Incremental and Adaptive Abnormal Behaviour Detection*

Tao Xiang† and Shaogang Gong
Department of Computer Science
Queen Mary, University of London, London E1 4NS, UK
{txiang,sgg}@dcs.qmul.ac.uk

Abstract

We develop a novel visual behaviour modelling approach that performs incremental and adaptive model learning for online abnormality detection in a visual surveillance scene. The approach has the following key features that make it advantageous over previous ones: (1) *Fully unsupervised learning*: both feature extraction for behaviour pattern representation and model construction are carried out without the laborious and unreliable process of data labelling. (2) *Robust abnormality detection*: using Likelihood Ratio Test (LRT) for abnormality detection, the proposed approach is robust to noise in behaviour representation. (3) *Online and incremental model construction*: after being initialised using a small bootstrapping dataset, our behaviour model is learned incrementally whenever a new behaviour pattern is captured. This makes our approach computationally efficient and suitable for real-time applications. (4) *Model adaptation to reflect changes in visual context*. Online model structure adaptation is performed to accommodate changes in the definition of normality/abnormality caused by visual context changes. This caters for the need to reclassify what may initially be considered as being abnormal to be normal over time, and vice versa. These features are not only desirable but also necessary for processing large volume of unlabelled surveillance video data with visual context changing over time. The effectiveness and robustness of our approach are demonstrated through experiments using noisy datasets collected from a real world surveillance scene. The experimental results show that our incremental and adaptive behaviour modelling approach is superior to a conventional batch-mode one in terms of both performance on abnormality detection and computational efficiency.

Keywords: Behaviour Analysis and Recognition, Visual Surveillance, Abnormality Detection, Incremental Learning, Likelihood Ratio Test, Dynamic Scene Modelling, Dynamic Bayesian Networks.

*An earlier version of this paper appeared in a conference proceeding [29].

†Corresponding author. Tel: (+44)-(0)20-7882-5201; Fax: (+44)-(0)20-8980-6533

1 Introduction

Abnormal behaviour detection in video is one of the most critical problems in visual surveillance. Although its importance has long been recognised and much effort has been made to tackle the problem [4, 5, 18, 14, 10, 8, 33, 23, 11, 27, 32, 3], it remains largely unsolved especially for cluttered busy scenes outside a well-controlled laboratory environment. This is due to not only the complexity and variety of visual behaviour in a realistic and unconstrained environment, but also the ambiguous nature in the definition of normality and abnormality, which is highly dependent on the visual context and can change over time. In particular, a behaviour can be considered as either being normal or abnormal depending on when and where it takes place. This causes problems for the conventional behaviour models learned in batch mode which remain static once trained.

In this paper, we develop a novel behaviour modelling approach that performs incremental and adaptive behaviour model learning for online abnormality detection. After initialisation using a small bootstrapping dataset, our behaviour model performs online abnormal behaviour detection and incremental model parameter updating simultaneously whenever a new behaviour pattern is captured. More importantly, our model is capable of detecting changes of visual context and definition of abnormality and carrying out model adaptation to reflect these changes. Our approach has a number of key features which distinguish it from previous approaches:

1. **Fully unsupervised learning.** Both feature extraction for behaviour pattern representation and behaviour model construction are fully unsupervised in our approach. In particular, our behaviour model learning is based on unlabelled data without knowing whether each training behaviour pattern is normal and to which normal behaviour class it belongs. Compared to existing supervised learning based approaches [18, 14, 10, 8], our approach is intrinsically more difficult but also offering a number of significant advantages: (a) The laborious, often impractical and unreliable process of manual labelling is avoided. (b) Abnormal behaviour patterns are commonly rare and unexpected, therefore difficult to define. Our approach lifts the burden of manually defining and selecting abnormal training samples.

2. **Robust abnormality detection.** A Likelihood Ratio Test (LRT) [16] based abnormal behaviour detection method is developed. Specifically, apart form a model for normal behaviour, an approximate abnormal behaviour model is also constructed. Both models are built based on mixtures of Dynamic Bayesian Networks (DBNs) [9]. Given a newly observed behaviour pattern, whether it is abnormal is determined using the likelihood ratio of generating the pattern using the two models. As a probabilis-

tic model, a mixture of DBNs can cope with behaviour representation errors occurred at individual frames. Moreover, the adoption of LRT takes into account the subtle and ambiguous nature of defining normality/abnormality especially given the inevitable errors in representation. Our method is thus robust to noise in behaviour representation.

3. **Online and incremental model construction.** After being initialised using a small bootstrapping dataset, our behaviour model is learned in an online and incremental manner. Specifically, given a newly observed behaviour pattern, it is detected as either normal or abnormal by the model learned using the patterns observed so far; the model parameters are then updated incrementally using only the new data based on an incremental Expectation-Maximisation (EM) algorithm. This is in contrast with most previous behaviour modelling techniques that operate in a batch mode where observing (and collecting) sufficiently large samples of behaviour patterns is necessary before model training. Online incremental learning is not only desirable but also necessary for processing large volume of unlabelled surveillance video data for which a batch-mode method is both computationally and logistically too expensive. Based on online incremental statistic learning, our approach is computationally efficient for modelling complex behaviours observed continuously over time and thus suitable for real-time surveillance applications.

4. **Model adaptation to reflect changes in visual context.** Whether a behaviour pattern is abnormal is highly dependent on the visual context. Existing methods assume what was considered to be normal/abnormal in the training dataset would continue to hold true regardless of the inevitable circumstantial changes over time. Our approach enables model adaptation to reflect these changes. This is achieved through online model structure updating and a bias towards more recent observations. For instance, when an unfamiliar behaviour pattern is observed, it would be initially considered to be an abnormality. However, if similar patterns were to appear repeatedly thereafter, they shall be deemed as normal. In this case, our behaviour model would adapt to this change of context and the model structure will be updated to accommodate the addition of a new normal behaviour class. Note that although both incremental learning and model adaptation are part of the online model updating processes, they focus on different aspects of a model, namely model parameters and model structure respectively. Furthermore, unlike incremental model learning which is triggered by the arrival of new data, model adaptation takes place only when visual context changes are detected.

The rest of the paper is structured as follows: Section 2 reviews related work to highlight the contributions of this work. Section 3 addresses the problem of behaviour representation. An event-based behaviour

representation is presented. The incremental and adaptive behaviour modelling algorithm is described in Section 4. It consists of three key steps: model initialisation using a bootstrapping dataset (Section 4.1), online abnormality detection based on LRT (Section 4.2), and online model updating based on incremental EM learning and model adaptation (Section 4.3). In Section 5, the effectiveness and robustness of our approach are demonstrated through experiments using noisy datasets collected from a real world surveillance scene. In particular, its performance on abnormality detection and computational efficiency is compared with a batch-mode method. The paper concludes in Section 6.

2 Related Work

Much work on abnormal behaviour detection[1] took a supervised learning approach [18, 14, 10, 8, 6, 22] based on the assumption that there exist well-defined and known *a priori* behaviour classes (both normal and abnormal). As demonstrated in [27], a supervised model can give inferior abnormality detection performance compared to that of an unsupervised model even though more efforts are required in manual labelling of data. Note that the approaches proposed in [6] and [22] are rule-based approaches, i.e. human knowledge on what is normal/abnormal in a scene is hand-crafted into the model. This differs from most other approaches which are based on statistical learning. These rule-based approaches provide an effective solution for detection abnormality in a simple and static scene. However, defining and hand-crafting rules become infeasible for modelling complex behaviour. Moreover, these approaches break down when the scene context and definitions of normality/abnormality change over time.

More recently, a number of techniques have been proposed for unsupervised learning of behaviour models [33, 11, 3, 27, 24]. They can be further categorised into two different types according to whether an explicit model is built. Approaches that do not model behaviour explicitly either perform clustering on observed patterns and label small clusters as abnormal [33, 11], or build a database of spatio-temporal patches using only regular/normal behaviours (manually labelled) and detect those patterns that cannot be composed from the database as being abnormal [3]. The approach proposed in [33] cannot be applied to any previously unseen behaviour patterns therefore is suitable for post-mortem analysis rather than on-the-fly abnormality detection. This problem is addressed by the approaches proposed in [11] and [3]. However, in these approaches all the previously observed normal behaviour patterns must be stored either in the form of sequences of discrete events [11] or ensembles of spatio-temporal patches [3] for detecting abnormality

[1]The notion of Abnormal Behaviour appeared in different names in the literature including unusual, suspicious, or surprising behaviour/events/activities, or simply anomaly, abnormality, irregularities or outliers.

from unseen data, which jeopardises the scalability of these approaches. Alternatively, an explicit model based on a mixture of Dynamic Bayesian Networks (DBNs) can be constructed to learn specific behaviour classes for automatic detection of abnormalities on-the-fly given unseen data [27]. However, since the model is trained in a batch mode, it cannot cope with changes of visual context.

There is also another approach that differs from both the supervised and unsupervised techniques above. A semi-supervised model was introduced by [32] with a two-stages training process. In stage one, a normal behaviour model is learned using labelled normal patterns. In stage two, an abnormal behaviour model is learned unsupervised using Bayesian adaptation. This approach still suffers from the laborious manual data labelling process.

The work presented in this paper is closely related to our earlier work [27] in the aspect of behaviour representation. However, in addition to the key advantage of online incremental and adaptive model learning, we develop a more principled criterion for abnormality detection based on a Likelihood Ratio Test (LRT) originally proposed for key-words detection in speech recognition [25]. This makes our approach more robust to errors in behaviour representation. It is also worth pointing out that both the approaches proposed in [11] and [3] are claimed to be incremental and online. Nevertheless, in [11] online abnormality detection only takes place after the model is built in a batch mode, while in [3] the incremental model learning process requires human intervention (i.e. manually defining a new class of normal behaviour and adding it to the database). In our approach, model learning/adaptation and abnormality detection are carried out simultaneously without human intervention as new data become available.

3 Behaviour Representation

A continuous video $\mathbf{V}$ is segmented into N video segments $\mathbf{V} = \{\mathbf{V}_1, \ldots, \mathbf{V}_n, \ldots, \mathbf{V}_N\}$ so that ideally each segment contains a single behaviour pattern that does not necessarily restrict to a single object (i.e. may consist of a group or interactive activity). The n-th video segment $\mathbf{V}_n$ consists of T_n image frames represented as $\mathbf{V}_n = \{\mathbf{I}_{n1}, \ldots, \mathbf{I}_{nt}, \ldots, \mathbf{I}_{nT_n}\}$ where $\mathbf{I}_{nt}$ is the t-th image frame. Note that in this paper, a behaviour pattern is defined as a sample of a class of behaviour visually captured in a video. For instance, in a supermarket, the behaviour of customer checking out at a counter can be captured many times as behaviour patterns over a short period of time on a surveillance video. Each of such behaviour patterns, although belonging to the same behaviour class, can exhibit considerable variations visually. This characteristic must be considered when a behaviour modelling approach is designed.

In this paper we focus on surveillance videos taken by fixed cameras. The most commonly used shot change detection based segmentation approach is thus not appropriate because a continuous surveillance video contains only a single shot. Depending on the nature of the video sequence to be processed, a number of approaches can be adopted to address the problem. In a not-too-busy scenario, there are often non-activity gaps between two consecutive behaviour patterns which can be utilised for activity segmentation. In the case where obvious non-activity gaps are not available, an on-line segmentation algorithm proposed in [26] can be adopted. More specifically, surveillance video contents are firstly represented holistically in space and over time based on discrete events detected automatically in the scene resulting in a high-dimensional video content trajectory (more details about the discrete scene events follows). The break points on the trajectory correspond to video content changes and can be detected using the on-line algorithm proposed in [26]. Alternatively, the video can be simply sliced into overlapping segments with a fixed time duration [33].

A discrete scene event based approach [10, 28] is adopted for behaviour pattern representation. It has been demonstrated in [28] that a discrete event based representation is much more effective for cluttered and busier scenes in comparison to a continuous trajectory based representation employed by most existing approaches. Firstly, an adaptive Gaussian mixture background model [23] is adopted to detect foreground pixels which are modelled using Pixel Change History (PCH) [30]. Secondly, the foreground pixels in a vicinity are grouped into a blob using the connected component method. Each blob with its average PCH value greater than a threshold is then defined as a scene event. A detected scene event is represented as a 7-dimensional feature vector

$$\mathbf{f} = [\bar{x}, \bar{y}, w, h, R_f, M_p x, M_p y] \tag{1}$$

where $(\bar{x}, \bar{y})$ is the centroid of the blob, w and h are the width and height of the bounding box associated with the blob respectively, R_f is the filling ratio of foreground pixels within the bounding box, and $(M_p x, M_p y)$ are a pair of first order moments of the PCH image within the bounding box. Among these features, $(\bar{x}, \bar{y})$ are location features, (w, h) and R_f are principally shape features but also contain some indirect motion information, and $(M_p x, M_p y)$ are motion features capturing the direction of object motion [2].

Thirdly, classification is performed in the 7D scene event feature space using a Gaussian Mixture Model (GMM). The number of scene event classes K_e captured in the videos is determined by automatic model order selection based on Bayesian Information Criterion (BIC) [21]. The learned GMM is used to classify each detected event into one of the K_e event classes. Finally, the behaviour pattern captured in the n-th

[2] Similar to the Motion History Image (MHI) introduced by Bobick and Davis (see [2]), PCH implicitly represents the direction of movement. First order moments based on PCH value distribution within the bounding box is thus capable of measuring the direction of movement quantitively.

video segment $\mathbf{V}_n$ is represented as a feature vector $\mathbf{P}_n$, given as

$$\mathbf{P}_n = [\mathbf{p}_{n1}, \ldots, \mathbf{p}_{nt}, \ldots, \mathbf{p}_{nT_n}], \tag{2}$$

where T_n is the length of the n-th video segment and the t-th element of $\mathbf{P}_n$ is a K_e dimensional variable:

$$\mathbf{p}_{nt} = [p_{nt}^1, ..., p_{nt}^k, ..., p_{nt}^{K_e}]. \tag{3}$$

$\mathbf{p}_{nt}$ represents the behaviour captured by the t-th image frame of $\mathbf{V}_n$ where p_{nt}^k is the posterior probability that an event of the k-th event class has occurred in the frame given the learned GMM. If an event of the k-th class is detected in the t-th image frame of $\mathbf{V}_n$, we have $0 < p_{nt}^k \leq 1$; otherwise, we have $p_{nt}^k = 0$.

In our approach, a behaviour pattern is represented as a sequence of semantically meaningful scene events. Instead of using low level image features such as location, shape, and motion directly for behaviour representation (e.g. Eqn. (1)), we represent a behaviour pattern using the probabilities of different classes of event occurring in each frame. Consequently, the behaviour representation is compact and concise. This is critical for a model-based behaviour profiling approach because model construction based upon concise representation is more likely to be computationally tractable for modelling complex behaviours. It is worth pointing out that different types of behaviour patterns can be distinguished by either the classes of events they are composed of, or the temporal orders of the event occurrences. For instance, behaviour patterns A and B are deemed as being different if 1) A is composed of events of classes a, b, and d, while B is composed of events of classes a, c and e; or 2) Both A and B are composed of events of classes a, c and d; however, in A, event (class) a is mostly followed by c, while in B, event (class) a is more likely followed by d.

4 Incremental and Adaptive Behaviour Modelling

An outline of our incremental behaviour learning algorithm is shown in Fig. 1 and each step of the algorithm is explained in details as follows.

Model initialisation (Section 4.1): Constructing an initial behaviour model using mixture of DBNs given a small bootstrapping training set;
for *any newly observed behaviour pattern* $\mathbf{P}_{new}$ **do**
Abnormality detection (Section 4.2): Detecting whether $\mathbf{P}_{new}$ is abnormal using both a normal behaviour model $\mathbf{M}_n$ and an approximate abnormal behaviour model $\mathbf{M}_a$ based on Likelihood Ratio Test (LRT);
Incremental model parameter learning and model structure adaptation (Section 4.3): Updating the parameters of $\mathbf{M}_n$ and $\mathbf{M}_a$ using $\mathbf{P}_{new}$ and performing model adaptation when visual context changes are detected;
end

Figure 1: Outline of our incremental and adaptive behaviour modelling algorithm.

4.1 Model Initialisation

4.1.1 Behaviour Affinity Matrix

Consider a small bootstrapping dataset $\mathbf{D}$ consisting of N feature vectors:

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

where $\mathbf{P}_n$ represents the behaviour pattern captured by the n-th video segment $\mathbf{V}_n$ (see Eqn. (2)). The problem to be addressed is to discover the natural grouping of the training behaviour patterns upon which an initial behaviour model can be built. We treat this as an unsupervised temporal string clustering problem. There are two aspects that make this problem challenging: (1) Different feature vectors, as multivariate strings, can be of different lengths because each behaviour pattern has a variable temporal duration. Conventional clustering algorithms such as K-means and mixture models require that each data sample is represented as a fixed length feature vector. These algorithms thus cannot be applied to our problem. (2) A definition of a distance/affinity metric among these temporal strings of variable lengths is nontrivial [17]. Measuring affinity between feature vectors of variable length often involves dynamic time warping [12].

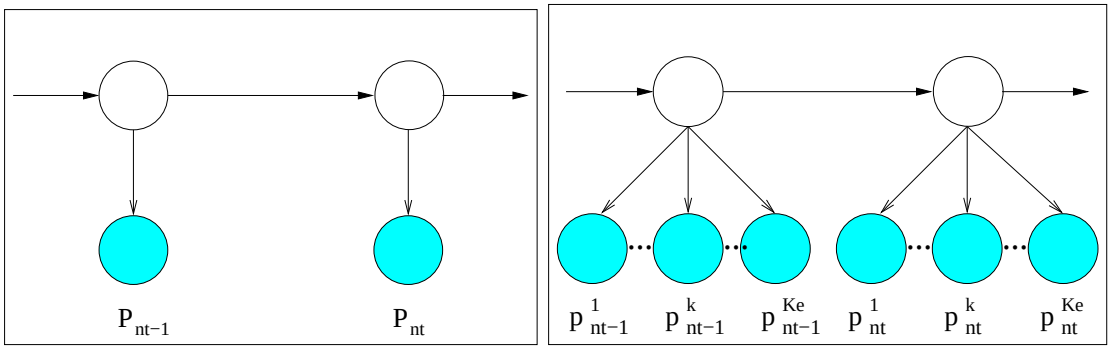
Dynamic Bayesian Networks (DBNs)[3] provide a solution for overcoming the above-mentioned difficulties. More specifically, each behaviour pattern in the training set $\mathbf{D}$ is modelled using a DBN. To measure the affinity between two behaviour patterns represented as $\mathbf{P}_i$ and $\mathbf{P}_j$, two DBNs denoted as $\mathbf{B}_i$ and $\mathbf{B}_j$ are trained on $\mathbf{P}_i$ and $\mathbf{P}_j$ respectively using the Expectation-Maximisation (EM) algorithm [7, 9]. The affinity

[3]Dynamic Bayesian Networks (DBNs), or Dynamic Probabilistic Networks, are graphical models suitable for temporal or time-series data [9]. Examples of DBNs include Kalman Filters, Hidden Markov Models (HMMs), and variations of HMMs such as Coupled Hidden Markov Models (CHMMs).

between $\mathbf{P}_i$ and $\mathbf{P}_j$ is then computed as:

$$S_{ij} = \frac{1}{2}\left\{\frac{1}{T_j}\log P(\mathbf{P}_j|\mathbf{B}_i) + \frac{1}{T_i}\log P(\mathbf{P}_i|\mathbf{B}_j)\right\}, \tag{5}$$

where $P(\mathbf{P}_j|\mathbf{B}_i)$ is the likelihood of observing $\mathbf{P}_j$ given $\mathbf{B}_i$, and T_i and T_j are the lengths of $\mathbf{P}_i$ and $\mathbf{P}_j$ respectively[4].



(a) HMM (b) MOHMM

Figure 2: Modelling a behaviour pattern $\mathbf{P}_n = \{\mathbf{p}_{n1}, \ldots, \mathbf{p}_{nt}, \ldots, \mathbf{p}_{nT_n}\}$ where $\mathbf{p}_{nt} = \{p^1_{nt}, ..., p^k_{nt}, ..., p^{K_e}_{nt}\}$ using a HMM and a MOHMM. Observation nodes are shown as shaded circles and hidden nodes as clear circles.

DBNs of different topologies can be employed. A straightforward choice would be a Hidden Markov Model (HMM) (Fig. 2(a)). In this HMM, the observation variable at each time instance corresponds to $\mathbf{p}_{nt}$ (Eqn. (3)), which represents the behaviour captured in the t-th frame of the n-th behaviour pattern. The observation variable is of dimension K_e, i.e. the number of event classes. The conditional probability distributions (CPD) of $\mathbf{p}_{nt}$ is assumed to be Gaussian for each of the N_s states of its parent node. However, a drawback of using a HMM is that too many parameters are needed to describe the model when the observation variables are of high dimension. This makes a HMM vulnerable to overfitting therefore generalising poorly to unseen data. It is especially true in our case because a HMM needs to be learned for every single behaviour pattern in the training dataset which could be short in duration. To solve this problem, we employ a Multi-Observation Hidden Markov Model (MOHMM) [10] shown in Fig. 2(b). Compared to a HMM, the observational space is factorised by assuming that each observed feature (p^k_{nt}) is independent of each other. Consequently, the number of parameters for describing a MOHMM is much lower than that for a HMM ($2K_eN_s + {N_s}^2 - 1$ for a MOHMM and $({K_e}^2 + 3K_e)N_s/2 + {N_s}^2 - 1$ for a HMM). In this paper, N_s, the number of hidden states for each hidden variables in the MOHMM, is set to K_e, i.e., the number of event classes. This is reasonable because the value of N_s should reflect the complexity of a behaviour pattern, so

[4]Note that there are other ways to compute the affinity between two sequences modelled using DBNs [19, 20]. However, we found through our experiments that different affinity measures make little difference for our behaviour modelling task.

should the value of K_e.

Now we have constructed an $N \times N$ affinity matrix $\mathbf{S} = [S_{ij}]$ where S_{ij} is computed using Eqn. (5) for the bootstrapping dataset $\mathbf{D}$. As we explained earlier, the aim of constructing a behaviour affinity matrix is to cluster the N behaviour patterns in $\mathbf{D}$. Let us denote the number of clusters discovered using the affinity matrix as K_c. Once the clustering is done, each cluster would correspond to one bootstrapping behaviour class and each of the N behaviour patterns in the bootstrapping dataset $\mathbf{D}$ can be labelled as one of the K_c behaviour classes. Data clustering using an affinity matrix can be performed using a spectral clustering algorithm. In this paper, the multiclass spectral clustering algorithm proposed by Yu and Shi [31] is employed.

4.1.2 Bootstrapping Behaviour Models

Now each of N behaviour patterns in the bootstrapping dataset $\mathbf{D}$ are labelled as one of the K_c behaviour classes. Bootstrapping behaviour models are then constructed as mixtures of MOHMMs based on the clustering result. First, we model the k-th ($1 \leq k \leq K_c$) behaviour class using a MOHMM denoted as $\mathbf{B}_k$. The parameters of $\mathbf{B}_k$, denoted as $\theta_{\mathbf{B}k}$, are estimated using all the patterns that belong to the k-th class in $\mathbf{D}$. Second, each of the K_c behaviour classes is labelled as being either normal and abnormal according to the number of patterns within the class. More specifically, the K_c classes are ordered in descending order according to the number of class members and the first K_n classes are labelled as being normal. K_n is computed as:

$$K_n = \arg\min_{K_n} \left(\sum_{k=1}^{K_n} \frac{N_k}{N} > Q \right), \tag{6}$$

where N_k is the number of members in the k-th class and Q corresponds to the minimum portion of the behaviour patterns in the bootstrapping training set to be deemed as being normal. We thus have $0 < Q \leq 1$. Third, a bootstrapping normal behaviour model $\mathbf{M}_n$ is constructed as a mixture of K_n MOHMMs for the K_n normal behaviour classes. An approximate abnormal model $\mathbf{M}_a$ is also constructed using the $K_a = K_c - K_n$ abnormal behaviour classes in the bootstrapping dataset. Let $\mathbf{P}$ be a sample of $\mathbf{M}_n$. The probability density function (pdf) of $\mathbf{M}_n$ can be written as:

$$P(\mathbf{P}|\mathbf{M}_n) = \sum_{k=1}^{K_n} w_{nk} P(\mathbf{P}|\mathbf{B}_{nk}), \tag{7}$$

where w_{nk} is the mixing probability/weight of the k-th mixture component with $\sum_{k=1}^{K_n} w_{nk} = 1$ and $\mathbf{B}_{nk}$ is the k-th MOHMM corresponding to the k-th normal behaviour class. Similarly for $\mathbf{M}_a$, we have:

$$P(\mathbf{P}|\mathbf{M}_a) = \sum_{k=1}^{K_a} w_{ak} P(\mathbf{P}|\mathbf{B}_{ak}). \tag{8}$$

The parameters of the normal behaviour model $\mathbf{M}_n$ are

$$\theta_{\mathbf{M}_n} = \left\{ K_n, w_{n1}, ..., w_{ni}, ..., w_{nK_n}, \theta_{\mathbf{B_{n1}}}, ..., \theta_{\mathbf{B_{ni}}}, ..., \theta_{\mathbf{B_{nK_n}}} \right\}.$$

Similarly, the parameters of the approximate abnormal behaviour model $\mathbf{M}_a$ are

$$\theta_{\mathbf{M}_a} = \left\{ K_a, w_{a1}, ..., w_{aj}, ..., w_{aK_a}, \theta_{\mathbf{B_{a1}}}, ..., \theta_{\mathbf{B_{aj}}}, ..., \theta_{\mathbf{B_{aK_a}}} \right\}.$$

In model initialisation, given a very small bootstrapping training set with poor statistics, we essentially perform abnormal behaviour detection for the initial training set simply according to the rarity of behaviours as there is no other meaningful discriminative information available in the small bootstrapping training set. For further abnormality detection as more data becomes available online, we formulate a more elaborate approach. The approach takes into consideration the generalisation capability of mixture models learned using an incremental Expectation-Maximalisation (EM) algorithm.

4.2 Online Abnormality Detection

Beyond the initial bootstrapping step, we address the problem of abnormality detection using the Likelihood Ratio Test (LRT) method [16] to achieve robustness in distinguishing abnormal behaviours from normal ones. Specifically, given a newly observed behaviour pattern represented as $\mathbf{P}_{new}$ and the current models $\mathbf{M}_n$ and $\mathbf{M}_a$, abnormality detection is performed based on a hypothesis test between

$$\begin{aligned} H_0 &: \mathbf{P}_{new} \text{ is from the hypothesised model } \mathbf{M}_n, \text{ i.e. normal} \\ H_i &: \mathbf{P}_{new} \text{ is from the a model other than } \mathbf{M}_n, \text{ i.e. abnormal} \end{aligned}$$

H_0 is called the null hypothesis while H_i is called the alternative hypothesis. $\mathbf{P}_{new}$ is accepted as a normal behaviour pattern if H_o hits; otherwise $\mathbf{P}_{new}$ is detected as being abnormal. The most popular solution to

this hypothesis test is LRT given by

$$\Lambda(\mathbf{P}_{new}) = \frac{P(\mathbf{P}_{new}; H_0)}{P(\mathbf{P}_{new}; H_i)} \begin{cases} \geq Th_\Lambda & \text{accept } H_0 \\ < Th_\Lambda & \text{accept } H_i \end{cases} \tag{9}$$

where $P(\mathbf{P}_{new}; H_0)$ and $P(\mathbf{P}_{new}; H_i)$ are the likelihood functions of the hypotheses H_0 and H_i respectively and Th_Λ is called a rejection threshold.

The key issue in LRT is how to accurately construct the alternative model which in our case is the abnormal behaviour model. As we pointed out earlier in this paper, an abnormal behaviour model is much more difficult if even possible to construct accurately compared with a normal one because abnormal behaviours are rare and unpredictable (e.g. there could be infinite number of ways of being abnormal). To overcome this problem, $\mathbf{M}_a$, constructed as a mixture of MOHMMs using the abnormal behaviour patterns observed so far, is employed to approximate the abnormal behaviour model. The LRT is then rewritten as:

$$\Lambda(\mathbf{P}_{new}) = \frac{P(\mathbf{P}_{new}|\mathbf{M}_n)}{P(\mathbf{P}_{new}|\mathbf{M}_a)} \begin{cases} \geq Th_\Lambda & \text{accept } H_0 \\ < Th_\Lambda & \text{accept } H_i \end{cases} \tag{10}$$

where $P(\mathbf{P}_{new}|\mathbf{M}_n)$ and $P(\mathbf{P}_{new}|\mathbf{M}_a)$ are computed using Equations (7) and (8) respectively.

4.3 Online Incremental Parameter Updating and Model Structure Adaptation

Now given that a newly observed behaviour pattern $\mathbf{P}_{new}$ has been classified as either normal or abnormal, the parameters of $\mathbf{M}_n$ and $\mathbf{M}_a$ are updated as follows:

4.3.1 Updating parameters of the normal behaviour model $\mathbf{M}_n$

If $\mathbf{P}_{new}$ was detected as being normal, $\mathbf{P}_{new}$ is matched with the mixture component of $\mathbf{M}_n$ that has the maximal posterior probability, i.e. the probability that $\mathbf{P}_{new}$ could be generated by the component. The best matched component is denoted as $\mathbf{B}_{ni}$ and the posterior probability for $\mathbf{B}_{ni}$ is computed as:

$$P\left(\mathbf{B}_{ni}|\mathbf{P}_{new}\right) = \frac{w_{ni} P\left(\mathbf{P}_{new}|\mathbf{B}_{ni}\right)}{P\left(\mathbf{P}_{new}|\mathbf{M}_n\right)} \tag{11}$$

where $P\left(\mathbf{P}_{new}|\mathbf{M}_n\right)$ is given by Eqn. (7).

Initialisation:

- set iteration counter $p = 0$;
- set $\theta_{\mathbf{B}_{ni}}^{[0]} = \theta_{\mathbf{B}_{ni}}^{[old]}$, the parameters of $\mathbf{B}_{ni}$ before seeing $\mathbf{P}_{new}$;

while *no convergence* **do**

E Step:

- given $\mathbf{P}_{new}$ and $\theta_{\mathbf{B}_{ni}}^{[p]}$, compute the sufficient statistics of $\mathbf{P}_{new}$, $S_{\mathbf{P}_{new}}^{[p+1]}$ using the forward/backward procedure over $\mathbf{P}_{new}$;
- compute the sufficient statistics for the complete data (i.e. all the behaviour patterns observed so far that belong to $\mathbf{B}_{ni}$) as $S^{[p+1]} = S^{[p]} + S_{\mathbf{P}_{new}}^{[p+1]} - S_{\mathbf{P}_{new}}^{[p]}$;

M Step:

- set $\theta_{\mathbf{B}_{ni}}^{[p+1]}$ to the $\theta_{\mathbf{B}ni}$ that yields the maximum likelihood given $S^{[p+1]}$;
- set $p = p + 1$;

end

Figure 3: An online incremental EM algorithm for updating the parameters of the mixture component of $\mathbf{M}_n$ matched by the newly observed normal pattern $\mathbf{P}_{new}$. Details on the forward/backward procedure and computing sufficient statistics can be found in [13] and [1]. Convergence of the algorithm is reached when $P\left(\mathbf{P}_{new}|\theta_{\mathbf{B}_{ni}}^{[p+1]}\right) - P\left(\mathbf{P}_{new}|\theta_{\mathbf{B}_{ni}}^{[p]}\right) < Th_p$ where Th_p is a threshold.

The parameters of $\mathbf{B}_{ni}$ (denoted as $\theta_{\mathbf{B}ni}$) are updated using an incremental EM algorithm. The general principle of incremental EM was originally introduced in [15]. Here we formulate an algorithm for online incremental learning of $\mathbf{B}_{ni}$ given the detected normal behaviour pattern $\mathbf{P}_{new}$, as outlined in Fig. 3. It has been proved that stable convergence is guaranteed for such an incremental EM algorithm (see [15]). Both a conventional (batch) EM algorithm [7] and an incremental one have the identical M step. The difference lies in the way in which sufficient statistics $S^{[p+1]}$ is computed in the E step. Specifically, the batch EM algorithm computes sufficient statistics on the whole dataset at each iteration. In contrast, the incremental algorithm updates $S^{[p+1]}$ incrementally using a subset of the dataset, in our case a single data item. As pointed out in [15], the main rationale behind an incremental EM algorithm in an off-line learning set-up is that faster convergence can be achieved because the information from the new data contributes to the parameter estimation more quickly than the batch EM algorithm. In other words, the parameter estimation efficiency is the main concern. In our on-line learning case, the main motivation for adopting the incremental EM algorithm is that data only become available sequentially, i.e. the whole data set never exist. Note that the E step of the algorithm only looks at a single data item $\mathbf{P}_{new}$. Furthermore, both the E step and the M step take constant time, regardless of the number of behaviour patterns observed so far. These characteristics make the algorithm computational and memory efficient, and therefore suitable for real-time applications.

After $\theta_{\mathbf{B}ni}$ are updated, the weight of the matched mixture component is updated as:

$$w_{ni}^{[new]} = w_{ni}^{[old]} + \alpha\left(1 - w_{ni}^{[old]}\right) \tag{12}$$

where $w_{ni}^{[old]}$ is the weight before seeing $\mathbf{P}_{new}$ and α with $0 \leq \alpha \leq 1$ is a learning rate. The weights for the components of $\theta_{\mathbf{M_n}}$ are then renormalised so that they satisfy $\sum_{k=1}^{K_n} w_{nk} = 1$. Consequently, the weight of the matched component has been increased whilst the weights for the other components of $\mathbf{M}_n$ have been decreased. The learning rate α will determine the speed at which the weights are updated.

4.3.2 Updating parameters of the approximate abnormal behaviour model $\mathbf{M}_a$

If $\mathbf{P}_{new}$ was detected as being abnormal, we need to establish whether $\mathbf{P}_{new}$ belongs to one of the existing abnormal behaviour classes. Specifically, the best matched component of $\mathbf{M_a}$ is determined using posterior probability as above and denoted as $\mathbf{B}_{aj}$. The similarity/distance between $\mathbf{P}_{new}$ and $\mathbf{B}_{aj}$ is then measured as the normalised log-likelihood of observing $\mathbf{P}_{new}$ given $\mathbf{B}_{aj}$:

$$d\left(\mathbf{P}_{new}, \mathbf{B}_{aj}\right) = \frac{1}{T_{\mathbf{P}_{new}}} \log P\left(\mathbf{P}_{new} | \theta_{\mathbf{B_{aj}}}\right)$$

where $T_{\mathbf{P}_{new}}$ is the length of $\mathbf{P}_{new}$ (total number of frames). If

$$d\left(\mathbf{P}_{new}, \mathbf{B}_{aj}\right) > Th_d, \tag{13}$$

$\mathbf{P}_{new}$ is determined as belonging to the best matched mixture component $\mathbf{B}_{aj}$ and both $\theta_{\mathbf{B_{aj}}}$ and w_{aj} are updated in the same way as $\theta_{\mathbf{B_{ni}}}$ and w_{ni} (see Fig. 3 and Eqn. (12)). Otherwise (i.e. $\mathbf{P}_{new}$ was detected as being abnormal and Eqn. (13) was not satisfied), a new component corresponding to a new abnormal behaviour class is added to $\mathbf{M_a}$ whose parameters are estimated using $\mathbf{P}_{new}$ and its weight is set to the smallest weight of the existing components of $\mathbf{M_a}$. Weight renormalisation is then performed to ensure that $\sum_{k=1}^{K_a} w_{ak} = 1$.

4.3.3 Model structure adaptation via mixture component trimming

Model adaptation is achieved through mixture component trimming. Unlike model parameters updating which is carried out whenever new data are available, model adaptation is performed only when changes

in visual context are detected. More specifically, when a normal behaviour class, represented as one of the mixture component of $\mathbf{M_n}$, has not been supported by any new observations, its weight would be decreased gradually following the model parameter updating procedure above. When its weight is smaller than a threshold Th_{w1}, it can be assumed that this behaviour class has become abnormal and the corresponding mixture component would be regrouped into the approximate abnormal behaviour model $\mathbf{M_a}$. In the meantime, when an abnormal behaviour class is matched repeatedly by new observations with Eqn. (13) being satisfied, the weight of the corresponding mixture component will increase gradually. When its weight becomes greater than a threshold Th_{w2}, it becomes normal and the corresponding mixture component would be regrouped into the normal behaviour mixture $\mathbf{M_n}$. The abnormal classes whose weights are smaller than Th_{w1} would then be discarded in order to impose a limit on the total number of abnormal behaviour classes that a model is designed to cope. This is because that in a realistic situation, there are always limited computational resources available while the total number of abnormal behaviour classes can potentially be unlimited. After component trimming, the mixture weights of $\mathbf{M_n}$ and $\mathbf{M_a}$ are renormalised. Component trimming makes our behaviour model adaptive to changes in visual context. Consequently the numbers of mixture components/behaviour classes for both the normal and abnormal models can vary over time.

4.3.4 Discussions

A number of issues deserve further discussions:

1. Two mixtures of MOHMMs, $\mathbf{M_n}$ and $\mathbf{M_a}$ are initialised and updated for modelling normal and abnormal behaviours respectively. Having two models for normal and abnormal behaviours rather than modelling normal behaviours alone is necessary and critical in our approach. This is because (a) it makes robust abnormality detection possible based on Likelihood Ratio Test (LRT), which is advantageous over the conventional Maximum Likelihood (ML) method as demonstrated by our experiments to be presented in Section 5; and (b) it makes it possible that our behaviour model can adapt to changes in visual context. Note that it could be impossible to build an exact model for abnormal behaviours in most cases because they are rare and unpredictable. However, it is possible to build an approximate one using a mixture model given the abnormal patterns detected so far (i.e. $\mathbf{M_a}$). In particular, as a mixture of MOHMMs, $\mathbf{M_a}$ is a generative model which is capable of generalising from a limited number of samples. Based on the observed abnormal behaviour patterns, our approximate abnormal model aims to capture the randomness and unexpectedness of those unseen abnormal behaviour

patterns therefore providing a good alternative model for $\mathbf{M_n}$ in LRT.

2. As emphasised above, $\mathbf{M_a}$ differs from $\mathbf{M_n}$ in that $\mathbf{M_a}$ is an approximate model. As a result, the parameters and structures of $\mathbf{M_a}$ and $\mathbf{M_n}$ are updated differently. In particular, when $\mathbf{P}_{new}$ is detected as being normal, it must be classified to one of the existing normal behaviour classes and the corresponding mixture component of $\mathbf{M_n}$ is to be updated. Nevertheless, when $\mathbf{P}_{new}$ is detected as abnormal, we update the best matched mixture component only when we have sufficient confidence (i.e. Eqn. (13) is satisfied). Otherwise, a new component will be added to reflect the fact that an unseen abnormal behaviour classes is observed. Again, this difference is caused by the rarity and unpredictability of abnormal behaviours.

3. Although It has been shown by Neal and Hinton [15] that stable convergence is guaranteed for each mixture component of $\mathbf{M_n}$ and $\mathbf{M_a}$, no theoretical proof can be given for the convergence of our behaviour model as a whole. In particular, our behaviour model is based on mixture models with changing component numbers. An incremental EM algorithm thus cannot be implemented directly to the two mixture models (i.e. estimating the parameters of each mixture component and the mixture weights simultaneously). In our solution, the mixture weight updating (Eqn. (12)) and component trimming parts of the algorithm are based on online approximations and therefore are slightly ad-hoc. Nevertheless, experimental results to be presented in Section 5 demonstrate empirically that our model converges to a satisfactory solution.

4. Although a discrete event based behaviour representation is adopted in our approach, other behaviour representations can also be used in our approach provided that a behaviour pattern can be represented as a feature vector.

5 Experiments

Dataset and behaviour representation — A CCTV camera was mounted on the ceiling of an office entry corridor, monitoring people entering and leaving an office area (see Fig. 4). The office area is secured by an entry door which can only be opened by scanning an entry card on the wall next to the door (see the middle frame of Fig. 4(b)). Two side-doors were located at the right hand side of the corridor. People from both inside and outside the office area have access to these two side-doors. Typical behaviours occurring in the scene would be people entering or leaving either the office area or the side-doors, and walking towards

Figure 4: Examples of behaviour patterns captured in a corridor entrance/exit scene. (a)–(f) show image frames of commonly occurred behaviour patterns belonging to the 6 behaviour classes listed in Table 1. (g)&(h) show examples of rare behaviour patterns captured in the scene. (g): One person entered the office following another person without using an entry card. (h): Two people left the corridor after a failed attempt to enter the door. The four classes of events detected automatically, 'entering/leaving the near end of the corridor', 'entering/leaving the entry-door', 'entering/leaving the side-doors', and 'in corridor with the entry door closed', are highlighted in the image frames using bounding boxes in blue, cyan, green and red respectively.

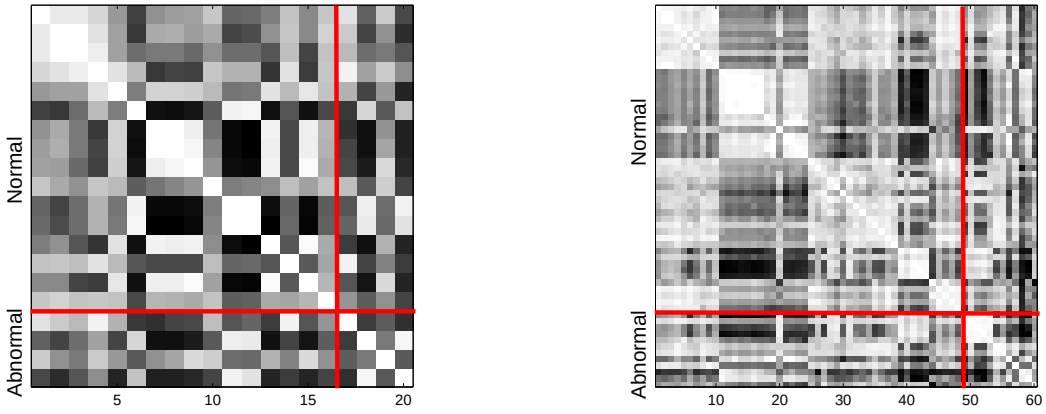
the camera. Most captured behaviour patterns involved 1-2 people. Each behaviour pattern would normally last a few seconds. For this experiment, a dataset was collected over 5 different days consisting of 6 hours of video totalling 432000 frames captured at 20Hz with 320×240 pixels per frame. This dataset was then automatically segmented into sections separated by any motionless intervals lasting for more than 30 frames. This resulted in 142 video segments of actual behaviour pattern instances. Each segment has on average 121 frames with the shortest 42 and longest 394. Examples of behaviour patterns captured in the 6 hour video are shown in Fig. 4.

Discrete events were detected and classified using automatic model order selection in clustering, resulting in four classes of events corresponding to the common constituents of all behaviours in this scene: 'entering/leaving the near end of the corridor', 'entering/leaving the entry-door', 'entering/leaving the side-doors', and 'in corridor with the entry door closed'. Examples of detected events are shown in Fig. 4 using colour-coded bounding boxes. It is noted that due to the narrow view nature of the scene, differences be-

tween the four common events are rather subtle and can be mis-identified based on local information (space and time) alone, resulting in errors in event detection. The fact that these events are also common constituents to different behaviour patterns means that local events treated in isolation hold little discriminative information for behaviour profiling. All experiments described below were conducted on an 3GHz platform.

C1	From the office area to the near end of the corridor
C2	From the near end of the corridor to the office area
C3	From the office area to the side-doors
C4	From the side-doors to the office area
C5	From the near end of the corridor to the side-doors
C6	From the side-doors to the near end of the corridor

Table 1: Six classes of commonly occurred behaviour patterns in the corridor scene.



(a) An ordered affinity matrix with $N = 20$

(b) An ordered affinity matrix with $N = 60$

Figure 5: Examples of spectral clustering for model initialisation. The spectral clustering results were illustrated using the ordered affinity matrices of the bootstrapping datasets. The affinity matrices were plotted such that "white" corresponds to the highest affinity value while "black" represents the lowest value. They were ordered according to the clustering results so that data points belonging to the same cluster formed a bright block along the diagonals of the matrices. The discovered clusters were then re-organised in a descending order in size from top-right to bottom-right along the matrix diagonal lines. The top K_n clusters corresponded to normal behaviour classes and were used to initialised the normal behaviour model $\mathbf{M}_n$, while the remaining clusters were used in building the abnormal model $\mathbf{M}_a$. The values of K_n, obtained using Eqn. (6), were 5 and 7 for (a) and (b) respectively.

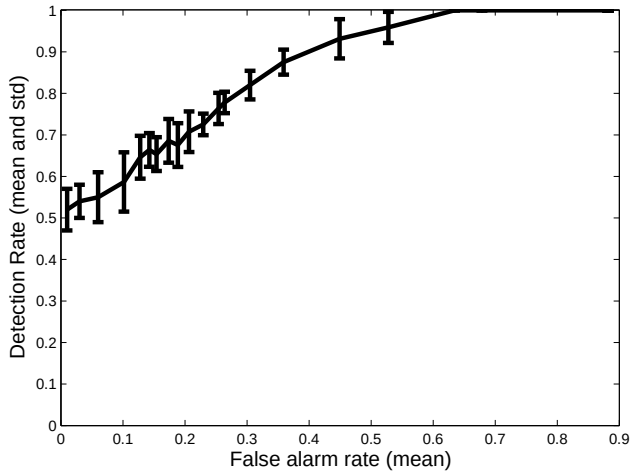
Model initialisation — A bootstrapping dataset consisting of N video segments was randomly selected from the overall 142 segments for model initialisation. N was set to either 20 or 60 in our experiments. The remaining segments ($142 - N$ in total) were used for incremental and adaptive model learning and abnormality detection later. This model initialisation exercise was repeated 20 times each for $N = 20$ and $N = 60$ and in each trial a different model was initialised using a different random dataset. This is in order to test the effect of the size of initial training set and avoid any bias in the abnormality detection results. The number of initial behaviour classes to be established through model bootstrapping K_c was set to 10 in

our experiments. Q (see Eqn. (6)) was set to 0.7 and on average the numbers of normal behaviour classes determined using Eqn. (6) were 5 when $N = 20$ and 6 when $N = 60$ over 20 trials. Fig. 5 shows examples of the model initialisation process. It is noted that given a small random initial training set ($N = 20$), mixture components in $\mathbf{M}_n$ often corresponded to only part of the 6 commonly occurred behaviour classes (Table 1). In this case, the rest of the 6 common behaviour classes were either labelled as being abnormal behaviour classes and modelled by $\mathbf{M}_a$ or did not form any cluster due to their rare occurrence in the small bootstrapping dataset. It is also observed that given a larger initial training set $N = 60$, all 6 commonly occurred behaviour classes can find their corresponding components in $\mathbf{M}_n$ in most trials (15 out of 20). It can be seen in Fig. 5 that there were fair amount of similarities among different clusters even between the normal and abnormal ones, as indicated by those bright cells off the affinity matrix diagonal blocks in Fig. 5(a)&(b). This was because (1) different behaviour classes shared the same events as constituents and often differed only in temporal orders of those events, and (2) there were considerable amount of noise/errors in event detection.

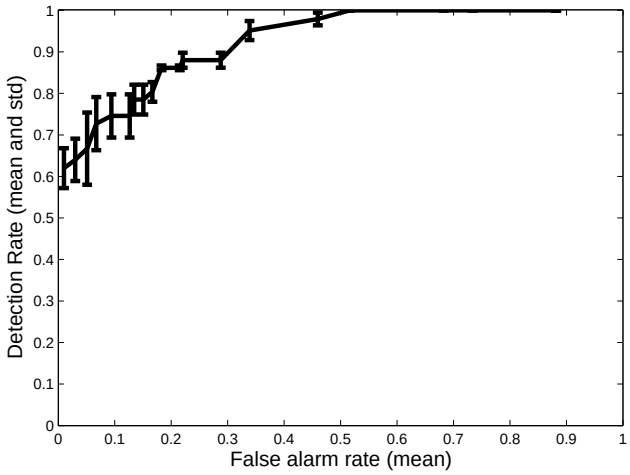
Online abnormality detection and incremental learning — After model initialisation, online abnormality detection and incremental model updating were performed. Parameters for incremental and adaptive model updating were set as: learning rate $\alpha = 0.1$ (Eqn. (12)), convergence threshold for parameter updating of matched mixture components using incremental EM $Th_p = 0.0001$ (see caption of Fig. 3), threshold for matching abnormal behaviour classes $Th_d = -0.5$ (Eqn. (13)), and thresholds for mixture component trimming: $Th_{w1} = 0.05$ and $Th_{w2} = 0.25$. It was observed in our experiments that the our algorithm were not sensitive to these parameters.

As new behaviour patterns were being presented to the model for abnormality detection and incremental learning, the numbers of mixture components in $\mathbf{M}_n$ and $\mathbf{M}_a$, denoted as K_n and K_a respectively, increased before stabilising around constant numbers. On average, K_n and K_a converged to 8 and 12 respectively in our experiments. The convergence took place after an average of 35 new behaviour patterns were observed when $N = 20$. The number of new observations needed for model convergence was down to 23 when $N = 60$.

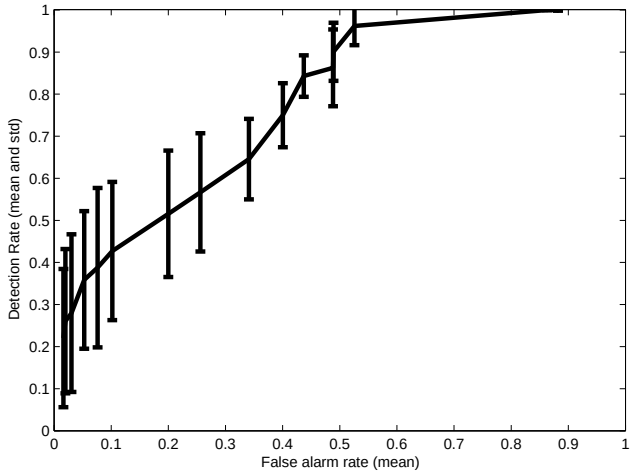
To evaluate the performance of the learned models on abnormality detection, ground truth was extracted by labelling the testing/incremental-learning datasets such that each behaviour pattern was labelled as being normal if there were similar patterns that have been seen before and abnormal otherwise. The performance of the models was measured using the detection rate and false alarm rate in abnormality detection which are functions of Th_Λ (see Eqn.(9)). Varying Th_Λ gave us a ROC curve in each trial. The ROC curves averaged



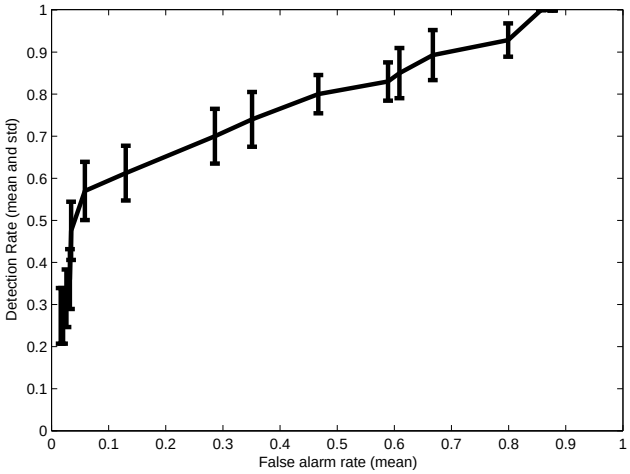
(a) $N = 20$, incremental



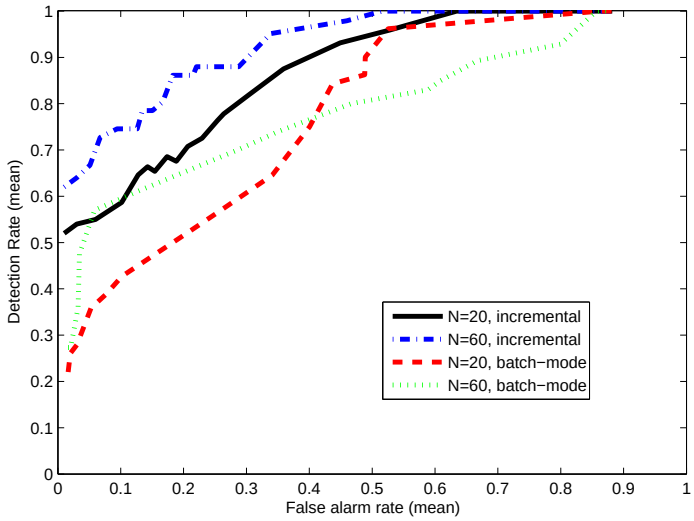
(b) $N = 60$, incremental



(c) $N = 20$, batch-mode



(d) $N = 60$, batch-mode



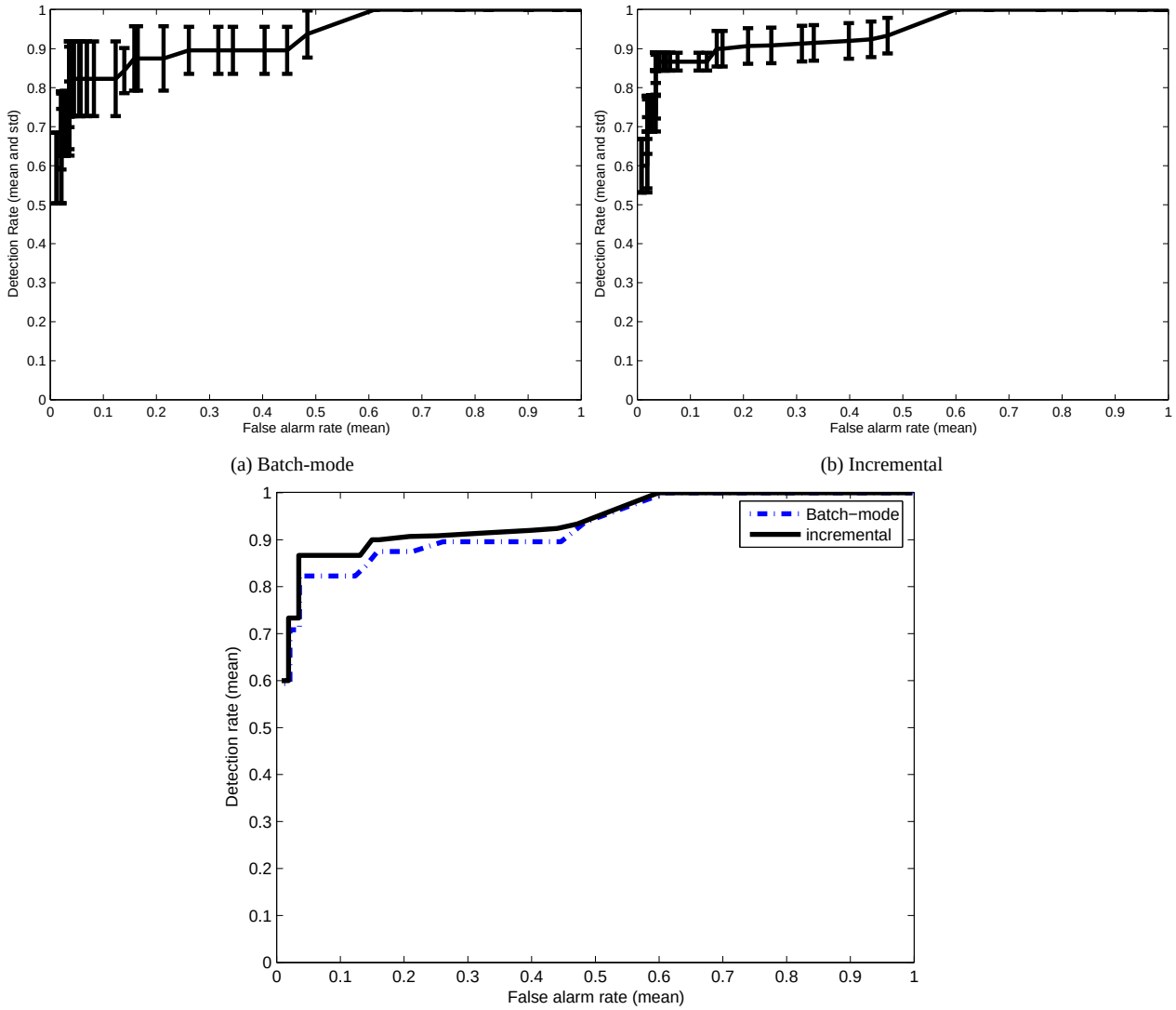
(e) Mean ROC curves in one plot

Figure 6: Comparing the performance of abnormality detection for models learned using different experimental settings. The performance was measured using detection rate and false alarm rate plotted in ROC curves. (a)–(d) show the mean and ± 1 standard deviation of the ROC curves obtained over 20 trials under different experimental settings. The mean ROC curves were also shown in a single plot in (e) to better illustrate the differences.

over 20 trials for $N = 20$ and $N = 60$ are shown in Fig. 6(a) and (b) respectively. The standard deviation of the ROC curves across different trials are also depicted in Fig. 6(a) and (b) to demonstrate the effects of the bootstrapping dataset selection on the model performance. Comparing Fig. 6 (a) with (b), it is clear that better performance was obtained using larger initial training sets. This is because the models were initialised poorly using small bootstrapping datasets. Poor initialisation is also the reason why the models initialised using smaller datasets needed more data to converge. Nevertheless, it is observed that even with small initial training sets, our models were able to discover all the normal behaviour classes and reach convergence when sufficient observations became available after model initialisation. In particular, when behaviour patterns belonging to one of the 6 typical behaviour classes in Table 1 were observed repeatedly, a new mixture component would be added to $\mathbf{M}_n$ to represent the class if it was not already there after model initialisation. The experimental results thus demonstrate that our incremental learning model can cope with changes of visual context (in this case, abnormal behaviour patterns becoming normal).

Comparative evaluation against batch-mode offline learning — We compared the performance of our incremental and adaptive behaviour modelling algorithm with the batch-mode algorithm proposed in [27]. This batch-mode algorithm was chosen for comparison in our experiments because it uses an identical feature extraction and behaviour representation method. The difference in the results would thus be caused solely by the different ways of model learning adopted by the two algorithms. Specifically, in the batch-mode algorithm, only a normal behaviour model is built using a training data set. A newly observed behaviour pattern is detected as being abnormal if the probability of observing the pattern given the model is below a threshold. As a batch-mode algorithm, no model updating is performed during testing. Therefore, there are two key differences between our algorithm and the bath-mode one: (1) on model construction: the former is incremental and adaptive while the latter is batch-mode, and (2) on abnormality detection: the former adopts LRT while the latter uses simple thresholding.

Two experiments were carried out. In the first experiment, the same bootstrapping datasets used in our online incremental algorithms above were used to construct the batch-mode models. The averaged ROC curves obtained using models trained in batch mode are shown in Fig. 6(c) and (d) for training datasets of sizes 20 and 60 respectively. Comparing Fig. 6(a)&(b) with Fig. 6(c)&(d), it is evident that the incrementally learned models outperform those learned in batch mode. The performance of the batch-mode behaviour models with $N = 20$ was especially poor (see Fig. 6(c)). This was mainly due to the fact that these models were learned poorly using the small training sets and, without model updating, cannot cope with the changes of visual context. It is also noted that the ROC curves obtained using our incremental models



(a) Batch-mode

(b) Incremental

(c) Mean ROC curves in one plot

Figure 7: Comparing our online incremental learning algorithm with a batch-mode algorithm given the same amount of data for model construction. (a) and (b) show the mean and ± 1 standard deviation of the ROC curves obtained over 20 trials using the batch-mode algorithm and our online incremental algorithm respectively. The mean ROC curves are plotted in a single figure in (c).

exhibited smaller variations across different trials indicated by the smaller standard deviations shown in Fig. 6(a)&(b). This again can be explained by the model adaptation feature of the incremental algorithm which makes the model less sensitive to the choice of initial training data.

As mentioned earlier, the two algorithms differs in both the way they construct the models and the way abnormality is detected. Both differences contributed to the superior results obtained using our incremental and adaptive algorithm. It is easier and more intuitive to understand the advantage brought by employing an online incremental model construction procedure in the above experiment. In particular, an incremental and adaptive model keeps updating itself whenever a new observation is captured. On the contrary, a batch-mode model remains fixed once the training has been done. The former one thus makes use of more data available

for model construction than the latter one. In the second experiment, we examine the performance of the two algorithms when they are presented with the same amount of data for model building.

In this experiment, an online incremental model was initialised using 20 randomly selected behaviour patterns. After being incrementally learned using another 80 patterns, it was tested for the remaining 42 patterns without model updating. For the batch mode algorithm, the same dataset consisting of a total of 100 samples was used for training. Again, the experiment was repeated for 20 trials using different datasets for training. The comparative results are presented in Fig. 7. It can be seen from Fig. 7 that our online incremental algorithm outperforms the batch-mode algorithm even using the same amount of data for model construction. Comparing Fig. 7(a) with (b), it can also be seen that the results of our online incremental algorithm exhibited less variations across different trials. This again shows that our algorithm is less sensitive to the choice of training datasets than the batch-mode algorithm. The results obtained in the second experiment demonstrate that a behaviour model can be built more efficiently and effectively even for off-line abnormality detection thanks to the incremental and adaptive learning feature of our algorithm.

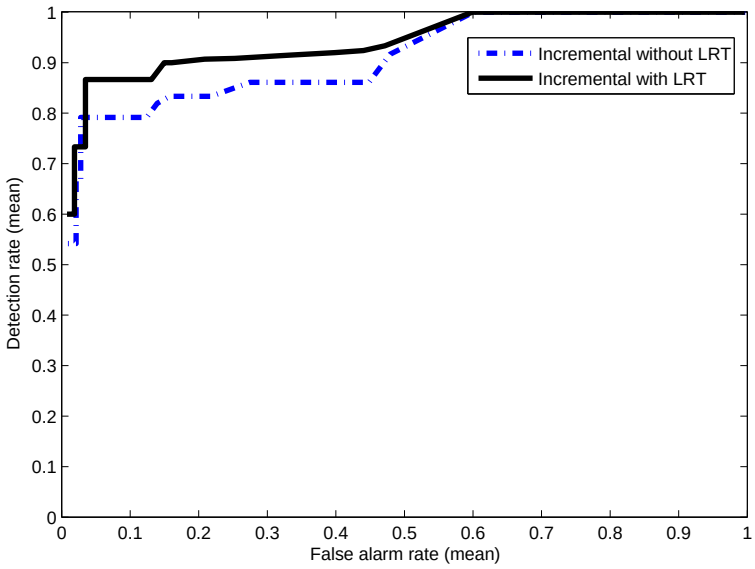


Figure 8: The performance of our incremental algorithm with and without using LRT for abnormality detection.

Comparative evaluation of the effectiveness of LRT — The following experiment was carried out to highlight the importance of using LRT in our algorithm. Online behaviour models were constructed using the identical datasets and following the same procedure as the second experiment described above except that different ways of abnormality detection were adopted. Specifically, instead of using LRT, abnormality detection was performed by thresholding the probability of observing a pattern given the normal model $\mathbf{M}_n$. The comparative results are presented in Fig. 8. The results show that better abnormality detection

performance can be achieved through the introduction of LRT. This validates our argument that the use of LRT makes our algorithm more robust against errors in behaviour representation.

Computational cost — After model initialisation, the computational cost for our online incremental algorithm was significantly lower compared to the offline batch-mode algorithm (see Table 2). This is because only one behaviour pattern is used to update a single mixture component of $\mathbf{M}_n$ or $\mathbf{M}_a$ at each time. More importantly, since our algorithm is also online, it can run in real time.

	computational cost(second per frame)
incremental	0.025
batch-mode	0.137

Table 2: Comparing the computational cost of incremental learning with that of a batch-mode learning method. These were for Matlab implementations.

6 Discussion and Conclusion

The results of the two comparative experiments suggest than both the incremental and adaptive learning aspect of the proposed approach and the use of LRT contribute to the better performance of our model on abnormality detection compared to a conventional batch-mode method. Our experiments also demonstrate that the proposed algorithm is capable of adapting to changes of visual context and can run in real-time. This makes our algorithm suitable for a real-world surveillance application processing 24/7 continuous flow of video data.

In our approach, the abnormal behaviour model is approximate by $\mathbf{M}_a$ using the abnormal behaviour patterns observed so far. Such an approximate model is necessary for both Likelihood Ratio Test (LRT) and model adaptation based on mixture component trimming. As pointed out earlier, the key for the success of using LRT is to provide an accurate approximation for the alternative model, in this case the abnormal behaviour model. Our experimental results suggest that $\mathbf{M}_a$, constructed as a mixture of MOHMMs, is able to provide such an accurate approximation. In particular, thanks to the generative nature of a mixture of Dynamic Bayesian Networks (DBNs), it captures the randomness and unexpectedness which are common features of any abnormal behaviour pattern. $\mathbf{M}_a$ thus better explains an abnormal behaviour pattern that has not been observed before compared to $\mathbf{M}_n$. This is the basis for LRT to work in our approach.

In spite of the improvement of performance brought by the incremental and adaptive behaviour learning feature of our approach, the detection and false alarm rates achieved by the approach may still struggle to

meet the requirements of a practical surveillance application. The performance would certainly be improved if a more complete/sophisticated set of features are employed to detect events and represent behaviour patterns. Nevertheless it is noted that for the particular office entry scene analysed in the paper, the modest performance was mainly caused by the poor surveillance camera setup. In particular, the camera was mounted low and close to the scene which gives a very narrow view. Such a narrow view is ideal for face recognition, but not for behaviour analysis. Moreover, the movements of people are largely towards or away from the camera. It is widely acknowledged that a side view would be more ideal for activity and behaviour monitoring because it will result in far less occlusions. For instance, in the current camera setup, it is impossible to detect a 'card-swiping' event due to occlusions. Such an event class could provide very useful information on the normality/abnormality of a behaviour pattern in this scene.

In conclusion, we proposed a fully unsupervised approach for visual behaviour modelling and abnormality detection. Our approach differs from previous techniques in that our model is learned incrementally and adaptively given a small bootstrapping training set. In addition, our model adapts to changes in visual context over time therefore catering for the need to reclassify what may initially be considered as being abnormal to be normal over time, and vice versa. Furthermore, our model adopts a LRT based abnormality detection method which makes our approach more robust to errors in behaviour representation. Our experimental results demonstrate that the proposed approach is superior to the conventional batch-mode ones in terms of both performance on abnormality detection and computational efficiency.

References

[1] J. Berger. *Statistical decision theory and Bayesian analysis*. Springer-Verlag, 1995.

[2] A. F. Bobick and J. W. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):257–267, 2001.

[3] O. Boiman and M. Irani. Detecting irregularities in images and in video. In *IEEE International Conference on Computer Vision*, pages 462–469, 2005.

[4] H. Buxton. Generative models for learning and understanding dynamic scene activity. In *International Workshop on Generative Model Based Vision*, 2002.

[5] H. Buxton and S. Gong. Visual surveillance in a dynamic and uncertain world. *Artificial Intelligence*, 78:431–459, 1995.

[6] H. Dee and D. Hogg. Detecting inexplicable behaviour. In *British Machine Vision Conference*, pages 477–486, 2004.

[7] A. Dempster, N. Laird, and D. Rubin. Maximum-likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39:1–38, 1977.

[8] T. Duong, H. Bui, D. Phung, and S. Venkatesh. Activity recognition and abnormality detection with the switching hidden semi-markov model. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 838–845, 2005.

[9] Z. Ghahramani. Learning dynamic Bayesian networks. In C.L. Giles and M. Gori, editors, *Adaptive Processing of Sequences and Data Structures*, volume 1387 of *Lecture Notes in Artificial Intelligence*, pages 168–197. Springer-Verlag, 1998.

[10] S. Gong and T. Xiang. Recognition of group activities using dynamic probabilistic networks. In *IEEE International Conference on Computer Vision*, pages 742–749, 2003.

[11] R. Hamid, A. Johnson, S. Batta, A. Bobick, C. Isbell, and G. Coleman. Detection and explanation of anomalous activities: Representing activities as bags of event n-grams. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1031–1038, 2005.

[12] J. Kruskal and M. Liberman. *The symmetric time-warping problem: From continuous to discrete*. Addison-Wesley, 1983.

[13] L.R.Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[14] R. J. Morris and D. C. Hogg. Statistical models of object interaction. *International Journal of Computer Vision*, 37(2):209–215, 2000.

[15] R. Neal and G. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In M. I. Jordan, editor, *Learning in Graphical Models*. Kluwer, 1998.

[16] J. Neyman and E.S. Pearson. On the use and interpretation of certain test criteria for purposes of statistical inference. *Biometrika*, 20A:263–294, 1928.

[17] J. Ng and S. Gong. Learning intrinsic video content using levenshtein distance in graph partition. In *European Conference on Computer Vision*, pages 670–684, 2002.

[18] N. Oliver, B. Rosario, and A. Pentland. A Bayesian computer vision system for modelling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):831–843, August 2000.

[19] A. Panuccio, M. Bicego, and V. Murino. A hidden markov model-based approach to sequential data clustering. In *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, pages 734–742, London, UK, 2002. Springer-Verlag.

[20] F. Porikli. Trajectory distance metric using hidden markov model based representation. In *The Sixth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, 2002.

[21] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.

[22] V. Shet, D. Harwood, and L. Davis. Multivalued default logic for identity maintenance in visual surveillance. In *European Conference on Computer Vision*, volume 4, pages 119–132, 2006.

[23] C. Stauffer and W. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–758, August 2000.

[24] X. Wang, K. Tieu, and E. Grimson. Learning semantic scene models by trajectory analysis. In *European Conference on Computer Vision*, volume 3, pages 111–123, 2006.

[25] J. Wilpon, L. Rabiner, C. Lee, and E. Goldman. Automatic recognition of keywords in unconstrained speech using hidden markov models. *IEEE Trans. Acoustic, Speech and Signal Proc.*, pages 1870–1878, 1990.

[26] T. Xiang and S. Gong. Activity based video content trajectory representation and segmentation. In *British Machine Vision Conference*, pages 177–186, 2004.

[27] T. Xiang and S. Gong. Video behaviour profiling and abnormality detection without manual labelling. In *IEEE International Conference on Computer Vision*, pages 1238–1245, 2005.

[28] T. Xiang and S. Gong. Beyond tracking: Modelling activity and understanding behaviour. *International Journal of Computer Vision*, 67(1):21–51, 2006.

[29] T. Xiang and S. Gong. Incremental visual behaviour modelling. In *IEEE International Workshop on Visual Surveillance*, pages 65–72, 2006.

[30] T. Xiang, S. Gong, and D. Parkinson. Autonomous visual events detection and classification without explicit object-centred segmentation and tracking. In *British Machine Vision Conference*, pages 233–242, 2002.

[31] S. Yu and J. Shi. Multiclass spectral clustering. In *IEEE International Conference on Computer Vision*, pages 313–319, 2003.

[32] D. Zhang, D. Gatica-Perez, S. Bengio, and I. McCowan. Semi-supervised adapted HMMs for unusual event detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 611–618, 2005.

[33] H. Zhong, J. Shi, and M. Visontai. Detecting unusual activity in video. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 819–826, 2004.