Behavior Clustering for Anomaly Detection

Xudong Zhu, Hui Li, and Zhijing Liu

Xidian University, 2 South Taibai Road, Xi'an, Shaanxi, China
zhudongxu@vip.sina.com

Abstract. This paper aims to address the problem of clustering behaviors captured in surveillance videos for the applications of online normal behavior recognition and anomaly detection. A novel framework is developed for automatic behavior modeling and anomaly detection without any manual labeling of the training data set. The framework consists of the following key components: 1) Drawing from natural language processing, we introduce a compact and effective behavior representation method as a stochastic sequence of spatiotemporal events, where we analyze the global structural information of behaviors using their local action statistics. 2) The natural grouping of behaviors is discovered through a novel clustering algorithm with unsupervised model selection. 3) A runtime accumulative anomaly measure is introduced to detect abnormal behaviors, whereas normal behaviors are recognized when sufficient visual evidence has become available based on an online Likelihood Ratio Test (LRT) method. This ensures robust and reliable anomaly detection and normal behavior recognition at the shortest possible time. Experimental results demonstrate the effectiveness and robustness of our approach using noisy and sparse data sets collected from a real surveillance scenario.

Keywords: Computer Vision, Anomaly Detection, Hidden Markov Model, Latent Dirichlet Allocation.

1 Introduction

In visual surveillance, there is an increasing demand for automatic methods for analyzing an extreme number of surveillance video data produced continuously by video surveillance system. One of the key goals of deploying an intelligent video surveillance system (IVSS) is to detect abnormal behaviors and recognize the normal ones. To achieve this objective, one need to analyze and cluster previously observed behaviors, upon which a criterion on what is normal/abnormal is drawn and applied to newly captured patterns for anomaly detection. Due to the large amount of surveillance video data to be analyzed and the real-time nature of many surveillance applications, it is very desirable to have an automated system that requires little human intervention. In the paper, we aim to develop such a system that is based on fully unsupervised behavior modeling and robust anomaly detection.

X. Lai et al. (Eds.): E-Forensics 2010, LNICST 56, pp. 14–27, 2011.

Let us first define the problem of automatic behavior clustering for anomaly detection. Given a collection of unlabeled videos, the goal of automatic behavior clustering is to learn a model that is capable of detecting unseen abnormal behaviors while recognizing novel instances of expected normal ones. In this context, we define an anomaly as an atypical behavior that is not represented by sufficient samples in a training data set but critically satisfies the specificity constraint to an abnormal behavior. This is because one of the main challenges for the model is to differentiate anomaly from outliers caused by noisy visual features used for behavior representation. The effectiveness of an behavior clustering algorithm shall be measured by 1) how well anomalies can be detected (that is, measuring specificity to expected patterns of behavior) and 2) how accurately and robustly different classes of normal behaviors can be recognized (that is, maximizing between class discrimination).

To solve the problem, we develop a novel framework for fully unsupervised behavior modeling and anomaly detection. Our framework has the following key components:

1. A event-based action representation. Due to the space-time nature of actions and their variable durations, we need to develop a compact and effective action representation scheme and to deal with time warping. We propose a discrete event-based image feature extraction approach. This is different from most previous approaches such as [1], [2], [3] where features are extracted based on object tracking. A discrete event-based action representation aims to avoid the difficulties associated with tracking under occlusion in noisy scenes. Each action is modeled using "bag of events" representation [4], which provides a suitable means for time warping and measure the affinity between actions.
2. Behavior clustering based on discovering the natural grouping of behavior using Hidden Markov Model with Latent Dirichlet Allocation (HMM-LDA). A number of clustering techniques based on local word-statistics of a video have been proposed recently [5], [4], [6]. However, these approaches only capture the content of a video sequence and ignore its order. But generally behaviors are not fully defined by their action-content alone; however, there are preferred or typical action-orderings. This problem is addressed by the approach proposed in [4]. However, since discriminative prowess of the approach proposed in [4] is a function of the order over which action-statistics are computed, it comes at an exponential cost of computation complexity. In this work, we address these issues by proposing the usage of HMM-LDA to classify action instances of an behavior into states and topics, constructing a more discriminative feature space based on the context-dependent labels, and resulting in potentially better behavior-class discovery and classification.
3. Online anomaly detection using a runtime accumulative anomaly measure and normal behavior recognition using an online Likelihood Ratio Test (LRT) method. A runtime accumulative measure is introduced to determine an unseen normal or abnormal behavior. The behavior is then recognized as one

of the normal behavior classes using an online LRT method which holds the decision on recognition until sufficient visual features have become available. This is in order to overcome any ambiguity among different behavior classes observed online due to insufficient visual evidence at a given time instance. By doing so, robust behavior recognition and anomaly detection are ensured as soon as possible, as opposed to previous work such as [7], [8], which requires completed behavior being observed. Our online LRT-based behavior recognition approach is also advantageous over previous ones based on the Maximum Likelihood (ML) method [8], [9]. An ML-based approach makes a forced decision on behavior recognition without considering the reliability and sufficiency of the visual evidence. Consequently, it can be error prone.

Note that our framework is fully unsupervised in that manual data labeling is avoided in both the feature extraction and the discovery of the natural grouping of behaviors. There are a number of motivations for performing behavior clustering: First, manual labeling of behaviors is laborious and often rendered impractical given the vast amount of surveillance video data to be processed. More critically though, manual labeling of behaviors could be inconsistent and error prone. This is because a human tends to interpret behaviors based on the a priori cognitive knowledge of what should be present in a scene rather than solely based on what is visually detectable in the scene. This introduces a bias due to differences in experience and mental states.

The rest of the paper is structured as follows: Section 2 addresses the problem of behavior representation. The behavior clustering process is described in Section 3. Section 4 centers about the online detection of abnormal behavior and recognition of normal behavior. In Section 5, the effectiveness and robustness of our approach is demonstrated through experiments using noisy and sparse data sets collected from both indoor and outdoor surveillance scenarios. The paper concludes in Section 6.

2 Behavior Representation

2.1 Video Segmentation

The goal is to automatically segment a continuous video sequence $\mathbf{V}$ into N video segments $\mathbf{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_i \ldots, \mathbf{v}_N\}$ such that, ideally, each segment contains a single behavior pattern. The nth video segment $\mathbf{v}_n$ consisting of T_n image frames is represented as $\mathbf{v}_n = [\mathbf{I}_{n1}, \ldots, \mathbf{I}_{nt}, \ldots, \mathbf{I}_{nT_n}]$, where $\mathbf{I}_{nt}$ is the tth image frame. Depending on the nature of the video sequence to be processed, various segmentation approaches can be adopted. Since we are focusing on surveillance video, the most commonly used shot change detection-based segmentation approach is not appropriate. In a not-too-busy scenario, there are often nonactivity gaps between two consecutive behavior patterns that can be utilized for behavior segmentation. In the case where obvious nonactivity gaps are not available, the online segmentation algorithm proposed in [3] can be adopted. Specifically, video

content is represented as a high-dimensional trajectory based on automatically detected visual events. Breakpoints on the trajectory are then detected online using a Forward-Backward Relevance (FBR) procedure. Alternatively, the video can be simply sliced into overlapping segments with a fixed time duration [5].

2.2 Behavior Representation

First, moving pixels of each image frame in the video are detected directly via spatiotemporal filtering of the image-frames:

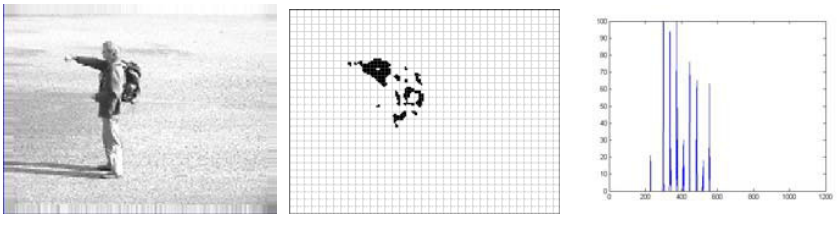
$$M_t(x,y,t) = (I(x,y,t) * G(x,y;\sigma) * h_{ev}(t;\tau,\omega))^2 + (I(x,y,t) * G(x,y;\sigma) * h_{od}(t;\tau,\omega))^2 > Th_a \quad (1)$$

where $G(x,y;\sigma) = e^{((\frac{x}{\sigma_x})+(\frac{y}{\sigma_y}))}$ is the 2D Gaussian smoothing kernel, applied only along the spatial dimensions (x,y), and h_{ev} and h_{od} are a quadrature pair of 1D Gabor filters applied temporally, which are defined as $h_{ev}(t;\tau,\omega) = -cos(2\pi t\omega)e^{-t^2/\tau^2}$ and $h_{od}(t;\tau,\omega) = -sin(2\pi t\omega)e^{-t^2/\tau^2}$. The two parameters σ and τ correspond to the spatial and temporal scales of the detector respectively. This convolution is linearly separable in space and time and is fast to compute.

Second, each frame is defined as a event. A detected event is represented as the spatial histogram of the detected objects. Let $H_t(i,j)$ be an $m \times m$ spatial histogram, with m typically equal to 10.

$$H_t(i,j) = \sum_{x,y} M(x,y,t) \cdot \delta(b_i^x \le x < b_{i+1}^x) \cdot \delta(b_i^y \le y < b_{i+1}^y) \quad (2)$$

where b_i^x,b_j^y $(i,j = 1,\ldots,m)$ are the boundaries of the spatial bins. The spatial histograms indicate the rough area of object movement. The process is demonstrated in figure 1(a)-(c).



(a) (b) (c)

Fig. 1. Feature extraction from video frames. (a) original video frame. (b) binary map of objects. (c) spatial histogram of (b).

Third, vector quantization is applied to the histogram feature vectors classifying them into a dictionary of K_e event classes $\mathbf{w} = \{w_1,\ldots,w_K\}$ using K-means. So each detected event is classified into one of the K_e event classes.

Finally, the behavior captured in the nth video segment v_n is represented as an event sequence $\mathbf{P}_n$, given as

$$\mathbf{w}_n = [w_{n1}, \ldots, w_{nt}, \ldots, w_{nT_n}] \tag{3}$$

where T_n is the length of the nth video segment. w_{nt} corresponds to the tth image frame of v_n, where $w_{nt} = w_k$ indicates that an event of the kth event class has occurred in the frame.

3 Behavior Clustering

The behavior clustering problem can now be defined formally. Consider a training data set $\mathbf{D}$ consisting of N feature vectors

$$\mathbf{D} = \{\mathbf{w}_1, \ldots, \mathbf{w}_n, \ldots, \mathbf{w}_N\} \tag{4}$$

where $\mathbf{w}_n$ is defined in (6), represents the behavior captured by the nth video $\mathbf{v}_n$. The problem to be addressed is to discover the natural grouping of the training behaviors upon which a model for normal behavior can be built. This is essentially a data clustering problem with the number of clusters unknown. There are a number of aspects that make this problem challenging: 1) Each feature vector $\mathbf{w}_n$ can be of different lengths. Conventional clustering approaches require that each data sample is represented as a fixed length feature vector. 2) Model selection needs to be performed to determine the number of cluster. To overcome the above mentioned difficulties, we propose a clustering algorithm with feature and model selection based on modeling each behavior using HMM-LDA.

3.1 Hidden Markov Model with Latent Dirichlet Allocation (HMM-LDA)

Suppose we are given a collection of M video sequences $D = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_M\}$ containing action words from a vocabulary of size V $(i = 1, \ldots, V)$. Each video $\mathbf{w}_j$ is represented as a sequence of N_j action words $\mathbf{w}_j = (w_1, w_2, \ldots, w_{N_j})$, where w_i is the action word representing the i-th frame. Then the process that generates each video $\mathbf{w}_j$ in the corpus D is:

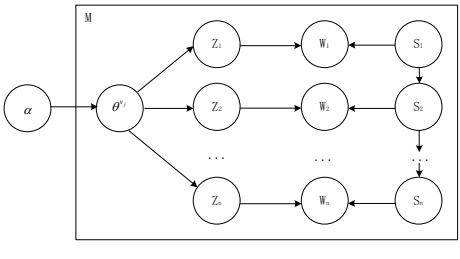


Fig. 2. Graphical representation of HMM-LDA model

1. Draw topic weights $\theta^{(\mathbf{w}_j)}$ from $Dir(\alpha)$
2. For each word w_i in video $\mathbf{w}_j$
   (a) Draw z_i from $\theta^{(\mathbf{w}_j)}$
   (b) Draw c_i from $\pi^{(c_{i-1})}$
   (c) If $c_i = 1$, then draw w_i from $\phi^{(z_i)}$, else draw w_i from $\phi^{(c_i)}$

Here we fixed the number of latent topic K to be equal to the number of behavior categories to be learnt. Also, α is the parameter of a K-dimensional Dirichlet distribution, which generates the multinomial distribution $\theta^{(\mathbf{w}_j)}$ that determines how the behavior categories (latent topics) are mixed in the current video $\mathbf{w}_j$. Each spatial-temporal action word w_i in video $\mathbf{w}_j$ is mapped to a hidden state s_i. Each hidden state s_i generates action words w_i according to a unigram distribution $\phi^{(c_i)}$ except the special latent topic state z_i, where the z_ith topic is associated with a distribution words $\phi^{(z_i)}$. $\phi^{(z_i)}$ corresponds to the probability $p(w_i|z_k)$. Each video $\mathbf{w}_j$ has a distribution over topic $\theta^{(\mathbf{w}_j)}$, and transitions between classes c_{i-1} and c_i follow a distribution $\pi^{s_{i-1}}$. The complete probability model is

$$\theta \sim Dirichlet(\alpha) \tag{5}$$

$$\phi^{(z)} \sim Dirichlet(\beta) \tag{6}$$

$$\pi \sim Dirichlet(\gamma) \tag{7}$$

$$\phi^{(c)} \sim Dirichlet(\delta) \tag{8}$$

Here, α, β, γ and δ are hyperparameters, specifying the nature of the priors on θ, $\phi^{(z)}$, π and $\phi^{(c)}$.

3.2 Learning the Behavior Models

Our strategy for learning topics differs from previous approaches [12] in not explicitly representing θ, $\phi^{(z)}$, π and $\phi^{(c)}$ as parameters to be estimated, but instead considering the posterior distribution over the assignments of words to topics, $p(\mathbf{z}|\mathbf{c}, \mathbf{w})$. We then obtain estimates of θ, $\phi^{(z)}$, π and $\phi^{(c)}$ by examining this posterior distribution. Computing $p(\mathbf{z}|\mathbf{c}, \mathbf{w})$ involves evaluating a probability distribution on a large discrete state space. We evaluate $p(\mathbf{z}|\mathbf{c}, \mathbf{w})$ by using a Monte Carlo procedure, resulting in an algorithm that is easy to implement, requires little memory, and is competitive in speed and performance with existing algorithms.

In Markov chain Monte Carlo, a Markov chain is constructed to converge to the target distribution, and samples are then taken from Markov chain. Each state of the chain is an assignment of values to the variable being sampled and transitions between states follow a simple rule. We use Gibbs sampling where the next state is reached by sequentially sampling all variable from their distribution when conditioned on the current values of all other variables and the data. To

apply this algorithm we need two full conditional distributions, $p(z_i|\mathbf{z}_{-i}, \mathbf{c}, \mathbf{w})$ and $p(c_i|\mathbf{c}_{-i}, \mathbf{z}, \mathbf{w})$. These distributions can be obtained by using the conjugacy of the Dirichlet and multinomial distributions to integrate out the parameters θ and ϕ, yielding

$$p(z_i|\mathbf{z}_{-i}, \mathbf{c}, \mathbf{w}) \propto \begin{cases} n_{z_i}^{w_j} + \alpha, & c_i \neq 1 \\ (n_{z_i}^{w_j} + \alpha)\dfrac{n_{w_i}^{(z_i)} + \beta}{n^{(z_i)} + W\beta}, & c_i = 1 \end{cases} \tag{9}$$

where $n_{z_i}^{(\mathbf{w}_j)}$ is the number of words in video $\mathbf{w}_j$ assigned to topic z_i, $n_{w_i}^{(z_i)}$ is the number of words assigned to topic z_i that are the same as w_i, and all counts include only words for which $c_i = 1$ and exclude case i.

$$p(c_i|\mathbf{c}_{-i}) = \frac{(n_{c_i}^{(c_{i-1})} + \gamma)(n_{c_{i+1}}^{(c_i)} + I(c_{i-1} = c_i)I(c_i = c_{i+1}) + \gamma)}{n_{\cdot}^{(c_i)} + I(c_{i-1} = c_i) + C_\gamma} \tag{10}$$

$$p(c_i|\mathbf{c}_{-i}, \mathbf{z}, \mathbf{w}) \propto \begin{cases} \dfrac{n_{w_i}^{(c_i)} + \delta}{n^{(c_i)} + W\delta} p(c_i|\mathbf{c}_{-i}), & c_i \neq 1 \\ \dfrac{n_{w_i}^{(z_i)} + \beta}{n^{(z_i)} + W\beta} p(c_i|\mathbf{c}_{-i}), & c_i = 1 \end{cases} \tag{11}$$

where $n_{w_i}^{(z_i)}$ is as before, $n_{w_i}^{(c_i)}$ is the number of words assigned to class c_i that are the same as w_i, excluding case i, and $n_{c_i}^{(c_i)}$ is the number of transitions from class c_{i-1} to class c_i, and all counts of transitions exclude transitions both to and from c_i. $I(.)$ is an indicator function, taking the value 1 when its argument is true, and 0 otherwise. Increasing the order of the HMM introduces additional terms into $p(c_i|\mathbf{c}_i)$, but does not otherwise affect sampling.

The z_i variables are initialized to values in $\{1, 2, \ldots, K\}$, determining the initial state of the Markov chain. We do this with an online version of the Gibbs samples, using Eq.12 to assign words to topics, but with counts that are computed from the subset of the words seen so far rather than the full data. The chain is then run for a number of iterations, each time finding a new state by sampling each z_i from the distribution specified by Eq.12. Because the only information needed to apply Eq.12 is the number of times a word is assigned to a topic and the number of times a topic occurs in a document, the algorithm can be run with minimal memory requirements by caching the sparse set of nonzero counts and updating them whenever a word is reassigned. After enough iteration for the chain to approach the target distribution, the current values of the z_i variables are recorded. Subsequent samples are taken after an appropriate lag to ensure that their autocorrelation is low.

With a set of samples from the posterior distribution $p(\mathbf{z}|\mathbf{c}, \mathbf{w})$, statistics that are independent of the content of individual topics can be computed by integrating across the full set of samples. For any single sample we can estimate θ, $\phi^{(z)}$, π and $\phi^{(c)}$ from the value $\mathbf{z}$ by

$$\widehat{\phi}^{(z)} = \frac{n_{w_i}^{(z_i)} + \beta}{n^{(z_i)} + W\beta} \tag{12}$$

$$\widehat{\phi}^{(c)} = \frac{n_{w_i}^{(c_i)} + \delta}{n^{(c_i)} + W\delta} \tag{13}$$

$$\theta = n_{z_i}^{w_j} + \alpha \tag{14}$$

$$\pi = \frac{(n_{c_i}^{(c_{i-1})} + \gamma)(n_{c_{i+1}}^{(c_i)} + I(c_{i-1} = c_i)I(c_i = c_{i+1}) + \gamma)}{n_{\cdot}^{(c_i)} + I(c_{i-1} = c_i) + C_\gamma} \tag{15}$$

3.3 Model Selection

Given values of α, β and γ, the problem of choosing the appropriate value for K is a problem of model selection, which we address by using a standard method from Bayesian statistics. For a Bayesian statistician faced with a choice between a set of statistical models, the natural response is to compute the posterior probability of the set of models given the observed data. The key constituent of this posterior probability will be the likelihood of the data given the model, integrating over all parameters in the model. In our case, the data are the words in the corpus, $\mathbf{w}$, and the model is specified by the number of topics, K, so we wish to compute the likelihood $p(\mathbf{w}|K)$. The complication is that this requires summing over all possible assignments of words to topics $\mathbf{z}$. However, we can approximate $p(\mathbf{w}|K)$ by taking the harmonic mean of a set of values of $p(\mathbf{w}|\mathbf{z}, K)$ when $\mathbf{z}$ is sampled from the posterior $p(\mathbf{z}|\mathbf{c}, \mathbf{w}, K)$. Our Gibbs sampling algorithm provides such samples, and the value of $p(\mathbf{w}|\mathbf{z}, K)$ can be computed.

4 Online Anomaly Detection and Normal Behavior Recognition

Given a unseen behavior pattern $\mathbf{w}$, we calculate the likelihood $l(\mathbf{w}; \alpha, \beta) = P(\mathbf{w}|\alpha, \beta)$. The likelihood can be used to detect whether an unseen behavior pattern is normal using a runtime anomaly measure. If it is detected to be normal, the behavior pattern is then recognized as one of the K classes of normal behavior patterns using an online LRT method.

An unseen behavior pattern of length T is represented as $\mathbf{w} = (w_1, \ldots, w_t, \ldots, w_T)$. At the tth frame, the accumulated visual information for the behavior pattern, represented as $\mathbf{w}_t = (w_1, \ldots, w_t)$, is used for online reliable anomaly detection. First, the normalized likelihood of observing $\mathbf{w}$ at the tth frame is computed as

$$l_t = P(\mathbf{w}_t|\alpha, \beta) \tag{16}$$

l_t can be easily computed online using the variational inference method.

We then measure the anomaly of $\mathbf{w}_t$ using an online anomaly measure Q_t

$$Q_t = \begin{cases} l_t, & if\ t = 1 \\ (1-\alpha)Q_{t-1} + \alpha(l_t - l_{t-1}), & otherwise \end{cases} \tag{17}$$

where α is an accumulating factor determining how important the visual information extracted from the current frame is for anomaly detection. We have $0 < \alpha \leq 1$. Compared to l_t as an indicator of normality/anomaly, Q_t could add more weight to more recent observations. Anomaly is detected at frame t if

$$Q_t < Th_A \tag{18}$$

where Th_A is the anomaly detection threshold. The value of Th_A should be set according to the detection and false alarm rates required by each particular surveillance application.

At each frame t, a behavior pattern needs to be recognized as one of the K behavior classes when it is detected as being normal, that is, $Q_t > Th_A$. This is achieved by using an online LRT method. More specifically, we consider a hypotheses test between the following

H_k:$\mathbf{w}_t$ is from the hypothesized model z_k and belongs to kth normal behavior class;

H_0:$\mathbf{w}_t$ is from a model other than z_k and does not belong to the kth normal behavior class;

where H_0 is called the alternative hypothesis. Using LRT, we compute the likelihood ratio of accepting the two hypotheses as

$$r_k = \frac{P(\mathbf{w}_t; H_k)}{P(\mathbf{w}_t; H_0)} \tag{19}$$

The hypothesis H_k can be represented by the model z_k, which has been learned in the behavior clustering step. The key to LRT is thus to construct the alternative model that represents H_0. In a general case, the number of possible alternatives is unlimited; $P(\mathbf{w}_t; H_0$ can thus only be computed through approximation. Fortunately, in our case, we have determined at the tth frame that $\mathbf{w}_t$ is normal and can only be generated by one of the K normal behavior classes. Therefore, it is reasonable to construct the alternative model as a mixture of the remaining of $K-1$ normal behavior classes. In particular, (4) is rewritten as

$$r_k = \frac{P(\mathbf{w}_t|z_k)}{\sum_{i \neq k} P(\mathbf{w}_t|z_i)} \tag{20}$$

Note that r_k is a function of t and computed over time. $\mathbf{w}_t$ is reliably recognized as the kth behavior class only when $1 \ll Th_r < r_k$. When there are more than one r_k greater than Th_r, the behavior pattern is recognized as the class with the largest r_k.

5 Experiments

In this section, we illustrate the effectiveness and robustness of our approach on behavior clustering and online anomaly detection with experiments using data sets collected from the entrance/exit area of an office building.

5.1 Dataset and Feature Extraction

A CCTV camera was mounted on a on-street utility pole, monitoring the people entering and leaving the building (see Fig.3). Daily behaviors from 9a.m. to 5p.m. for 5 days were recorded. Typical behaviors occurring in the scene would be people entering, leaving and passing by the building. Each behavior would normally last a few seconds. For this experiment, a data set was collected from 5 different days consisting of 40 hours of video, totaling to 2880,000 frames. A training set consisting of 568 instances was randomly selected from the overall 947 instances without any behavior class labeling. The remaining 379 instances were used for testing the trained model later.

5.2 Behavior Clustering

To evaluate the number of clusters K, we used the Gibbs sampling algorithm to obtain samples from the posterior distribution over $\mathbf{z}$ for K values of 3, 4, 5, 6, 7, 8, and 12. For all runs of the algorithm, we used $\alpha = 50/T$, $\beta = 0.01$ and $\gamma = 0.1$, keeping constant the sum of the Dirichlet hyper-parameters, which can be interpreted as the number of virtual samples contribution to the smoothing of θ. We computed an estimate of $p(\mathbf{w}|K)$ for each value of K . For all values of K, we ran 7 Markov chains, discarding the first 1,000 iterations, and then took 10 samples from each chain at a lag of 100 iterations. In all cases, the log-likelihood values stabilized within a few hundred iterations. Estimates of $p(\mathbf{w}|K)$ were computed based on the full set of samples for each value of K and are shown in Fig.3.

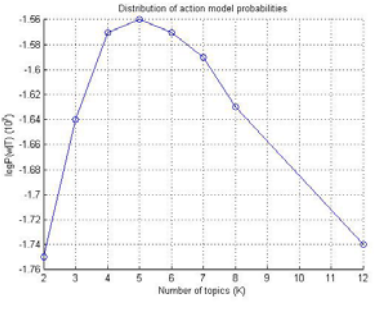


Fig. 3. Model selection results

The results suggest that the data are best accounted for by a model incorporating 5 topics. $p(\mathbf{w}|K)$ initially increases as function of K, reaches a peak at $K = 5$, and then decreases thereafter. By observation, each discovered data cluster mainly contained samples corresponding to one of five behavior classes listed in Table 1.

Table 1. The Five Classes of Behaviors that Most Commonly Occurred in the entrance/exit area of an office building

C1	going into the office building
C2	leaving the office building
C3	passing by the office building
C4	getting off a car and entering the office building
C5	leaving the office building and getting on a car

5.3 Anomaly Detection

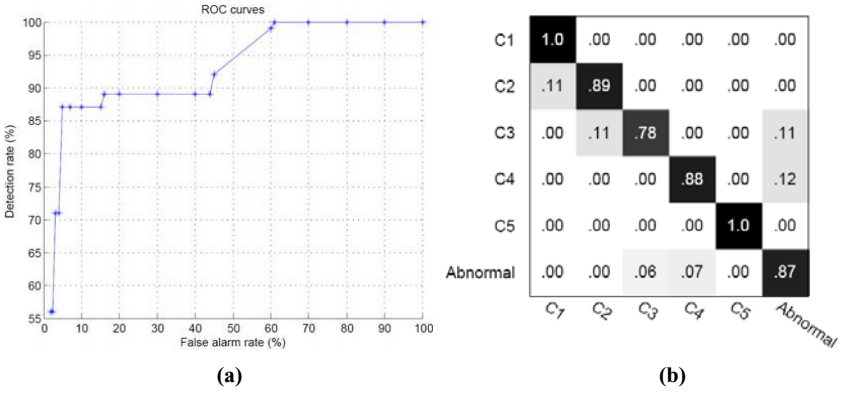
The behavior model built using both labeled and unlabeled behaviors were used to perform online anomaly detection. To measure the performance of the learned models on anomaly detection, each behavior in the testing sets was manually labeled as normal if there were similar behaviors in the corresponding training sets and abnormal otherwise. A testing pattern was detected as being abnormal when (18) was satisfied. The accumulating factor α for computing Q_t was set to 0.1. Fig.4. demonstrates one example of anomaly detection in the entrance/exit area of an office building.

We measure the performance of anomaly detection using the anomaly detection rate, which equals to $\frac{\#(abnormal\ detected\ as\ abnormal)}{\#(abnormal\ patterns)}$, and the false alarm rate, which equals to $\frac{\#(normal\ detected\ as\ abnormal)}{\#(normal\ patterns)}$. The detection rate and false alarm rate of anomaly detection are shown in the form of a Receiver Operating Characteristic (ROC) curve by varying the anomaly detection threshold Th_A, as Fig.5(a).

5.4 Normal Behavior Recognition

To measure the recognition rate, the normal behaviors in the testing sets were manually labeled into different behavior classes. A normal behavior was recognized correctly if it was detected as normal and classified into a behavior class containing similar behaviors in the corresponding training set by the learned

Fig. 4. Example of anomaly detection in the entrance/exit area of an office building. (a) An abnormal behavior where one people attempted to destroy the car parking the area. It resembles C3 in the early stage. (b) The behavior was detected as an anomaly from Frame 62 till the end based on Q_t.



(a) **(b)**

Fig. 5. (a) the mean ROC curves for our dataset. (b)confusion matrix for our dataset; rows are ground truth, and columns are model results.

behavior model. Fig.5(b) shows that when a normal behavior was not recognized correctly by a model trained using unlabeled data, it was most likely to be recognized as belonging to another normal behavior class. On the other hand, for a model trained by labeled data, a normal behavior was most likely to be wrongly detected as an anomaly if it was not recognized correctly. This contributed to the higher false alarm rate for the model trained by labeled data.

5.5 Result Analysis and Discussion

To compare our approach with six other methods, we use exactly the same experiment setup and list the comparison results in Table 2. Each of these is a anomalous behavior detection algorithm that is capable of dealing with low resolution and noisy data. We implement the algorithms of Xiang *et al.* [3], Wang *et al.* [6], Niebles *et al.* [13], Boiman *et al.* [7], Hamid *et al.* [4] and Zhong *et al.* [5]. The key findings of our comparison are summarized and discussed as follows:

1. Table 2 shows that the precision of our HMM-LDA is superior to the HMM method [3], the LDA method [6], the MAP-based method [7] and two

Table 2. Comparison of different methods

methods	Anomaly Detection Rate (%)
Our method	89.26
Xiang *et al.* [3]	85.76
Wang *et al.* [6]	84.46
Niebles *et al.* [13]	83.50
Boiman *et al.* [7]	83.32
Hamid *et al.* [4]	88.48
Zhong *et al.* [5]	85.56

co-clustering algorithms [5],[4]. HMM [3] outperforms the LDA [6] on our scenario, but HMM [3] require explicit modeling of anomalous behaviors structure with minimal supervision. Some recent methods ([5] using Latent Semantic Analysis, [13] using probabilistic Latent Semantic Analysis, [6] using Latent Dirichlet Allocation, [4] using n-grams) extract behavior structure simply by computing local action-statistics, but are limited by their ability to capture behavior structure only up to some fixed temporal resolution. Our HMM-LDA provided the best account, being able to efficiently extract the variable length action-subsequence of behavior, constructing a more discriminative feature space, and resulting in potentially better behavior-class discovery and classification.

2. Work done in [5] clusters behaviors into its constituent sub-class, labeling the clusters with low internal cohesiveness as anomalous cluster. This makes it infeasible for online anomaly detection. The anomaly detection method proposed in [4] was claimed to be online. Nevertheless, in [4], anomaly detection is performed only when the complete behavior pattern is observed. In order to overcome any ambiguity among different behavior classes observed online due to different visual evidence at a given time instance, our online LRT method holds the decision on recognition until sufficient visual features have become available.

6 Conclusions

In conclusion, we have proposed a novel framework for robust online behavior recognition and anomaly detection. The framework is fully unsupervised and consisted of a number of key components, namely, a behavior representation based on spatial-temporal actions, a novel clustering algorithm using HMM-LDA based on action words, a runtime accumulative anomaly measure, and an online LRT-based normal behavior recognition method. The effectiveness and robustness of our approach is demonstrated through experiments using data sets collected from real surveillance scenario.

References

1. Yamato, J., Ohya, J., Ishii, K.: Recognizing human action in time-sequential images using hidden markov model. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (1992)
2. Bobick, A.F., Wilson, A.D.: A state-based approach to the representation and recognition of gesture. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(12), 1325–1337 (1997)
3. Xiang, T., Gong, S.: Beyond tracking: Modelling activity and understanding behaviour. International Journal of Computer Vision 67(1), 21–51 (2006)
4. Hamid, R., Johnson, A., Batta, S., Bobick, A., Isbell, C., Coleman, G.: Detection and Explanation of Anomalous Activities: Representing Activities as Bags of Event n-Grams. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1031–1038 (2005)

5. Zhong, H., Shi, J., Visontai, M.: Detecting Unusual Activity in Video. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 819–826 (2004)
6. Wang, Y., Mori, G.: Human Action Recognition by Semi-Latent Topic Models. IEEE Transactions on Pattern Analysis and Machine Intelligence (2009)
7. Boiman, O., Irani, M.: Detecting irregularities in images and in video. In: IEEE International Conference on Computer Vision, pp. 462–469 (2005)
8. Oliver, N., Rosario, B., Pentland, A.: A Bayesian computer vision system for modelling human interactions. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(8), 831–843 (2000)
9. Zelnik-Manor, L., Irani, M.: Event-based video analysis. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 123–130 (2001)
10. Comaniciu, D., Meer, P.: Mean Shift Analysis and Applications. In: Proceedings of the International Conference on Computer Vision, Kerkyra, pp. 1197–1203 (1999)
11. Efros, A.A., Berg, A.C., Mori, G., Malik, J.: Recognizing action at a distance. In: IEEE International Conference on Computer Vision, pp. 726–733 (2003)
12. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of Machine Learning Research 3, 993–1022 (2003)
13. Niebles, J.C., Wang, H., Fei-Fei, L.: Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words. In: Proc. British Machine Vision Conference, pp. 1249–1258 (2006)