COMS 4721 Final Project: Anomaly Detection

Gabrielle Agrocostea (gda2108), Josh Plotkin (jsp2014), Celene Chang (chc2155)

January 15, 2015

I Introduction

Unusual behavioral patterns or inconsistent data points are challenges that are encountered in virtually every instance of real metric collection. How does one determine if a large bank transaction is 'normal' for a customer? What does it mean that the most recent value from a time series is a few standard deviations from the mean of the past week? Anomaly detection, a Machine Learning technique, can be used to help answer these questions.

What is an anomaly?

An 'anomaly' is a general term, and can be defined in different ways depending on the context of the problem. If one were attempting to fit a model to a dataset, then an anomaly may be an outlier that skews the model or biases the distribution a certain direction. In this instance, anomalies can be detected for the purpose of removing them, so that the validity of the model or the 'truthfulness' of the distribution can be improved.

In contrast, if one was tracking a trend, then an unusual datapoint would be a novelty and may suggest that something interesting or alarming is happening. Anomaly detection can thus be used to identify novel points and alert the user to take further action. An anomaly may also be an extraordinary instance, such as a Michael Jordan-esque athlete; a diagnostic tool in medical applications; suspicious behavior, such as in cyber security or fraud (currently the most common use), and so on. The meaning of anomalies is boundless across different applications, and thus the usefulness of optimizing methods for detecting them is indisputable.

Why is this difficult?

An obvious challenge in defining an anomaly is distinguishing it from noise in a data set. Depending on the type of data, thresholds would need to be set such that a minimum level of noise is tolerable. However, in many applications false positives are more tolerable than false negatives; that is, we would be okay with having some occurrences falsely labeled as anomalies but it would be very costly to allow anomalies to go undetected. In addition,

some applications produce data with high variances while for others even a tiny deviation is a problem, so the parameters would need to be very specific and finely tuned.

Another difficulty is the number of occurrences in training data: by definition, anomalous data are inconsistent with normal behavior and thus are rare. As a result, when attempting to build a classifier that will separate 'normal' from 'not normal', other techniques may be necessary to boost the latter class and strengthen the model.

A third challenge for data that are continuously updated is that the meaning of 'normal' may be changing over time. In case that shifts in 'normal' happen often, it would be beneficial to have a number of additional data points before determining whether an unusual point was indeed anomalous, or just the start of a new set of behavior. However in the case of credit card fraud, delayed detection in an effort to be conservative can lead to large financial loss for the credit card company.

II Practical Importance of Anomaly or Novelty Detection

Anomaly detection is a task as broad as machine learning itself. Our research revealed methods that spanned nearly every machine learning algorithm we've learned in class. This is due to the wide range of circumstances in which outlier detection is required.

Security risks

For many people, anomaly detection is synonymous with fraud. Josh had a recent experience related to this. In January, he took a trip to London. United Airlines misplaced his bags and so he had to go shopping to replace his clothes. At some point toward the end of the shopping spree, his bank card was declined. Evidently, the bank has a threshold that was not reached until a certain number of transactions, frequency, or dollars spent was reached.

One might wonder, "how can the bank detect in Josh's spending habits if Josh has never been a victim of fraud?" Indeed, this is a challenge, and brings up the notion of **novelty** detection. How can we study normal activity and then distinguish what is not normal? We will investigate this later.

There are many other cases in which outliers present a security risk. Computer networks typically have a "normal" level of activity. A spike in packets sent or users might represent an attack.

Outlier detection is not limited to inspecting data only. It can be used to detect unusual activity on a security camera as well as in a steam of photos.

In these cases, the outliers are mostly something users are trying to identify and then handle afterwards. This is not always the case.

Learning from and predicting outliers

Another approach to anomaly detection comes in trying to predict the outliers themselves. To paraphrase Abe Stanway, creator of the Etsy Skyline algorithm, it's much more efficient to be proactive and find warning signs of an impending anomaly than it is to be reactionary.

One such case where being proactive is critical is in detecting potential outbreaks. Given the nature of an outbreak, the risk increases exponentially over time. It is essential to find the initial warning signs and take action before the outbreak occurs.

In a similar way, predicting anomalous weather patterns can save lives and money. A recent local example was in the case of Hurricane Sandy. While a storm cannot be prevented, measures can be taken to fortify certain areas. Sandy was an extreme outlier that was detected too late. In many recent examples across the world, the cost has been hundreds or thousands of lost lives.

From a machine learning perspective, this means modeling outliers. It is not enough to simply point them out. We must understand their make-up. What are their characteristics, why do they occur, and what are the covariants?

Outlier Removal as a Pre-processing step

In the cases of predicting outbreaks or security risks, the outliers themselves are of interest. A common use for outlier detection is simply removing data points from a training set. This is because outliers are capable of throwing off many otherwise effective algorithms.

AdaBoost is one of the most effective algorithms we learned this semester. Think for a moment how an anomaly would be treated by the algorithm. The anomalous point would be repeatedly misclassified and its weight increased. This would encourage the weak learners to label this point correctly at the expense of labeling the other data points correctly. If there is one anomalous data point, this might present an enormous problem. As the number of outliers increase, the more of a problem this presents.

Regression techniques can also be adversely affected by the presence of outliers. We have provided an example at the end of this paper that demonstrates this. Once again, think for a moment about how ordinary least squares works. It attempts to minimize the mean of the squared errors. Since the errors are squared, the algorithm is encouraged to put the line closer to the outlier to mitigate that expense. This is because squared error is not a robust metric. We will discuss robust techniques at a later point.

A logical idea for detecting outliers is the 3σ rule, which will be discussed later in this paper. The general intuition is "let's find the mean and standard deviation and calculate the probabilities of each point occurring." The problem is similar to the linear regression problem. Outliers can wildly affect these statistics. Typically, the outliers will both move the mean toward those points and away from the distribution, and due to the squared errors again, increase the standard deviation significantly. One can imagine how this would yield statistics that would make the outliers appear as simple noise rather than a point that needs to be removed.

As mentioned, outlier detection is a vast area of research due to its wide range of uses. Just as there are many practical uses for this, there are many possible solutions. We will now investigate these.

III Implementation

As with most Machine Learning techniques, the variety of ways that one can implement an anomaly detection algorithm is expansive. Here we provide examples of different approaches that can be used to detect anomalies.

Supervised learning

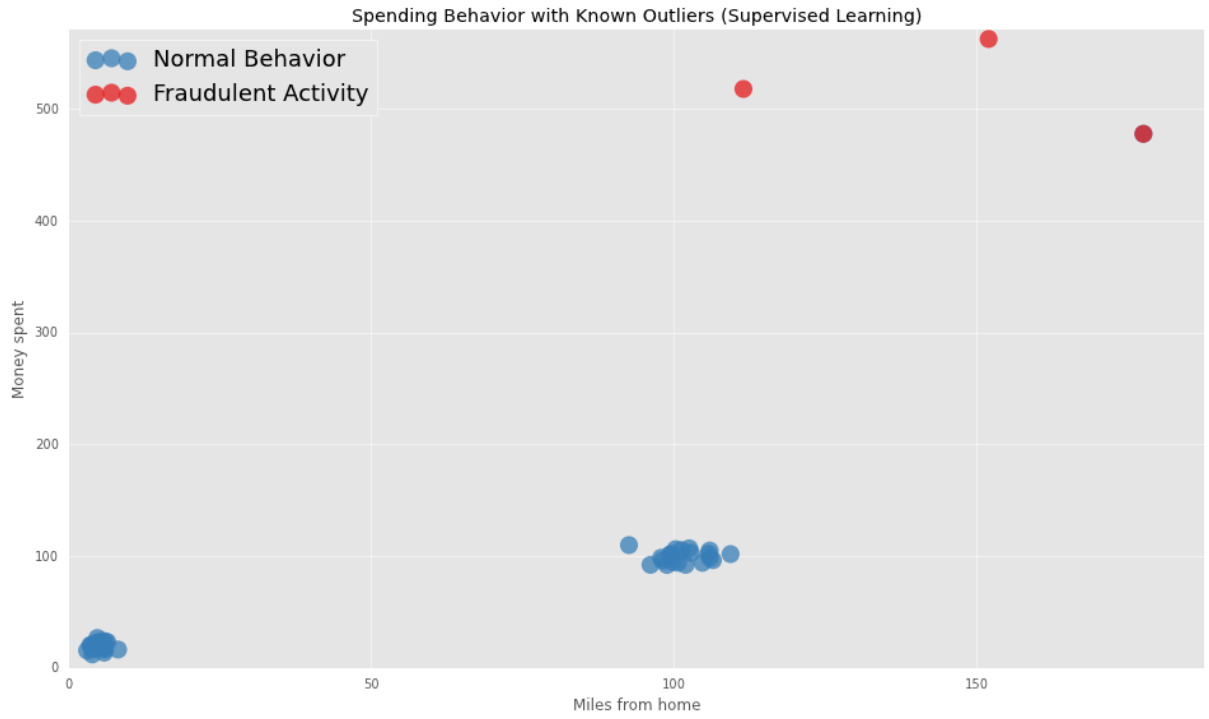

Figure 1:

Supervised outlier detection is simliar to classification problems in supervised learning. Among many classification methods, anomalities can be modeled using techniques such as SVM, Neural Networks and Bayesain Networks. Models are built for normal events based on labeled training data and data that does not match is labeled as outliers.

In addition, models can also be built for outliers and data not matching could be treated as normal. However, in general, there can be imbalanced classes where there aren't sufficient outliers and it is beneficial to make up artifical outliers to rebalance the classes. A key performance metric in supervised learning for outlier detection is Recall, and, it is more important than accuracy.

For example, as in our presentation, we have labels for normal behavior as well as fraudulent behavior. To separate the points, one could run any classification method and find the decision boundary. Also, since there arent many outliers in the dataset we have imbalanced classes and need to perform boostrapping or boost the anomaly class.

Semi-supervised learning

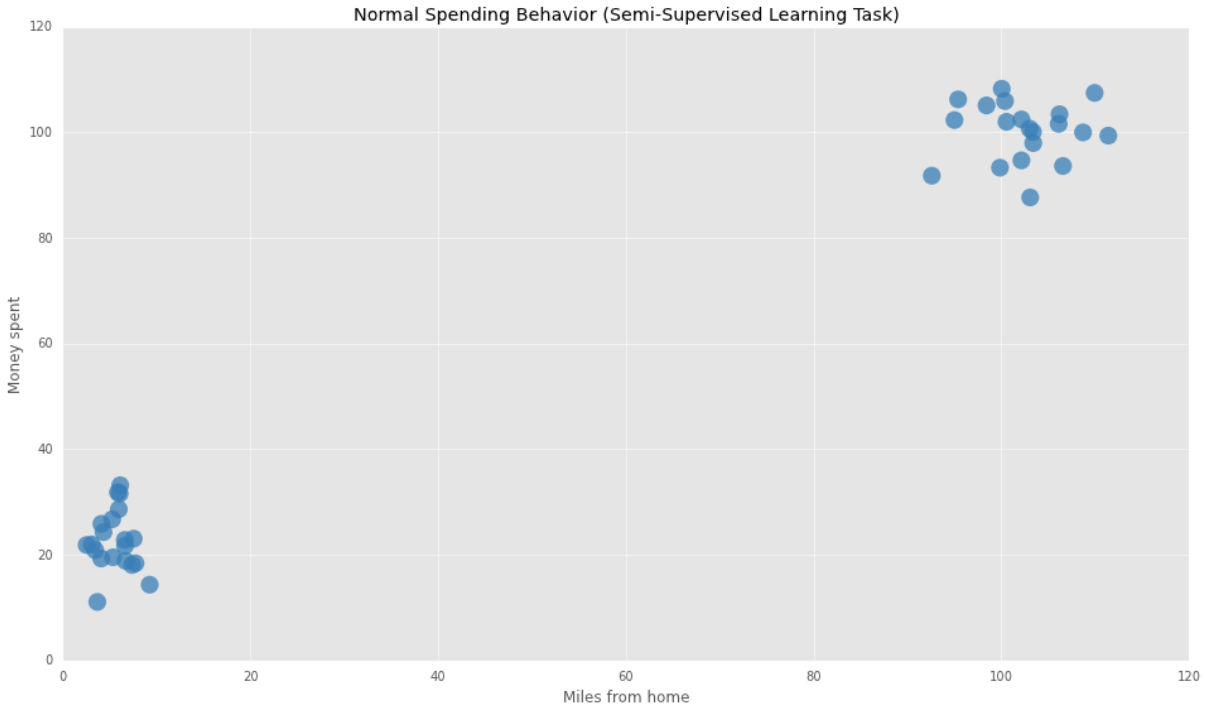

Figure 2:

Semi-supervised learning falls somewhere between unsupervised learning and supervised learning. Often, labels are only available for normal data, and outliers are not present.

For the above example, we have data points for normal activity but are unaware of what anomalous behavior looks like. K-means can be used to cluster the normal data. New data points far from the centroids can be labeled as outliers. In a similar way, Gaussian Mixture Models can be used to cluster the data, and find the probability of a new data point belonging to its closest Gaussian cluster. An innovative method for modeling normal data is called one-class SVM. It creates a decision boundary around the normal data. New data points not fitting within the decision boundaries can be considered anomalous. This can be seen below, where we have modeled one-class SVM on our toy data set.

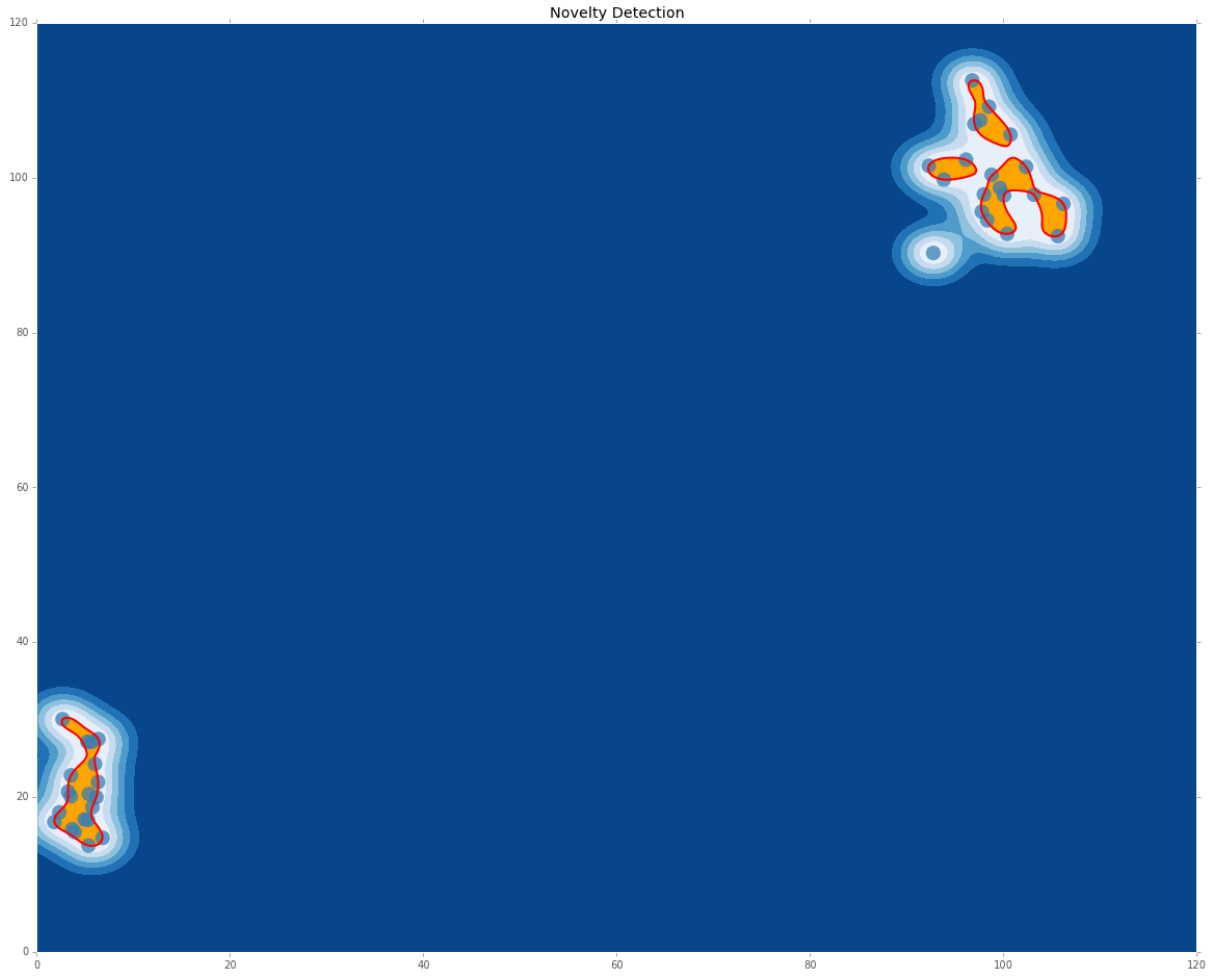

Figure 3:

Unsupervised learning

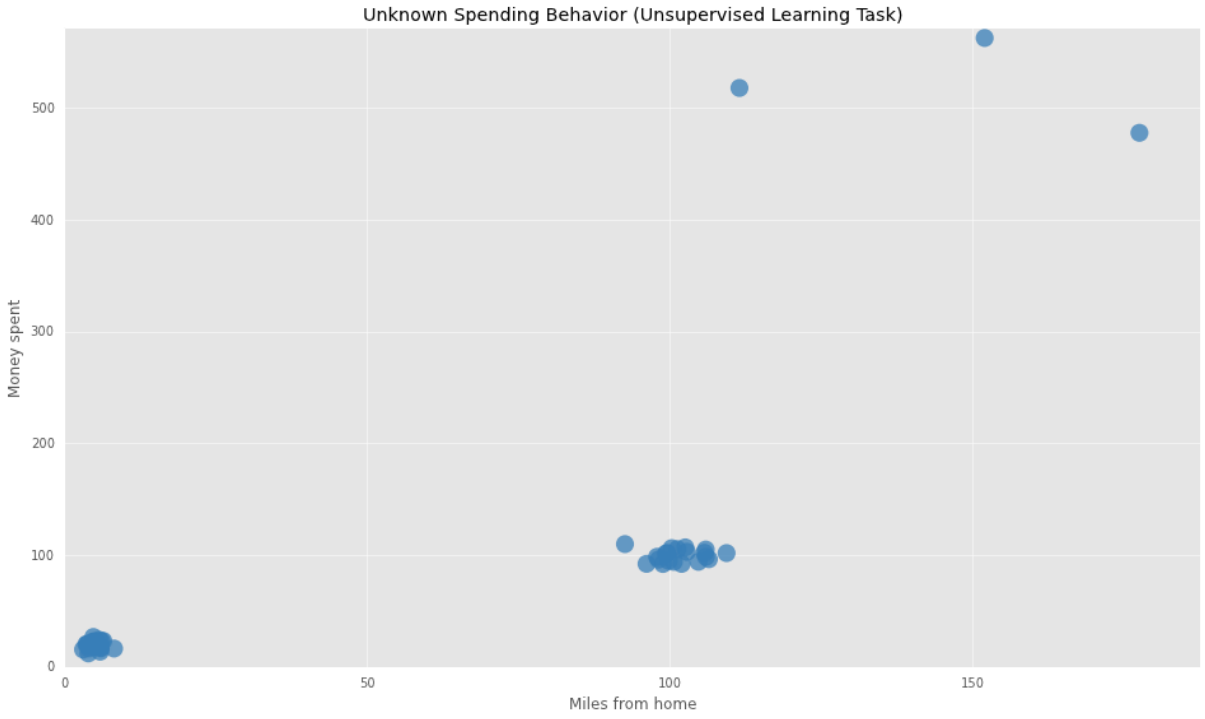

Figure 4:

In the context of anomaly detection, unsupervised learning occurs when outliers may exist in the data set, but we're unsure of which points are anomalous. The assumption in unsupervised learning methods for outlier detection is that anomalies are rare relative to normal data. It is also assumed that the normal data is somehow clustered into groups, and that each cluster has distinct features. Since there are many clustering methods, these can be adapted and applied to unsupervised outlier detection. One example might be using K-means to find each point's distance to its group's centroid. Higher distances correspond to outliers. We can also use a method similar to K-nearest neighbors which finds the distance to its k_{th} nearest neighbor. This allows us to find clusters of outliers.

One disadvantage is that collective outliers - when a subset of the data collectively deviates from the rest of the data even if the individual data instance is not itself and outlier - can not be detected and could easily be mistaken for a cluster. In addition, it can be difficult to distinguish outlier from noise in the data. The advantages of using supervised learning in outlier detection is that the models are easily be understood and are highly accurate in detecting known anomalies.

As with many supervised learning methods, the curse of dimensionality causes problems.

K-means relies on Euclidian distance, which becomes increasingly useless in higher dimensions. Solutions do exist, however. Instead of using Euclidian distance, we can introduce methods using Mahalanobis distance or the angles between points. We can attempt to reduce the number of data points we have, using a method like PCA.

Statistical methods

Another approach to unsupervised learning involves statistical methods. Statistical methods assume that the data can be modeled by some statistical/stochastic process (generative model). The typical process is to learn a generative model that fits the data and then find objects with low probability. The low probability objects are considered outliers. For example one can use Gaussian distribution to model normal data, then estimate the probability of the data fitting normal distribution for each point $g_D(y)$. If the probability $g_D(y)$ is low then y does not belong to the Gaussian model and is an outlier. In general, data that not follow the statistical model are considered outliers and as such the performance of statistical model highly depends on whether real data fits the assumed statistical model.

For statistical methods, there are two types of methods: Parametric and non-parametric. In parametric methods, it is generally assumed that the normal data is generated from a parametric distribution. One of the tasks is to learn the parameters from the normal sample and then use tests such as the Grubbs statistic of Chi Squared to determine outliers. In contrasts, non-parametric methods do not assume any parameters and they are used to learn the parameters. An advantage of non-parametric methods is that there are fewer assumptions made about the model so the method might be more applicable.

IV Anomaly Detection and the future

Anomaly detection is becoming increasingly important over time. Websites like Etsy, a site where people from around the world buy and sell products, have a need for methods to quickly detect fraud. Abe Stanway, a former employee of Etsy, developed the Skyline application which is an ensemble method for finding potential outliers. This is built to handle the massive stream of data they deal with around the clock. If anomaly detection is like finding a needle in a haystack, then the haystack is growing exponentially over time.

As we have seen, outlier detection can use virtually any machine learning technique. As new machine learning methods and algorithms are developed over time, we can utilize these on the never-ending task of finding and learning from anomalies.