EC719 Final Project: Group Anomaly Detection

Gregory Starr

May 15, 2017

1 Introduction

Anomaly detection is the problem of identifying behavior that deviates from the norm. Traditionally this involves finding individual points that are far removed from the clusters of a dataset. In contrast to point anomaly detection, group anomaly detection aims to identify groups of data points that display anomalous aggregate behavior. This is a more difficult problem because individual points within an anomalous group may not be spatially separated from the rest of the data. In this case it is the distribution of the points within the group that makes it anomalous.

Anomaly detection is an important problem because it can help identify special cases that scientists and specialists would want to study. Mechanisms that cause anomalies are often not entirely understood and by studying these special cases scientists could learn more about their fields. Anomalies are generally caused by very rare phenomena and manually searching for occurrences would be an infeasible task. This is why automated anomaly detectors are necessary to develop. Group anomaly detection is necessary because in some cases there are different learning opportunities when individual data points are considered in groups. Often times larger scale phenomena can be seen in the aggregate behavior of individual data points. For example, when looking at astronomical data, an anomalous group of galaxies may be the effect of some large scale gravitational interaction that scientists would benefit from studying. Another area which could benefit from group anomaly detection is the search for low-altitude ionization of the ionosphere, visible in measurements from incoherent scatter radar. Anomalies such as these don't show up as a single data point but as an aggregate behavior, and sifting through large databases of radar data is impossible by hand. Therefor group anomaly detection has the potential to benefit many parts of the scientific community.

While point anomaly detection has been studied extensively, group anomaly detection has not. Some previous attempts, such as in [1], have relied on first finding point anomalies, then if the number of anomalies in a group is higher than expected, the group is flagged as anomalous. The problem with this approach is that it assumes that individual points within anomalous groups will themselves be anomalous. There have also been a few attempts, namely in [2] and [5], to detect anomalies at the group level relying only on their distributions.

In [2], the authors attempt to encode all of the information from the distribution into a kernel-mean mapping from the distribution space into a reproducing kernel Hilbert space. This is convenient because then traditional, efficient inner-product based algorithms can be used to identify outliers. In [2], they apply the mapping, then use a one-class support vector machine (OCSVM) to find an efficient classifier, which they call the one-class support measure machine (OCSMM). Although they reported good results, they may have over estimated the effectiveness of the kernel-mean mapping.

In this paper, I try to improve upon [2], by incorporating a K nearest neighbors score based on the kernel-mean mapping. K nearest neighbor graphs have been used with good success in the point anomaly detection setting. I will be borrowing methods from [4], in which the authors use a euclidean distance K nearest neighbors as a measure of anomalousness. Then they rank the training data based on the KNN score and estimate a ranking function using a ranking SVM. Then, they are able to define a false alarm rate α and declare a new point η anomalous if it ranks below the α-percentile when compared to the training data. They then prove optimality for the resulting detector. See [4] for further explanation and proofs.

My method will try to reap the benefits of both [2] and [4] by replacing the euclidean distance in the K nearest neighbors graph with the kernel product of two groups. This will extend the results of [4] to work on groups of data points. I will show that this method significantly outperforms both a baseline test and the OCSMM on synthetic data. As this project is not finished I will then discuss possible future direction for the project.

2 Problem Statement

This section will set up the problem and will use the same notation as in [2] and [4]. I won't go into much detail about the problem setup, to see the full formulations and proofs, see [2] and [4].

Let $\mathcal{X}$ be the input space and $\mathfrak{P}_{\mathcal{X}}$ be the set of all probability distributions on $\mathcal{X}$. We assume that $\mathscr{P}$ is a distribution on $\mathfrak{P}_{\mathcal{X}}$ and $\mathbb{P}_1, ..., \mathbb{P}_\ell$ are distributions on $\mathcal{X}$ drawn i.i.d. according to $\mathscr{P}$. The group of points S_i has n_i points which are drawn i.i.d. according to $\mathbb{P}_i$. Therefor each group is defined as $S_i = \{x_k^{(i)}\}_{1 \leq k \leq n_i}$ for i = 1,...,ℓ. The main concept is that the nominal groups are generated frequently but with a certain variance between them according to $\mathscr{P}$. The goal if this problem is to find a minimum subset of $\mathfrak{P}_{\mathcal{X}}$ that contains as much probability mass as possible. In this way, we consider groups that are generated with very low probability anomalous. Instead of working in the space of distributions, I made use of the kernel mean map described in [2].

3 Methods

3.1 Kernel Mean Map

In order to work with the probability distributions, $\mathbb{P}_i$ is mapped into a reproducing kernel Hilbert space $\mathcal{H}$ by taking the expected kernel of the distribution.

$$\mu : \mathfrak{P}_{\mathcal{X}} \to \mathcal{H}, \qquad \mathbb{P}_i \to \int_{\mathcal{X}} k(x, \cdot) \mathrm{d}\mathbb{P}_i(x) \tag{1}$$

This is equivalent to taking the kernel function at each point in the input space and doing a weighted average of them using a probability distribution. You can think of $\mu_{\mathbb{P}_i}$ as just a feature representation of a group S_i associated with kernel K.

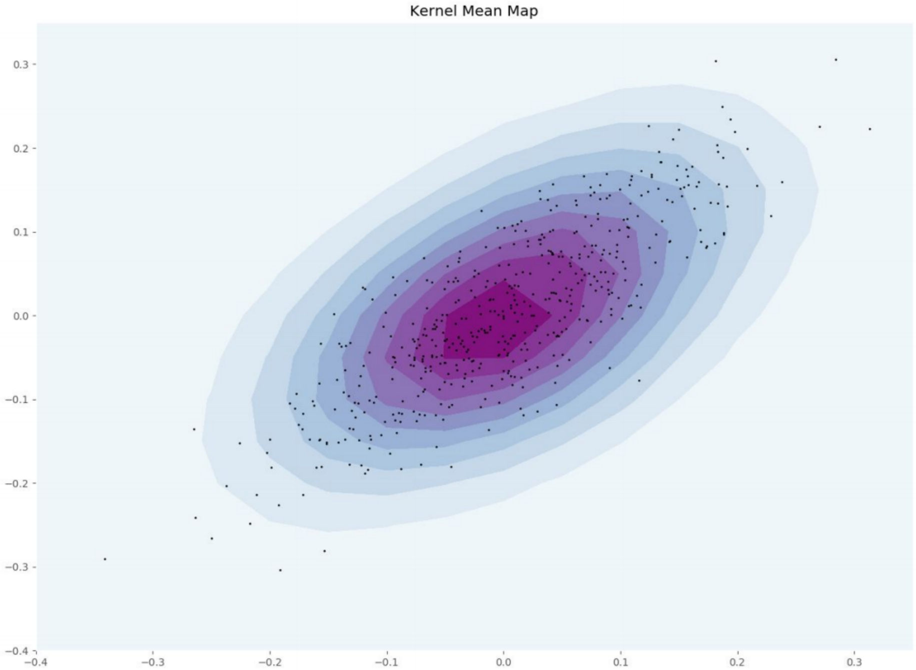

Figure 1: Visualization of kernel mean map. Each data point's kernel function is a 2D Gaussian distribution (RBF kernel). When the average of every sample's RBF kernel is taken, the resulting distribution approximates the distribution that the points were sampled from

Using the kernel mean map allows us to take inner products of probability distributions and approximate an inner product with groups of data points. The inner product of two distributions according to [2] is:

$$\langle \mu_{\mathbb{P}_i}, \mu_{\mathbb{P}_j} \rangle = \iint k(x,y) \mathrm{d}\mathbb{P}_i(x) \mathrm{d}\mathbb{P}_j(y) \tag{2}$$

and, letting $K(\mathbb{P}_i, \mathbb{P}_j) = \langle \mu_{\mathbb{P}_i}, \mu_{\mathbb{P}_j} \rangle$, can be approximated by:

$$K(\hat{\mathbb{P}_i}, \hat{\mathbb{P}_j}) = \frac{1}{n_i \cdot n_j} \sum_{k=1}^{n_i} \sum_{l=1}^{n_j} k(x_k^{(i)}, x_l^{(j)}). \tag{3}$$

One important result of [2] is that the kernel mean map preserves all of the information about the distribution, and therefor using the feature representation associated with the kernel incorporates all of the higher order statistics from the distribution.

For the lower level kernel k, called the embedding kernel in [3], I use the Gaussian RBF kernel which is parameterized by the bandwidth parameter σ and given by:

$$k_\sigma(x, x') = \exp\Big(- \frac{\|x - x'\|^2}{2\sigma^2} \Big) \tag{4}$$

One issue with the kernel mean map is that in order to distinguish different distributions with the same mean, the bandwidth parameter σ has to be very small (in my experiments I found $\sigma = .005$ worked well but this is dependent on the nominal distribution). With a small bandwidth parameter, the kernel product between two groups decays rapidly with displacement. When trying to reproduce the experiments on synthetic data in [2], I found that most of the time, an anomalous group distributed such that it straddled many nominal groups would have a medium-large kernel product with those groups, and that especially when the number of groups ℓ was low, the anomalous groups had higher kernel products than many of the nominal groups. This made it impossible for a one class support vector machine type algorithm to distinguish them.

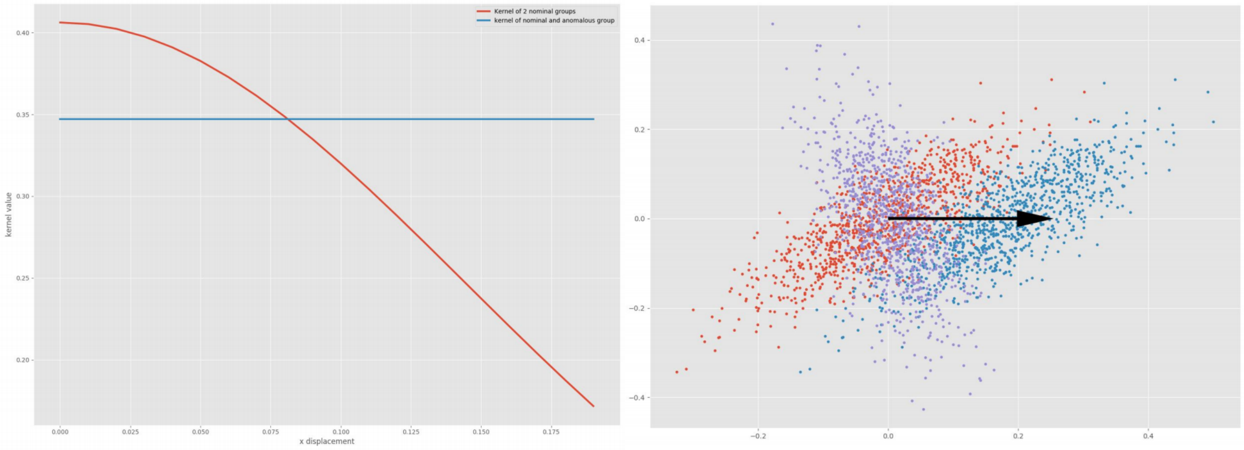

Figure 2: As the lateral displacement of two nominal groups (red and blue) increases, their kernel product decays rapidly causing them to be indistinguishable to the OCSMM

For this reason, other methods were required to make the kernel more robust.

3.2 KNN Scoring

The problem with the kernel was that when an anomalous group straddled several nominal groups, it had a moderate kernel product with all of them which made it look nominal to an OCSMM. In contrast, a nominal group usually overlapped with one or two groups substantially but not with any other groups. The intuitive solution to the kernel issue is to employ a K nearest neighbors score to determine how nominal a group is. let $\mathbf{S}$ be the set of training groups and let $D_{(i)}(S)$ be the ith highest kernel product that S has with another group in $\mathbf{S}$ not including itself. Then the "K nearest neighbors" score can be defined as:

$$G_{\mathbf{S}}(S) = \frac{1}{K}\sum_{j=1}^{K} D_{(j)}(S) \tag{5}$$

The anomalous groups who's probability density is spread across many groups will be penalized because only the first few highest kernel products can contribute. Since in this case, the group only has moderate kernel products with many groups, its $G_{\mathbf{S}}(S)$ value should be low. I then modify the score function from [4] to rank larger values of $G_{\mathbf{S}}(S)$ higher. The score function for a group η then becomes:

$$R_\ell(\eta) = \frac{1}{\ell}\sum_{i=1}^{\ell} \mathbf{1}_{[G_{\mathbf{S}}(\eta)<G_{\mathbf{S}}(S_i)]} \tag{6}$$

This score function quantifies the level of similarity to the training groups of a test group η. It is also the proportion of the training groups whose K nearest neighbors score is less than that of η. Following [4], η is flagged as an anomaly if $R_\ell(\eta) < \alpha$. In [4], the authors show that $R_\ell(\eta)$ is a consistent estimator of the p-value, so that by defining α, we also define a confidence level with which we identify anomalies.

4 Experiment

4.1 Setup

To show the effectiveness of the method described in this paper, I created synthetic testing and training sets and compared the accuracy and ROC curves of a few different methods.

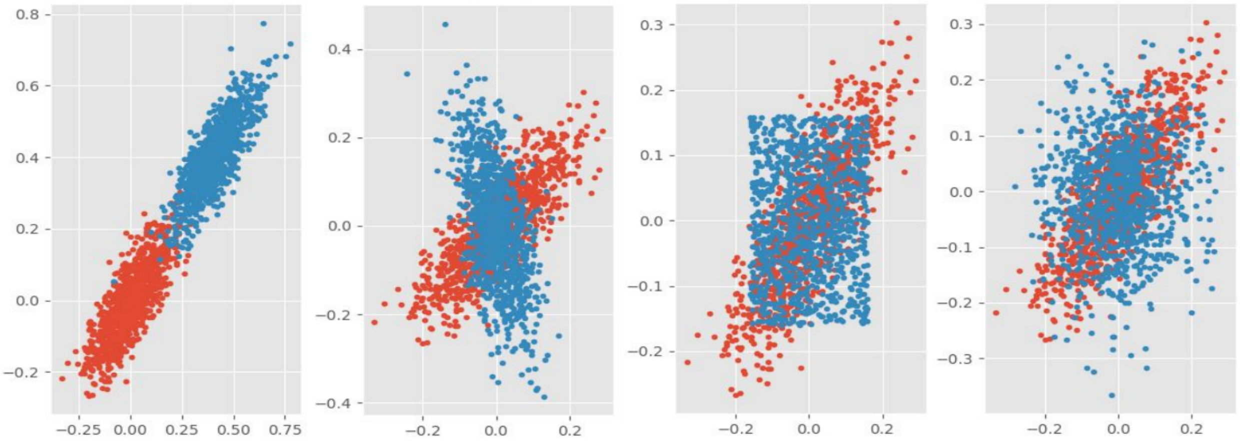

Figure 3: Nominal group (red) compared to the various anomaly groups (blue)

The training set and test set were generated in an identical way. Both consisted of 2000 groups of 100 two-dimensional points. The nominal data are multivariate Gaussian distributions with the following parameters:

$$\Sigma_{nominal} = \begin{bmatrix} 0.01 & 0.008 \\ 0.008 & 0.01 \end{bmatrix} \qquad \mu_{nominal} = 0.3 * \mathcal{N}([0,0], \mathbf{I}_2) \tag{7}$$

To generate anomalies, first I chose the anomaly likelihood to be .02, then for the groups selected to be anomalous (by a Bernoulli random variable), I randomly and uniformly chose from four different equally likely distributions: nominal displaced by $+1$ in both dimensions, nominal rotated by 60 deg counter clockwise, 2D uniform distribution with the same variance and mean vector as nominal and Gaussian with covariance matrix $\Sigma = [[0.1, 0], [0, 0.1]]$ and the nominal mean vector. The different anomalies are displayed in figure 3.

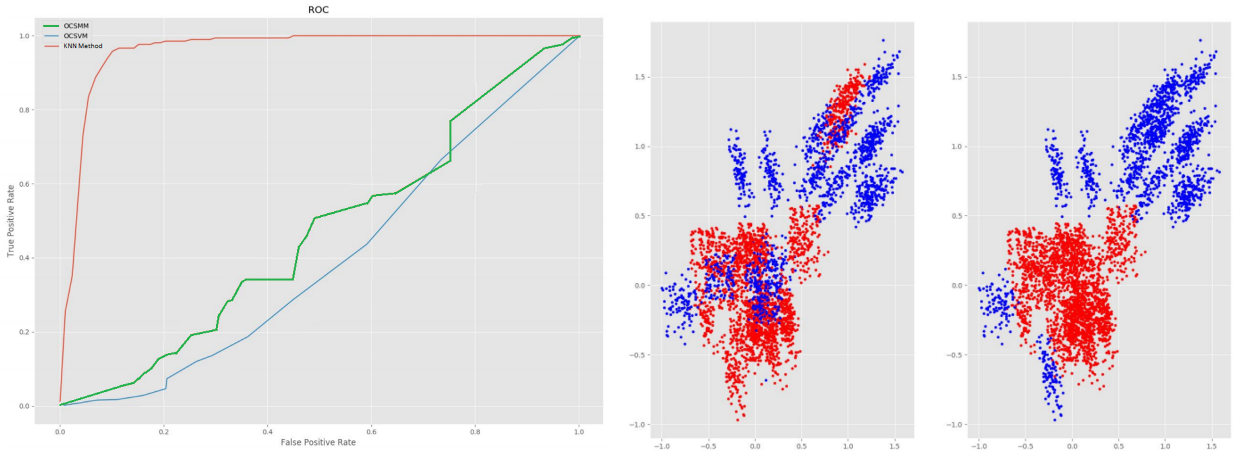

Figure 4: (LEFT) ROC curves of the KNN approach(red), One class support vector machine(blue) and one class support measure machine(green). Notice that the support measure machine fails to beat random guessing because has no way of detecting group anomalies unless they are spatially separated from the nominal groups. (RIGHT) Anomalous groups (plotted without nominal groups) seen by the OCSVM (left) and the OCSVM (right). Correctly identified groups are in blue.

Through some experimentation, I determined that a good choice for the embedding kernel bandwidth σ should be 0.005. This is obviously highly dependent on the variance of the nominal distributions, but as of yet I don't have a way of incorporating this parameter into the learning algorithm. I also set the K nearest neighbors parameter $K = 3$. Intuitively the higher this gets, the less this algorithm should work because the nominal groups will have their nearest neighbor scores averaged over more low values, whereas the anomalous groups' scores will remain about the same.

4.2 Discussion

My approach was tested against two other methods. The minimum baseline is a one class support vector machine where the groups are represented by their means. This obviously did poorly because it only has first order information about the distributions of the groups. You can see clearly from the rightmost plot in figure 4 that the OCSVM misclassified any group whose mean was near the middle. The next best detector was the one class support measure machine, which used the kernel on distributions. This didn't do as well as I had hoped and barely beat random guessing. It's pitfalls have already been discussed in previous sections, but you can see from the middle plot in figure 4 that it was able to correctly identify some anomalies located near the nominal groups. Its false alarm rate is very high regardless of what bandwidth is chosen for the embedding kernel.

To generate the ROC curves in figure 4, I ran the algorithm on training and test data and averaged the performance on the test data over 5 rounds. The ROC curve illustrates that the approach taken in this paper performs far better than either of the other methods on synthetic data. A perk of the algorithm introduced in [4] is that you can select the approximate false alarm rate for the detector.

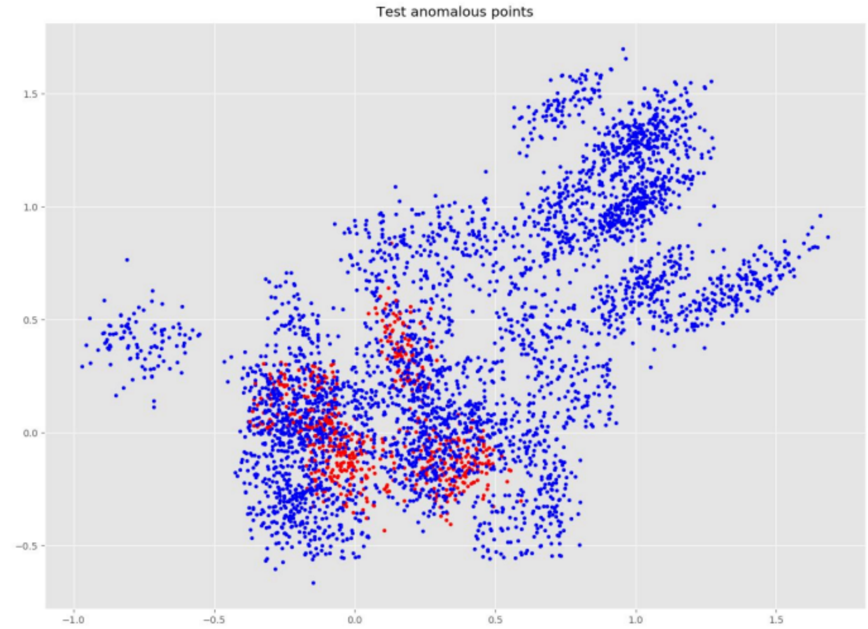

Figure 5: Anomalous test points plotted without nominal points to avoid clutter. For this trial the precision was 0.86 and the false alarm rate was 0.05

Figure 5 shows that all of the anomalies that I tested generally ranked in the lowest 5-th percentile compared to the training groups. It makes sense that the ones it missed were closest to the middle of the nominal distribution of groups, as this makes it harder to tell them apart.

5 Conclusion/Future Work

In this paper I discussed my approach for group anomaly detection. It is primarily based on the work done in [2] and [4].

Although the algorithm seems to have a great degree of accuracy, it needs improvement in terms of computational complexity. As it stands, training and testing both take the same amount of time with complexity $O(\ell^2)$. The training complexity is difficult to reduce because the kernel matrix K needs to be computed in full. Unless there is a way to reduce the number of training groups without sacrificing performance, the training complexity will stay the same. The test stage complexity could probably be reduced in a number of different ways. Since the kernel product is especially taxing to compute, the best way to increase the efficiency of this method is to reduce the number of times it is computed. One way would be to use the algorithm in [4] and learn an estimator for the score function $R_\ell(\eta)$. Because of the representer theorem, the estimator is simply a weighted subset of the training groups. In other words, learning an estimator should simply identify which training groups are necessary to adequately represent the nominal distribution. Another approach could be to prune the set of training groups based on some kind of utility metric. If removing a training group doesn't affect how the training groups are classified, then obviously that group is redundant.

This paper has shown that the approach taken here, while computationally intensive, has a very desirable ROC curve, and with a little more work could be a valuable tool to the scientific community. The code for this project is available on Github at:

https://github.com/gregstarr/anomaly-detection.

References

[1] K. Das, J. Schneider, and D. B. Neill, "Anomaly pattern detection in categorical datasets," in *Proc. 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. USA: ACM, 2008, pp. 169–176.

[2] K. Muandet and B. Schölkopf. One-class support measure machines for group anomaly detection. *arXiv preprint arXiv:1303.0309*, 2013.

[3] K. Muandet, K. Fukumizu, F. Dinuzzo, and B. Schölkopf. Learning from distributions via support measure machines. In *Advances in Neural Information Processing Systems (NIPS)*, pages 10–18. 2012.

[4] J. Root, J. Qian, and V. Saligrama, "Learning efficient anomaly detectors from K-NN graphs," in *Proc. of the Int. Conf. on Artificial Intelligence and Statistics*, 2015, pp. 790–799

[5] L. Xiong, B. Poczos, and J. Schneider. Group anomaly detection using flexible genre models. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2011a.