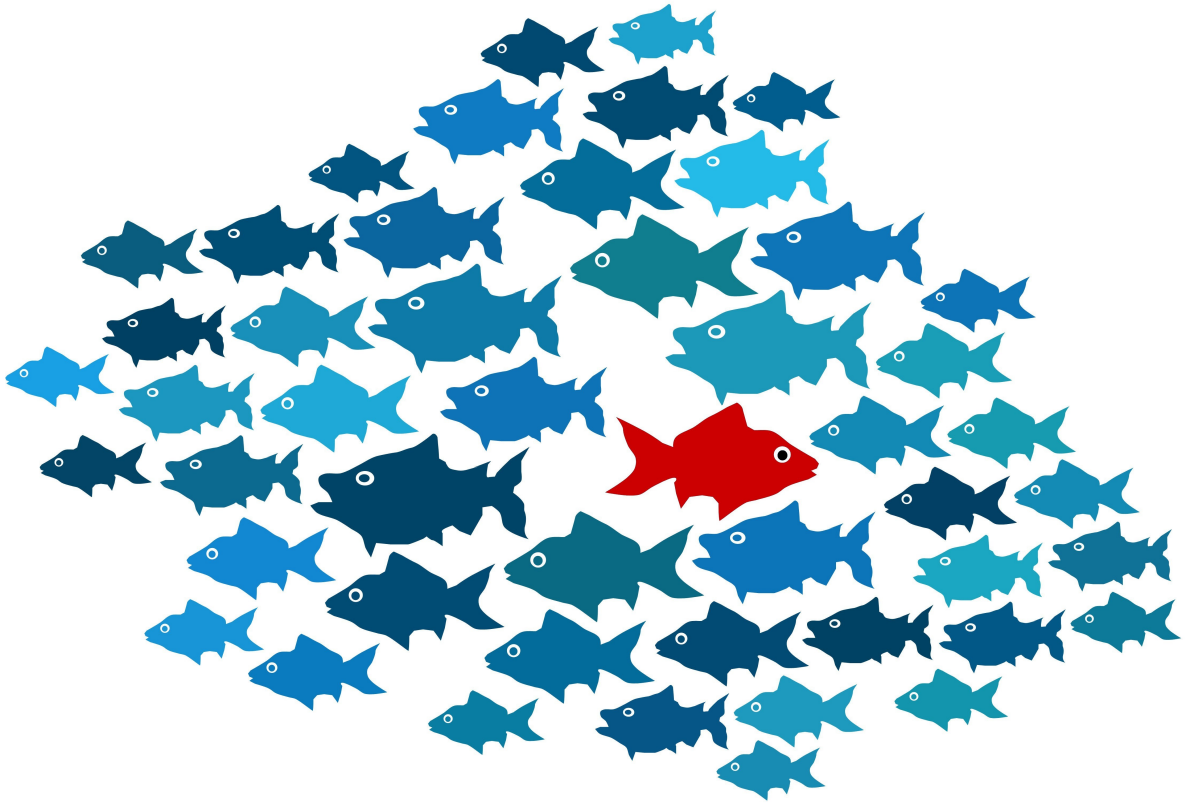
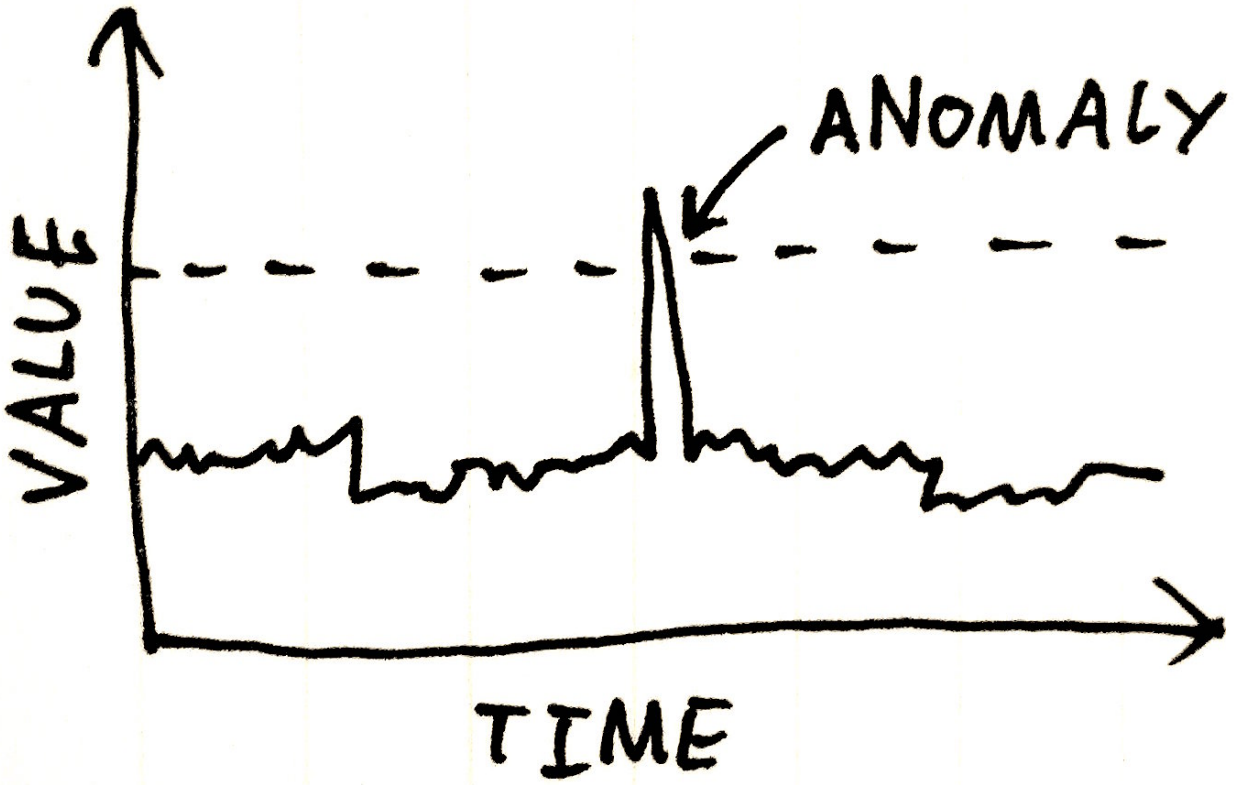
Anomaly Detection

Alex Lin



in Time Series

Melissa Yu



Asking the Question

Problem Statement: Given a sequence of random variables $\{Z_i : 1 \leq i < \infty\}$, we wish to develop a method for determining the existence of *anomalies*.

Specifically, as the variables arrive one by one, we test the *exchangeability assumption* for the sequence $z_1 \ldots z_n$, which states that the joint distribution for these variables is invariant under any permutation of the indices.

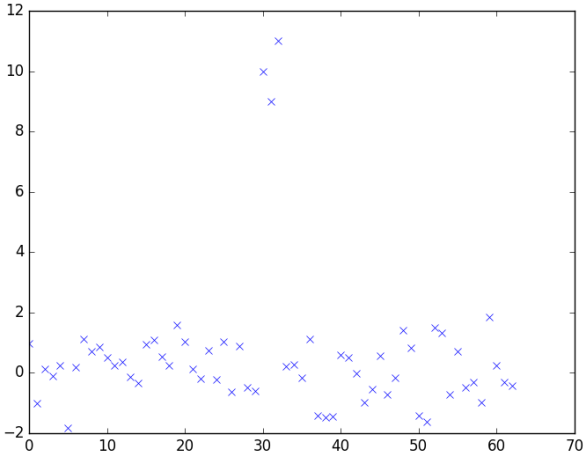
One method for anomaly detection involves using a *Martingale*, a sequence of random variables $\{M_i : 0 \leq i < \infty\}$ such that $E(M_{n+1}|M_1, \ldots, M_n) = M_n$

In particular, M_n models the degree to which $z_1 \ldots z_n$ violates the null hypothesis given by the exchangeability assumption.

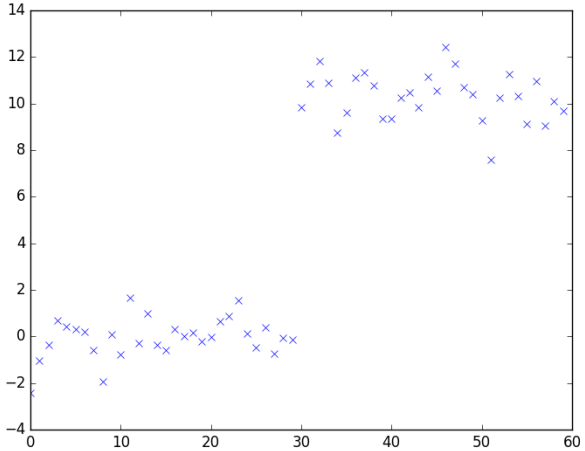
Key Questions: How can we appropriately construct Martingales that will successfully detect the existence of different kinds of anomalous data? What is the best Martingale threshold level for rejecting the null hypothesis? Is there a tradeoff between the accuracy and the confidence of our methods?

Getting the Data

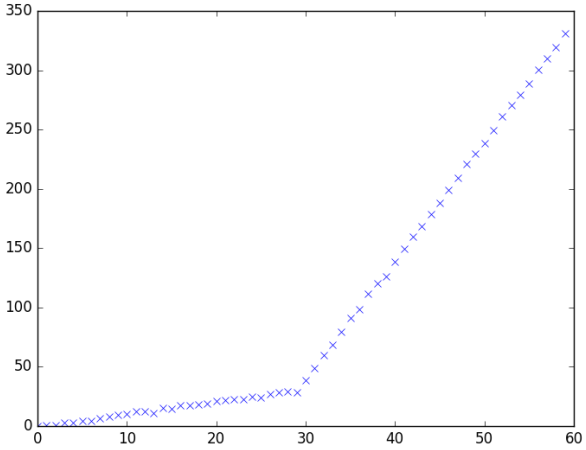
Anomaly Types: Our study focuses on three general types of anomalies.



I. Random Outliers



II. Sudden Gap



III. Slope Change

We construct various datasets with these three structures to evaluate the efficacy of our methods.

We also examine real-life time series datasets from Twitter giving counts for the number of Tweet mentions of large, publicly-trade companies (e.g. AAPL, AMZN, CVS, FB) over five-minute intervals.

Standardizing the Algorithm

Here is our generalized, two-step procedure for analyzing a dataset, constructing Martingales, and detecting anomalies. We wrap all of our methods in an `AnomalyDetector` class, whose objects can be instantiated with two main customizable options – (1) a strangeness function and (2) a Martingale construction method.

Part One: Importing the training examples one by one and generating a list of corresponding *p*-values.

- We select a *strangeness function* f that inputs a fixed set of training examples and outputs a list of corresponding, relative *alpha values* that signify the degree to which each example deviates from the rest.
- Different types of strangeness functions can be found in our `strange` module; these include metrics based on average distance, range percentile, residuals calculated from OLS, and changes in slope/trend calculated from OLS
- For each new example z_n, we compute $\alpha_1, \ldots, \alpha_n = f(z_1, \ldots, z_n)$ and calculate our new *p*-value p_n as

$$p_n = \frac{\#\{i : \alpha_i > \alpha_n\} + \theta_n \#\{i : \alpha_i = \alpha_n\}}{n}$$

where θ_n is a random number from the interval [0, 1] and $\#\{\}$ denotes the cardinality of a set.

- Thus, lower *p*-values will be given to unusual examples that do not quite fit the general trend of the data up until that point.

Part Two: Constructing a Martingale from the sequence of *p*-values.

- We select a *Martingale construction method* that inputs the set of *p*-values generated in Part One and uses a particular procedure to calculate/plot the Martingales.
- Specific methods will employ the use of a *betting function*, which inputs a *p*-value and outputs a representative value for Martingale calculation. The array of betting functions can be found in our `betting` module.
- Different types of Martingale construction methods can be found in our `martingale` module; these include power-based approaches in which we select some exponential $\epsilon \in [0, 1]$, a simple-mixture approach that is ϵ independent, and a plug-in approach that estimates the PDF of the sequence of *p*-values

This general algorithm was first introduced by Vovk et al. in [1]. In this project, we add our own customizations to answer the specific questions we posed.

Evaluating Strangeness Functions

A First Experiment: We compare the relative efficacy of different strangeness functions for Part One of the algorithm in generating Martingales.

- Each dataset is self-constructed, contains 110 data points and 2 'change points' that signify the onset of an anomaly.
- For this part of the analysis, we standardize Part Two of the algorithm by using a *power* method with a *fixed* betting function and $\epsilon = 0.8$.

Here are the four different strangeness functions we examine. Note that they all performed well on *non-anomalous* `Norm(0, 1)` data, returning very low Martingale values.

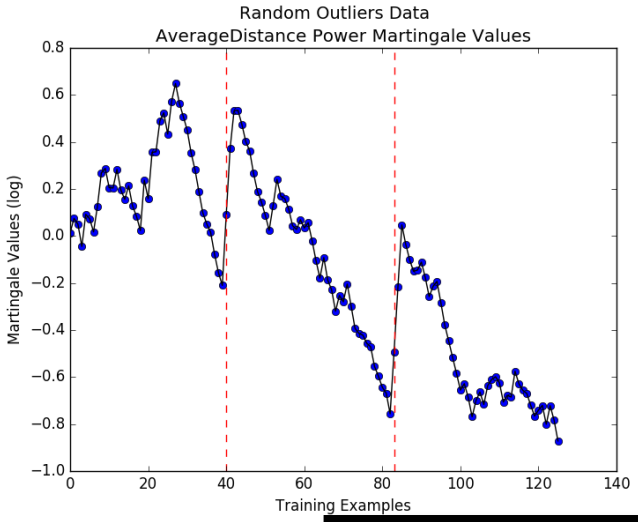
Average Distance – returns avg. squared distance from z_i to all other points

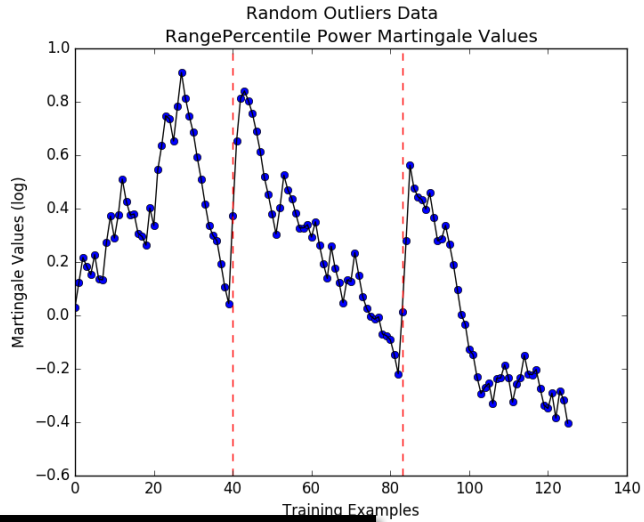
Range Percentile – returns ratio of $z_i - \min_j z_j$ to $\max_j z_j - \min_j z_j$

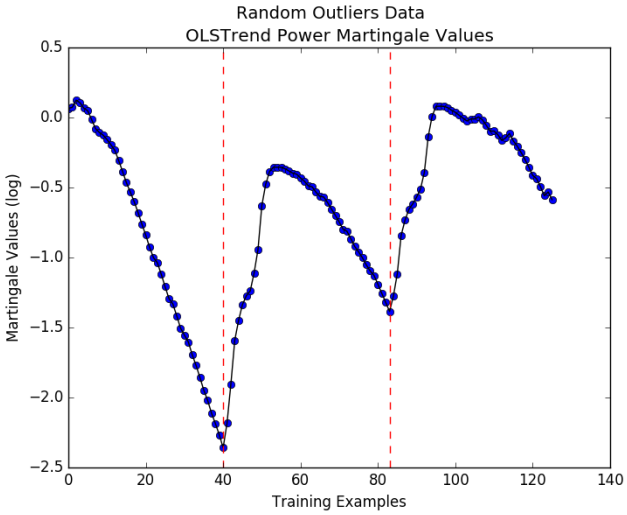
OLS Trend – returns average squared distance from OLS slope trained on points $z_{i-w} \ldots z_i$ to OLS slope trained on all other sets of w points

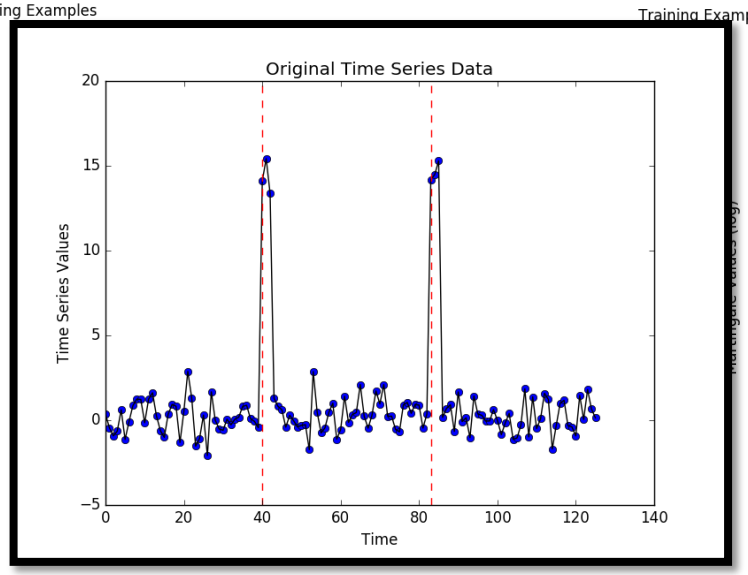
OLS Residual – returns absolute diff. between z_i and the value $\hat{z}_i$, predicted from OLS trained on points $z_{i-w} \ldots z_{i-1}$ for some w

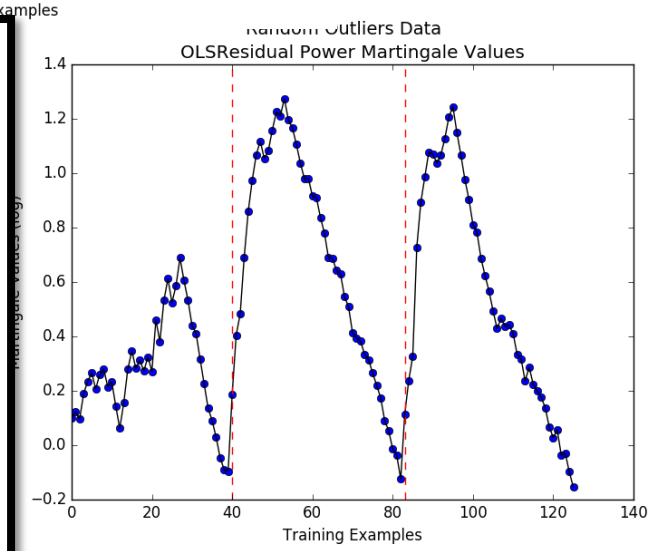
I. Random Outliers



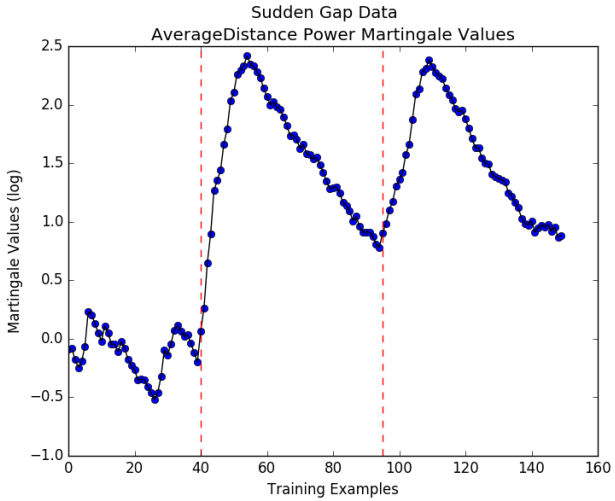


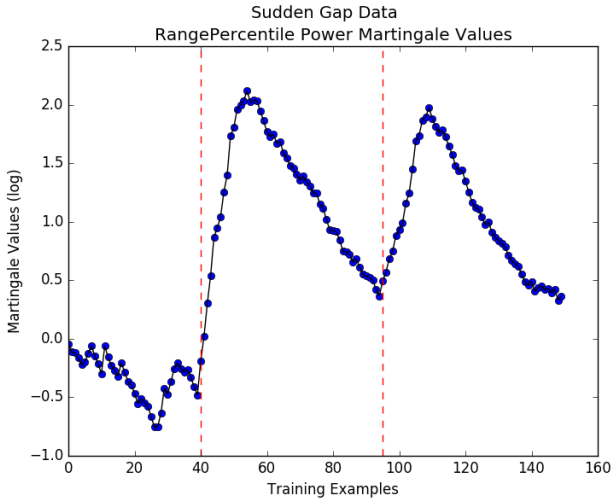


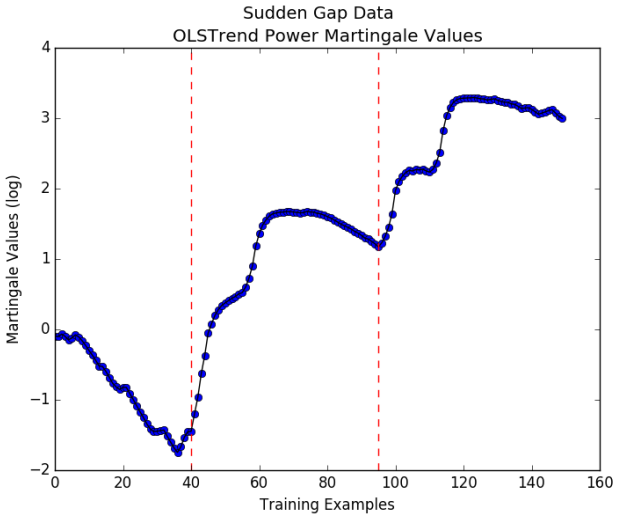


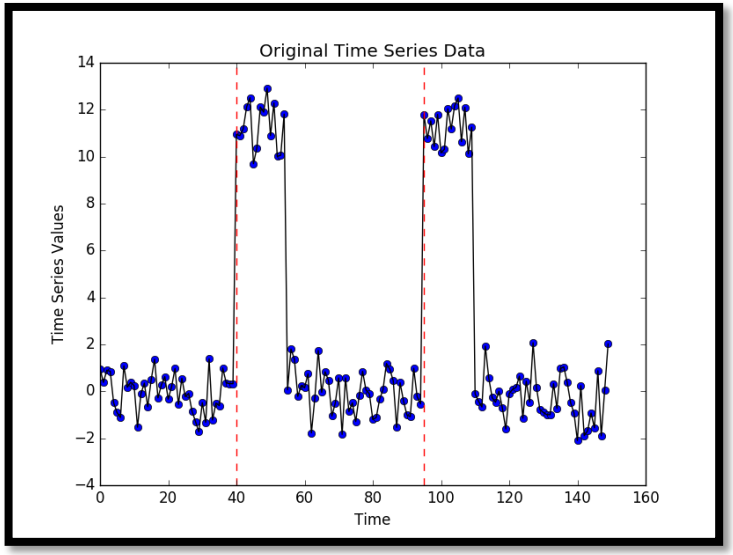


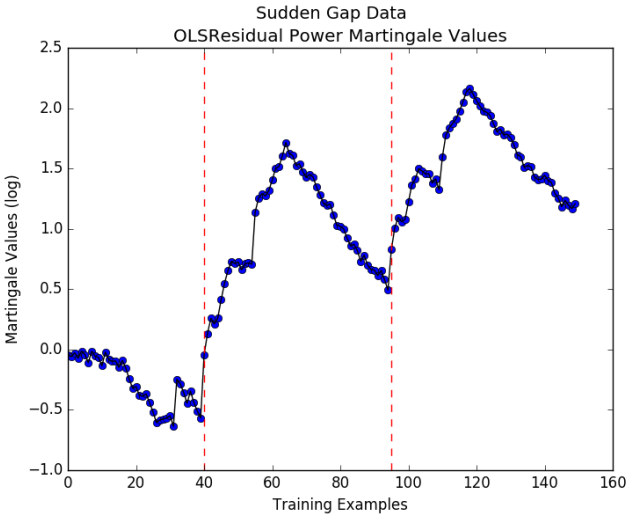
II. Sudden Gap



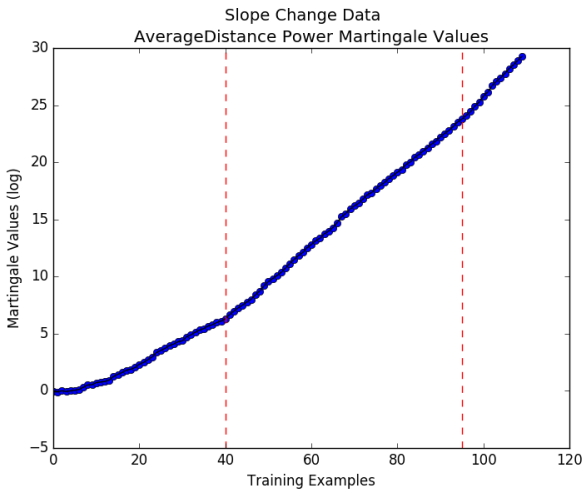


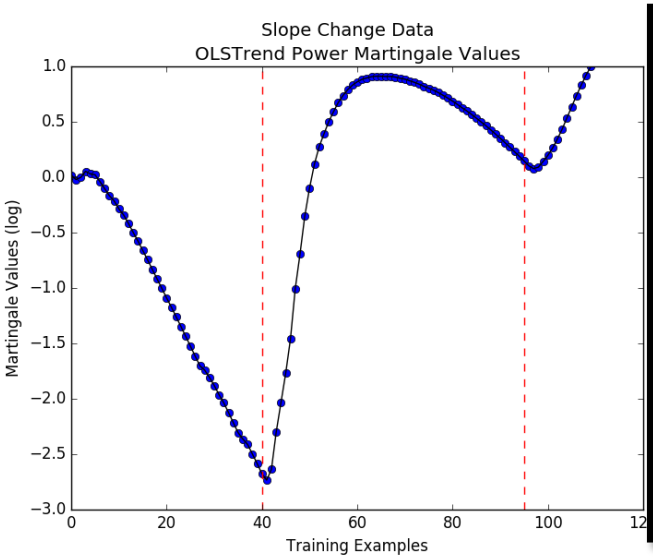


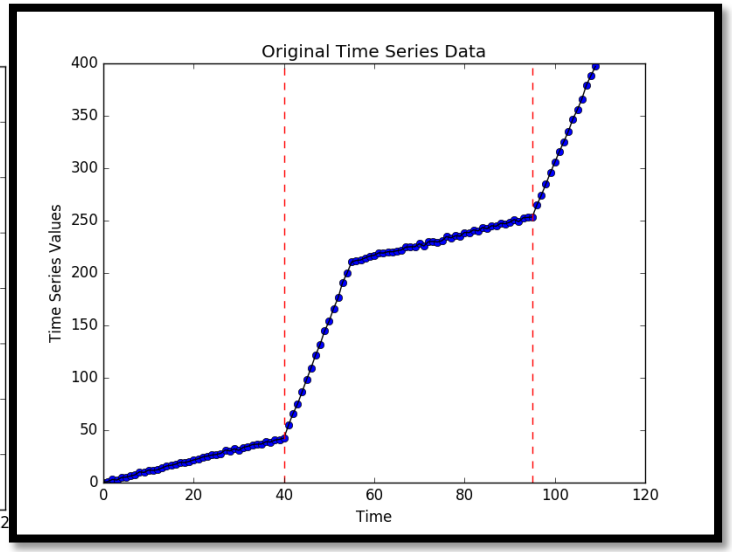


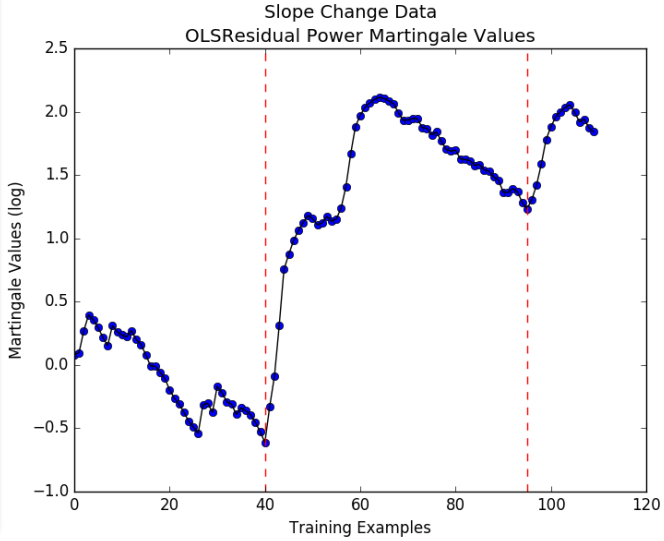


III. Slope Change









Discussing the Results

Random Outliers: All four strangeness functions produce Martingales that locally peak right after an outlier occurs in the time series data. However, due to the fact that there are very few outliers, these peaks for Average Distance, Range Percentile, and OLS Trend are not actually high enough to be global maximums, which is undesirable. Instead, we see that **OLS Residual** performs the best, which is expected since the difference between a random outliers' expected value and its predicted value should be significantly large.

Sudden Gap: All four methods seem to perform well for this type of anomaly; they all have peak Martingales after the appearance of a gap. Note that **Average Distance** produces the highest-valued Martingale.

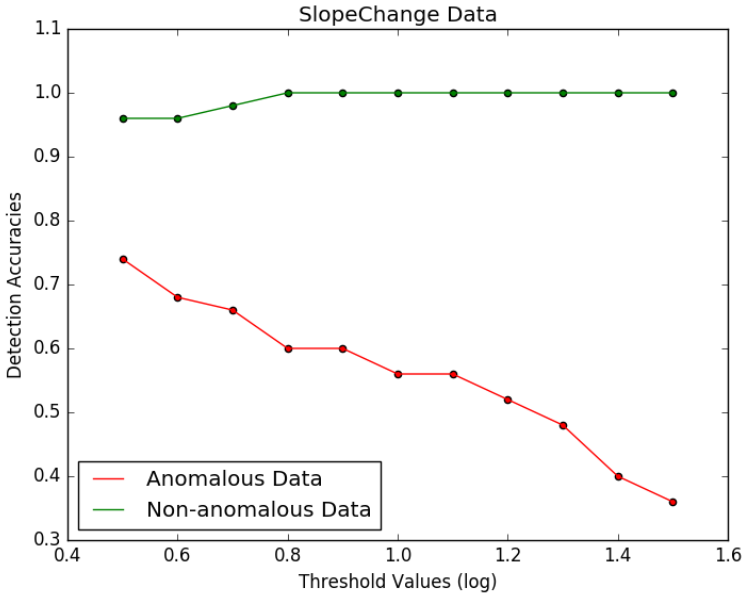
Slope Change: Average Distance and Range Percentile perform poorly, as they start increasing before the slope changes. **OLS Trend** seems to most accurately reflects the patterns of the slope changes, which is understandable given that this metric measures slope fluctuations.

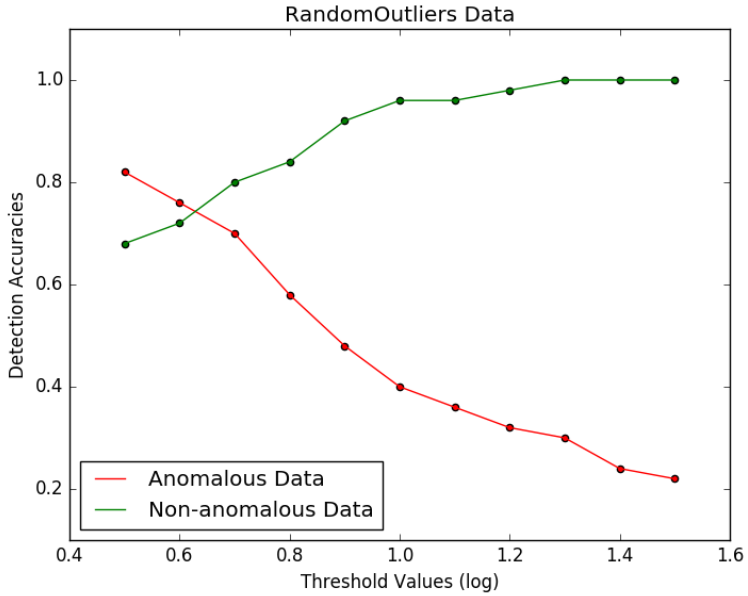
Comparing Different Thresholds

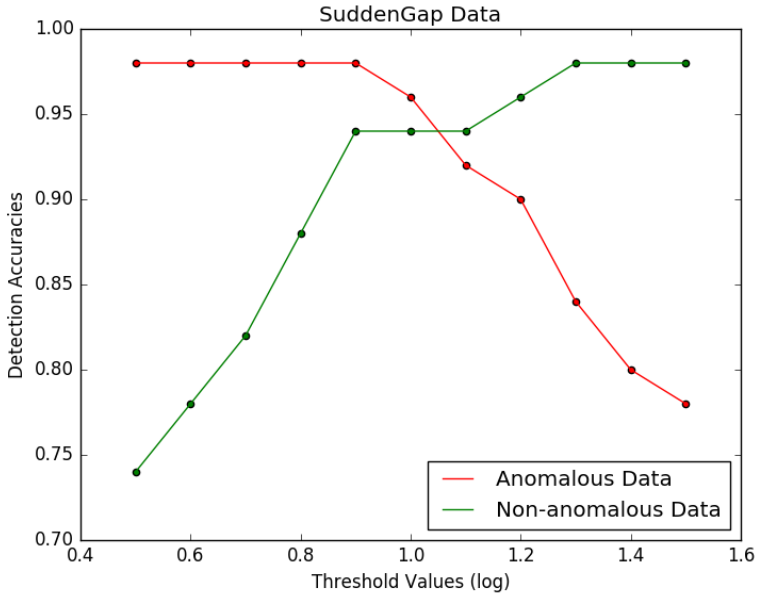
Methodology: We evaluate the efficacy of anomaly detection for different Martingale thresholds. We expect there to be a tradeoff between the accuracy and the confidence of our methods. Here are some general points:

- We focus on the three aforementioned anomaly types. We use a different strangeness function for each, based on the previous results.
  - *Random Outliers* → **OLS Residual**
  - *Sudden Gap* → **Average Distance**
  - *Slope Change* → **OLS Trend**
- For each anomaly type, we randomly generate 50 anomalous examples and 50 non-anomalous examples and test the algorithm with the appropriate strangeness function. We record accuracy percentages for thresholds of 10^m, where $0.5 \leq m \leq 1.5$.
- Again, we standardize Part Two of the algorithm by using a *power* method with a *fixed* betting function and $\epsilon = 0.8$.

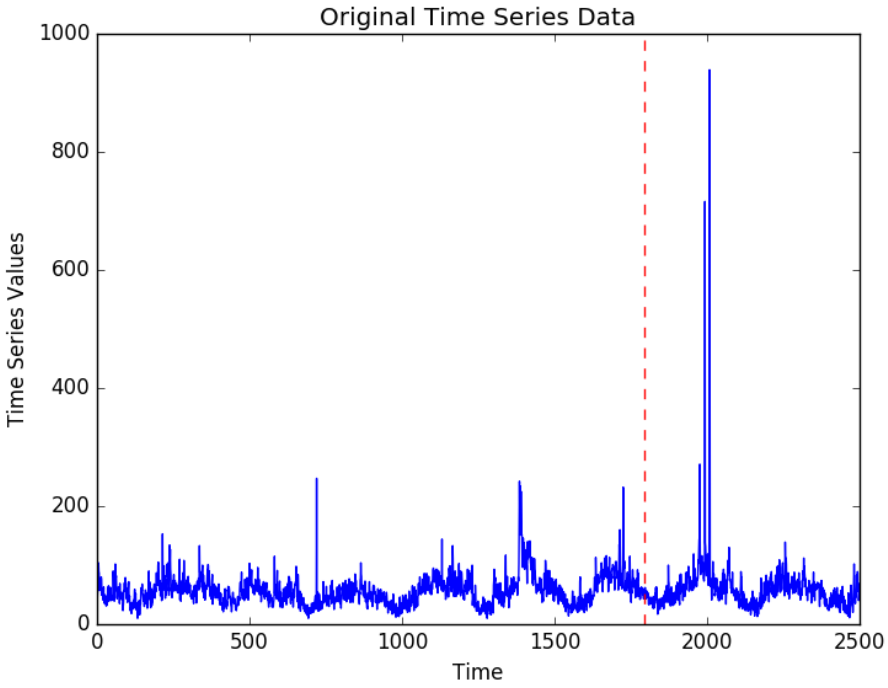
Threshold Tradeoffs

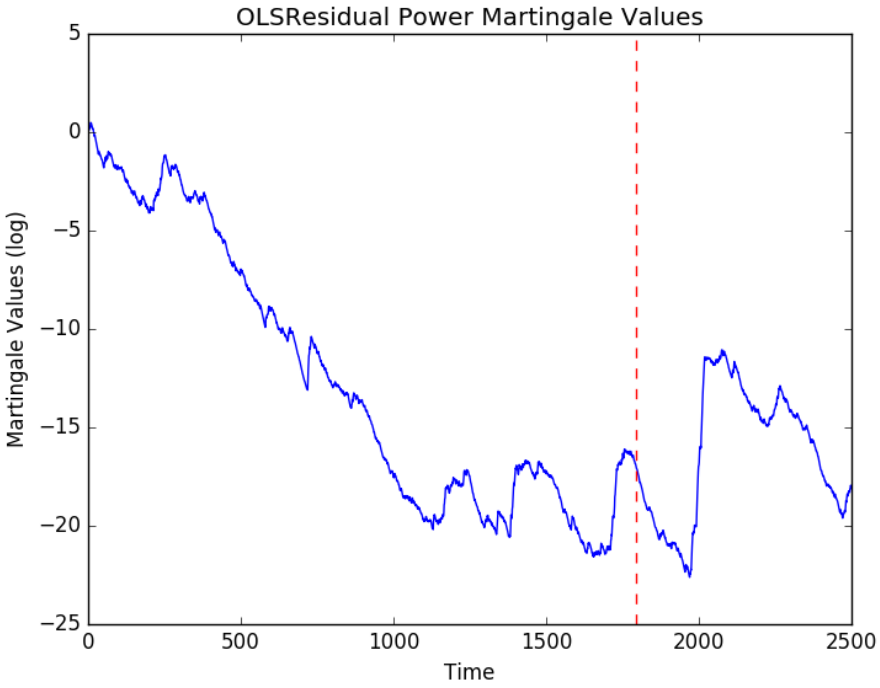






Twitter Data (AMZN)





This is an example of our algorithm running on the AMZN set of the Twitter dataset. Note that it exhibits a Random Outliers structure. Thus, we chose to use the OLS Residual strangeness function. Here we plot the first 2500 points of the dataset.

Discussing the Results (II)

As expected, the results from plotting the accuracies for the anomalous and non-anomalous data for various values of the threshold show that there is a tradeoff between accurately detecting anomalies and classifying "normal" data as non-anomalous.

- Setting the threshold value too low will result in very high accuracies for anomalous data, since nearly all data streams will be labeled as violating the exchangeability condition, but the tradeoff is that non-anomalous data streams will be mistakenly classified as anomalous too.
- Note that the Slope Change data run with the OLS Trend strangeness function shows that nearly all threshold values produce ~100% accuracy on non-anomalous data. We expect this given that previous results show that OLS Trend for non-anomalous data almost monotonically drops from 0.
- Optimal threshold can be chosen by considering which of the 2 accuracies (anomalous and non-anomalous) is more important to the user/application at hand.

Conclusions and Future Work

Choose Strangeness Function Based on Data: Different strangeness functions are better suited to detecting different anomaly types.

Choose Threshold Based on User Priorities: There is a tradeoff between detecting anomalies accurately and classifying non-anomalous data streams correctly. Choose a threshold based on the user's priorities.

Future Work:

- Explore performance of various strangeness functions for *labeled* data (1-Nearest Neighbor, SVM, Decision Trees) since our current work focuses only on unlabeled data streams.
- Evaluate performance of current methods on *multidimensional* datasets and make improvements as necessary.
- Explore other martingale construction methods, including Sleepy Jumper, etc.
- After hitting a change point, remove previous points from memory to improve accuracy of detecting future anomalies.

References

1. Vovk, V., Nouretdinov, I., and Gammerman, A. Testing exchangeability on-line. In *Proceedings of the 20th International Conference on Machine Learning*, pp. 768–775, 2003.
2. Ho, S.-S. A martingale framework for concept change detection in time-varying data streams. In *Proceedings of the 22nd International Conference on Machine Learning*, pp. 321–327, 2005.
3. Fedorova et. al. Plug-in martingales for testing exchangeability on-line. In *Proceedings of the 29th International Conference on Machine Learning.*