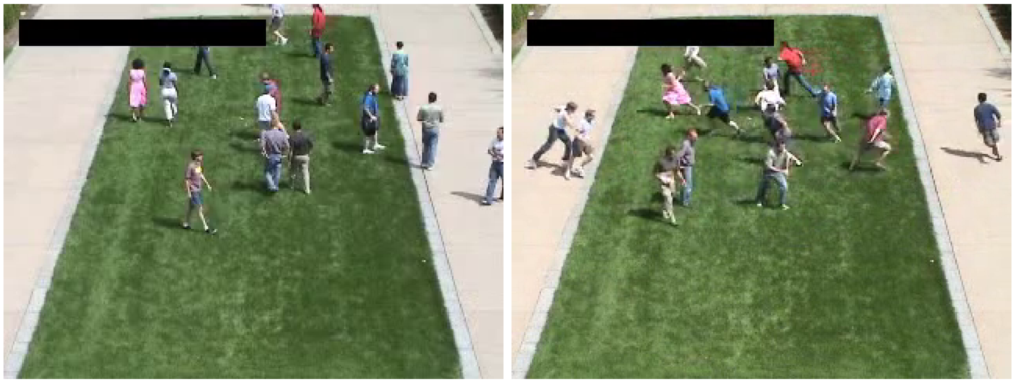
16-831 Final Report:
Online Anomaly Detection in Videos

Allie Del Giorno, Hanzhang Hu, Nick Rhinehart

December 12, 2014

1 Introduction

This project address the problem of detecting interesting or anomalous frames in lengthy surveillance videos in an online unsupervised manner. The motivation of this project is based on the fact that long stretches of video are becoming a prevalent data source. It is estimated that there are 210 million cameras in use today, generating over 1 billion gigabytes of data per day [1]. These lengthy videos pose a challenge for anomaly frame detections because first, it is difficult to acquire ground truth labels for the hours long lengthy videos for anomaly detection. Second, labeled anomalous frames may not be beneficial for test-time predictions since new anomalies tend to differ from known scenes. Third, anomaly frames are rare, making training data highly imbalanced. Last, learning with video data is difficult because of the dataset size. To meet these challenges we explore online unsupervised learning methods for anomaly detection. In this work we aim to form models of normality and use them to detect anomalous, or novel, frames.



(a) Normal activity

(b) Anomalous activity

Figure 1: Two frames of the `ca01` sequence [1]

1.1 Possible Problem Formulation

More formally, given a video clip, an anomaly detector scores each frame of the clip with a real number, or alternatively provides a binary classification of the frame as “anomalous” or “normal”. There are multiple ways to formulate the learning steps. First, we can view the anomaly detection as a novelty detection problem, in which we train the detector with known normal video frames to report novel/abnormal scenes. This approach is weakly supervised, in that we need not provide labels for all of the types of normal events in a video; instead we need only initialize the framework with normal data. Second, we can view the detection as a classification or filter problem, in which we train with known anomalous and normal frames to score future frames. This approach is less appealing, as it is quite constraining to assume what types of anomalies

[1] We overlayed black boxes in the top corner, as the original videos overlay a banner when the frame is labelled as anomalous

will occur. Lastly, we can leverage the intra-class similarity and inter-class dissimilarity of normal and abnormal frames to form unsupervised normal and abnormal clusters. In the work we present, we perform online learning using the first and last of the above views of anomaly detection and compare their results.

1.2 Dataset and Feature Set-up

We use the University of Minnesota Unusual Crowd Activity Dataset [2] as our starting dataset. It contains 10 scenes of short clips of normal crowd activities turning into abnormal ones: in each clip, a fixed birds eye camera records that people first wander randomly in a public area, such as hallway or lounge, and then suddenly run out of the scene.

We consider each frame of the video clips as a sample point, and compute *Dense Trajectory*[4] as its base feature. More specifically, Dense Trajectory views a video as a 3-D array of pixels, and uses Dense Optical Flow to compute trajectory of pixel patches at interest points. Afterwards, Dense Trajectory computes for each patch trajectory its 3-D histogram of gradient (HOG), histogram of dense optical flow (HOF), and motion boundary histograms (MBH). To form a fixed length descriptor of a video frame, we perform a K-means-hard-quantization on each of the three basic features, HOG, HOF, and MBH, to form three bags of visual words (BOV), which are concatenated to be the final descriptor.

To better analyze the features, we compute RBF and uniform kernels between frames of one clip, as shown in Figure 3, and notice strong intra-class similarity and inter-class dissimilarity of normal and abnormal frames.

2 Approaches

An intuitive way to build an anomaly detection system is to build borders around "normal" things and classify everything outside as abnormal. We begin with this approach. Frame descriptors that lie within the area the SVM classifies as "normal" will also be marked normal, but those outside will be classified as abnormal. The SVM can also give us a continuous-valued score that reveals how far the point is from the classification border. We began with standard one-class SVM, and extend it with kernel functions.

2.1 One-Class SVM

We can formulate the normality model building problem as a one-class Support Vector Machine (SVM) problem, which is formulated as follows:

$$\min_{w,\xi,\rho} \frac{1}{2}\|w\|^2 + \frac{1}{\nu n}\sum_{i=1}^{n}\xi_i - \rho, \qquad \text{such that} \tag{1}$$

$$w^T x_i \geq \rho - \xi_i \qquad \text{and} \qquad \xi_i \geq 0 \qquad \text{for all } i = 1, \ldots, n \tag{2}$$

This looks similar to the standard two-class SVM, except that the label of the data point is not required, and the margin is a parameter rather than a fixed value. Because the margin can move, we can loosen the margin if the data is difficult to classify, but we pay a penalty in the loss function. In this setting, a continuous-valued score can be generated by computing the value $w^T x - \rho$. A prediction is classified as anomalous if this value is less than 0, and normal otherwise. Intuitively, this score can be thought of as how likely it is that the new data point belongs the class of points seen before. Because this is a convex loss function with linear constraints, it can be solved in an online fashion. In the online setting, each point gets "added" to the SVM after a prediction score is given to it.

2.1.1 NORMA Online Kernelized One Class SVM

Following Figure 1 of [3], we extend one-class SVM with kernels and online updates. In particular, we found that a proper-bandwidth RBF kernel (see Figure 3a) performs well for distinguishing novel scenes. At each step, we first evaluate for the new sample x its prediction $f(x)$, using the kernel trick of $f(x) = \sum_i \alpha_i K(x_i, x)$. Then, if the new sample is detected as novel ($f(x) < \rho$), we record the sample and its associated weight α, which is based on the gradient of loss w.r.t to current f. In general, since we may not keep track of all

Trial ID	Precision	Recall	Accuracy	#Train	#Test	Test Anomalies	Test Normalities
Data from Figure 4a	0.653	1.000	0.716	100	500	91	409
Data from Figure 4b	0.732	1.000	0.813	300	300	91	209
Data from Figure 4c	0.103	1.000	0.783	480	120	91	29
Data from Figure 4d	0.841	0.995	0.875	480	920	202	718
Data from Figure 5a	0.342	1.000	0.411	100	540	57	483
Data from Figure 5b	0.862	1.000	0.885	300	340	57	283
Data from Figure 5c	0.592	1.000	0.738	480	160	57	103
Data from Figure 5d	0.651	1.000	0.695	480	820	104	716

Figure 2: Results table

previous samples, we can apply a decaying weight to each recorded sample, so that we only consider samples within a recent window of time. The parameter to tune in this algorithm is one-class SVM novelty parameter ν, regularization constant λ, learning rate η, and the RBF kernel bandwidth σ_{RBF}.

2.2 Online Mean-shift Algorithm

Since abnormal frames usually appear in short bursts, their frame feature are in general very different from normal frame features. Thus, in an online clustering setting, the abnormal frames usually creates new cluster centers, or are close to recently created novel cluster centers. We leverage this intuition to detect abnormal frames by designing an online mean-shift algorithm.

In classic mean-shift algorithm, we initialize a number of seeds of modes of the feature distribution, and we iteratively update each seed by an average of its surrounding points weighted by a proximity measure between the seed and these points until the seeds converge as follows:

$$\hat{x}^{(t+1)} = \frac{\sum_i^N x_i K(x_i, \hat{x}^{(t)})}{\sum_i^N K(x_i, \hat{x}^{(t)})}, \tag{3}$$

where K is a kernel function, x_i's are data samples, and $\hat{x}^{(t)}$ is a seed of mode indexed by iteration number t.

We augmented this classic mean shift algorithm to be an online clustering algorithm. Classic mean shift requires all data points to detect the modes of the distribution of data. To avoid this requirement, we keep a constant number of random samples from the seen data point. To achieve this, we keep a W number of stored samples, and when we see sample number s, we replace a random stored sample chosen with probability $1/s$ with the new sample. This procedure can be proven to guarantee that at each timestep s, all previous samples have equal probability to be stored in the buffered samples.

We then perform regular mean-shift algorithm on these stored samples along with a small batch of new data points to find the modes associated with the new points. After updating the modes location and weight, we report the anomaly level of the points based on the novelty of the modes: each learned mode has a weight equal to number of data samples associated with the mode, and the novelty of the mode is one minus the ratio of the weight of the assigned mode to the total weights of all modes.

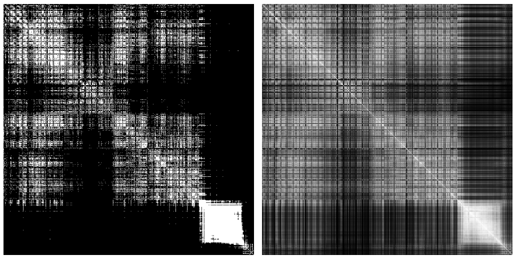
3 Experiments

3.1 One-Class SVM

We implemented both nonkernelized online one-class SVM and NORMA kernelized SVM, and were able to tune the NORMA parameters with an RBF Kernel to yield reasonable performance. We visualize our tuned kernels on the `ca01` sequence in Figure 3. The parameters used for all tests are

$$\nu = 1e^{-2}, \lambda = .3, \eta = 1e^{-3}, \sigma_{RBF} = 5e^{-3}$$

We found that if we let the algorithm learn with the anomalies, the significant intra-class similarity that the anomalies exhibit quickly bias the algorithm to treat new anomalies as looking very similar to previously seen data, and thus they are not properly classified. This is expected behavior of the algorithm, as it treats new



(a) Uniform Kernel (b) RBF Kernel

Figure 3: Kernels computed across all pairs of the `ca01` sequence. The dark stripe in the columns and rows to the right and bottom of the matrix represent the dissimilarity between the anomalous and non-anomalous sections of the video. The bright block in the lower right corner represents the similarity of anomalous frames with each other.

training examples as part of the normal class. Therefore, we train on some portion of the beginning data, which we assume is normal, and test on the rest. In Figures 4a, 4b, and 5a, the algorithm has not seen enough training data to properly classify future standard test data as non-anomalous. In Figures 4c, 5b, and 5c, we find that the algorithm has a strong belief that the test anomalous data it sees is indeed anomalous. In Figures 4d and 5d, we train on the normal data from sequences `ca01` and `ca09` (respectively), and test on the anomalous data from each sequence, as well as the full sequences of `ca02` and `ca10` (each of which corresponds to the same physical scene but different recordings). These experiments are good tests of precision, as the algorithm is presented with reasonable amounts of both anomalous and non-anomalous data. Statistics of these tests are presented in Figure 2, and we see that all tests yield perfect or near-perfect recall of anomalous events, and that the cross-sequence tests exhibit reasonable precision (0.841 and 0.651, respectively).

3.2 Online Mean-Shift

3.2.1 Synthetic Data

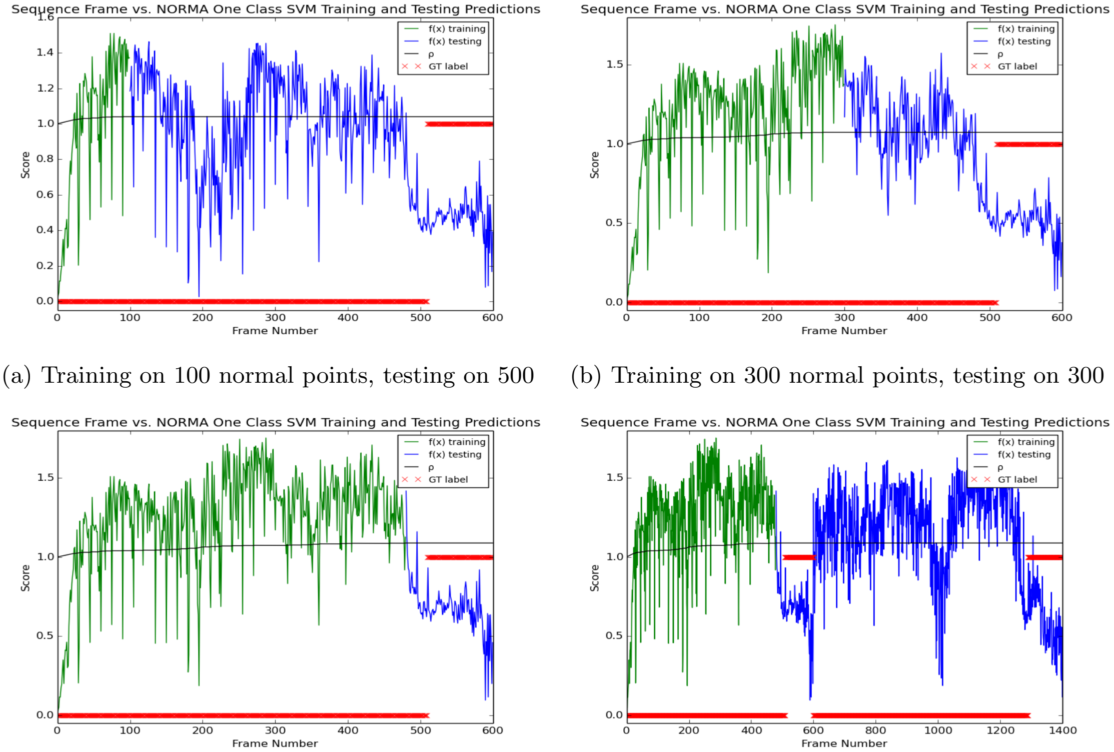
The synthetic data is shown and described in Figure 6. We generated data that allowed us to make sure the algorithm could learn a model for normality that contained two different normal clusters. We could also make sure the algorithm detected the first few instances of a new cluster as abnormal. The synthetic data are generated from three clearly separated Gaussian in 3D space, and we feed them to our online mean-shift to find the modes of the distributions, as shown in Figure 6b.

3.2.2 Crowd Activity Dataset

To detect anomaly frames in a video, we not only need to find modes of the frame distribution a frame is closest to, we also need information from the mode about its novelty. We use the ratio of the number of points associated with the mode to the total number of points as the normality score of the mode. This approach is good if the data has a clear main mode, explaining most of the normal data, and the abnormal frames are located significantly far away from this main mode. It can fail if the normal data contains multiple low weight modes.

We again use the Crowd Activity Dataset to experiment our method. In Figure 7, we show the unsupervised learning results of our online mean-shift algorithm. Blue lines represent the anomaly score of frames, and red lines are the ground truth anomaly score (0.25 for normal, 0.75 for abnormal for display purpose). The black line is the threshold for classifying a frame as anomaly. It is interesting to note that novel scenes have high anomaly scores, but the score diminishes if we continue to observe similar scenes. This is a beneficial feature of our unsupervised detection, because the algorithm reports novel scenes while having the capability to learn the scene is normal after enough learning. On scene 01 and scene 09, online mean-shift achieves 74.83% and 76.71% accuracy respectively.

The parameters to tune in this algorithm are the RBF kernel width for mean-shift algorithm, and the width θ for deciding whether modes learned from different starting seeds are indeed one single mode. We



(a) Training on 100 normal points, testing on 500

(b) Training on 300 normal points, testing on 300

(c) Training on 480 normal points, testing on 120

(d) Training on 480 normal points, testing on 900

Figure 4: Training NORMA SVM on normal data from `ca01` and testing it on normal and anomalous data. In Figure 4d, the additional data comes from `ca02`. The `GT` line is 0 for normal data, and 1 for anomalies. The classification as anomalous is given by $f(x) < \rho$.

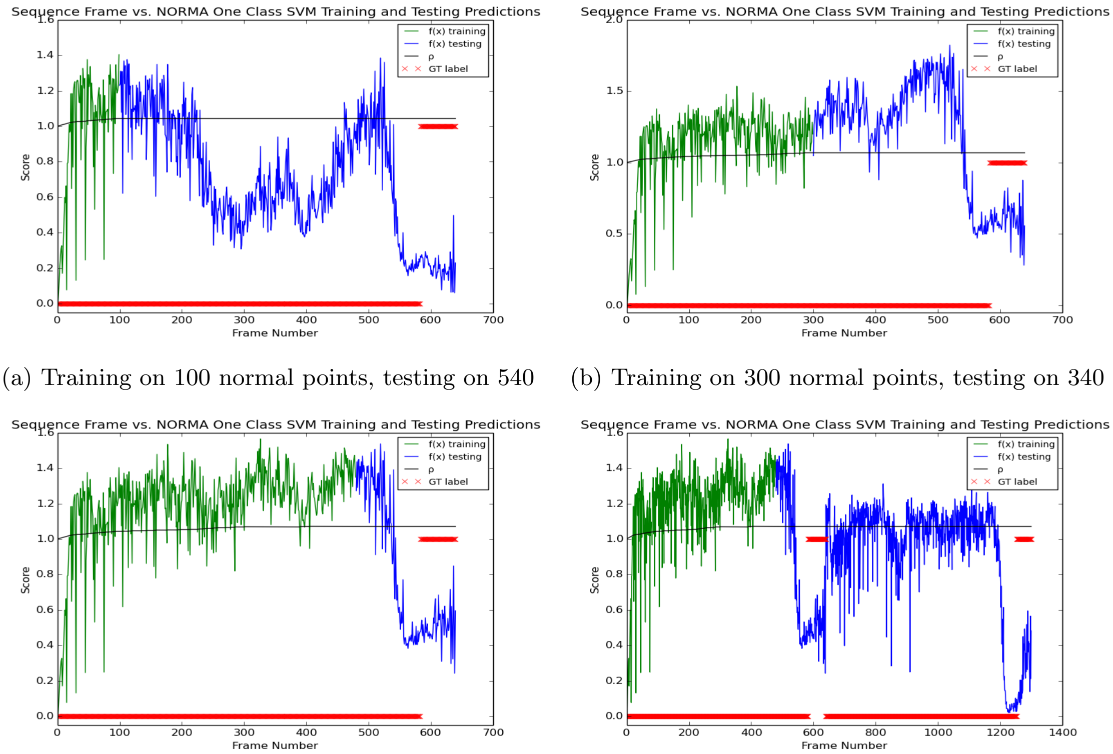
chose the median of pairwise feature Euclidean distances as the kernel width and we used a hold-out video scene to tune θ.

3.3 Comparison

While both one-class SVM and mean-shift can be used to detect anomaly frames in videos as shown in previous figures, they have some difference in behavior. One-class SVM requires some training data that contains only the normal frames, so it is not entirely unsupervised, unlike the mean-shift-based algorithm. After deployment, one-class SVM also cannot learn that a novel scene is normal if we are given enough examples from the scene.

4 Future Work

Our anomaly score assignment in online mean-shift based algorithm is crudely based on the weight of each learned distribution mode. This score assignment can be further improved with learning-based methods. Both of our one-class SVM and mean-shift algorithms assume that normal frames form a dominate region in the full feature space. While this assumption seemed to hold in our investigated data-set, it is not, however, guaranteed in all datasets.



(a) Training on 100 normal points, testing on 540

(b) Training on 300 normal points, testing on 340

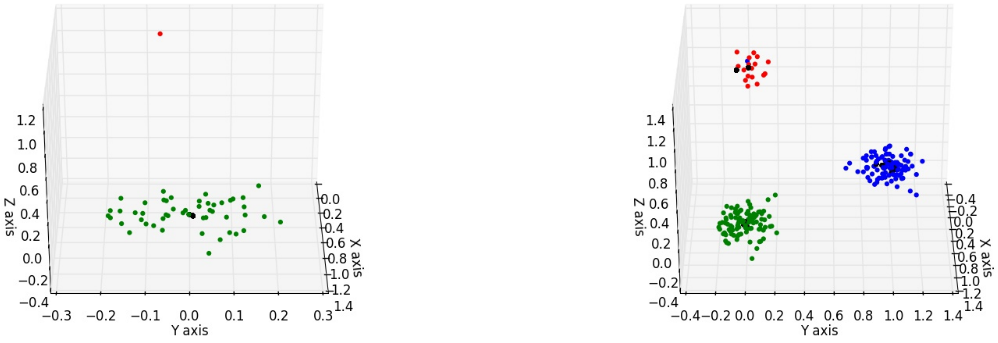
(c) Training on 480 normal points, testing on 160

(d) Training on 480 normal points, testing on 924

Figure 5: Training NORMA SVM on normal data from `ca09` and testing it on normal and anomalous data. In Figure 5d, the additional data comes from `ca10`. The `GT` line is 0 for normal data, and 1 for anomalies. The classification as anomalous is given by $f(x) < \rho$.

5 Results and Discussion

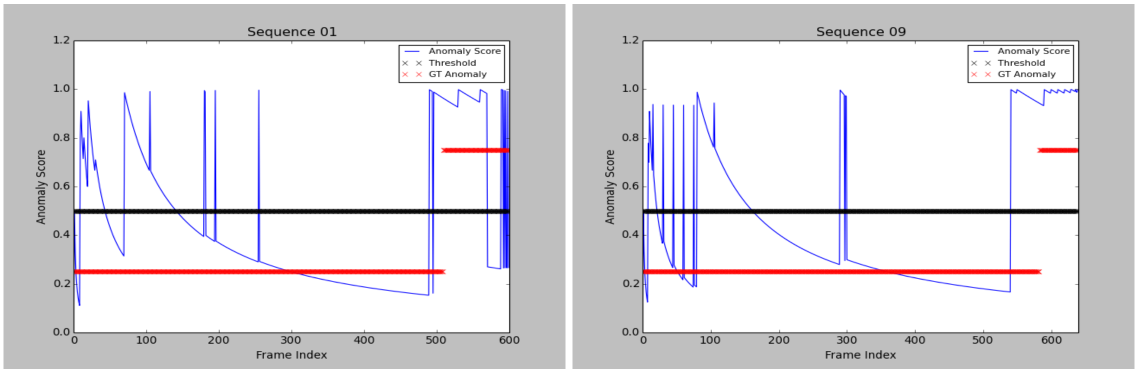
In this project we implemented two different online methods for anomaly detection in videos, using one-class SVM and clustering respectively. Our methods have shown to be effective in an easy real-world data-set.



(a) The first 51 data points

(b) All 220 data points

Figure 6: The synthetic data set with discovered cluster centers. Points are generated from three classes of Gaussians (red, blue, green). The dataset is, in sequential order, {50 green, 1 red, 50 blue, 50 green, 50 blue, 20 red}. All red points are considered anomalous in this example, and the first few blue points are also abnormal. Discovered cluster centers are marked in black.



(a) Unsupervised classification of sequence 01

(b) Unsupervised classification of sequence 09

Figure 7: Anomaly scores learned from our unsupervised detection algorithm based on online mean-shift algorithm. Different than in one-class svm, a frame of a score above the threshold is classified as anomaly. Novel scenes are initially with high anomaly score, and the score decays as we have more of this scene.

References

[1] http://storageservers.wordpress.com/2014/07/30/how-many-video-surveillance-cameras-are-there-in-thi

[2] http://mha.cs.umn.edu/Movies/Crowd-Activity-All.avi

[3] Kivinen, Jyrki, Alexander J. Smola, and Robert C. Williamson. "Online learning with kernels." Signal Processing, IEEE Transactions on 52.8 (2004): 2165-2176.

[4] Heng Wang et al. "Action Recognition by Dense Trajectories." International Journal of Computer Vision (2013) 103 (1): 60-79. http://lear.inrialpes.fr/people/wang/dense_trajectories

[5] Anna Choromanska and Claire Monteleoni. "Online clustering with experts." In International Conference on Artificial Intelligence and Statistics (2012), pp. 227-235.