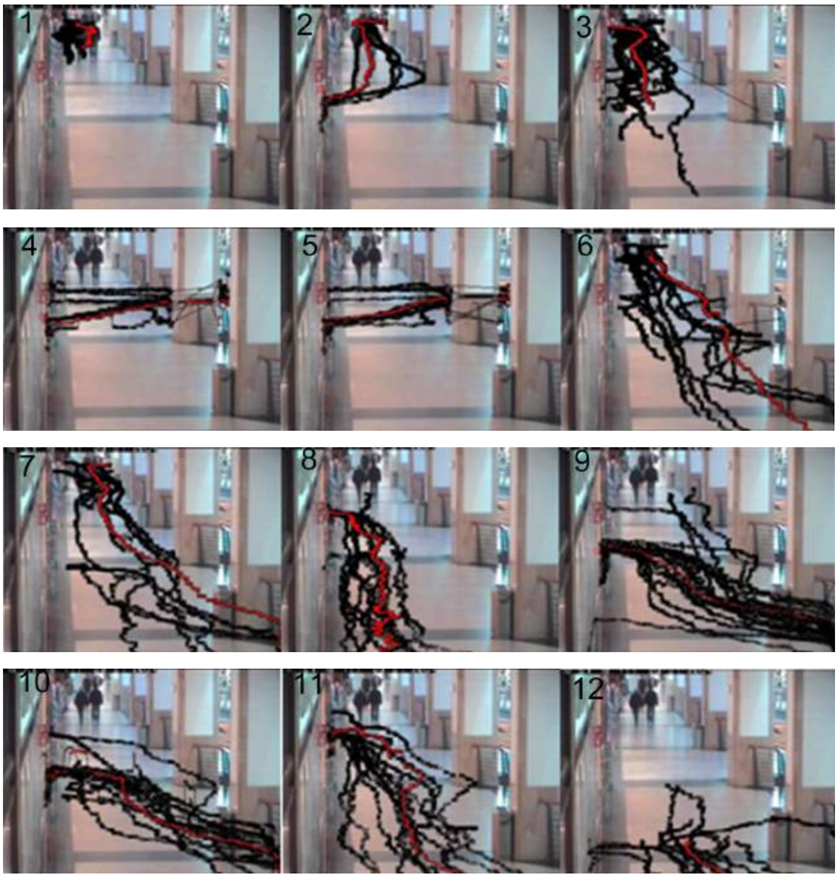


Fig. 4. Motion trajectory clustering of CAV-CORR dataset

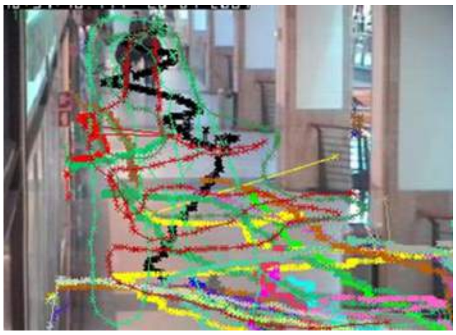


Fig. 5. Trajectories, identified as anomalous, during learning of patterns by applying Iterative HSACT-LVQ algorithm on CAV-CORR dataset

6.3 Experiment 2: Comparison of Iterative HSACT-LVQ with Competitive Techniques

The purpose of this experiment is to compare the performance of proposed Iterative HSACT-LVQ algorithm with the adaptation of spectral clustering [32][22][33] and adaptive affinity propagation (Adaptive AP) [42]. The matlab code for implementation of Adaptive AP is obtained from [43]. Comparative evaluation is provided in terms of detection of correct number of clusters, quality of clustering, robustness to the presence of different number of anomalous samples in training data and response time.

Datasets	**HSACT-LVQ**			**Spectral**			**Adaptive AP**		
	# of clusters	CH	Dunn	# of clusters	CH	Dunn	# of clusters	CH	Dunn
SIM_3	3	1667.3	3.17	2	1173.7	2.98	2	1278.1	3.19
SIM_5	5	1891.3	2.78	4/5	1463.4	1.85	4	1254.6	2.74

Table 2
Comparison of Iterative HSACT-LVQ, Adaptive AP and spectral clustering based on the number and quality of clusters using clean training data from simulated datasets

Datasets	**HSACT-LVQ**			**Spectral**			**Adaptive AP**		
	# of clusters	CH	Dunn	# of clusters	CH	Dunn	# of clusters	CH	Dunn
SIM_3	3	1516.7	2.91	6/7	823.9	1.16	3	1092.9	2.11
SIM_5	5	1974.5	2.97	7	1105.6	1.27	8	1165.7	1.89

Table 3
Comparison of Iterative HSACT-LVQ, Adaptive AP and spectral clustering based on the number and quality of clusters using corrupted training data (with anomalies) from simulated datasets

The experiment has been conducted on simulated SIM_3 and SIM_5 datasets. Learning of patterns is conducted separately using Iterative HSACT-LVQ, Adaptive AP and spectral clustering, and two cluster validity indices are employed to evaluate the quality of clustering results. These include *Dunn*'s index [35] and Calinski-Harabasz (*CH*) index [36].

The experiment is performed on clean (without anomalies) as well as corrupted (with anomalies) training data to investigate the effect of presence of anomalies on the performance of learning algorithms. The number of clusters identified by different clustering algorithms along with the quality of clustering using clean training data, as indicated by three different cluster validity indices, are presented in Table 2. Similar results using corrupted training data are presented in Table 3. Higher values for *Dunn* and *CH* index indicate better clustering.

As evidenced from Table 2 and 3, Iterative HSACT-LVQ algorithm performs consistently better than Adaptive AP and spectral clustering for all the datasets in the presence of clean and corrupted training data. Comparing results from

Table 2 and 3 show good consistency in the performance of Iterative HSACT-LVQ in the presence of clean and corrupted training data. However, the performance of Adaptive AP and spectral clustering degrades significantly in the presence of anomalies. Moreover, the number of clusters identified using our proposed approach is consistent with the number of groupings hidden in the dataset. On the other hand, Adaptive AP and spectral clustering is not able to identify the correct number of clusters. This is verified by matching the identified number of clusters with the actual number of groupings hidden in classified SIM_3 and SIM_5 datasets.

Effectiveness of Iterative HSACT-LVQ algorithm, as compared to competitive clustering algorithms, is now demonstrated graphically for SIM_5 dataset. The training data from SIM_5 dataset is shown in Fig. 6. Gaussian parameters used to generate each of the clusters in Fig. 6 is presented in Table 4. The training data is obtained by generating 70 samples from each of the Gaussian distribution. The anomalies are induced in SIM_5 dataset by generating different number of data points from a uniform distribution such that $(x, y) \in (U(1, 12), U(1, 12))$. The data points that lie within 3 standard deviations of normal clusters are then removed from the set of anomalous data points. Results of learning patterns using Iterative HSACT-LVQ, Adaptive

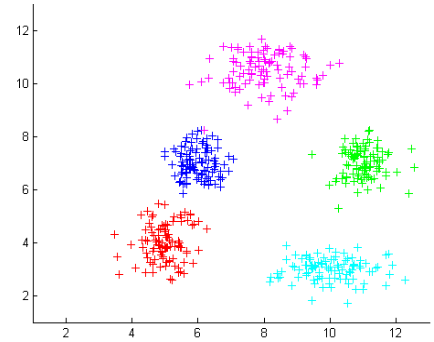


Fig. 6. SIM_5 dataset sampled from five Gaussians

Cluster Colour	Blue	Green	Red	Magenta	Cyan
Mean	(6,7)	(11,7)	(5,5)	(8,10)	(10,3)
Covariance	$\begin{pmatrix} 0.2 & 0 \\ 0 & 0.3 \end{pmatrix}$	$\begin{pmatrix} 0.2 & 0 \\ 0 & 0.3 \end{pmatrix}$	$\begin{pmatrix} 0.3 & 0 \\ 0 & 0.4 \end{pmatrix}$	$\begin{pmatrix} 0.7 & 0 \\ 0 & 0.4 \end{pmatrix}$	$\begin{pmatrix} 0.6 & 0 \\ 0 & 0.2 \end{pmatrix}$

Table 4
Gaussian parameters used to generate 5 clusters

AP and spectral clustering is demonstrated graphically in Fig. 7. Fig. 7(a) presents learning result for SIM_5 dataset in the presence of 60 anomalies in training data and Fig. 7(b) presents learning results in the presence of 120 anomalies. Data points belonging to same class are represented with similar colour and marker to ease the visualisation of learned clusters. Comparing

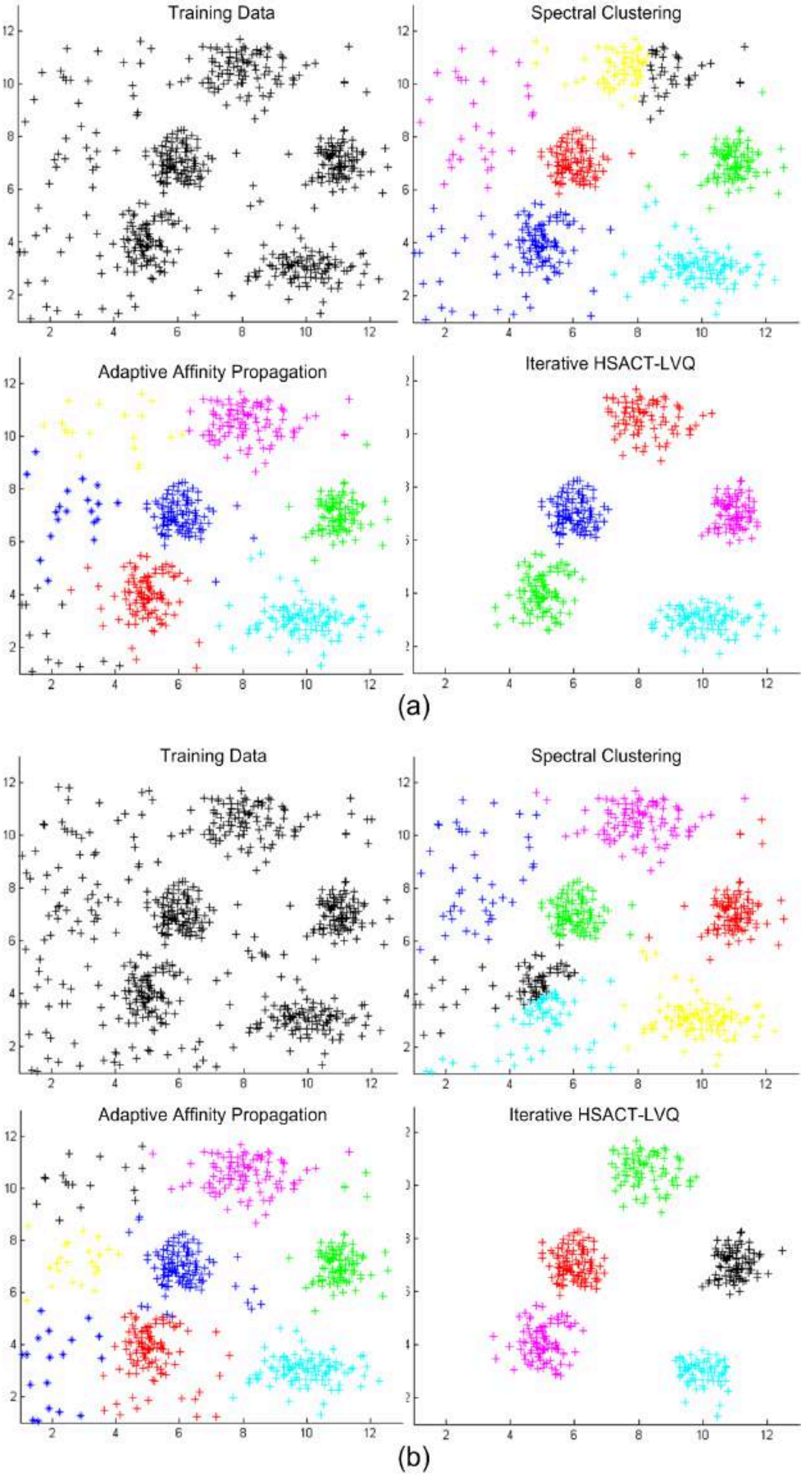


Fig. 7. Learning of patterns from SIM_5 dataset using Iterative HSACT-LVQ, Adaptive AP and spectral clustering in the presence of different number of anomalies ($\#anomalies$) in training data (a) $\#anomalies = 60$ (b) $\#anomalies = 90$

clustering results with the ground truth for SIM_5 dataset shows that Iterative HSACT-LVQ identifies the right number of clusters even in the presence of significant number of anomalies. The overall distribution of normal clusters learned using proposed algorithm, remains unaffected by the presence of anomalies in training data. On the other hand, quality of Adaptive AP and spectral clustering is significantly affected by the presence of anomalies. The proposed technique is also robust to the presence of higher number of anomalous samples in training data.

Comparison of different clustering algorithms is now provided by investigating the scalability of these algorithms to the number of options from which to identify the correct number of clusters present in the dataset. We have implemented these algorithms using MATLAB 7 and running times are noted on an Intel Pentium IV 1.73 GHz machine with 504 MB of RAM. Experiment has been conducted on SIM_5 dataset and the response time of clustering algorithms, for different number of candidate clusters, are presented in Table 5. It is evident from the results in Table 5 that our proposed approach is scalable to number of options from which to select the right number of patterns hidden in the dataset. This is one of the important advantages of incorporating HSACT component with LVQ based learning that the proposed clustering algorithm takes same amount of time for any number of cluster options. On the other hand, spectral clustering requires repeating the complete clustering algorithm for each potential number of clusters. As a result, spectral clustering exhibits increasing time complexity with increasing number of cluster options from which to identify the number of groupings hidden in training dataset. Adaptive affinity propagation exhibits consistent time complexity for different number of cluster options but its response time is significantly higher as compared to Iterative HSACT-LVQ. In Table 6, comparison of different clustering algorithm is now presented based on the response time for different number of samples in training dataset. It is evident from the results in Table 6 that our proposed approach is scalable to the increasing number of samples in training dataset. On the other hand, Adaptive AP and spectral clustering exhibits increasing time complexity with increasing number of training samples. Adaptive AP suffers from its quadratic complexity in function of the number of training samples.

6.4 Experiment 3: Evaluation of Proposed Model-based Classification and Anomaly Detection

The purpose of this experiment is to evaluate the performance of proposed model-based approach for classification of unseen data samples to one of the known patterns. The experiment demonstrates the ability of proposed classification system to act as an anomaly detection system. The experiment has been

	Response Time (sec.)		
# of cluster options	**Iterative HSACT-LVQ**	**Spectral**	**Adaptive AP**
2	11.74	6.91	273.45
4	11.62	11.93	273.45
6	11.61	15.70	275.12
8	11.32	21.31	275.50
10	11.31	28.85	271.37
12	11.46	36.51	273.72
14	11.10	45.98	274.27
16	11.57	52.43	269.97
18	11.27	59.64	274.91
20	11.67	74.36	275.15

Table 5
Comparison of clustering algorithms based on the response time for different number of cluster options

	Response Time (sec.)		
# of training samples	**Iterative HSACT-LVQ**	**Spectral**	**Adaptive AP**
300	9.54	63.91	117.81
500	9.61	130.37	469.92
900	9.71	217.23	1706.12

Table 6
Comparison of different clustering algorithm based on the response time for different number of samples in training dataset.

conducted on SIM_5 and LAB datasets. For simulated SIM_5 datasets, classified training data is obtained by generating 50 samples from each class using the distribution as specified in Table 4. Test data is obtained by generating 500 samples from a uniform distribution such that $(x, y) \in (U(1, 12), U(1, 12))$. On the other hand, LAB dataset is a classified motion dataset and contain anomalous trajectories within the dataset itself. Classified training data for this dataset is obtained by randomly selecting half of the trajectories from each of the normal patterns in the dataset. The remaining half of the trajectories from normal patterns along with anomalous trajectories are extracted and used as a test data.

Feature vector representation of simulated SIM_5 and trajectory-based LAB dataset is obtained as specified in previous experiments. Members of each class from the training data are used to generate model of normality associated to each normal pattern, using the algorithm as presented in Fig. 2. Classes are modelled using 30 mediods per class. Once the m-Mediods (with $m = 30$) based model for all the classes have been learnt, classification of samples from the test data is done using the classifier as proposed in section 5.2. We have used different values of significance parameter τ for SIM_5 datasets. For LAB dataset, classification and anomaly detection is carried out by setting the value of $\tau = 10$. The classification and anomaly detection results for SIM_5 dataset, using different values of significance parameter τ, are presented in Fig. 8. Training data is represented using '+' marker whereas classified normal samples are represented by small circles. For ease of visualisation, data points belonging to same class are represented with same colour. Samples from test data that are identified as anomalous are represented using black 'x' marker. It is apparent from Fig. 8 that proposed classification system correctly classifies test samples to known classes whilst identifying anomalies in the test data. Another important observation from Fig. 8 is that setting higher values of τ results in acceptance of only those instances as normal that are tightly bounded to normal classes. As τ decreases, we are less likely to detect anomalous patterns because it results in acceptance of more unusual data instances as normal members of one of the known classes. Using different significance levels therefore enables our proposed system to be adaptive to the density of data within a class.

After demonstrating the efficacy of proposed classification and anomaly detection approach on synthetic data, the experiment is then repeated on real life LAB dataset. Classification obtained by applying the proposed approach on LAB dataset is shown in Fig. 9. The matching of classification obtained for each trajectory with its ground truth shows that no trajectory is misclassified. Trajectories identified as anomalous using the value of $\tau = 10$ are shown in Fig. 10. It is clear from Fig. 10 that anomalous trajectories are significantly different from the normal motion patterns as shown in Fig. 9. These experimental results give evidence to the claim that the proposed model-based classification and anomaly detection system is an effective and robust approach that works well with real life motion datasets.

6.5 Experiment 4: Quantitative Evaluation of Proposed Model-based Anomaly Detection

The purpose of this experiment is to provide a quantitative evaluation of proposed model-based approach for anomaly detection. The experiment has been conducted on real life ASL dataset. We have selected signs from 6 different words including 'alive', 'all', 'crazy', 'drink', 'god' and 'go'. Classified training

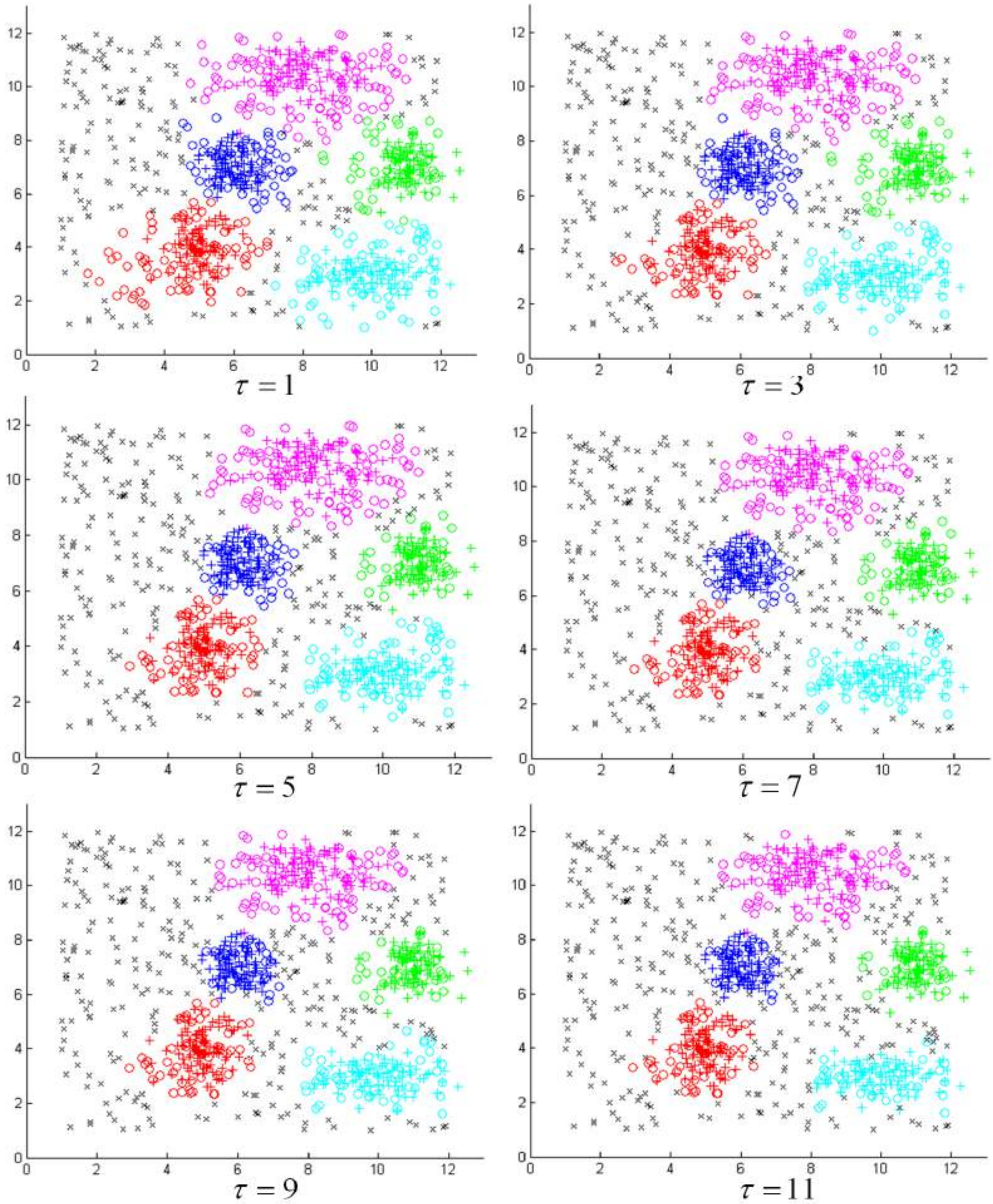


Fig. 8. Classification of test data, based on SIM_5 classes, using different values of significance parameter τ

data for ASL dataset is obtained by randomly selecting half of the trajectories belonging to each of the 6 selected words. The remaining half of the trajectories are then selected and used as test data. Feature vector representation of ASL dataset is obtained as specified in earlier experiments. Patterns are modelled using 20 mediods per pattern. We have used the value of significance parameter $\tau = 8$ for anomaly detection. Test dataset is then passed through the anomaly detection system. We would expect that few instances drawn from class X would be recorded as anomalous when tested against the same class, whereas nearly all instances would be detected as anomalous when

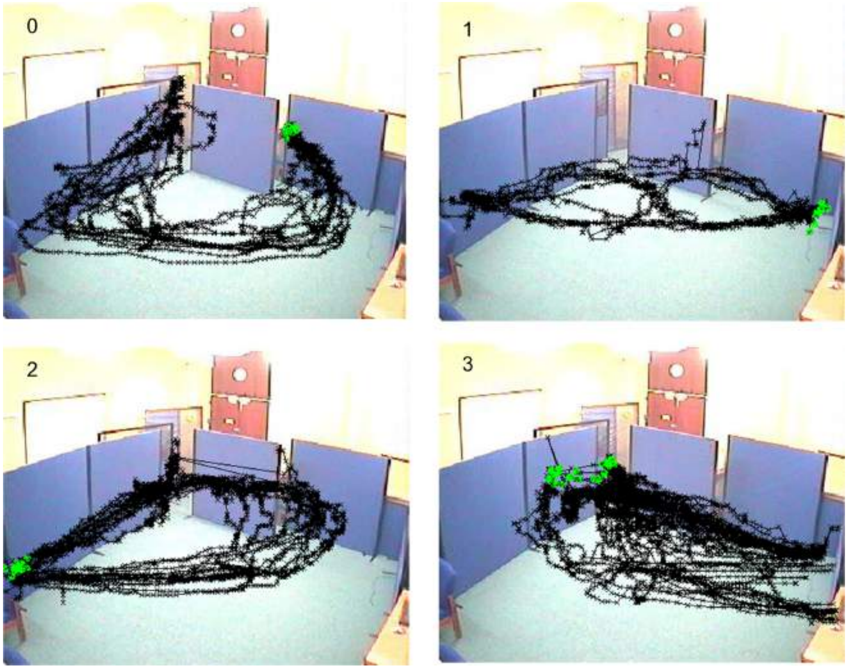


Fig. 9. Classification of test trajectories from LAB dataset

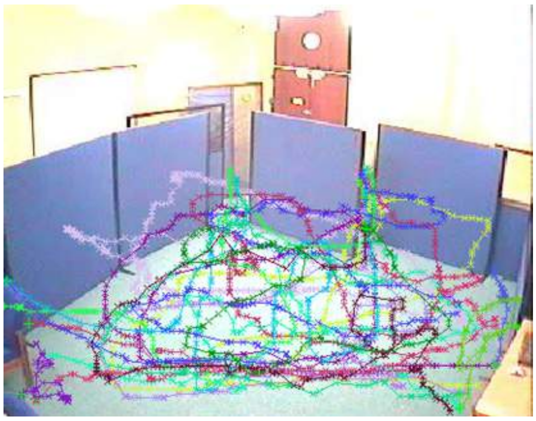


Fig. 10. Trajectories identified as anomalous from LAB dataset using proposed anomaly detection mechanism with $\tau = 10$

tested against a different class Y.

The percentages of instance vectors from ASL dataset, detected as anomalous, are shown in Table 7. Test class (X) is represented by top row whereas modelled class (Y) is represented by left column in Table 7. If the classes were completely separable, the diagonal table entries would be zero and the off-diagonal entries would be 100. The accuracies of proposed anomaly detection system on different ASL signs show that almost all signs from a particular class appears to be anomalous when its membership is tested against the m-Mediods model of class representing another sign. Very few samples are identified as anomalous when they are tested again the model of the same

Modelled/Test Class	alive	all	crazy	drink	go	god
alive	0	100	100	100	100	100
all	100	7.4	100	100	100	100
crazy	100	100	3.7	96.3	100	100
drink	100	100	100	0	100	100
go	100	100	100	96.3	0	100
god	96.3	100	100	100	100	7.4

Table 7
Percentage of instance vectors detected as anomalous for ASL dataset

class as evident from Table 7.

6.6 Experiment 5: Comparison of Proposed Classifier with Competitive Techniques

The purpose of this experiment is to compare the performance of proposed m-Mediods model-based approach for classification with competitive techniques. To establish a base case, we have implemented two different systems for comparison including GMM and Mahalanobis classifier. The experiment has been conducted on real life ASL dataset. Signs from different number of word classes are selected. m-Mediods based models from the classified training data for ASL dataset is obtained as specified in Experiment 4. Modelling of patterns for Mahalanobis classifier is done by estimating a single multivariate Gaussian PDF for each class. Modelling of patterns and classification of unseen samples using GMM is based on the approach as described in [37]. Once the models for all the classes have been learnt, the test data is passed to different classifiers and the class labels obtained are compared with the ground truth. The experiment is repeated with different numbers and combinations of word classes. Each classification experiment is averaged over 50 runs to reduce any bias resulting from favourable word selection.

The classification accuracies obtained for different classifiers using various numbers of word classes from ASL dataset are shown in Table 8. Based on these results, we see that the proposed m-Mediods model-based classification yields superior classification accuracies. GMM produces good performance for lower number of classes. Increasing the number of classes results in degrading the performance of GMM-based classifier. The proposed classification approach gives better classification accuracies than Mahalanobis classifier as well. From Table 8, it can also be noted that the relative accuracy of the proposed classifier compared with GMM and Mahalanobis classifier increases with an increase

	ASL (#classes : #samples)				
	2 : 70	**4 : 140**	**8 : 280**	**16 : 560**	**24:840**
***m*-Mediods**	0.98	0.92	0.88	0.83	0.78
Mahalanobis	0.95	0.88	0.82	0.75	0.71
GMM	0.97	0.92	0.83	0.74	0.69

Table 8
Percentage classification accuracies for different number of classes from ASL dataset

in the number of classes; thus making it more scalable for larger number of classes.

Similar experiment with ASL dataset has been conducted by Bashir *et al.* [22] using their proposed GMM and HMM-based classification system. They reported classification accuracies of 0.96, 0.92, 0.86 and 0.78 for 2, 4, 8 and 16 word classes respectively. Comparing these classification accuracies with the results obtained using our approach, we see that m-Mediods model-based classifier performs better than GMM and HMM-based recognition system [22] even though our proposed classification approach is conceptually simpler and computationally less expensive.

7 Discussion and conclusions

In this paper, we have presented a detailed discussion on unsupervised learning of patterns in the presence of anomalies in training data. A novel Iterative HSACT-LVQ algorithm has been proposed for learning of motion patterns whilst filtering anomalous samples. The paper also addresses the problem of modelling normal motion patterns. A novel approach, referred to as m-Mediods modelling, is proposed that models the class containing n members with m-Mediods known *a-priori*. Once the m-Mediods model for all the classes have been learnt, the classification of new trajectories and anomaly detection can be performed by checking the closeness of said trajectory to the models of different classes using hierarchical classifier.

Experimental results are presented to show that Iterative HSACT-LVQ based learning mechanism gives better clustering results than competitive techniques such as adaptive affinity propagation and spectral clustering. Learning of patterns, using proposed approach, is unaffected by anomalies in training data until the presence of very high number of anomalies results in development of groupings within anomalous data, thus resulting in identification of ghost clusters. The approach is also scalable to number of options from which to select the right number of hidden patterns and the size of training dataset.

Experiments are also conducted to show the effectiveness of proposed m-Mediods based classification and anomaly detection system. Matching the classification results with labelled training data shows that the test instances are classified correctly and filtered instances (anomalies) are sufficiently distant from all the known classes. Comparison of proposed classifier with competitive techniques demonstrates the superiority of our proposed approach as it performs consistently better than commonly used Mahalanobis, GMM and HMM-based classifiers.

References

[1] Z. Aghbari, K. Kaneko, A. Makinouchi, Content-trajectory approach for searching video databases, IEEE Transanction on Multimedia, vol. 5, no. 4, December 2003, pp. 516-531.

[2] S.F. Chang, W. Chen, J.M. Horace, H. Sundaram, D. Zhong, A Fully Automated Content based Video Search Engine Supporting Spatiotemporal Queries, IEEE Transactions on Circuits and System for Video Technology, vol. 8, no. 5, September 1998, pp. 602-615.

[3] S. Dagtas, W. Ali-Khatib, A. Ghafor, R.L. Kashyap, Models for motion-based video indexing and retrieval, IEEE Transactions on Image Processing, vol. 9, no. 1, 2000, 88-101.

[4] C.T. Hsu, S.J. Teng, Motion trajectory based video indexing and retrieval, IEEE International Conference on Image Processing, vol. 1, 2002, pp. 605-608.

[5] Y. Jin, F. Mokhtarian, Efficient video retrieval by motion trajectory, Proceedings of British Machine Vision Conference, Kingston, September 2004, pp. 667-676.

[6] S. Khalid, A. Naftel, Evaluation of matching metrics for trajectory based indexing and retrieval of video clips, Proceedings of IEEE WACV, Colorado, USA, January 2005, pp. 242-249.

[7] C. Shim, J. Chang, Trajectory based video retrieval for multimedia information systems, Proceedings of ADVIS, 2004, pp. 372-382.

[8] N. Johnson, D. Hogg, Learning the distribution of object trajectories for event recognition, Proceedings of British Conference on Machine Vision, 1995, pp. 582-592.

[9] W. Hu, X. Xiao, D. Xie, T. Tan, S. Maybank, Traffic accident prediction using 3-D model based vehicle tracking, IEEE Transactions on Vehicular Tech, vol. 53, no. 3, May 2004, pp. 677-694.

[10] J. Owens, A. Hunter, Application of the Self-Organising Map for Trajectory Classification, Proceedings of Third IEEE International Workshop on Visual Surveillance, Dublin, Ireland, July 2000, pp. 77.

[11] F.I. Bashir, A.A. Khokhar, D.Schonfeld, View-invariant motion trajectory based activity classification and recognition, ACM Multimedia Systems, special issue on Machine Learning Approaches to Multimedia Information retrieval, 2006, pp. 45-54.

[12] F.I. Bashir, A.A. Khokhar, D.Schonfeld, HMM based Motion Recognition System using Segmented PCA, IEEE International Conference on Image Processing, Genova, Italy, Sept. 11-14, 2005, pp. 1288-1291.

[13] W. Hu, D. Xie, T. Tan, S. Maybank, Learning activity patterns using fuzzy self-organizing neural networks, IEEE Transactions on Systems, Man & Cybernetic, vol. 34, no. 3, June 2004, pp. 1618-1626.

[14] C. Faloutsos, M. Ranganathan, Y. Manolopoulos, Fast Sub-sequence Matching in Time-Series Databases, Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data, 1994, pp. 419-429.

[15] K. Chan, A. Fu, Efficient time series matching by wavelets, Proc. of International Conference on Data Engineering, Sydney, March 1999, pp. 126-133.

[16] Y. Cai, R. Ng, Indexing Spatio-Temporal Trajectories with Chebyshev Polynomials, ACM SIGMOD/PODS Conference, France, June 13-18, 2004, pp. 599-610.

[17] C. Shim, J. Chang, Content based retrieval using trajectories of moving objects in video databases, Proceedings of IEEE 7th International Conference on Database Systems for Advanced Applications, 2001, pp. 169-170.

[18] R. Agarwal, C. Faloutsos, A. Swami, Efficient Similarity Search in Sequence Databases, 4th International Conference of Foundations of Data Organization and Algorithms, Evanston, Illinois, USA, October 1993, pp. 69-84.

[19] J. Alon, S. Sclaroff, G. Kollios, V. Pavlovic, Discovering clusters in motion time-series data, Proc. IEEE CVPR, vol. 1, June 2003, pp. I-375- I-381.

[20] D. Buzan, S. Sclaroff, G. Kollios, Extraction and Clustering of Motion Trajectories in Video, International Conference on Pattern Recognition, Cambridge, UK, 2004, pp. 521-524.

[21] M. Vlachos, G. Kollios, D. Gunopulos, Discovering Similar Multidimensional Trajectories, Proceedings of the International Conference on Data Engineering, San Jose, CA, 2002, pp. 673-684.

[22] F.I. Bashir, A.A. Khokhar, D.Schonfeld, Object Trajectory-Based Activity Classification and Recognition Using Hidden Markov Models, IEEE Transactions on Image Processing, vol. 16, no. 7, 2007, 1912-1919.

[23] D. Zhang, Gatica-Perez, S. Bengio, I. McCowan, Semi-supervised adapted hmms for unusual event detection, Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2005, 611-618.

[24] C. Abraham, P-A. Cornillon, E. Matzner-Lober, N. Molinari, Unsupervised curve clustering using b-splines, Scandinavian Journal of Statistics, vol. 30, no. 3, September 2003, pp. 581-595.

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

[26] J. Owens, A. Hunter, Novelty Detection in Video Surveillance Using Hierarchical Neural Networks, Proceedings of ICANN, Madrid, Spain, August 28-30, 2002, pp. 1249-1254.

[27] C. Stauffer, E. Grimson, Learning Patterns of Activity Using Real-Time Tracking, IEEE Transactions on Pattern Recognition and Machine Intelligence, vol. 22, no. 8, 2000, pp. 747-757.

[28] N. Sumpter, A.J. Bulpitt, Learning spatio-temporal patterns for predicting object behaviour, Image and Vision Computing, vol. 18, 2000, 697-704.

[29] T. Kohonen, Learning Vector Quantization, Neural Network,1988.

[30] CAVIAR test sequences [Online]. Available: http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/

[31] A. Zisserman, R. Harley, Multiple View Geometry in Computer Vision, Cambridge University Press, 2003.

[32] F. Porikli, T. Haga, Event Detection by Eigenvector Decomposition using Object and Frame Features, International Conference on Computer Vision and Pattern Recognition, 2004.

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

[34] D.L. Davie, D.W. Bouldin, A Cluster Separation Index, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 1, 1979, pp. 224-227.

[35] J.C. Dunn, A Fuzzy Relative of ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters, Journal of Cybernetics, vol. 3, 1973, pp. 32-57.

[36] R.B. Calinski, J. Harabasz, A Dendrite Method for Cluster Analysis, Comm. in Statistics, vol. 3, 1974, pp. 1-27.

[37] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, Oxford, New York, 1995.

[38] Y. Yacoob, M.J. Black, Parameterized Modeling and Recognition of Activities, Computer Vision and Image Understanding, vol. 73 (2), Feb. 1999, pp. 232-247.

[39] N. Rea, R. Dahyot, A. Kokaram, Semantic Event Detection in Sports through motion understanding, Proceedings of Conference on Image and Video Retrieval, Dublin, Ireland, July 21-23, 2004.

[40] S. Khalid, A. Naftel, Classifying Spatiotemporal Object Trajectories using Unsupervised Learning in the Coefficient Feature Space, Multimedia Systems, Vol. 12(3), Dec. 2006, pp. 227-238.

[41] B.J. Frey, D. Dueck, Clustering by Passing Messages between Data Points, Science, 2007, 315(5814), 972-976.

[42] K. Wang, J. Zhang, D. Li, X. Zhang, T. Guo, Adaptive Affinity Propagation Clustering, Acta Automatica Sinica, 33(12), 2007, 1242-1246.

[43] http://www.mathworks.com/matlabcentral/fileexchange/loadAuthor.do?objectType=authorobjectId=1095267

[44] Y.N. Andrew , I.J. Micahel, Y. Weiss, On spectral clustering: Analysis and an algorithm, In Advances in Neural Information and Processing Systems, volume 14, 2001.