Semi-Supervised Learning for Anomalous Trajectory Detection

Conference Paper · January 2008

DOI: 10.5244/C.22.103 · Source: DBLP

Robert Fisher
The University of Edinburgh
351 PUBLICATIONS **9,394** CITATIONS

Semi-supervised Learning for Anomalous Trajectory Detection

R. R. Sillito and R. B. Fisher
School of Informatics, University of Edinburgh, UK

Abstract

A novel learning framework is proposed for anomalous behaviour detection in a video surveillance scenario, so that a classifier which distinguishes between normal and anomalous behaviour patterns can be incrementally trained with the assistance of a human operator. We consider the behaviour of pedestrians in terms of motion trajectories, and parametrise these trajectories using the control points of approximating cubic spline curves. This paper demonstrates an incremental semi-supervised one-class learning procedure in which unlabelled trajectories are combined with occasional examples of normal behaviour labelled by a human operator. This procedure is found to be effective on two different datasets, indicating that a human operator could potentially train the system to detect anomalous behaviour by providing only occasional interventions (a small percentage of the total number of observations).

1 Introduction

The problem of identifying abnormal behaviour from surveillance video footage has become a key focus in computer vision research. The task of continually and reliably monitoring large numbers of video streams presents an insurmountable challenge for an individual human operator, and stands to gain a great deal from computational automation. In particular, there is a clear potential role for computer vision in providing first-order interpretations of video data so that the relative salience of different video streams can be quantified, and the attention of human operators appropriately prioritised. This work explores a procedure for utilising minimal input from a human operator in the training of an abnormal behaviour classifier and illustrates the benefits that this could provide.

In order to automatically identify events worthy of human scrutiny, it is necessary to represent video data in terms of features which allow us to reliably distinguish unusual behaviour from ordinary occurrences. One such feature consists in the motion trajectories that result from tracking the movements of pedestrians and vehicles over time: indeed, motion trajectories have provided the basis for a large body of work on automated surveillance. One of the earliest approaches to behaviour classification, proposed by Johnson and Hogg [8], identified unusual behaviour by comparing new trajectories with a set of clusters representing typical sequences of typical local motion vectors in a given scene. A similar approach has recently been adopted by Hu et al. in [7], where typical trajectories are modelled with a more complex hierarchical clustering strategy. In a different vein, recent work by Dee and Hogg [1] has shown that unusual trajectories can also be identified using a rule-based approach, inspired by cognitive science, which quantifies the extent to which the movements of a given individual could be regarded as goal-directed.

A limitation of trajectory-based approaches is that they depend on the existence of reliable methods for tracking moving objects: while tracking is possible in certain scenarios it is, in the general sense, an unsolved problem. In this light, one of the most promising recent approaches to automatic behaviour analysis consists of decomposing video data in terms of some low level representational primitive, and modelling the sequential topology of behaviours in terms of such primitives. The low-level representational currencies which have been employed range from the global representations of changes in scene content employed by Xiang and Gong in [14], to the local optic-flow based motion descriptors employed by Robertson and Reid in [11]. Sequences of such low-level primitives are typically represented using Hidden Markov Models (and variants thereof) or Bayesian Networks, which provide a powerful probabilistic framework for identifying anomalous behaviour.

A good overview of automated surveillance approaches can be found in [2]. Despite the diversity of models employed, all algorithms for identifying anomalous behaviour ultimately rely on quantifying the extent to which a new example of behaviour can be explained by a particular model. Thus, outside the unprecedented circumstance of having a large corpus of examples of anomalous behaviour (or having a rule based model eg. [1]), the detection of anomalous activity essentially corresponds to an unsupervised one-class learning problem, where the goal is to construct a definition of normal behaviour from a large set of examples of ordinary behaviour whose distribution - it is hoped - will yield a low probability for the abnormal activity which we wish to detect.

In this paper we propose an incremental semi-supervised learning framework for modelling the distribution of normal motion trajectories occurring in a particular scene. In the proposed framework, the approval of a human operator is requested before incorporating any behaviour pattern that appears novel with respect to the model. The underlying motivation for this is that, while it would be infeasible label every instance of normal behaviour used to build a model, it would still be desirable to have some control over the creation of such models. In particular it would be important to be able to prevent examples of anomalous behaviour being inadvertently incorporated through gradual repetition. Given that current automated surveillance approaches are intended to assist, rather than replace, human operators, it would be desirable to make use of the opportunity for occasional human feedback when training such systems.

Using the semi-supervised learning procedure proposed in Section 2.1, we show that only a very small cost in terms of human classification effort is required to filter the data needed to create a useful model of normal behaviour. This method could be regarded as providing a low cost “safety net” to prevent the possible inclusion of anomalies when training a behaviour classification algorithm. A key component of the proposed framework is a novel incremental one-class learning algorithm, described in Section 2.2 (originally proposed in [13]) which provides a means to gradually build a model of normal behaviour as new examples are added. This algorithm is trained on motion trajectories which are parametrised by the control points of cubic spline curves (see Section 2.3). While the proposed trajectory modelling/learning approach is in itself, novel, it is worth noting that the semi-supervised learning framework we demonstrate in this paper could be applied as a wrapper to any existing anomaly detection algorithm capable of incremental unsupervised learning.

2 Method

2.1 Semi-supervised Learning Framework Given a sequence of motion trajectories obtained from a pedestrian detection/tracking algorithm, we propose a method for incrementally building a model of motion patterns corresponding to normal behaviour, utilising occasional input from a human operator. At any stage of training, the model can be used to make predictions about whether or a not a new example is normal. In the proposed framework, illustrated in Figure 1, the approval of an human operator is requested before incorporating new motion patterns which are anomalous according to the existing model. Conversely, when a new example is classified as normal, it is automatically incorporated into the model.

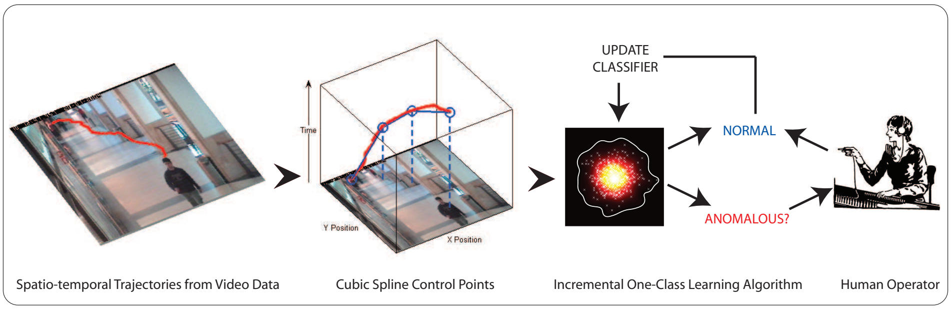

UPDATE CLASSIFIER
Time
Y Position
X Position
NORMAL
ANOMALOUS?
Spatio-temporal Trajectories from Video Data
Cubic Spline Control Points
Incremental One-Class Learning Algorithm
Human Operator

Figure 1: Self-training framework for (normal) motion trajectory modelling: Trajectories produced by a tracking algorithm are represented with vectors of cubic spline control points (see Section 2.3) and then assessed by a classifier. New examples classified as normal are automatically used to train the classifier (see Section 2.2), while anomalous examples are passed to a human operator for approval.

The proposed learning procedure provides a way of using both labelled and unlabelled examples to train a one-class classifier. This corresponds to a type of semi-supervised learning known as *self training*, which has previously been used with success in a variety of domains eg. [12]. Self training allows unlabelled data to be incorporated into supervised learning problems (such as one-class learning) by labelling it with the (high confidence) predictions of the classifier being trained. Here we are using the self training framework as a means of parsimoniously requesting labels from a human operator for the (predominantly normal) data used to build an anomalous trajectory classifier, by focusing only on the relatively unusual examples.

2.2 Incremental One-Class Learning Algorithm A key component of the proposed learning framework is an incremental one-class learning algorithm which can be trained on a sequence of examples and, at any stage during training, estimate the likelihood that a new example is normal. This is achieved by incrementally building a Gaussian mixture model to describe the underlying distribution of the data (ie. data labelled as normal by the operator/classifier), so that new examples with a sufficiently low likelihood with respect to the model can be flagged as potentially anomalous.

At the early stages of training we model the underlying data distribution by placing a Gaussian kernel function on each item of training data. This approach, known as kernel density estimation, has been shown to be an effective way of modelling unknown distributions for anomaly detection problems in a variety of domains [9]. At this stage, each Gaussian component has an identical covariance matrix $\Sigma(\sigma) = I^d \cdot \sigma^2$. In the absence of any information about the data, we initially set σ to an arbitrarily small value of 0.001 and, as more observations become available, we start to use a value determined by maximising the following leave-one-out likelihood function for the dataset[1]. (We use the median of the log likelihood values, rather than the sum as originally proposed by Duin [3], to prevent unnecessarily large values of sigma being estimated when the distribution of initial training data is unusually sparse.)

$$\sigma_{est} = \arg\max_{\sigma} \left[Median_{\{x_1,\dots x_N\}} \left(\log \left(\frac{1}{(2\pi\sigma)^{\frac{d}{2}}} \cdot \frac{1}{N-1} \cdot \sum_{\forall x \neq x_n} e^{-\frac{||x_n - x||^2}{2\sigma^2}} \right) \right) \right] \tag{1}$$

As the size of the training set increases, the computational cost of evaluating new data w.r.t. the model (which scales linearly with the number of training examples N) reaches a maximum feasible limit ($N = N_{max}$). At this point, whenever a new component (ie. a new datapoint $\mu_{new} = x_{new}$ with covariance matrix $\Sigma_{new} = I^d \cdot \sigma^2_{N_{max}}$ and weight $w_{new} = \frac{1}{N+1}$) is added, a pair of components (which may include the new one) are merged. We choose the pair of components $G_i = \{\mu_i, \Sigma_i, w_i\}$ and $G_j = \{\mu_j, \Sigma_j, w_j\}$ to merge (see [13] for details) by minimising the following information-theoretic cost function, as proposed by Goldberger and Roweis in [4].

$$cost(G_i, G_j) = w_i KL(G_i || G_{merge(i,j)}) + w_j KL(G_j || G_{merge(i,j)}) \tag{2}$$

This cost function is a weighted combination of the Kullback-Leibler divergences between each member of a hypothetical pair of Gaussian components and their merged counterpart. The KL divergence represents the expected information loss per sample when replacing one distribution with another, and can be easily calculated for a pair of Gaussians:

$$KL(G_p || G_q) = \frac{1}{2} \left(\log \frac{|\Sigma_q|}{|\Sigma_p|} + \mathrm{Tr}(\Sigma_q^{-1} \Sigma_p) + (\mu_p - \mu_q) \Sigma_q^{-1} (\mu_p - \mu_q)^T - d \right) \tag{3}$$

The proposed mechanism allows us to incrementally build a Gaussian mixture model to represent the class of normal data, while placing minimal constraints on the complexity of the model. To detect anomalies, we use a method proposed by Roberts et al. in [10] in which the Gumbel distribution is used to place novelty thresholds on a Gaussian mixture model. The key insight is that when drawing samples from a Gaussian distribution, the expected distance of the most extreme sample changes according to the number of observations made. This results in a cumulative probability function (Equation 4) for the distance of the most extreme value out of N samples drawn from a multivariate Gaussian G (where $D_M(z)$ is the Mahalanobis distance of a vector z from G, and N is the number of observations used to estimate G). This function can therefore be used to determine if a new example is anomalous (see next paragraph).

[1] Every time a new observation is added, we re-estimate σ by searching over a range of values. We phase in the estimated value of σ according to the number of unique distances between data points observed so far: $\sigma_N = 0.001(1 - \frac{N(N-1)}{N_{max}(N_{max}-1)}) + \sigma_{est} \cdot (\frac{N(N-1)}{N_{max}(N_{max}-1)})$ where N_{max} is the maximum number of kernels that will be added to the model, and $N \leq N_{max}$ is the number of kernels currently added.

$$P(D_M(z)|N) = \exp\left(-\exp\left(-\frac{D_M(z) - \left((2\ln N)^{\frac{1}{2}} - \frac{\ln\ln N + \ln 2\pi}{2(2\ln N)^{\frac{1}{2}}}\right)}{(2\ln N)^{\frac{1}{2}}}\right)\right) \tag{4}$$

The extent to which a new example can be explained by the model can hence be quantified by determining its Mahalanobis distance w.r.t. the closest component in the model, and then evaluating Equation 4. However, in order to classify new examples as normal or anomalous, an arbitrary threshold must be set. Preliminary experiments have led us to set a threshold at $P = 0.8$. Although the threshold is arbitrarily chosen, the advantage of this method is that it allows a single threshold value to effect more/less conservative classification boundaries for components representing fewer/more observations. To prevent the possible inclusion of anomalies in the semi-supervised framework, a more conservative threshold could be set for self-training on unlabelled data.

2.3 Trajectory Representation The proposed learning algorithm requires each observation to be encoded with a vector of fixed length – this poses a problem as each motion trajectory consists of coordinate sequences of arbitrary length/time. We therefore need a representation capable of encoding both the shape and spatiotemporal profile of a trajectory in a consistent parametric form: this is achieved by approximating each spatiotemporal trajectory with a uniform cubic B-spline curve parametrised by time and defined by a set of 7 control points ($d = 3$, $p = 7$).

$$S(t) = \{X(t), Y(t)\} = \{\sum_{i=0}^{p-1} C_i^X B_{i,d+1}(t), \sum_{i=0}^{p-1} C_i^Y B_{i,d+1}(t)\} \tag{5}$$

Every control point $C_i = \{C_i^X, C_i^Y\}$ corresponds to a B-spline basis function $B_{i,d+1}(t)$ which is defined by a knot vector[2] $\vec{\tau}$ and the following recursive formulae:

$$B_{i,1}(t) = \begin{cases} 1 & \text{if } \tau_i \leq t < \tau_{i+1} \\ 0 & \text{otherwise} \end{cases} \qquad B_{i,m}(t) = \frac{t-\tau_i}{\tau_{i+m-1}-\tau_i} B_{i,m-1} + \frac{\tau_{i+m}-t}{\tau_{i+m}-\tau_{i+1}} B_{i+1,m-1} \tag{6}$$

An approximation for the coordinates $\vec{X} = \{x_1, \dots x_N\}$ and $\vec{Y} = \{y_1, \dots y_N\}$ of a pedestrian at times $\vec{T} = \{t_1, \dots t_N\}$) can be expressed in terms of unknown control points $\vec{C^X}, \vec{C^Y}$ and an $N \times p$ matrix Φ where $\Phi_{n,i} = B_{i,d+1}(t_n)$, so that $\vec{X} \approx \Phi\vec{C^X}$ and $\vec{Y} \approx \Phi\vec{C^Y}$. Thus the control points which minimise the sum of squared errors between the original trajectory and its approximation can be found using the Moore-Penrose pseudoinverse operator $\Phi^\dagger = (\Phi^T\Phi)^{-1}\Phi^T$ as follows:

$$\vec{C^X} = \Phi^\dagger \vec{X} \quad \vec{C^Y} = \Phi^\dagger \vec{Y} \tag{7}$$

To represent each trajectory we find $\vec{C^X}$ and $\vec{C^Y}$ using a normalised time sequence $\frac{1}{t_N}.\vec{T}$ so that $\Phi_{n,i} = B_{i,d+1}(\frac{t_n}{t_N})$, and construct the following 15 dimensional vector, where the

[2] For a uniform cubic spline defined on the interval $t \in [0,1)$ with 7 control points, $\vec{\tau} = \{0,0,0,0,\frac{1}{4},\frac{1}{2},\frac{3}{4},1,1,1,1\}$. See Gueziec et al. [5] for an overview of spline fitting.

final element is the total time taken. Figure 2 shows an example of a trajectory and its reconstruction from the proposed representation, and illustrates the 7 basis functions defined by Equation 6. A key assumption of our approach is that differences between trajectories will be reliably reflected by distances in this representational space.

$$\begin{bmatrix} x_1 & y_1 & t_1 \\ \vdots & \vdots & \vdots \\ x_N & y_N & t_N \end{bmatrix} \rightarrow \{C_1^X, \ldots, C_7^X, C_1^Y, \ldots, C_7^Y, t_N\} \tag{8}$$

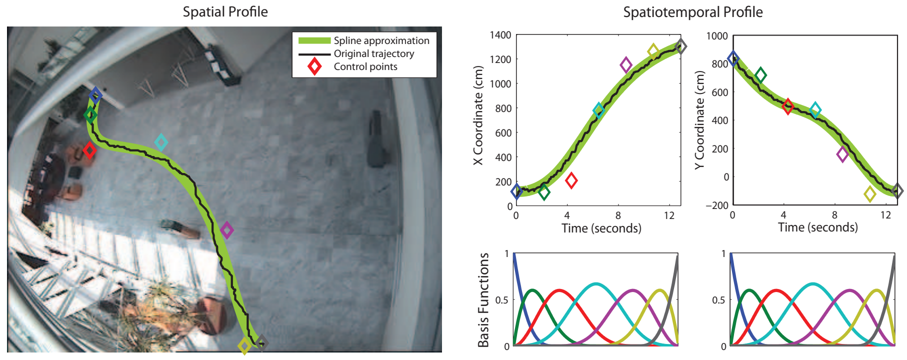


Figure 2: Spatiotemporal trajectory approximation using cubic B-spline curves.

3 Experiments

In this section we demonstrate the proposed trajectory learning algorithm on a two different datasets, and show that it is possible - with only a small number of operator interventions - to train a classifier that correctly recognises a large proportion of normal behaviour without misclassifying any anomalies. In each instance we define a classification problem with a small test set of motion trajectories corresponding to normal and unusual activities. We then use another larger set of normal trajectories to incrementally train our algorithm, and measure classification performance on the test set at every training iteration. For each dataset, we use the same settings for the learning algorithm: a total of $N_{max} = 100$ Gaussian kernels are added to the model before merging commences, and we set a classification threshold at $P = 0.8$ (see Section 2).

3.1 CAVIAR "INRIA" Dataset The publicly available CAVIAR dataset[3] consists of video footage and tracking data for a range of behaviours performed by actors in the entrance lobby of INRIA Labs, and contains around 60 complete tracks. We selected a subset of 21 tracks to represent normal behaviour, consisting of people walking directly from one exit point to another. We then selected a further subset of 19 tracks to define anomalous behaviour, consisting of actors fighting, falling down, and leaving/collecting packages. Although this data, shown in Figure 3, clearly encapsulates the type of problem

[3] Available at: `http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/`

an anomalous trajectory detection algorithm should be able to solve, it is not sufficient for testing our algorithm as there are no further examples available for training.

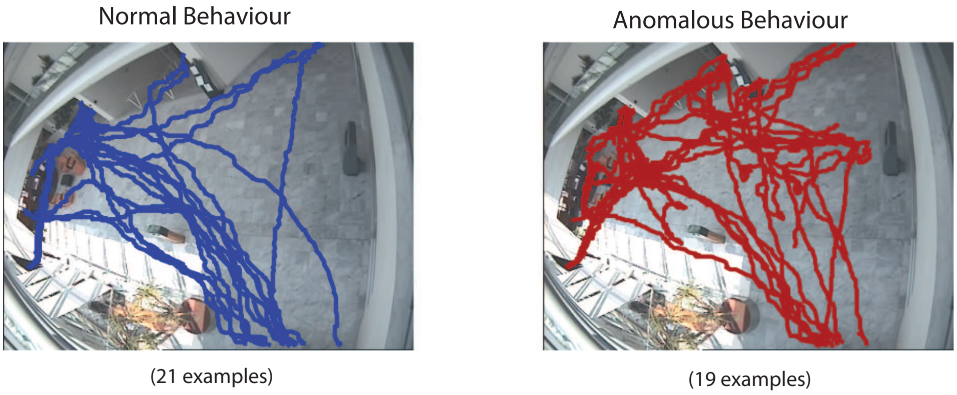


Figure 3: Test trajectories from the CAVIAR dataset.

We address the absence of additional training data by simulating a large set of ordinary walking behaviour between the entry/exit points featured in the test sets. For each pair of entry/exit locations, a route is hand-defined by a set of elliptical regions which represent entry/exit points and way points. This allows us to generate a diverse collection of possible paths by drawing sets of samples from these regions. Each set of samples is then interpolated to form a series of subgoals, which are used to generate a realistic trajectory in conjunction with the model for instantaneous pedestrian dynamics proposed by Helbing and Molnar in [6]. We define a route model for each of the 11 entry-exit pairs which appear in the test data and generate 100 simulated tracks for both traversal directions of each route, resulting in a total of 2200 tracks. Figure 4 shows the elliptical regions defining one of the 11 routes (in ground plane coordinates) between two exit points and a random subset of simulated tracks (10 from each route) projected onto the image plane.

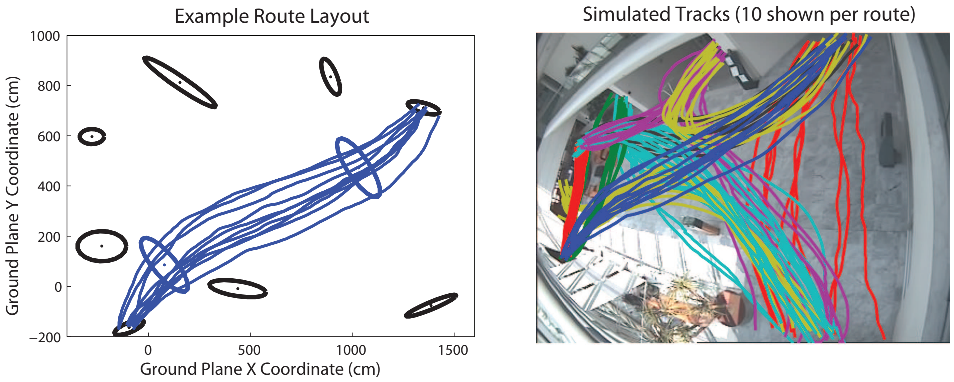


Figure 4: Normal behaviour simulation for CAVIAR scenario. Left: ellipses indicate all entry/exit points, and way points for a given route. Right: example simulated tracks coloured by route.

The proposed algorithm was trained on 10 different random orderings of the simulated tracks. As each (simulated) training example is added, three things are measured: the proportion of the (real) normal test data correctly recognised as normal (True Positive rate); the proportion of (real) anomalous test data misclassified as normal (False Positive rate), and the proportion of the last 20 training examples that would have required human approval before being incorporated (Intervention rate). The left-hand plot in Figure 5 shows how these measures (averaged over 10 trials) change as more data is added to the model: classification performance steadily improves until, at the end of training, an average True Positive rate of $TP = 80\%$ $(\pm 5.85\%)$ is achieved, with an average False Positive rate of $FP = 0\%$. Varying the classification threshold (on the cumulative probability - given by Equation 4 - of a given instance belonging to its closest mixture component) yields the ROC curve shown on the right hand side of Figure 5, which shows that a maximum value of $TP = 83\%$ $(\pm 4.05\%)$ can be achieved while $FP = 0\%$.

The intervention rate drops very rapidly, with 75.5 ± 1.9 interventions requested during the first 100 training iterations, but only 63.3 ± 23.9 interventions during the remaining 2100 iterations - corresponding to an intervention rate of 3% $(\pm 1.1\%)$. For this dataset, it is clear that only a small proportion of the training data needs to be labelled to achieve a high level of classification performance.

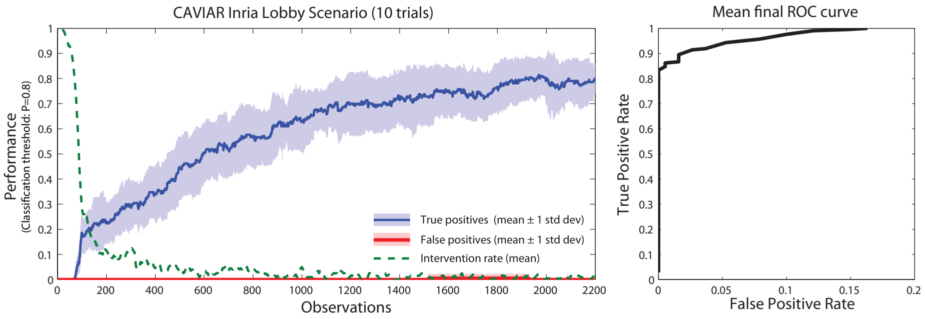
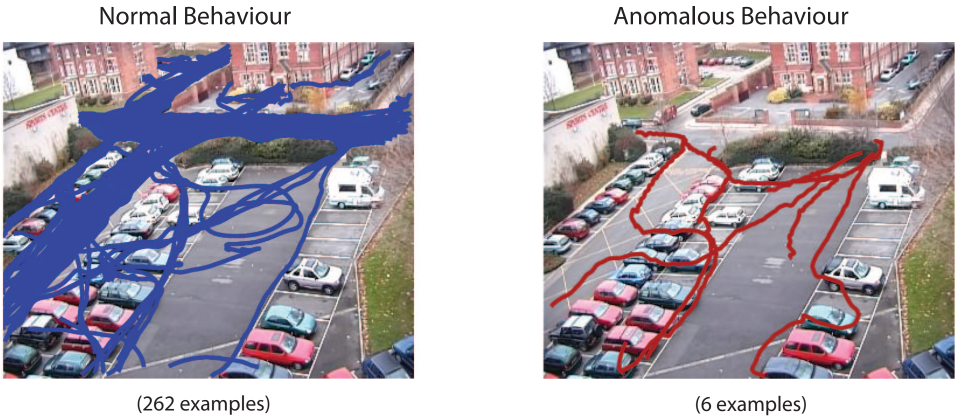


Figure 5: Classification performance on the CAVIAR dataset. See text for description (note that the False Positive rate in the left hand plot rarely exceeds zero.)

3.2 Carpark Dataset While the simulated normal tracks used in Section 3.1 allowed us to correctly classify real examples of normal and unusual behaviour, it is important to ascertain that similar behaviour would result when training on real tracking data instead of simulations. To establish this we train/test the algorithm on an additional (real) dataset, shown in Figure 6, which consists of 262 trajectories documenting ordinary behaviour taking place in a car park scenario, and a further set of 6 deliberately circuitous trajectories corresponding to the behaviour of actors[4]. While this is still relatively small set of data, it affords us the possibility of examining the classification performance/intervention rates attained during the early stages of training with the proposed algorithm. We split the data so that 235 (ie. 90%) of the normal examples are used for training our algorithm and 27 are retained for testing, along with the 6 anomalous examples.

The algorithm is trained for 10 different training/testing permutations of the available normal data and, as for the previous dataset, performance measures are taken at each

[4] This dataset, kindly donated by Hannah Dee, was originally used for behaviour classification in [1].



(262 examples)

(6 examples)

Figure 6: Trajectories from the carpark dataset.

training iteration. Figure 7 illustrates the changes in performance observed: classification performance steadily improves, reaching $TP = 72.2\%$ ($\pm 10.3\%$) after 235 training iterations (it is interesting to note that for the CAVIAR dataset $TP = 24.7\%$ at this point), with $FP = 0\%$. The ROC curve obtained at the end of training indicates a maximum of $TP = 75.6\%(\pm 10.7\%)$ can be achieved while $FP = 0\%$. The intervention rate drops steadily, albeit more slowly than for the CAVIAR dataset, with 65.5 ± 3 interventions occurring during the first 100 training iterations, and 46.6 ± 12 during the remaining 135. A total of 112.1 ± 11.7 interventions occur during 235 training iterations, compared to 93.7 ± 5.2 during this period for the CAVIAR dataset.

While it would be unwise to extrapolate these observations any further, they certainly do not rule out the possibility that the same low intervention rate observed for the CAVIAR dataset might - given a larger set of training examples - be achieved in reasonable time. At the very least, this data has provided an indication that the proposed trajectory representation/learning approach can be effectively applied to real data as well as simulations. It is additionally worth noting that the same learning/classification parameters have resulted in good classification performance for two different datasets, indicating the efficacy of the underlying learning algorithm.

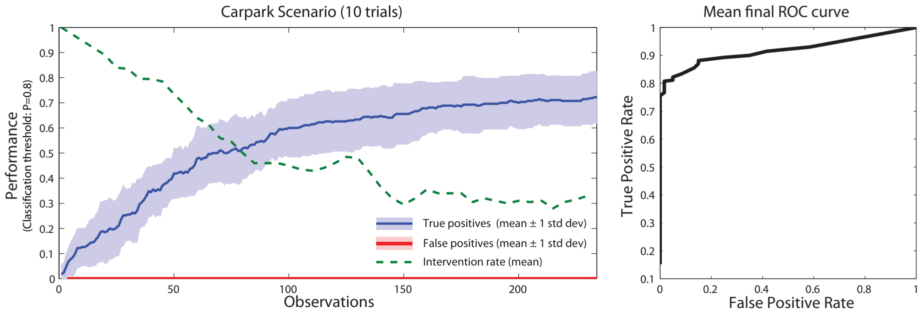


Figure 7: Classification performance on the carpark dataset. See text for description (note that only 235 training examples are available as opposed to the 2200 used in Figure 5.)

4 Conclusion

We have demonstrated a novel human-assisted learning/classification framework for identifying anomalous behaviour on the basis of motion trajectories. Using a novel incremental one class learning algorithm to model the distribution of typical motion trajectories, we have demonstrated a mechanism that potentially allows a human operator to train an anomaly detection classifier by providing very occasional interventions. A key criticism that could be made is that the proposed method provides no principled safeguards against anomalies being incorporated into the learning algorithm. However, if regarded as an alternative to an entirely unsupervised learning algorithm, it is clear that our method could at worst perform equivalently to such algorithms. In this vein, it is interesting to note that the proposed learning framework could potentially be used as a wrapper for existing unsupervised behaviour modelling algorithms. Future work seeks to further elucidate the benefits of the proposed system through extensive testing on larger real-world datasets.

Acknowledgments

RRS was supported by an EPSRC/MRC Neuroinformatics DTC studentship. The authors would like to thank Hannah Dee for providing valuable data.

References

[1] H. Dee and D. Hogg. Detecting inexplicable behaviour. In *Proc. BMVC*, pages 477–486, 2004.

[2] H. Dee and S. Valestin. How close are we to solving the problem of automated visual surveillance? *Machine Vision and Applications*, 2007.

[3] R. Duin. On the choice of smoothing parameters for Parzen estimators of probability density functions. *IEEE Trans. Computers*, C-25:1175–1179, 1976.

[4] J. Goldberger and S. Roweis. Hierarchical clustering of a mixture model. In *Advances in Neural Information Processing Systems 17*, pages 505–512. 2005.

[5] A. Gueziec and N. Ayache. Smoothing and matching of 3D space curves. *International Journal of Computer Vision*, 12:79–104, 1994.

[6] D. Helbing and P. Molnar. Social force model for pedestrian dynamics. *Physical Review E*, 51:4282, 1995.

[7] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and S. Maybank. A system for learning statistical motion patterns. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28:1450–1464, 2006.

[8] N. Johnson and D. Hogg. Learning the distribution of object trajectories for event recognition. *Image and Vision Computing*, 14:609–615, 1995.

[9] M. Markou and S. Singh. Novelty detection: a review - part 1: statistical approaches. *Signal Processing*, 83:2481–2497, 2003.

[10] S. Roberts. Novelty detection using extreme value statistics. *IEE Proceedings - Vision, Image & Signal Processing*, 146(3):124–129, 1999.

[11] N. Robertson and I. Reid. A general method for human activity recognition in video. *Computer Vision and Image Understanding*, 104:232–248, 2006.

[12] C. Rosenberg, M. Hebert, and H. Schneiderman. Semi-supervised self-training of object detection models. In *Proc. IEEE WACV/MOTION*, pages 29–36, 2005.

[13] R. R. Sillito and R. B. Fisher. Incremental one-class learning with bounded computational complexity. In *Proc. ICANN (LNCS 4668)*, pages 58–67, 2007.

[14] T. Xiang and S. Gong. Video behavior profiling for anomaly detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 30:893–908, 2008.