Active Online Anomaly Detection using Dirichlet Process Mixture Model and Gaussian Process Classification

Jagannadan Varadarajan[1], Ramanathan Subramanian[2], Narendra Ahuja[1,3],
Pierre Moulin[1,3] and Jean-Marc Odobez[4]
[1] *Advanced Digital Sciences Center, Singapore*, [2] *International Institute of Information Technology, India*
[3] *University of Illinois at Urbana-Champaign, IL USA*, [4] *Idiap Research Institute, Martigny, Switzerland*
vjagan@adsc.com.sg, s.ramanathan@iiit.ac.in, {moulin.ifp, n-ahuja}@illinois.edu, odobez@idiap.ch

Abstract

We present a novel anomaly detection (AD) system for streaming videos. Different from prior methods that rely on unsupervised learning of clip representations, that are usually coarse in nature, and batch-mode learning, we propose the combination of two non-parametric models for our task: i) Dirichlet process mixture models (DPMM) based modeling of object motion and directions in each cell, and ii) Gaussian process based active learning paradigm involving labeling by a domain expert. Whereas conventional clip representation methods adopt quantizing only motion directions leading to a lossy, coarse representation that are inadequate, our clip representation approach results in fine grained clusters at each cell that model the scene activities (both direction and speed) more effectively. For active anomaly detection, we adapt a Gaussian Process framework to process incoming samples (video snippets) sequentially, seek labels for confusing or informative samples and and update the AD model online. Furthermore, the proposed video representation along with a novel query criterion to select informative samples for labeling that incorporates both exploration and exploitation criteria is proposed, and is found to outperform competing criteria on two challenging traffic scene datasets.

1. Introduction

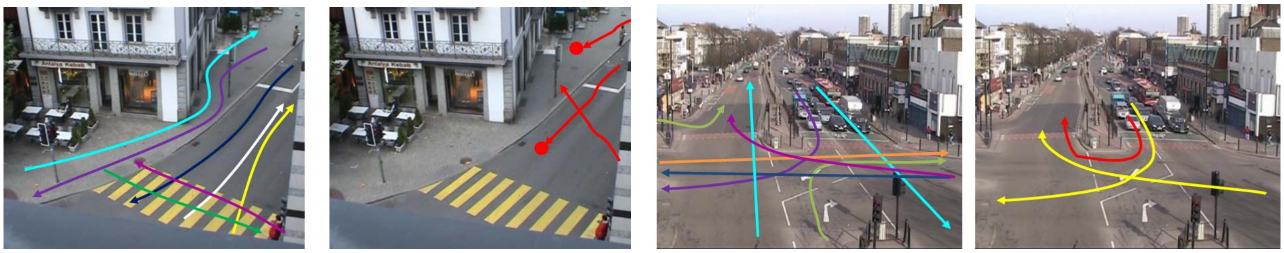
Heightened security concerns in the present day environment have led to the proliferation of CCTV camera networks monitoring public spaces such as airports, metro stations, traffic junctions and shopping malls. This in turn, has led to the stunning rise of surveillance feeds, and necessitated a strong demand for methods that can automatically detect *anomalous* or *suspicious* events to generate video highlights for future inspection and examination.

This work expressly focuses on ***abnormal event detection*** from surveillance feeds of traffic scenes (Fig.2). Analyzing traffic scenes is challenging as a variety of events occur simultaneously, with vehicles moving in different directions at varying speeds along with pedestrians moving on sidewalks or crossing the road. Events of interest in such scenes include accidents, traffic congestion, abrupt changes in traffic patterns, pedestrian movement in prohibited areas and jay-walking. Automated detection of abnormal traffic events is formidable as they are often subtle and highly contextual, and therefore difficult to model. Moreover, factors such as low video resolution, camera perspective, lighting changes, occlusions, variance in object size and motion and the rarity of anomalous exemplars strongly impede detection.

Prior traffic anomaly detection (AD) approaches [3, 31, 35] typically adopt an unsupervised, one-class learning approach, where a model of normal behaviors is learned first, and used to subsequently detect abnormalities during the test phase. However, these methods do not effectively deal with the AD problem because of two main reasons: i) lack of effective video representations, and ii) the model of normal behaviors is learned offline and *not updated* as new data arrive. To address this issue, we propose a non-parametric approach to anomaly detection that models local features more effectively and explores a *human-in-the-loop* AD system relying on ***active*** and ***online*** learning framework, where a domain expert labels ***confusing*** (or alternatively, ***informative***) examples, which are then employed for refining the AD model (Fig. 1).

This work makes two research contributions. As our first contribution, we propose a ***Dirichlet process mixture model*** (DPMM)-based modeling of object motion and directions within each cell of pixels to generate a fine-grained representation of scene activities. In contrast, traditional methods achieve unsupervised learning of coarse-grained scene activities which are often inadequate for AD. Also, since it would be unreasonable and expensive to query the

978-1-5090-4822-9/17 $31.00 © 2017 IEEE
DOI 10.1109/WACV.2017.74



(a) Idiap Junction data [31]

(b) QMUL Junction data [14]

Figure 2: **Problem Illustration:** One exemplar image from two traffic datasets is shown above. Arrows indicate direction of normal/abnormal activities. For each dataset, the left image shows normal activities, while the right image shows some abnormal activity patterns (red arrows). Typical abnormal activities include jay-walking by pedestrians, cars entering pedestrian zone (solid red dot), vehicles taking an illegal U-turn, near collisions *etc.* (see Table 1).

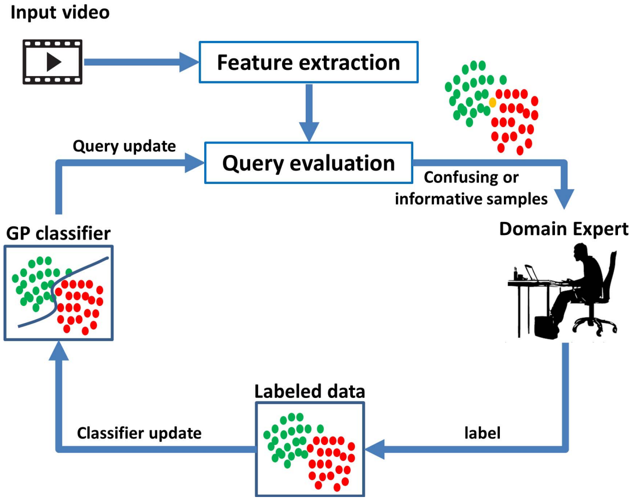


Figure 1: **Online and active anomalous event detection overview:** Each video clip, represented in a feature space, is first put through a query criteria to *decide whether* it is to be presented to the expert for labeling. Clips thus selected (as denoted by the orange dot) are actively labeled, and added to the existing list of labeled samples (events) for updating the classifier model.

expert for every clip (sample), *confusing* samples are determined such that AD performance improves after every query. To this end, our second contribution involves the proposition of a novel criterion, termed $\mathcal{Q}_{\mathrm{rel}}$, to evaluate if a sample is to be queried for labeling. $\mathcal{Q}_{\mathrm{rel}}$ incorporates and evaluates two related criteria, namely, (i) ***exploration***– where sample labeling enables discovering unseen regions in the feature space, and (ii) ***exploitation***, where sample labeling refines inter-class boundaries as (orange sample in Fig.1). Furthermore, unlike AD methods (such as [5, 11]) that rely on batch-based active learning (AL), we propose an AL paradigm where samples are *sequentially* processed by the expert as in a real-life scenario. Different from previous works [17, 18] which rely on simple classifiers, our AL module employs a more powerful Gaussian Process (GP) classifier, and several interesting properties of GP motivated its choice for active AD: (i) Uncertainty measures such as *predictive mean* and *covariance* of GPs naturally facilitate an AL paradigm [4, 11, 12]; (ii) Bayesian formulation of a GP allows for a sequential update of the AD model in closed-form [4], and (iii) The feature distributions of normal activities varies smoothly over time, which is encapsulated by the GP covariance. Employing a GP classifier with our $\mathcal{Q}_{\mathrm{rel}}$ criterion enables superior performance relative to other AL methodologies, as confirmed by experiments on two traffic surveillance datasets (illustrated in Fig. 2).

2. Related Work

Research areas closely related to this work are i) video representation for AD and ii) stream-based active learning. Below, we present a review of each of these topics.

Most existing work on abnormality detection rely on unsupervised methods to derive a representation of scene activities due to difficulty in labeling several hours of video data. Conventional AD methods are mostly trajectory based [19,21,25]. Here, object trajectories are used to learn dominant motion patterns of normal activities, which are then used for identifying outlier trajectories using likelihood measures [13,31] or classifiers such as one-class SVM [21]. Due to difficulties in obtaining reliable object trajectories, recent trends have explored variants of probabilistic topics models [15,31] and dynamic Bayesian networks [3,7,30] to describe a video clip via a learned patterns of activities. The activity patterns are in turn learned by applying topic models such as probabilistic latent semantic analysis (pLSA) on low-level visual features from foreground pixels and their optical flow.

All aforementioned methods quantize (only) motion angles computed from optical flow vectors by determining the range of the quantization bins *a-priori*. While this approach is simple to use, it i) completely ignores object speed information that is readily available, ii) is not well adapted to the scene, and iii) results in large vocabularies when additional contextual cues are added. Furthermore, when scene activities are represented by mid-level topics learned from the scene, anomaly measures become less sensitive to changes or violations in low-level features.

To address these issues, we propose to quantize the flow vectors arising from each non-overlapping block of pixels[1] using a Dirichlet process mixture model (DPMM), a non-parametric approach to learn mixture models that also infers the number of clusters in a data-driven manner. Learning a DPMM from optical flows results in a scene-centric vocabulary, while also incorporating both direction and speed information without increasing its size. Non-parametric methods, especially Hierarchical Dirichlet Processes [28] have been used by many earlier works [2, 17, 34] to model scene activities. Nevertheless, they are geared towards high-level learning of patterns and use pre-defined vocabularies.

Another important problem faced by existing AD methods is that they train a model in *batch mode*, which precludes the interactive labeling of samples making them unsuitable for streamed data as typically encountered in real-life scenarios. In such cases, **active learning (AL)** would be an apt approach, where the aim is to improve a classifier incrementally by seeking labels on confused examples. AL paradigm has been used in computer vision to address problems including object classification [11], scene classification [4, 12] and domain adaptation [16], where the focus is mainly on designing efficient querying strategies that strike a trade-off between *exploration* and *exploitation*. For instance, methods in [10, 29] predictive uncertainities for instance labeling and thus can be called exploitative as they aim to refine boundaries of known classes. On the other hand, explorative methods [20, 26] look for unknown regions in the feature space. This idea has been also employed to detect anomalies [20], traffic intrusions [26] and discover rare classes [5, 8]. Furthermore, several hybrid approaches including the unified theory in [9] combine both exploration and exploitation measures [4, 18] so that classification of known classes improves simultaneously with the discovery of new classes. For a more detailed review of AL methods, we refer to [24, 33]

Research works applying AL for surveillance have been very limited, as most well-known AL methods work on pool or batch-based settings, where samples are selected from a large pool of unlabeled samples for annotation by the domain expert, which only account for offline learning paradigms. In batch-based learning, a query criteria that gives a relative measure with respect to rest of the samples (*e.g.*, ranked distance from the classification margin) is sufficient.

Surveillance settings are inherently stream based. Therefore, our goal is to decide *on-the-fly* whether or not to request a label for an incoming sample. This requires more informative measures than distance ranks to be formulated. Our work is closely related to the work by [18], where a stream-based AL framework is developed via simple naive Bayes classifier which assumes that the individual features (activities in different scene regions) are independent given the class label. However, this assumption simplifies the fact that traffic scenes involve *complex interactions* among activities in different regions, and that scene activities are inherently correlated. In contrast, we make use of predictive mean and covariance functions, as well as confidence intervals offered within the Bayesian formulation of GPs to formulate principled query strategies resulting in improved detection performance as shown in Section 5. The following section describes our proposed AD framework.

3. Problem Formulation

Consider a video stream composed of clips[2] υ_t indexed by time $t \in \{1, \ldots, \infty\}$. Each clip represented (in terms of events) as $\boldsymbol{x}_t \in \mathcal{X}, \mathcal{X} \subset I\!R^D$ lying in a D dimensional space, belongs to one of two classes $y_t \in \{-1, 1\}$, with labels -1 and 1 denoting *normal* and *abnormal* events respectively. We seek to learn a classifier $\mathcal{C}$ that best predicts a label y_t for each clip $\boldsymbol{x}_t$. In order to acquire labels for classifier training and updation, we request a domain expert to label incoming samples (clips) for abnormal events (see Fig.1). Given a new sample, the decision to query the expert for a label is taken based on a query function $\mathcal{Q}$ and a budget **B**. The goal of function $\mathcal{Q}$ is to select ***informative*** samples that will help explore unseen regions in the feature space, while also refining decision boundaries to improve AD performance. **B** is the limit on the number of queries made to build $\mathcal{C}$.

Fig. 3 presents the overall flowchart for our approach. The first step in our approach is to learn a DPMM model for each cell (10×10) of pixels. This model is learned from a training set comprising *normal* activities. A feature vector that consolidates the activations from local DPMM models is used to describe each video clip, which is input to our GP-based active learning framework. Here, we first briefly review the basic concepts involved in our non-parametric modeling, *i.e.*, i) Dirichlet process mixture model and ii) Gaussian Processes, followed by more precise details regarding clip representation and Gaussian process learning.

3.1. Dirichlet Process Mixture Model

Dirichlet process mixture models (DPMM) can be considered as an infinite extension of the finite mixture model. Therefore, it may be easy to understand a DPMM starting from a finite mixture model. Fig. 4(a) is a graphical representation of a finite mixture model with K components. The β vector gives the weight of each mixture component

[1] Each non-overlapping block of pixels also called cells is our basic unit for clip representation.

[2] snippets capturing concurrent events some of which may be abnormal.

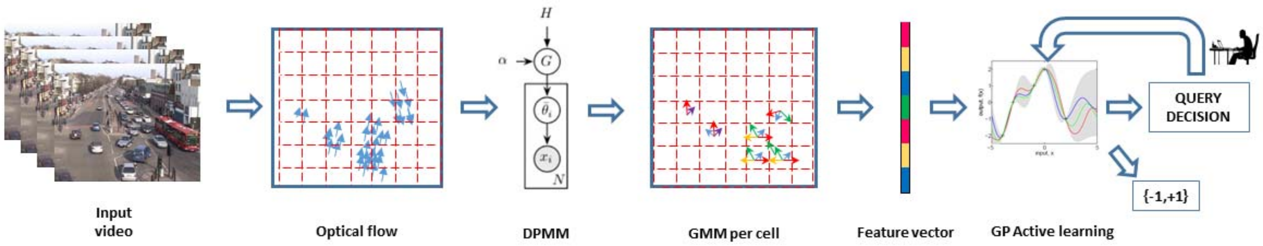


Figure 3: Flowchart of our non-parametric active online AD framework. Optical flow vectors observed from each cell of 10×10 pixels over normal clips are used to learn a DPMM model for each cell. Every video clip is then represented by a feature vector obtained by concatenating the activations in each cell's DPMM, which is fed into the Gaussian process-based active learning framework. A decision is made instantaneously whether or not to query an expert for a label corresponding to the sample.

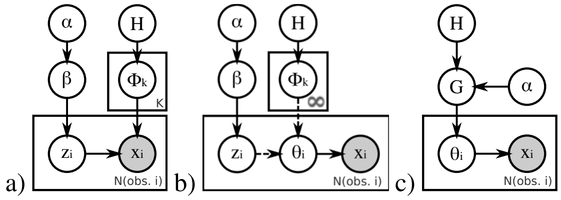


Figure 4: Finite mixture and Dirichlet Process (infinite mixture): a) finite mixture with K elements; b) mixture representation for DP; c) compact representation for DP.

and α is a prior on these weights. Each Φ_k represents the parameters of a mixture component and z_i represents the index of the mixture component for each observation x_i. The mixture components we are using here are Gaussian distributions ($\Phi_k = (\mu_k, \Sigma_k)$), finally resulting in a Gaussian mixture model (GMM).

$$\beta \sim GEM(\alpha) \tag{1}$$

$$\forall k \qquad \phi_k \sim H \tag{2}$$

$$\forall i \qquad z_i \sim \text{Categorical}(\beta) \tag{3}$$

$$x_i \sim \phi_{z_i} = \mathcal{N}(x_i|\mu_{z_i}, \Sigma_{z_i}) \tag{4}$$

By letting K go to infinity, we obtain an infinite mixture model as shown in Fig. 4(b). We can explicitly represent the mixture component selected by each observation noted θ_i. We use dashed arrows to indicate deterministic relations, here $\theta_i = \Phi_{z_i}$ (or, expressed as a draw from a Dirac distribution: $\theta_i \sim \delta_{\Phi_{z_i}}$). To adapt to this infinite mixture elements, the weight vector β is of infinite length and the prior α takes a specific form. The α prior is now a single positive real value used as the parameter of a "GEM" (Griffiths, Engen, McCloskey) also known as a "stick breaking" process. This process produces an infinite list of weights that sum to 1: the first weight $\beta_1 = \beta_1'$ is drawn from a beta distribution $Beta(1, \alpha)$, the second weight is drawn in the same way but only from the remaining part, *i.e.*, $\beta_2 = (1 - \beta_1) * \beta_2'$ with β_2' drawn from $Beta(1, \alpha)$, and so on for the other weights, hence the "stick breaking" name. For each mixture component, the parameters Φ_k are independently drawn from a prior H. We thus have the following:

A more compact equivalent notation can be used to represent a Dirichlet Process. While the mixture representation is well adapted for deriving the Gibbs sampling scheme, a more compact representation as shown in Figure 4c is widely used to represent a DPMM. Here, individual mixture components are not shown and instead their weighted countable infinite mixture $G = \sum_{k=1}^{\infty} \beta_k \delta_{\phi_k}$ is used. The corresponding representation, using a DP notation, is given as:

$$G \sim DP(\alpha, H) \tag{5}$$

$$\forall i \qquad \theta_i \sim G \tag{6}$$

$$x_i \sim \theta_i \tag{7}$$

We need to specify the base distribution H to complete the model. We use a Normal- Inverse Wishart distribution for the base distribution H as it acts as a conjugate for Normal distribution and simplifies the inference process. The hyper parameters for the base distribution are $\mu_0, \lambda_0, \Sigma_0, \nu_0$ and the concentration parameter α, which controls the number of Gaussian components. The DPMM is solved by estimating the posterior distribution using Gibbs sampling. We refer to [27] for more details on this.

3.2. Gaussian Process

Given a dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ comprising N feature-label pairs where $\boldsymbol{x}_i$ and y_i are defined as above, our objective is to predict the label y_* of an unseen sample $\boldsymbol{x}_*$. We assume that the relationship between $\boldsymbol{x}_i$ and labels y_i is given by a latent function $f : \mathcal{X} \rightarrow I\!R$ and additive Gaussian noise leading to $y_i = f(\boldsymbol{x}_i) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$. Instead of restricting f to be from a certain parametric family of functions, we assume that f is drawn from a specific probability distribution $p(\mathbf{f})$. This enables a Bayesian treatment of our problem, *i.e.*, we infer the probability of y_* given $\boldsymbol{x}_*$ and old observations $\mathcal{D}$ by integrating out the corresponding function values $f_* = f(\boldsymbol{x}_*)$ and $\mathbf{f} = \{f_1, \ldots, f_N\}$:

$$p(y_*|\boldsymbol{x}_*, \mathcal{D}, \theta) = \int p(f_*|\boldsymbol{x}_*, \mathcal{D}, \theta) p(y_*|f_*) df_*, \qquad (8)$$

$$p(f_*|\boldsymbol{x}_*, \mathcal{D}, \theta) = \int p(f_*|\boldsymbol{x}_*, \mathbf{f}, \theta) p(\mathbf{f}|\mathcal{D}, \theta) d\mathbf{f}, \qquad (9)$$

where θ denotes model hyper-parameters. In GP, we assume that the prior distribution over function values is given by a multivariate Gaussian distribution, denoted as $p(\mathbf{f}) = \mathcal{N}(m(X), \kappa(X, X))$, where $m(.)$ is the mean function and $\kappa(.,.)$ is the covariance function (an $N \times N$ matrix whose $(i,j)^{\text{th}}$ element is given by a kernel function $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$), which describes the coupling between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ as a function of their distance. A popular choice for the kernel function is the squared exponential given by,

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sigma_n^2 \exp(-\sum_{k=1}^{D} \frac{(\boldsymbol{x}_{ik} - \boldsymbol{x}_{jk})^2}{2\sigma_k^2}), \qquad (10)$$

where σ_n^2 represents signal noise and σ_k denotes scaling parameter for dimension k. The posterior predictive distribution given in Eq.(9) is again Gaussian with moments:

$$\mu_*(\boldsymbol{x}_*) = \boldsymbol{k}_*^{\mathrm{T}} (K + \sigma_n^2 \boldsymbol{I})^{-1} \boldsymbol{Y} \qquad (11)$$

$$\sigma_*^2(\boldsymbol{x}_*) = k_{**} + \sigma_n^2 - \boldsymbol{k}_*^{\mathrm{T}} (K + \sigma_n^2 \boldsymbol{I})^{-1} \boldsymbol{k}_*, \qquad (12)$$

where $\boldsymbol{Y} = [y_1, \ldots, y_N]^{\mathrm{T}}$, $\boldsymbol{K}$, k_* and k_{**} respectively denote the $N \times N$ kernel matrix containing covariances of training samples, the $(N \times 1)$ kernel vector containing covariances between training samples and the test sample, and the covariance of the test sample to itself.

We can deduce from Eq.(10) that $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$ approaches its maximum value σ_n^2 when $\boldsymbol{x}_i \approx \boldsymbol{x}_j$ denoting that f_i and f_j are nearly perfectly correlated, whereas $k(\boldsymbol{x}_i, \boldsymbol{x}_j) \approx 0$ when $\boldsymbol{x}_i$ is far from $\boldsymbol{x}_j$ indicating minimum influence $\boldsymbol{x}_i$ has on $\boldsymbol{x}_j$. For binary classification problem, probability of y is independent of all other quantities given the value of $f(\boldsymbol{x})$, *i.e.*, $p(y = 1|\mathcal{D}, f(\boldsymbol{x})) = p(y = 1|f(\boldsymbol{x}))$. The likelihood function $p(y|f(\boldsymbol{x}))$ is usually modeled using cumulative normal (CN). This function maps high values of f to ≈ 1 and low f to ≈ 0.

Note that due to the Gaussian noise assumption that links y_* and f_*, their expected values are the same. However, their variances differ owing to observational noise. Due to this fact, a test sample x_* can be classified based on the sign of $\mu_*(\boldsymbol{x}_*)$, and the absolute predictive mean $|\mu_*(\boldsymbol{x}_*)|$ indicates how close or far the sample is to the classification boundary. A small absolute mean indicates that the sample lies close to the boundary and hence is a *confusing* case and vice-versa. Labeling samples near the class boundary is critical, as these may indicate abnormal events that may be potentially confused with normal events or false negatives. Also, labeling such samples can refine the class boundary leading to an *exploitative* strategy. In contrast to the pool-based setting, where a sample for querying is chosen by ranking absolute predictive mean values obtained for a pool of unlabeled data, we need to derive an absolute criterion to decide on-the-fly if a sample needs querying or not. We formulate this by placing a threshold on $|\mu_*(\boldsymbol{x}_*)|$, *i.e.*, we decide to query for a sample label if $|\mu_*(\boldsymbol{x}_*)|$ is close to zero or, $\mathcal{Q}_\mu : |\mu_*(\boldsymbol{x}_*)| \leq \tau_1$, where τ_1 is a threshold. In particular, we are interested in points for which CN is ≈ 0.5, which happens when $|\mu_*(\boldsymbol{x}_*)| \approx 0$.

Another criterion is to query for samples with large predictive variance $\mathcal{Q}_{\text{var}}(\boldsymbol{x}_*) = \sigma_*^2(\boldsymbol{x}_*)$. A large $\sigma_*^2(\boldsymbol{x}_*)$ indicates that the sample lies in an unexplored region of the feature space, potentially denoting outlier clips. Labeling such samples help us explore unknown abnormality types leading to an *explorative* strategy. An uncertainty measure which considers both the predictive mean and variance is proposed in [11]. Specifically, the query criterion is given by: $\mathcal{Q}_{\text{unc}}(\boldsymbol{x}_*) = \frac{|\mu_*(\boldsymbol{x}_*)|}{\sqrt{\sigma_*^2(\boldsymbol{x}_*) + \sigma_n^2}}$ This criterion combines both exploration and exploitation. However, this method focuses more on *outlier* samples that are far from training samples than confusing ones [10]. As abnormal events typically occur with other normal events and have limited spatial and temporal support, they lie close to normal events and denote confusing samples rather than outliers.

We formulate a new and more severe criterion that combines both exploration and exploitation, and ranks both predictive mean and variance for label querying: $\mathcal{Q}_{\text{rel}}(\boldsymbol{x}_*) = \min\{2|\mu_*(\boldsymbol{x}_*)|, \frac{2}{\sigma_*(\boldsymbol{x}_*)}\}$. A label is sought for samples when $\mathcal{Q}_{\text{rel}}(\boldsymbol{x}_*) < \tau_2$, where τ_2 denotes a user-defined threshold. Here, the idea is to choose samples for which at least one of the (mean or variance) criteria indicate that the sample needs to be queried. When $\tau_2 = 1$, samples are either very close to the class boundary or have a predictive standard deviation greater than 2, indicating they are far from ($> 95\%$) of training samples. Since $\mathcal{Q}_{\text{rel}}$ decides to query for labels on checking whether the mean or variance is relatively more important, it is a ***Relative Importance*** criterion. Empirical results confirm the suitability of this criterion for active AD.

4. Video representation

Since videos need to be characterized in terms of activities for our problem, we derive a clip representation derived from low-level motion cues. Upon splitting a video into short clips, we track moving objects via densely sampled feature points [32], with the maximum trajectory length set to 15 frames. Location and motion information available

Table 1: Abnormal events from the two considered datasets.

	Description	# clips (%)
Idiap	Car stopping abruptly after traffic light	21 (1.58)
	Pedestrians Jaywalking	146 (11.0)
	Car entering pedestrian area	47 (3.5)
QMUL	Illegal U-turn	29 (1.61)
	Emergency vehicles using incorrect lane	3 (0.17)
	Traffic halt due to fire engine	12 (0.67)

from trajectory observations are quantized to a feature vector representation for each clip. Quantization steps for vocabulary creation are:

Activities in surveillance videos captured by fixed cameras can be characterized by their location. We quantize pixel positions into non-overlapping cells of 10×10 pixels. *E.g.,* we obtain 29×36 cells from a 288×360 pixel video. From consecutive trajectory observations, we compute motion vectors (u_x, u_y). The inputs to our DPMM model for each cell are motion vectors observed in the cell from a set of normal clips. The DPMM components are then learned using the Chinese restaurant process and Gibbs sampling as detailed in [27]. Additionally, in order to capture unforeseen motion within each cell (which possibly might indicate anomolies), we add a background Gaussian component with parameters $(\mu_{bg} = 0, \Sigma_{bg} = 4, \beta_{bg} = 0.1)$ to every cell. For cells with a DPMM model, this component is simply added to the existing mixture components followed by a renormalization of the prior weight parameter.

Each cell c is then represented using a weight vector $\boldsymbol{w}_c$ of length $d_c + 1$, where d_c is the number of Gaussian components discovered for each cell c. The weight vector $\boldsymbol{w}_c$ is obtained by summing the posterior probability vectors from every single observation in that cell. The video clip is finally represented by stacking all the weight vectors $(\boldsymbol{w}_1, \cdots, \boldsymbol{w}_{N_c})$ followed by a normalization so that the sum of all weights is one. Here N_c is the number of cells in the image.

5. Experimental Results

Datasets: We report experiments on two public video datasets specified in Table 1. The **Idiap Junction data** [31] (Fig.2(a)), is a video from a busy road junction. The video is 44 minutes long, and recorded at 25 fps with a frame size of 360×288. Activities at the junction include (a) people walking on the pavement, (b) people waiting for vehicles to cross, (c) people crossing at zebra crossings, (d) vehicles moving in different directions *etc.*The **QMUL Junction data** [14] (Fig.2(b)) is filmed at a four-road junction. The video is 1 hour (90000 frames) long, recorded at 25 fps at 360×288 resolution. The junction is regulated by traffic lights and dominated by four types of traffic flows. Table 1 describes the abnormal activities in these two datasets. The number of clips involving abnormal activities, and the proportion of such clips over the video length (in %) are specified in the right column. Both videos were segmented into short clips of 50 frames. This results in 1327 and 1800 clips for the Idiap and QMUL data respectively. For Idiap, events in each clip were manually annotated as *normal/abnormal*, while annotations were obtained from [18] for QMUL. Ground-truth labels were used for simulating the *expert label* during queries, and for performance evaluation.

Settings: The DPMM model is first learned using 450 normal clips corresponding to 15 minutes of video. We use a Normal-inverse Wishart for the based distribution of the model with the parameters set as $\mu_0 = (0, 0), \lambda_0 = 4, \Sigma_0 = I, \nu_0 = 2$. The concentration parameter of DPMM, which controls the number of clusters, is set to a default value of $\alpha = 0.1$. The number of mixture components learned for each cell varies between 2-12 in the Idiap junction dataset and 2-8 in the QMUL junction dataset. The relatively higher number of components for the Idiap dataset may be due to the large number of pedestrian motion observed in the scene, which is more unstructured compared to vehicles. This results in a total of 3882 and 3218 components for the Idiap and QMUL Junction dataset, which has fewer dimensions compared to a predefined quantization[3].

Clips from each video were partitioned into training and test sets– 60% clips were used for training and the remaining for test. To begin with, a simple GP classifier was learned using two clips containing normal activities and one clip involving unusual event(s). The remaining training samples were visited sequentially, and were either queried for a label or discarded. Upon adding a newly (actively) labeled sample to the training set, the classifier was updated and its performance evaluated on the test set. For each dataset, we created 10 different sequences by randomly shuffling the training clips. Performance was measured by computing the area under the receiver operator curve (AUROC) after each query. Final curves (Fig.6) were obtained on averaging the AUROC values over the 10 runs. Kernel computation is a key component of GP classification. Histogram intersection kernel (HIK), given by $K_{\text{HI}}(\mathcal{Z}, \mathcal{Z}') = \sum_{i=1}^{N_A} \min\{z_i, z'_i\}$, which generates a positive-definite matrix was efficiently used for covariance computation for GP [23]. Since the feature vector representing each clip is a topic weight vector or normalized histogram, we used HIK for covariance computation. We used the GPML library [22] for active learning and implemented the DPMM model from scratch in Matlab.

Baselines: We perform experiments to evaluate i) our clip representation method and ii) our query criteria. First, to perform comparative evaluation of our clip representation method, we consider two other baseline approaches. a)

[3] From a frame size of 360×288 pixels, we get 4176 words by quantizing the location into 10×10 pixels and motion direction into 4 bins.

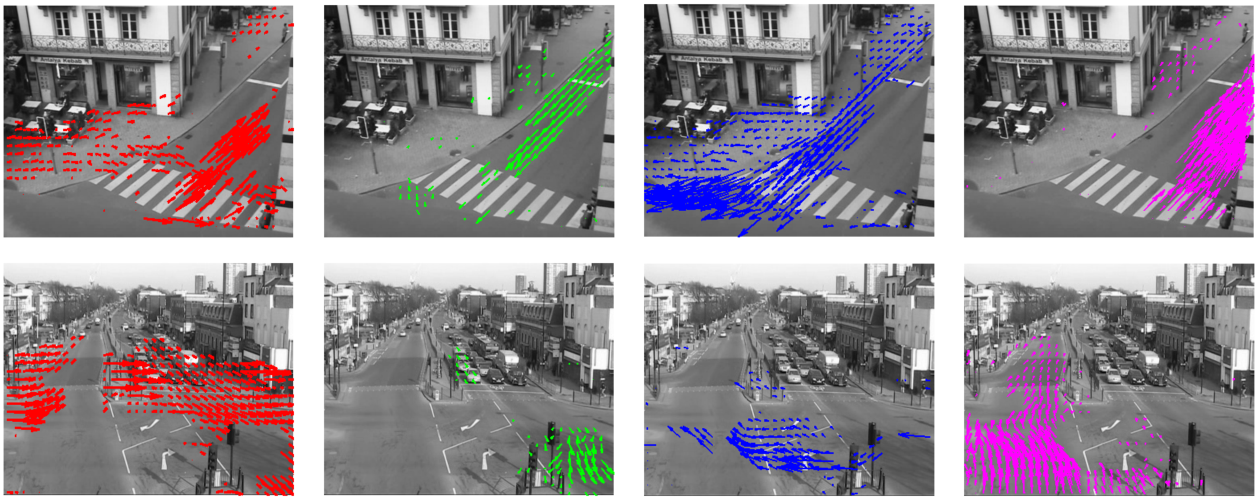
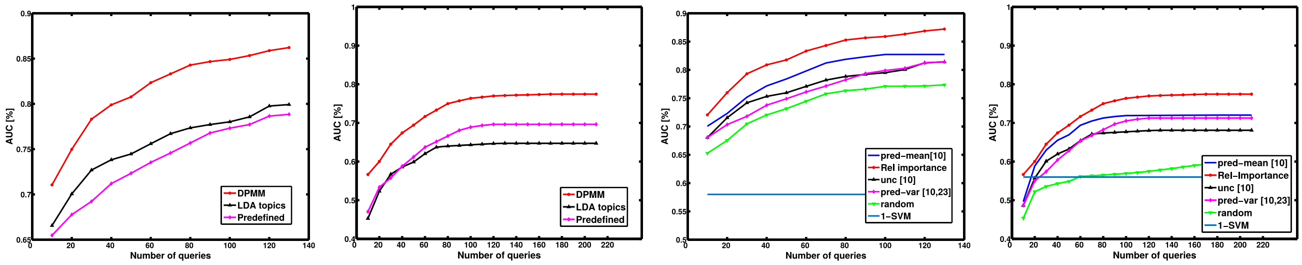


Figure 5: **DPMM result from TJ and QMUL datasets:** For each cell, the top ranking Gaussian component is demonstrated using arrows emerging from the location. For convenience, they are separated into different directions and color coded. a) column 1 - Red ($270° - 45°$), column 2 - egreen ($270° - 225°$), column 3 - blue ($225° - 135°$) and column 4 - magenta ($135° - 45°$). Length of the arrows are scaled based on their magnitude indicating speed.



(a)

(b)

(c)

(d)

Figure 6: **Comparison of clip representation methods and AL query criteria:** (a,b) results from different clip representation methods using the proposed Rel-Importance query strategy on (a) QMUL and (b) Idiap junction data. Results from different AL query strategies within the Gaussian Process framework for (c) QMUL and (d) Idiap junction data (best viewed in color and under zoom).

we implemented a predefined coarse vocabulary creation method followed in several earlier works [31, 34] Here, location is quantized into 10x10 cells and motion direction is quantized into four labels (*left*, *right*, *up*, *down*) corresponding to the cardinal motion directions. Each video clip (or *document*) v is represented by the frequency $n(v, w)$, of a word w occurring in v to obtain the bag-of-words representation. We denote this by the name **Predefined**. b) For the second baseline, we apply a dimensionality reduction method using Latent Dirichlet Allocation (LDA) [1] learned on a few normal activity clips to discover N_A dominant topics indicated by $\mathcal{Z} = \{z_1, \ldots, z_{N_A}\}$. Subsequently, the feature vector $\boldsymbol{x}$ for every v is given by the topic weight vector $(p(z_1|v), \ldots, p(z_{N_A}|v))$ obtained by the folding-in procedure [6], where each entry indicates the extent of topic z_i present in v. We used 30 topics to represent each clip in our experiments. We denote this by the name **LDA topics** in our evaluation.

We compare the AUROC obtained using $\mathcal{Q}_{\text{rel}}$ with various other query criteria. They include predictive mean (*pred-mean*/ $\mathcal{Q}_{\mu}$), predictive variance (*pred-var*/$\mathcal{Q}_{\text{var}}$) and the uncertainty criteria (*unc*/$\mathcal{Q}_{\text{unc}}$) proposed by [11, 23]. Thresholds to select samples for $\mathcal{Q}_{\mu}$, $\mathcal{Q}_{\text{var}}$ and $\mathcal{Q}_{\text{unc}}$ were fixed as 1, 2 and 0.5 respectively. We also use a *random*/$\mathcal{Q}_{\text{rand}}$ criteron, where samples are queried or rejected with equal probability. Furthermore, we also learned a one-class SVM (referred as 1-SVM in Fig.6) using only normal documents that are used by the aforementioned active learning methods.

Results: In Fig. 5, we demonstrate the dominant motion directions corresponding to each cell obtained by applying DPMM on the two datasets. Note that the length of the arrows are weighted by the magnitude of the mean motion vector. Interestingly, we see motion vectors with higher speed from cells close to the camera (cf. last column in Fig. 5) due to perspective effect of the camera.

In Fig. 6(a,b) we evaluate the performance of our active learning method using different clip representation methods, by fixing the query criteria to $\mathcal{Q}_{\rm rel}$. Thanks to the effective modeling of local activities by DPMM, we observe a much higher AUR by the proposed method compared to the other two baselines. Interestingly, the performance due to the predefined vocabulary and LDA topics are quite similar. This can be due to the fact that LDA topics are learned from the same predefined vocabulary.

Fig. 6(c,d) compares AUROCs obtained from classification accuracies evaluated on the abnormal clips with different query strategies on the two datasets, where our proposed DPMM based clip representation is used. Firstly, we observe that the classifier performance improves as queries are progressively made, however saturating after about 80 queries indicating that further labeled samples do not improve AD performance. Best performance is obtained using our $\mathcal{Q}_{\rm rel}$ criterion, resulting in a peak AUC performance of 87% for QMUL, and 76% AUC for Idiap. $\mathcal{Q}_{\mu}$ is the next best performing criterion, thereby revealing that query criteria focusing on refining class boundaries and resolving confusions perform best for the AD problem. This is in line with our understanding that most of the abnormal clips remain close to the normal clips in the feature space and hence contribute to confusions between the classes. We also see that $\mathcal{Q}_{\rm var}$ and $\mathcal{Q}_{\rm unc}$ [11] perform similarly, but with lower accuracy than $\mathcal{Q}_{\rm rel}$ and $\mathcal{Q}_{\mu}$. This is mainly because normal and abnormal events are closely clustered in the feature space, and hence looking for outliers as in $\mathcal{Q}_{\rm var}$ does not yield the best results. Expectedly, the batch mode 1-SVM performs just better than random with about 57% and 56% accuracy on the QMUL and Idiap datasets respectively. The random query criteria $\mathcal{Q}_{\rm rand}$ still improves over 1-SVM, but performs worse than others. Finally, a comprehensive comparison combining different clip representation methods with different query criteria is presented in Fig. 7. Again, we see that the combination of DPMM and $\mathcal{Q}_{rel}$ criteria gives the best results in both the cases.

Discussion: In order to better understand the effect of the threshold on $\mathcal{Q}_{rel}$ to select a query instance, we experimented with a range of threshold values from the interval $[0.1, 2]$. From our results, we found that our method is not too sensitive to this threshold and the performance remains unchanged until reduced below 0.5 or increased above 1.5. In the former case (< 0.5), a conservative threshold missed several interesting samples for label query leading to small performance improvements. In the latter case (> 1.5), several uninformative samples are selected for label query. This exhausts the budget quickly with only little improvement. In our case, setting this to 1 was a good compromise. Since our approach combines two different measures, it is also interesting to understand which measure triggers most of the queries. Our analysis revealed that the predicitive mean (first factor in $\mathcal{Q}_{rel}$) triggered nearly 85% and 82% of the queries in the QMUL and Idiap junction datasets respectively, with the remaining queries triggered by higher predicitive variance. This concurs with our observation that anomalies are subtle and often co-occur with other normal activities, leading to clips that are confusing (determined by uncertainty criteria) rather than being an outlier. This is also reflected in the performance curves presented in Fig. 6(c,d), where we see that predictive mean is the second best performing query criteria and often close to the proposed approach.

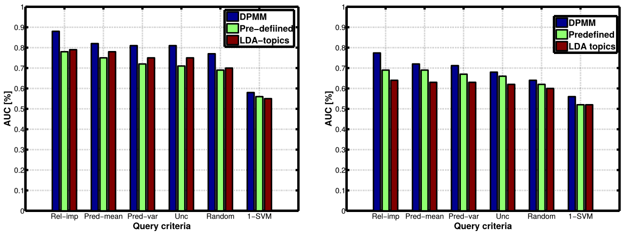


Figure 7: **Comprehensive comparison.** comparative study of various combinations of $\mathcal{Q}$ and clip representation methods (left) QMUL [14] and (right) Idiap [31] junction data (best viewed in color). The results shown are obtained after running AL with a budget $B = 60\%$ of training data.

6. Conclusion

This paper proposes an active and online anomaly detection system. Different from prior active learning methods proposed for surveillance scenarios which employ batch-based model training, our methodology accounts for real-life situations where video snippets are processed sequentially, followed by evaluation via the query criterion $\mathcal{Q}_{\rm rel}$ to decide if an event needs to be labeled by the domain expert. The criterion $\mathcal{Q}_{\rm rel}$ used to identify *informative* samples, incorporates the twin criteria of exploration and exploitation. Furthermore, a fine-grained representation of scene activities is extracted via a Dirichlet process mixture model that enables better context modeling in terms of speed and motion direction. Experiments on two traffic datasets show that $\mathcal{Q}_{\rm rel}$ outperforms competing query criteria for active learning.

Acknowledgments: This work is supported by the research grant for the Human-Centered Cyber-physical Systems Programme at the Advanced Digital Sciences Center from Singapore's Agency for Science, Technology and Research (A*STAR). We thank NVIDIA for GPU donation.

References

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3(4-5):993–1022, 2003.

[2] R. Emonet, J. Varadarajan, and J.-M. Odobez. Extracting and locating temporal motifs in video scenes using a hierarchical non parametric bayesian model. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[3] R. Emonet, J. Varadarajan, and J.-M. Odobez. Multi-camera open space human activity discovery for anomaly detection. In *IEEE Conference on Audio and Video Signal based Surveillance*, 2011.

[4] A. Freytag, E. Rodner, P. Bodesheim, and J. Denzler. Labeling examples that matter: Relevance-based active learning with gaussian processes. In *German Conference Pattern Recognition*, 2013.

[5] T. S. F. Haines and T. Xiang. Active rare class discovery and classification using dirichlet processes. *International Journal of Computer Vision*, 106(3):315–331, 2013.

[6] T. Hofmann. Unsupervised learning by probability latent semantic analysis. *Journal of Machine Learning Research*, 42:177–196, 2001.

[7] T. Hospedales, S. Gong, and T. Xiang. A Markov clustering topic model for mining behavior in video. In *IEEE International Conference on Computer Vision*, Kyoto, Japan, 2009.

[8] T. M. Hospedales, S. Gong, and T. Xiang. Finding rare classes: Adapting generative and discriminative models in active learning. In *Advances in Knowledge Discovery and Data Mining: 15th Pacific-Asia Conference*, 2011.

[9] T. M. Hospedales, S. Gong, and T. Xiang. A unifying theory of active discovery and learning. In *European Conference on Computer Vision*, 2012.

[10] P. Jain and A. Kapoor. Active learning for large multi-class problems. In *CVPR*, 2009.

[11] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell. Active learning with gaussian processes for object categorization. In *CVPR*, 2007.

[12] M. Kemmler, E. Rodner, E.-S. Wacker, and J. Denzler. One-class classification with gaussian processes. *Pattern Recognition*, 46(12):3507 – 3518, 2013.

[13] J. Kim and K. Grauman. Observe locally, infer globally: a space-time mrf for detecting abnormal activities with incremental updates. In *CVPR*, 2009.

[14] J. Li, S. Gong, and T. Xiang. Global behaviour inference using probabilistic latent semantic analysis. In *British Machine Vision Conference*, 2008.

[15] J. Li, S. Gong, and T. Xiang. Discovering multi-camera behaviour correlations for on-the-fly global activity prediction and anomaly detection. In *IEEE International Workshop on Visual Surveillance*, Kyoto, Japan, 2009.

[16] G. Liu, Y. Yan, R. Subramanian, J. Song, G. Lu, and N. Sebe. Active domain adaptation with noisy labels for multimedia analysis. *World Wide Web*, 19(2):199–215, 2016.

[17] C. C. Loy, T. M. Hospedales, T. Xiang, and S. Gong. Stream-based joint exploration-exploitation active learning. In *CVPR*, 2012.

[18] C. C. Loy, T. Xiang, and S. Gong. Stream-based active unusual detection. In *ACCV*, 2012.

[19] V. Mirge, K. Verma, and S. Gupta. Dense traffic flow patterns mining in bi-directional road networks using density based trajectory clustering. *Advances in Data Analysis and Classification*, pages 1–15, 2016.

[20] D. Pelleg and A. Moore. Active learning for anomaly and rare-category detection. In *NIPS*, pages 1073–1080. MIT Press, 2004.

[21] C. Piciarelli, C. Micheloni, and G. L. Foresti. Trajectory-based anomalous event detection. *IEEE Trans. Cir. and Sys. for Video Technol.*, 18(11):1544–1554, Nov. 2008.

[22] C. E. Rasmussen and H. Nickisch. Gaussian processes for machine learning (gpml) toolbox. *J. Mach. Learn. Res.*, 11:3011–3015, 2010.

[23] E. Rodner, A. Freytag, P. Bodesheim, and J. Denzler. Large-scale gaussian process classification with flexible adaptive histogram kernels. In *ECCV*, 2012.

[24] B. Settles. Active learning literature survey. Technical report, University of Wisconson-Madison, 2010.

[25] R. Sharma and T. Guha. A trajectory clustering approach to crowd flow segmentation in videos. In *IEEE International Conference on Image Processing (ICIP)*, pages 1200–1204, 2016.

[26] J. W. Stokes, J. C. Platt, J. Kravis, and M. Shilman. Aladin: Active learning of anomalies to detect intrusions, microsoft research. Technical report, MSR, 2008.

[27] E. Sudderth. *Graphical models for visual object recognition and tracking*. PhD thesis, MIT, 2006.

[28] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[29] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.*, 2:45–66, 2002.

[30] J. Varadarajan, R. Emonet, and J. Odobez. Bridging the Past, Present and Future; Modeling Scene Activities from Event Relationships and Global Rules. Providence, USA, 2012.

[31] J. Varadarajan and J. Odobez. Topic models for scene analysis and abnormality detection. In *IEEE International Workshop on Visual Surveillance*, Kyoto, Japan, 2009.

[32] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action Recognition by Dense Trajectories. In *CVPR*, 2011.

[33] M. Wang and X.-S. Hua. Active learning in multimedia annotation and retrieval: A survey. *ACM Trans. Intell. Syst. Technol.*, 2(2):10:1–10:21, Feb. 2011.

[34] X. Wang, X. Ma, and E. L. Grimson. Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(3):539–555, 2009.

[35] G. Zen and E. Ricci. Earth mover's prototypes: A convex learning approach for discovering activity patterns in dynamic scenes. In *CVPR*, pages 3225–3232. IEEE, 2011.