1 General Concepts

I begin with a list of general approaches used in fraud detection. I pull these from [1] and from [3]. It should be noted that most commercial grade fraud detection techniques are proprietary (for obvious reasons), and as such, details on some of the more robust methods for fraud detection are not always readily available.

- **Outlier detection.** Simply the process of finding transactions which deviate from what a "normal" transaction would look like. Transactions that spend more than three or four standard deviations above the average transaction price. Depending on the setting, this can be highly predictive of fraud, or rather inaccurate. Examining outliers usually requires a well defined "normal" purchase. For example, an individual who spends $350 on dry cleaning in a month could be considered an outlier due to his or her excessively large payment for a service (dry cleaning) that usually costs less than $50 a month. However, a computer store may see one customer spend $2500 on a nice computer, but also have customers who only spend $6 for a new network cable. In the case of the computer store, it is much harder to construct a "normal" customer.

  ⋆ One technique that is used in outlier detection is unsupervised learning (such as clustering). Unsupervised learning methods can be tuned to detect fraudulent transactions which differ in some way from "normal" baseline transaction. In most cases of outlier detection, unsupervised learning techniques are preferred to supervised learning techniques, since supervised learning techniques require previous knowledge of which transactions were fraudulent and which were not. Such information is often difficult to obtain and maintain, and moreover requires labeling transactions as fraudulent or not. This labeling process can be time consuming and may inadvertently introduce inaccuracies in the data (e.g. false positives). Supervised machine learning method also perform relatively poorly when

confronted with previously unseen types of fraud.

For more on unsupervised learning application in fraud detection, see [2].

- **Behavior outlier detection.** This method is a modified approach to detecting fraud by outliers in the data. This method specifically focuses on tracking user behavior longitudinally and watching for outliers in behavior. For example, if a credit card user only uses his or her card to buy gasoline once a week, a sudden $1600 charge for a new computer may arouse suspicion.

  On the whole, behavior outlier detection is one type of a more generalized category of local outlier detection methods. These methods focus on one specific metric (or a small collection of similar metrics) to detect fraud via outliers relative to the given metric. Local outlier detection is effective in situations where the population is heterogeneous, such as is the case with credit card usage. A technique outlined in [2] (in the setting of credit card fraud) is to determine peer groups for each account. That is, accounts are clustered based on previous spending and payment habits. Accounts are then tracked *within peer group* for abnormalities. This allows for detection of fraudulent transactions while at the same time accounting for the large variety of purchasing habits that can be found among users of any major credit card.

- **Meta-data outliers.** Another variation on outlier detection is using meta data associated with a purchase to detect fraud. For example, online purchases transacted at 3 a.m. local time may be an indicator of fraud. Another example of using transaction meta data to detect fraud would be comparing the billing address to the shipping address for a purchase. For an account registered to a billing address in Kansas, USA, it would be highly unusual if a purchase of iPads was purchased by said user and shipped to Uzbekistan.

- **Neural-Networks.** Sometimes neural-networks are applied to the space of fraud detection. Commercial usage of neural-networks is found in products like CARDWATCH, Falcon, and CyberSource. Most of the details pertaining to the implementation of neural-networks in these settings are proprietary.

- **Logistic regression.** We can model the probability of a transaction being fraudulent

based on a collection of parameters. Especially in settings where users are relatively homogeneous, logistic regression can prove a reliable method for detecting fraud.

- **Bayesian Belief Network** - A Bayesian belief network (BBN) represents a set of random variables and their conditional independencies using a directed acyclic graph (DAG), in which nodes represent random variables and missing edges encode conditional independencies between the variables. While quite accurate in theory, BBN can be hard to implement in practice without extensive experience.

Source [3] contains an extensive table aggregating resources for these above detailed methods.

1.1 Commercial Anti-Fraud Detection Services

Here are a few commercial fraud detection services I have found:

- LexisNexis Risk Solutions. They have a number of products that help combat fraud. Most of their products deal with Multi-factor authorization and identity fraud.
- Falcon. Users will need to enter in buyer information prior to viewing any specifics of the product line.
- CyberSource. Offers real time solutions to a variety of fraud problems. Many of them deal with identity theft in one form or another. Especially geared towards implementation in an eCommerce pay and ship enterprise.
- Palantir Anti-Fraud. Has several accessible graphical user interfaces. Has features for tracking fraud using spacial data.

As previously mentioned, these service do not disclose many algorithmic details pertaining to what they are doing behind the scenes. Also be aware that there are "anti-fraud" platforms that are really just phishing schemes posing as legitimate, so do not just blindly type in personal information. These scams prey on people or companies who do not have fraud detection software and may currently be vulnerable to cyber attack.

2 Detecting Anomalies

2.1 Generalized ESD Theory

One simple method that is commonly used in fraud detection is the extreme studentized deviate (ESD) test (see [4] and [5]). The generalized (ESD) test is used to detect outliers in a univariate data set. It is usually assumed that the data follows an approximately normal distribution. Although modern uses of this method as a stand alone rarely

Given an upper bound k, the pair of hypotheses uses in the ESD are as follows:

$$H_0 : \text{The data contains no anomalies}$$

$$H_A : \text{The data contains one or more (up to } k\text{) anomaly}$$

Compute the following value:

$$R_1 = \max_{i=1,2,\ldots,n} \frac{|x_i - \overline{x}|}{s},$$

where $\overline{x} = \frac{1}{n}\sum x_i$ is the sample mean and $s = \sqrt{\frac{1}{n-1}\sum (x_i - \overline{x})^2}$ is the sample standard deviation for a sample of size n.

Let $x_{(1)}$ be the maximizer for R_1. We now calculate R_2 like we did R_1, yet we omit $x_{(1)}$ and compute R_2 on the subset of $n-1$ observations. This process is repeated inductively to create R_3, R_4,..., R_r, each time removing all the previous maximizers $x_{(2)}$, $x_{(3)}$, ..., $x_{(r-1)}$ of the earlier computed R_i values. (So this process is done on subsets of size decreasing by one observation each step). We now have a set of r test statistics $\{R_1, R_2, R_3, \ldots, R_r\}$.

For a chosen significance level α, define the value $p_i = 1 - \frac{\alpha}{2(n-i+1)}$ for $i = 1, 2, \ldots, r$. Compute the following ancillary values:

$$\lambda_i = \frac{(n-i)t_{n-i-1,p_i}}{\sqrt{(n-i-1+t^2_{n-i-1,p_i})(n-i+1)}},$$

where $t_{\nu,p}$ represents the $p\times 100\%$th percentile of the t-distribution with ν degrees of freedom.

Finally, the number of outliers/anomalies is given by

$$\text{argmax}_i\{R_i > \lambda_i\}.$$

2.2 Seasonal Hybrid ESD

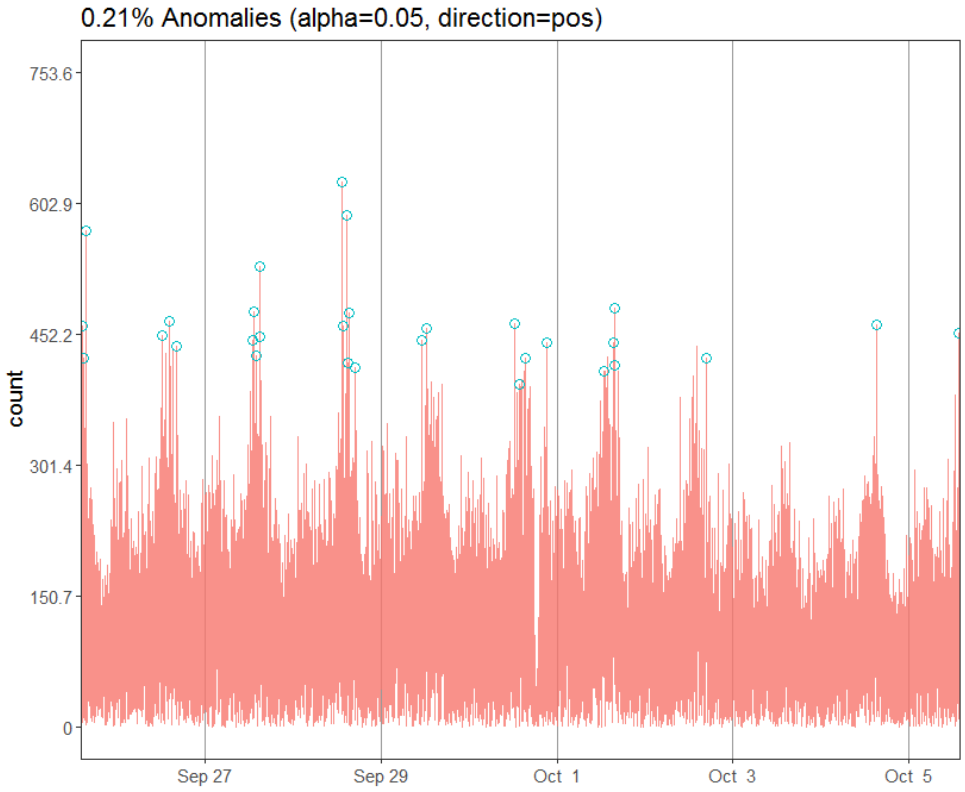
We can build on the methods of the generalized ESD methods to create what is called a "Seasonal Hybrid ESD." This test better accounts for data that is seasonal or periodic (such as credit card purchases, which tend to follow a seasonal pattern). The development of these methodologies was done mainly by Twitter. See [7] Twitter has implemented an open source package in R called `AnomalyDetection` that demonstrates seasonal hybrid ESD. There also is a Python implementation of this method. See `pyculiarity` at [8]. The Python implementation of this algorithm is essentially identical to that used in R. Python users should be able to easily translate anything given here in R to the equivalent Python. (Note that Python users will need to have `rpy2` installed).

```
##### Detecting anomalies in univariate data.
##### Applications to time series.

devtools::install_github("twitter/AnomalyDetection")
library(AnomalyDetection)

data(raw_data)
tsLength = nrow(raw_data)

multipliers = abs(rnorm(tsLength,mean=1, sd = 0.65))
raw_data[,2] = raw_data[,2]*multipliers
res = AnomalyDetectionTs(raw_data, max_anoms=0.05,
                         direction='pos', plot=TRUE)
res$plot
```

After calling the library `AnomalyDetection`, I load some internal time series data from the package (dummy data). The toy data set here is not given in a specific context, since it is anonymized data from Twitter. However, we could, for example, look at this data as a data set of wait times (in seconds) at a certain stop sign. This will have a periodic or "seasonal" aspect to it throughout the day.

I introduce some further randomness into the data by creating a vector of random multipliers. The function `AnomalyDetectionTs()` is then called on the time series data. I tell the function to allow for at most 5% of the data to be considered outliers (controlled by `max_anoms`). The direction parameter can be set to `'pos'`, `'neg'`, or `'both'` depending on what direction in which we wish to detect outliers.

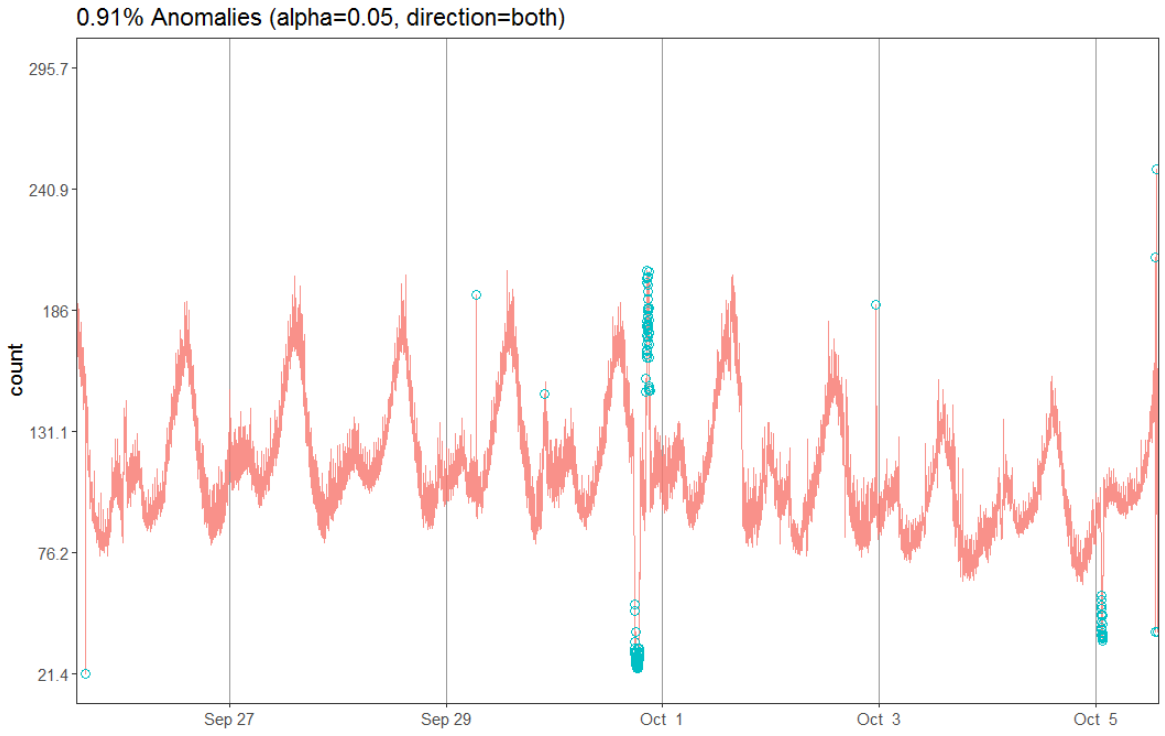
Here is an example visualization of a typical result:



The outliers or anomalies in the data set are circled.

This same type of analysis can also performed on the original raw data from the `AnomalyDetection` package (without the randomly generated multipliers).

```
##### Detecting anomalies in univariate data.
##### Applications to time series.
data(raw_data)

res = AnomalyDetectionTs(raw_data, max_anoms=0.05, direction='both', plot=TRUE)
res$plot
```



Here I have selected that I want to detect anomalies in both the positive and the negative direction (`direction='both'`).

2.3 Theory Behind Seasonal Hybrid ESD

From here on, I will refer to Seasonal Hybrid ESD as SH-ESD. The underlying statistical theory for SH-ESD is as follows. See [7] in particular for the specifics of the implementation.

For removal of the seasonal component from the time series, standard methods for periodic flattening are used. As is essentially always the case with time series data, we will actually be working with the residuals of the time series after adjusting for trend and seasonality. In short, a well accepted method called STL ascertains (using LOESS) the seasonality of the data and standardizes the data accordingly. See [9] for further details. (Note that this method for removing seasonality is not readily explainable in a few sentences. The details are being left only to those who are truly interested).

In the original generalized ESD method detailed above, the sample mean and standard deviation are employed. However, since these measures are especially vulnerable to skewness from anomalies and outliers, using them in a method for detecting those very anomalies is naturally dangerous. As such, the "hybrid" aspect of SH-ESD uses the sample *median* in lieu

of the sample mean. Furthermore, instead of using the simple sample standard deviation, we will now use the *median absolute deviation* (MAD). The MAD is the median of the absolute deviations from the sample median. Written formally, that is

$$\text{MAD} = \text{median}_i \left(|x_i - \text{median}_j(x_j)| \right).$$

This can be used to estimate the standard deviation of the data. In order to do this, we must multiply the MAD by a scalar β:

$$\hat{\sigma} = \beta \cdot \text{MAD}.$$

When the data is (approximately) normally distributed, $\beta = 1.4826$ can be used. This is the scalar used in `AnomalyDetection`. In general, for univariate time series data with more than 200 or so observations, the Central Limit Theorem will be quite sufficient to assume normality. For generalized applications, if we can determine the underlying distribution of the data, $\beta = \frac{1}{\Phi(0.75)}$ is usually considered standard. (Note: $\Phi(q)$ is the value of the $q \times 100$th percentile of the given distribution).

After having found these values, SH-ESD is performed in the same manner as the simpler generalized ESD given in Section 2.1, except we now use the median and the $\beta \cdot \text{MAD}$ in place of the mean and sample standard deviations respectively.

3 Sources

(Source title is a hyperlink to the reference when available. Some papers required paid access [which I have] and I downloaded a hard copy PDF of them for later).

[1] Yufeng Kou; Chang-Tien Lu; S. Sirwongwattana; Yo-Ping Huang. Survey of fraud detection techniques. *Proceedings of the 2004 IEEE International Conference on Networking, Sensing & Control*. Taipei, Taiwan, March 21-23, 2004

[2] Richard J. Bolton and David J. Hand. Unsupervised Profiling Methods for Fraud Detection. (Published by authors on their website)

[3] Anuj Sharma and Prabin Kumar Panigrahi. A Review of Financial Accounting Fraud Detection based on Data Mining Techniques. *International Journal of Computer Applications* Volume 39-No.1, February 2012.

[4] K. R. Nair. The Distribution of the Extreme Deviate from the Sample Mean and Its Studentized Form. *Biometrika*, Vol. 35, No. 1/2 (May, 1948), pp. 118-144.

[5] Rosner, Bernard. Percentage Points for a Generalized ESD Many-Outlier Procedure. *Technometrics*, vol. 25, no. 2, 1983, pp. 165172.

[6] GitHub Repository for `AnomalyDetection`. Accessed Jun 13, 2017.

[7] Jordan Hochenbaum Owen S. Vallis Arun Kejariwal. Automatic Anomaly Detection in the Cloud Via Statistical Learning. Published by Twitter Inc.

[8] GitHub Repository for `pyculiarity`. Accessed Jun 20, 2017.

[9] Robert B. Cleveland, William S. Cleveland, Jean E. McRae, and Irma Terpenning. STL: a seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics*, 6(1):373, 1990.