ETH
Swiss Federal Institute of Technology Zurich

Seminar for Statistics

Department of Mathematics

Master Thesis Summer 2012

Amanda Strong

A review of anomaly detection with focus on changepoint detection

Submission Date: August 24th 2012

Co-Adviser Markus Kalisch
Adviser: Prof. Dr. Sara van de Geer

Thanks Mom and Dad!

Abstract

Anomaly detection has the goal of identifying data that is, in some sense, not "normal." The definition of what is anomalous and what is normal is heavily dependent on the application. The unifying factor across applications is that, in general, anomalies occur only rarely. This means that we do not have much information available for modeling the anomaly generating distribution directly. We will describe several ways of approaching anomaly detection and discuss some of the properties of these approaches.

Changepoint detection can be considered a subtopic in anomaly detection. Here the problem setting is more specific. We have a sequence of observations and we would like to detect whether their generating distribution has remained stable or has undergone some abrupt change. The goals of a changepoint analysis may include both detecting that a change has occurred as well as estimating the time of the change. We will discuss some of the classic approaches to changepoint detection.

As very large datasets become more common, so do the instances in which it is difficult or impossible for humans to heuristically monitor for anomalous observations or events. The development and improvement of anomaly detection methods is therefore of ever-increasing importance.

Contents

List of Figures

Chapter 1

Introduction

If we have a situation where we have some idea of what ”normal” is, and we would like to detect any event that does not conform to this idea of normalcy, we are performing anomaly detection. As data collection and storage technologies have improved, the importance of developing anomaly detection techniques have increased. There are more and more problem settings where, because of the size or complexity of the data set, it is either impossible or impractical for humans to heuristically discover anomalous data, especially not with the speed that many applications require.

Anomaly detection techniques have a broad range of applications. They have been applied to, among other things, intrusion detection (in computer networks), credit card fraud detection, medical monitoring, and industrial process monitoring and quality control. One unifying factor across applications is that the distribution of anomalous data is not well described. In most cases, anomalies, by definition, occur only rarely. In other words, we have unbalanced anomaly and normal classes. This is what distinguishes anomaly detection from the usual classification problem. However, beyond this, the specifics of the approach are heavily dependent on the application.

In terms of input, anomaly detection methods must of course operate on many kinds of data, dependent on the application: Discrete, continuous, time series, etc. Perhaps the more important question about the dataset is whether it is labeled. Anomaly detection methods can operate in a supervised, semi-supervised, or unsupervised context. If we have both normal and anomalous labeled data, we have a supervised situation. However, it is unusual to have both types of labeled data available. It is much more common to have only a set of labeled normal data available, in which case we call the situation semi-supervised. Of course, if we have a completely unlabeled dataset, we have an unsupervised situation. In the unsupervised setting, we usually have to assume that anomalies are rare in the dataset in order to reasonably construct a model describing normality.

Anomaly detection methods also can have one of two main types of output. Some methods have binary output and simply classify a data instance as anomalous or non-anomalous. Others assign a score to each data instance. The score indicates in some way the degree to which the method considers the point anomalous. Even if the score has no formal statistical meaning, it can be useful. It is possible to tune a detection method by adjusting the threshold score over which we declare an instance to be anomalous.

Another consideration is whether or not the method is designed to operate on streaming

data. Many applications require real time assessment of incoming new data, such as intrusion detection or credit card fraud detection. Of course, any method can operate on streaming data if we have the computational resources to retrain a model after every new data point. However, that is usually not possible.

In Chapter 2, we will discuss several popular approaches to anomaly detection. We can assign many techniques to five broad categories: Nearest neighbor/density based approaches, clustering based approaches, classification based approaches, statistical approaches, and spectral approaches. These categories are suggested in the excellent survey article by Chandola, Banerjee, and Kumar (2009). We will further talk about how the above-mentioned characteristics apply within each category of approach.

Changepoint detection is a more specific subtopic in anomaly detection. In this case we have a sequence of observations. The observations are assumed to come from one distribution initially, but at some point that distribution may change. The two main goals of changepoint analysis are to determine if such a change has occurred and, if it has, estimate the time of the change. Methods vary based on what restrictions are placed on the pre and post change distributions and what information we have about potential changes.

Changepoint detection can also take place in a batch or online setting. The basic approach to the retrospective problem is to find a test statistic appropriate for testing between the hypothesis that a change has occurred and the hypothesis that no change has occurred. This statistic is usually based on a likelihood ratio, but other approaches exist. The online setting has the goal of detecting a change as quickly as possible once it has occurred. The most basic approach to this is also based on likelihood ratios. We will go into more details about this in Chapter 3.

Chapter 2

Anomaly Detection

2.1 Nearest Neighbor Based Approaches

2.1.1 Overview

We can use a nearest neighbor analysis to arrive at an anomaly score for a point in two ways: We can base a data point's anomaly score on either the distance to its kth nearest neighbor or on its local density relative to the local density of its neighbors. These are complementary ideas which both work under the assumption that normal data has close neighbors (i.e. occurs in dense regions) and anomalous data occurs in less dense regions.

The first step for any method built on nearest neighbors is to define a distance measure on the data space. The most basic choice is euclidean distance, but of course this only works for data with only continuous features. For categorical features, a matching coefficient is often used. One possibility for mixed features is Gower's similarity coefficient. Although these are common choices, there are many possible choices of measure.

The main advantage of nearest neighbor based methods is that they can operate on many different types of data once an appropriate distance measure has been defined. They don't require any assumptions about the data distribution. Another advantage is that they can work well in unsupervised settings. Additionally, they output an anomaly score that allows for application-dependent appropriate thresholds to be determined.

Their main disadvantage is their computational expense. Calculating an anomaly score for a test data point usually involves computing the distances between the point and all points in the test or training data, unless some optimization methods are used. Another weakness of nearest neighbor techniques is their dependence on a meaningful distance measure, which is not always trivial to define. In high dimensions this can be especially difficult and requires good feature selection to reduce the dimension.

2.1.2 Distance Based Methods

A pared down distance based approach may be the simplest anomaly detection technique. Data points are ranked by the distance to their kth nearest neighbor. Then, the highest

ranked points are identified as potential anomalies by a user defined parameter - Either a threshold distance is used or the top n data points are taken.

The method of ranking the data can be changed without modifying the essential technique. Several papers propose an anomaly score based on a sum of the distances to a point's k nearest neighbors. For example, Zhang and Wang (2006) present a subspace search algorithm for finding anomalies in high dimensions, HighDOD, that is based on this sum of distances.

2.1.3 Density Based Methods

In density based approaches, the fundamental idea is to use the distance to a point's kth nearest neighbor as a measure of the inverse of the density at that point. If we just ranked data points by this density estimate the idea is not very different from the basic distance method. If the normal data has different densities in different regions, this method fails. For example, if there are two clusters of normal data but one cluster is more diffuse than the other, the points in this cluster will always end up with a higher anomaly score than points in the other cluster. This is illustrated in the figure below. We can see that $o1$ is probably an outlier, while $p1$ is not. However, the relative densities at these points are similar in a global comparison, so $o1$ might not be detected as an outlier. To correct for this, many techniques compare the local density at a point with the local densities of its nearest neighbors to arrive at an anomaly score.

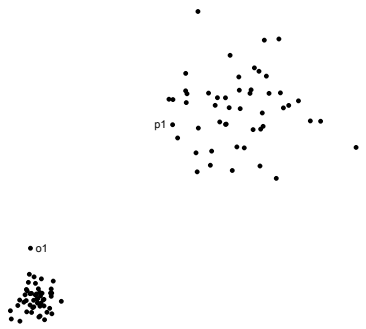

Figure 2.1: A situation where Local Outlier Factor is useful

Breunig, Kriegel, Ng, and Sander (2000) introduce the local outlier factor (LOF), which allows for points to be outliers relative to their immediate surroundings rather than only relative to the entire data set. First, the k-distance is calculated for each data point, essentially the distance to the kth nearest neighbor. This defines the k-distance neighborhood of the point, the neighborhood that contains every object closer to the point than the k-distance. Then the reachability distance of a point p with respect to point q is defined as reach-dist$(p, q) = \max\{\text{k-distance}(q), d(p, q)\}$. The local reachability distance of a point p is then the inverse of the average reachability distance based on the $MinPts$ nearest neighbors of p, where $MinPts$ is a user-defined parameter. The local outlier factor of p is then the average of the ratio between p's local reachability distance and the reachability distance of p's $MinPts$ nearest neighbors. Thus, a point has a high LOF score if its local

reachability density is lower than its neighbors. Therefore, $o1$ from our earlier example will have a high LOF score.

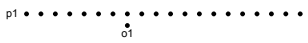

Figure 2.2: A situation where Connectivity-based Outlier Factor is useful

Tang, Chen, Fu, and Cheung (2002) propose an extension of the LOF idea, Connectivity-based Outlier Factor (COF). The COF again compares the local density at a point with the local density of its neighbors. However, instead of defining a neighborhood of a point in one step, points are added incrementally. The point and its closest neighbor initially form the neighborhood set. The next point to be added will be the point closest to this set, that is, the point closest to any member of the set. This proceeds until the set includes the predetermined (by k) number of points. This neighborhood structure makes it possible to recognize structures in the data like lines or rings. For example, in the figure below, $o1$ seems to be an outlier, while $p1$ does not. However, $p1$ might have a higher LOF score than $o1$. However, the COF will detect $o1$.

Papadimitriou, Kitagawa, Gibbons, and Faloutsos (2003) introduce another variation on LOF, the Multi-Granularity Deviation Factor. The r-neighborhood of a point p is defined as the set of all data points within the radius r of point p. The MDEF for a point p with respect to radius r represents the relative deviation of its local density from the average local density for points in its r-neighborhood. This method has shown success in detecting anomalous micro-clusters, which are hard to discover with many of the previous density-based methods.

2.2 Clustering Based Approaches

2.2.1 Overview

Clustering based anomaly detection techniques work under the assumption that normal data points fit well in large, dense clusters and anomalous points either don't belong in any cluster, are far from their cluster centroid, or occur in small or sparse clusters.

The basic clustering procedure is simple: The data is clustered and then each point is analyzed with respect to its closest cluster.

Clustering based techniques have the advantage of being well suited to unsupervised problems. It is also true that any clustering algorithm can be used; we don't require a special anomaly detection algorithm. We therefore have a wide range of clustering methods from which we can choose based on what works for a given data type and problem setting.

The fundamental disadvantage of clustering techniques is that of course the data must have some kind of detectable clustering structure. They can therefore be a bit less general than nearest neighbor techniques. On the other hand, some of the same problems arise in clustering and nearest neighbor based anomaly detection. There is still the problem of defining a meaningful distance measure on the data space and clustering methods can also be computationally expensive.

2.2.2 Methods Using Distance to Cluster Centroid

For these methods, the data is clustered using any existing clustering algorithm. Then, the anomaly score of a point is simply its distance to its closest cluster centroid. This can be done in either an unsupervised or semi-supervised fashion. In the unsupervised case, all points are clustered and then scored by their distances from their cluster centroids. In the semi-supervised case, all normal data points in the training set are clustered, then test points are compared to those clusters. In the unsupervised setting, this method fails to detect anomalies that themselves form clusters.

2.2.3 Methods Comparing Cluster Size and Density

These techniques attempt to detect groups of anomalies that form clusters under the assumption that small or sparse clusters represent anomalies. Any cluster that falls below a user-defined size threshold is identified as potentially anomalous.

An elaboration of this idea is proposed by He, Xu, and Deng (2003), the Cluster-Based Local Outlier Factor (CBLOF). After data is clustered, the CBLOF is calculated for each point. The CBLOF calculation depends on the size of the cluster the point is in. Within a small cluster, the CBLOF is defined as a product of the size of the cluster and the distance between the point and the closest large cluster. For points within a large cluster, the CBLOF is a product of the size of the cluster and the distance between the point and its own cluster centroid. We see that this method can't deal with cases where an anomaly does not form part of a cluster or forms part of a very small cluster, because its small cluster size can offset its distance from nearby larger clusters, even if that distance is large.

An often-used way to detect anomalies using cluster size, with much less computational expense, is to use fixed-width clustering. This idea was introduced by Eskin, Arnold, Prerau, Portnoy, and Stolfo (2002). Clusters are built incrementally: If a point is further away from the preceding point than a pre-defined threshold, a new cluster is formed. Otherwise, it is added to the current cluster. This can be done with a single pass over the data.

2.2.4 Methods Leaving Some Points Unclustered

Some clustering algorithms allow for residuals rather than including every point in a cluster. Although most of these were not designed specifically for anomaly detection, some

have been used for that purpose. Since their primary purpose is not finding anomalies, performance is generally suboptimal.

Yu, Sheikholeslami, and Zhang (1999) reports successful results with their anomaly detection method, FindOut, which is based on the WaveCluster algorithm. The data are first transformed into multidimensional signals using the wavelet transformation. The idea is that high frequency parts of the signal correspond to areas where the data distribution is changing rapidly, i.e. the boundaries of clusters. The low frequency/high amplitude parts of the signal correspond to areas where the data is concentrated, i.e. within clusters. Removing the dense regions in the transformed data space corresponds to removing data clusters. The presumption is that remaining points are anomalies. If the data is very noisy, this technique may treat noise as anomalies and have a high false positive rate.

2.3 Classification Based Approaches

2.3.1 Overview

Many ordinary classification techniques can be applied in anomaly detection. The assumption here is only that it is possible to learn a classifier in the data space. If we have data where only the normal points are labeled (semi-supervised), the classifier learns one class for the normal instances. Any test point that falls outside this class boundary is identified as anomalous. If we have data with both normal and anomalous class labels, we can use multi-class classification.

If we can perform supervised classification, classification based anomaly detection methods have the advantage of achieving high accuracy in discovering known anomaly types. The downside is that these methods tend to fail to detect new anomaly types. The biggest limitation is that, of course, we require labeled data, which is rare in anomaly detection problems.

The semi-supervised approach is easier in practice. Classifiers can accurately learn the normal behavior but are less limited in detecting previously-unseen types of anomalies. It is also far more common to have labeled normal data.

2.3.2 Methods for Balancing Classes

Even if fully labeled data is available, the intrinsic rarity of anomalies means that the classes will be significantly unbalanced. Standard decision theory based procedures may not give good results, since high classification accuracy can be achieved just by classifying every point as normal. Therefore, to directly apply a standard classification method that has not been designed for anomaly detection, it is beneficial to somehow balance the classes. Aside from adjusting a loss function, this can be achieved by over-sampling from the anomaly class or under-sampling from the normal class. Chawla, Bowyer, Hall, and Kegelmeyer (2002) propose SMOTE: Synthetic Minority Over-sampling Technique, which combines both over- and under-sampling to achieve better classifier performance. SMOTE, like several other existing methods for over-sampling the anomaly class, also involves generating artificial anomalies. In this case, these artificial anomalies are generated within the region of the existing anomalies.

2.3.3 Support Vector Machine Based Methods

Support Vector Machines can be directly applied to anomaly detection if there is labeled data available, possibly in conjunction with some kind of class balancing. However, it is also possible to use them in an unsupervised context based on an idea proposed by Steinwart, Hush, and Scovel (2005). The underlying assumption is again that anomalies occur in less dense regions compared to normal data instances. We can create a boundary between normal and anomalous regions by finding the level sets of the density on the data space. A Support Vector Machine can be applied to classify data by its local density.

2.3.4 Rule Based Methods

Again, many existing rule based classification techniques have been applied to anomaly detection problems. However, some algorithms have been developed specifically for unbalanced class problems and anomaly detection. The general idea of using association rule mining in anomaly detection is that rules with high support may be said to characterize normal behavior, while anomalous data points occur in relatively few frequent itemsets compared to normal data.

Joshi, Agarwal, and Kumar (2001) introduce the PN-rule algorithm, which is further specialized in Joshi and Kumar (2004). The PN-rule algorithm is a two phase procedure that focuses on modeling the rare anomaly class rather than on overall accuracy. In the first phase, P-rules are learned, which are rules that predict presence of the target class. In the P phase, the goal is to learn rules that cover the anomaly class data examples with high support and with good recall. In the second phase, N-rules are learned, which are rules that predict absence of the anomaly class and remove false positives. The CREDOS algorithm includes a growth and pruning phase to improve generalization.

2.3.5 Neural Network Based Methods

Several types of neural networks have been used in anomaly detection but for a multi-class setting with labeled training data, the basic procedure is the same. The neural network is trained on the normal training data and then test instances are classified as anomalous if they are rejected by the neural network.

Neural network based anomaly detection has been expanded beyond this basic approach by Hawkins, He, Williams, and Baxter (2002). Here, a replicator neural network serves as a form of dimensionality reduction. A multi-layer feed forward neural network with the same number as input and output nodes which are the data features. Once the neural network is trained, test data that can not be well reconstructed by this data is considered to be anomalous. The reconstruction error is the difference between the original value for a data feature and the output of the neural network at the output corresponding to that feature, summed over all features.

2.4 Statistical Approaches

2.4.1 Overview

When taking a statistical approach to anomaly detection, most methods make assumptions used in more traditional outlier detection. That is, the data is generated by a stochastic model, and an anomaly is something not generated by this underlying model. This means that we assume we can fit a model to the normal data and then use some kind of test to determine how probable it is that a test point was generated by this model. Both parametric and non-parametric techniques can be used to estimate the model.

If we have a simple data structure with low dimensionality, it is simple to apply basic statistical techniques to anomaly detection. We can simply fit some distribution and then assume that the anomaly score of a point is the inverse of the density at that point. We can also construct a hypothesis test with the null hypothesis that the test point was generated by the assumed distribution. If the null hypothesis is rejected, the point is assumed to be anomalous. However, in high dimensions it is no longer trivial to fit models or find the distribution of test statistics.

The primary advantage of statistical techniques is that anomaly scores have a statistical meaning and sometimes have an associated confidence interval. The main disadvantage is of course the requirement that we assume a particular distribution for the data. Then, even if we are successful in finding and fitting a distribution, we still have the problem of finding appropriate anomaly score cut offs or test statistics, which is difficult for more complex distributions.

2.4.2 Parametric Methods

If the data is generated by a Gaussian distribution, then Grubbs' test Grubbs (1969) can be used to find anomalies in univariate data. The distance between a data point and the mean can be incorporated into a test statistic that follows a t distribution. This test can be generalized to multivariate data by using the Mahalanobis distance of a point to the mean as in, for example Laurikkala, Juhola, and Kentala (2000). This reduces the problem to a univariate one: Grubbs' test is applied to the Mahalanobis distance to decide if a point is anomalous.

Typical outlier detection techniques from regression analysis can be used and elaborated on here. The basic method is to fit a regression model, then use the residual associated with a point as its anomaly score. Methods from robust regression can help deal with training data that may contain unlabeled anomalies. These methods are applied in Bianco, Garcia Ben, Martinez, and Yohai (2001) and Chen, Shao, Hu, and Su (2005) for Autoregressive Moving Average Models. Tsay, Pena, and Pankratz (2000) generalize to multivariate ARIMA models.

Some statistical techniques attempt to model two separate distributions, one for the normal data and one for the anomalous data. Eskin (2000) proposes such a mixture of distributions, where the distribution of the data, D, is given by:

$$D = (1 - \lambda)M + \lambda A$$

where M is the distribution of the normal data and A is the distribution of the anomalies. λ is the a priori probability of a data point being an anomaly. Initially, all data points are assigned to M. Then, for each point in M, the following procedure is performed: First, parameters for both M and A are estimated. Then, the log likelihood of the data set, D, is calculated. Next, the point is removed from M and assigned to A and parameters for the two distributions are estimated again. The log likelihood of D is calculated again. If the log likelihood is increased more than a predefined threshold, the data point is considered an anomaly and stays in A. Otherwise it is re-assigned to M.

2.4.3 Nonparametric Methods

A common way to model normal data in fraud detection or network security applications is to use histograms to model a normal use profile. For univariate data, applying this method is trivial. All that we do is create a histogram from normal training data values. Then, if a test data point falls outside of the histogram bins with normal data, that point is considered to be anomalous. The most important consideration is of which bin size to use, which is a tradeoff between lower anomaly detection rate with large bins and high false positive rate with small bins. Many anomaly detection studies have generalized this technique to multivariate data by simply making univariate histograms feature by feature and then constructing an overall anomaly score by combining feature-wise anomaly scores. Of course, this method can fail to detect anomalies that do not have extreme values for any one feature.

Yamanishi, Takeuchi, and Williams (2000) present SmartSifter, an anomaly detection method that can handle multivariate data with both discrete and continuous features. A histogram density is found over the domain of the categorical features. For each cell in this histogram density, a finite mixture model is fitted for the continuous features. The model is updated incrementally, as each data point is entered. Then, the anomaly score of that data point is based on the change to the model after the update associated with that point.

2.5 Spectral Approaches

2.5.1 Overview

These types of approaches assume that data can be projected onto some lower dimensional subspace and that normal data and anomalies can be distinguished in this subspace. This means that these techniques have the advantage of being well suited to unsupervised application. On the other hand, they assume that anomalous and normal points can be well distinguished in some subspace that we can find which may not be the case.

2.5.2 Principal Component Analysis Based Methods

Principal component analysis can provide us with such subspaces.The concept underlying many of these methods is the idea that if we find some combination of data features that explain most of the variability in the data, this lower dimensional set of attributes will be able to describe normal instances well but anomalous data only poorly. Parra, Deco,

and Miesbach (1995) project each data point onto the low variance principal components. Points with a high value in such a projection are considered to be anomalous.

Shyu, Chen, Sarinnapakorn, and Chang (2003) introduce an anomaly detection scheme based on robust principal component analysis. Robust estimators are used for the training data correlation matrix, which makes this method applicable for unsupervised anomaly detection. PCA is carried out on this training data correlation matrix. It is proposed that we take the first q major components that explain about 50 percent of the total variation in the data features. The sum of the squares of the first q principal component scores is the first part of this principal component classifier. If this value exceeds a threshold, then the point is declared to be anomalous. This usually discovers anomalies that have outlying values for some of the original features. The second part of the classifier is based on the sum of the squared p minor principal component scores for a data point. If this exceeds a threshold, then the point is declared to be anomalous. This tends to discover anomalies whose feature values are not outliers in themselves, but whose correlation structure does not agree with the rest of the data.

2.6 On-line Methods

2.6.1 Overview

Many of the methods mentioned above are able to classify a new data point or assign it an anomaly score as it arrives in a data stream, without recalculating a new model. Of course, the more important questions usually are when, if, and how we will update the model as we get new data. This is an especially important consideration if we have reason to believe that the nature of non-anomalous data may be changing significantly over time. One convenient way to break down suggestions for updating schemes is into three categories: Periodic, Incremental, and Reactive.

2.6.2 Periodic Update

The most basic idea is to periodically refit the model after we have received a pre-defined number of data points. How frequently this can be done depends on the computational expense of the original modeling procedure. The obvious advantage of using this kind of update schedule is that in its simplest form we do not need any special method other than the one we have already used, and that we can adjust how frequently we update to suit the computing capacity we have available. The disadvantage is that, if we wait too long to update the model, we may end up incorporating anomalous structures as normal data. In other words, if we receive several similar anomalous points between updates, the model update might result in them being classified as a non-anomalous cluster. This is a real problem in many applications where something might happen to cause anomalies to appear in clusters.

2.6.3 Incremental Update

We can also update the model incrementally, that is, after every new data point. In most cases it is too computationally expensive to fully refit a model after every new observation,

but we can use methods with relatively efficient incremental update methods. For example, there exist incremental versions of the LOF and the COF. The Incremental LOF and COF are presented in Pokrajac (2007) and Pokrajac, Reljin, Pejcic, and Lazarevic (2008). SmartSifter, mentioned previously, is also a powerful method with an incremental update step based on the EM algorithm. One downside to an incremental update scheme is that updating after every new observation may not necessarily improve performance and therefore might waste computational resources.

2.6.4 Reactive Update

We can also base our decision to update on some criterion for the incoming data rather than on a fixed time between updates. We might choose to update if a new data point has an anomaly score that is beyond some threshold. Fu, Zhou, and Wu (2008) propose a reactive update method based on LOF and back propagation neural networks.

2.7 R Implementations

Because the range of anomaly detection methods is so broad, many of the techniques that underlie them are already at least partially implemented in R. There are many existing implementations of nearest-neighbor based classification, clustering, support vector machine or rule based classifiers, principal component analysis, etc. However, these implementations are often not explicitly designed for anomaly detection and might work in some cases for building a model of normal data, but require some further development to define what constitutes an anomaly. We mention some packages with more explicit anomaly detection functionality:

The package **Rlof** (Hu, Murray, with Strategic Data Mining Team, of Human Services, and Australian. (2011)) contains a (parallel) implementation of Local Outlier Factor. **DMwR** (Torgo (2010)) is a package meant to accompany "Data Mining with R, learning with case studies" by Luis Torgo. It also contains methods for computing LOF.

Anomaly detection methods that use points' distance to their nearest cluster centroid can be relatively easy to do using one of the many existing implementations of clustering methods like **cluster** (Maechler, Rousseeuw, Struyf, Hubert, and Hornik (2 02)), **mclust** (Fraley and Raftery (2006)),etc.

The **DMwR** package also includes an implementation of the SMOTE algorithm for unbalanced classes. With some method of class balancing, it may be possible to use one of the many classification methods existing in R in packages like **rminer** (Cortez (2011)) and **RODM** (Tamayo and Mozes (2011)). **arules** (Hahsler, Gruen, and Hornik (2005)) provides an interface to algorithms for mining frequent item sets and association rules as well as an interface for analyzing the results.

There are also many methods for finding outliers from a more traditional standpoint, including **mvoutlier** (Filzmoser and Gschwandtner (2012)) for multivariate outlier detection based on Mahalanobis distance via robust methods and **faoutlier** (Chalmers (2012)) for factor analysis-based outlier detection.

Chapter 3

Changepoint Analysis

3.1 Offline Hypothesis Testing

In the offline case, we are working with the entirety of a sample of fixed size. As we mentioned in the introduction, interesting tasks fall into two separate areas: Detection and estimation.

In the offline setting, detection generally takes the form of hypothesis testing. Here, we look at the simplest case of a sequence of independent random variables $x_1, x_2, ..., x_n$ with associated probability distributions $F_1, F_2, ..., F_n$. Our task is to test the null hypothesis

$$H_0 : F_1 = F_2 = ... = F_n$$

versus the alternative

$$H_1 : F_1 = ... = F_{k_1} \neq F_{k+1} = ... = F_{k_2} \neq F_{k_2+1} = ... = F_{k_q} \neq F_{k_q+1} = ... = F_n$$

where $1 < k_1 < k_2 < ... < k_q < n$. q represents the unknown number of changepoints and $k_1, k_2, ..., k_q$ are the unknown positions of the changepoints. The null hypothesis is also sometimes called the hypothesis of stability.

We often assume that the distributions $F_1, ..., F_n$ fall within the same exponential or parametric family and that we only need to test whether there is a change in one or more parameters. Then the hypothesis testing problem is simply

$$H_0 : \theta_1 = \theta_2 = ... = \theta_n$$

versus the alternative

$$H_1 : \theta_1 = ... = \theta_k \neq \theta_{k+1} = ... = \theta_{k_2} \neq \theta_{k_2+1} = ... = \theta_n$$

where $\theta \in \mathbb{R}^p$, q is again the unknown number of changepoints and $k_1, k_2, ..., k_q$ represent the unknown positions of these changepoints.

Of course, we can simplify a bit further if we are only looking for at most one changepoint. The only two possibilities are that the parameter determining the distribution of each observation is constant, equal to θ_1, or that there is a changepoint so that θ_1 is not equal to θ_n. We can formulate the testing problem as

$$H_0 : \theta_i = \theta_1 \text{ for i from 1 to n}$$

and

$$H_i : \theta_i = \theta_1 \text{ for } 1 \leq i \leq k \text{ and } \theta_i = \theta_n \text{ for } k < i \leq n$$

We must then test H_0 against the composite hypothesis

$$\mathcal{H}_1 : \cup_{i \geq 1} H_i$$

It is standard to take a maximum likelihood approach to these testing problems. If we say that our sequence of random variables x_i have densities $p_\theta(x_i)$, characterized by θ, we can find the likelihood ratio corresponding to the hypotheses H_0 and H_i, in this general case

$$\Lambda_1^n(i) = \frac{\Pi_{i=1}^k p_{\theta_1}(x_i) \Pi_{i=k+1}^n p_{\theta_n}(x_i)}{\Pi_{i=1}^n p_{\theta_1}(x_i)}$$

We will henceforth formulate the testing problem as a simple hypothesis

$$\theta_1 = ... = \theta_k \neq \theta_{k+1} = ... = \theta_n$$

but we will reflect the fact that there are really multiple alternative hypotheses by maximizing the test statistic over all possible changepoints. The decision rule is therefore based on the test statistic

$$\Lambda_n = \max_{1 \leq j \leq n} \Lambda_1^n(j)$$

so that

$$d = \begin{cases} 0 \text{ if } \ln \Lambda_n < h \\ 1 \text{ if } \ln \Lambda_n \geq h \end{cases}$$

where h is some threshold.

Of course, we can only apply this directly if we know the value of the parameters θ_1 and θ_n. In many practical settings some or all of these are unknown, in which case we must replace them with their maximum likelihood estimators. This means we have to maximize over the unknown parameter estimates as well, and the decision function is based on the statistic

$$\tilde{\Lambda}_n = \max_{1 \leq k \leq n} \sup_{\theta_1} \sup_{\theta_n} \Lambda_1^n(k, \theta_1, \theta_n)$$

It should be noted that testing for the existence of a changepoint does not intrinsically provide an estimate for the location of that changepoint. However, once we have decided to assume a changepoint exists, we can again use a maximum likelihood approach to arrive at an estimate for the change time k:

$$(\hat{k}, \hat{\theta_1}, \hat{\theta_n}) = \arg \max_{1 \leq k \leq n} \sup_{\theta_1} \sup_{\theta_n} \ln \left[\Pi_{i=1}^{k-1} p_{\theta_1}(x_i) \Pi_{i=k}^n p_{\theta_n}(x_i) \right]$$

In the following subsections we will look more specifically at using this approach for detecting changes in mean and variance in the normal case as well as discuss some alternatives and extensions to this basic maximum likelihood approach. We will then mention some other model settings and more complex changes.

3.1.1 The Univariate Normal Case

We assume a sequence of observations $x_1, x_2, ..., x_n$ where each $x_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$. First we use the previously explained maximum likelihood approach to detect changes in the mean or the variance, either separately or together.

We begin by looking at **changes to the mean**. Each $x_i \sim \mathcal{N}(\mu_i, \sigma^2)$, where here all x_i have the same variance and σ^2 can be seen as a nuisance parameter. To restate the general hypothesis testing problem for this specific question, we are testing the null hypothesis

$$H_0 : \mu_1 = \mu_2 = ... = \mu_n$$

versus

$$H_1 : \mu_1 = ... = \mu_k \neq \mu_{k+1} = ... = \mu_n$$

How we can perform this test depends on whether or not we assume the variance, σ^2, is known. If so, we implement the general maximum likelihood procedure as follows. We follow the procedure from Chen and Gupta (2012) in the book *Parametric Statistical Change Point Analysis*, which summarizes much of the background literature on retrospective changepoint analysis.

Without loss of generality, assume $\sigma^2 = 1$. Then, under H_0, the likelihood function is

$$L_0(\mu) = \frac{1}{(\sqrt{2\pi})^n} e^{-\sum_{i=1}^{n}(x_i-\mu)^2/2}$$

and the maximum likelihood estimator of μ is

$$\hat{\mu} = \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Under H_1, the likelihood function is

$$L_1(\mu_1, \mu_n) = \frac{1}{(\sqrt{2\pi})^n} e^{-\left[\sum_{i=1}^{k}(x_i-\mu_1)^2+\sum_{i=k+1}^{n}(x_i-\mu_n)^2\right]/2}$$

and the maximum likelihood estimators of μ_1 and μ_n are

$$\hat{\mu_1} = \bar{x}_k = \frac{1}{k}\sum_{i=1}^{k} x_i \text{ and } \hat{\mu}_n = \bar{x}_{n-k} = \frac{1}{n-k}\sum_{i=k+1}^{n} x_i$$

To arrive at an expression for the likelihood test statistic, let

$$S_k = \sum_{i=1}^{k}(x_i - \bar{x}_k)^2 + \sum_{i=k+1}^{n}(x_i - \bar{x}_{n-k})^2$$

and

$$V_k = k(\bar{x}_k - \bar{x})^2 + (n-k)(\bar{x}_{n-k} - \bar{x})^2$$

If we let

$$S = \sum_{i=1}^{n}(x_i - \bar{x})^2$$

then

$$V_k = S - S_k$$

Algebraic manipulation, as shown in detail in Lehman(1986), shows that

$$U^2 = V_{k^*} = \max_{1 \le k \le n-1} V_k$$

is the likelihood test statistic for our testing problem as stated.

Hawkins (1977) derives the exact and asymptotic null distribution of U. Here we simply state the result: The null probability density function of U is given by

$$f_U(x) = 2\phi(x, 0, 1) \sum_{k=1}^{n-1} g_k(x, x) g_{n-k}(x, x)$$

where $\phi(x, 0, 1)$ is the pdf of $\mathcal{N}(0, 1)$, $g_1(x, s) = 1$ for $x, s \ge 0$, and $g_k(x, s) = P\left[|T_i| < s, i = 1, ..., k-1 || T_k\right.$ for $x, s \ge 0$.

If we let

$$T_k = \sqrt{\frac{n}{k(n-k)} \left[\sum_{i=1}^{k} (x_i - \bar{x})\right]}$$

rearranging shows that $T_k^2 = V_k$ so $|T_k| = \sqrt{V_k}$. We can use $U = \sqrt{V_{k^*}} = \max_{1 \le k \le n-1} \sqrt{V_k} = \max_{1 \le k \le n-1} |T_k|$ as the test statistic now that we have a null distribution. However, to use the null distribution we would have to evaluate $g_k(x, s)$ which can be somewhat computationally expensive due to the recursive definition. Details on evaluating the recursion as well as on the derivation of the null distribution as a whole can be found in the literature or in Chen and Gupta (2012).

Deriving the distribution of U also leads to the distribution of the changepoint position k:

$$p\left[\hat{k} = k\right] = \int_0^\infty g_k(x, x) g_{n-k}(x, x) \phi(x, 0, 1) dx$$

For reasons of computational efficiency, for sufficiently large n, it may be preferable to use the asymptotic null distribution, as derived in Yao and Davis (1986), rather than the exact null distribution of U. Under the null hypothesis

$$\lim_{n \to \infty} P\left[a_n^{-1}(U - b_n) \le x\right] = exp\left\{-2\pi^{1/2} e^{-x}\right\}$$

for $-\infty < x < \infty$, where $a_n = (2 \log \log n)^{-1/2}$ and $b_n = a_n^{-1} + \frac{1}{2} a_n \log \log \log n$.

Next we look at the case where we **do not assume the variance is known**. Under H_0, the likelihood function is

$$L_0(\mu, \sigma^2) = \frac{1}{(\sqrt{2\pi}\sigma)^n} e^{-\sum_{i=1}^n (x_i - \mu)^2 / 2\sigma^2}$$

and the maximum likelihood estimators of μ and σ^2 are

$$\hat{\mu} = \bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \text{ and } \hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2$$

Under H_1, the likelihood function is

$$L_1(\mu_1, \mu_n, \sigma_1^2) = \frac{1}{(\sqrt{2\pi})^n} e^{-\sum_{i=1}^{k}(x_i-\mu_1)^2/2\sigma_1 - \sum_{i=k+1}^{n}(x_i-\mu_n)^2/2\sigma_1^2}$$

and the maximum likelihood estimators of μ_1, μ_n, and σ_1^2 are

$$\hat{\mu}_1 = \bar{x}_k = \frac{1}{k}\sum_{i=1}^{k} x_i, \hat{\mu}_n = \bar{x}_{n-k} = \frac{1}{n-k}\sum_{i=k+1}^{n} x_i$$

and

$$\hat{\sigma}_1^2 = \frac{1}{n}\left[\sum_{i=1}^{k}(x_i - \bar{x}_k)^2 + \sum_{i=k+1}^{n}(x_i - \bar{x}_{n-k})^2\right]$$

Let

$$S = \sum_{i=1}^{n}(x_i - \bar{x})^2 \text{ and } T_k^2 = \frac{k(n-k)}{n}(\bar{x}_k - \bar{x}_{n-k})^2$$

Then, the likelihood based test statistic is

$$V = \max_{1\le k\le n-1} \frac{|T_k|}{S}$$

The null distribution of V, as well as Bonferroni approximations, are derived in Worsley (1977).

Still within the univariate Gaussian model, we look at **testing for changes in variance while the mean remains constant**. For a sequence of independent normally distributed random variables $x_1, x_2, ..., x_n$ with associated parameters (μ, σ_i^2), we want to test

$$H_0 : \sigma_1^2 = \sigma_2^2 = ... = \sigma_n^2 = \sigma^2$$

versus

$$H_1 : \sigma_1^2 = ... = \sigma_k^2 \neq \sigma_{k+1}^2 = ... = \sigma_n^2$$

where $k, 1 < k < n$, is the position of the changepoint. Under H_0, the log likelihood function is

$$\log L_0(\sigma^2) = -\frac{n}{2}\log 2\pi - \frac{n}{2}\log\sigma^2 - \frac{\sum_{i=1}^{n}(x_i-\mu)^2}{2\sigma^2}$$

and the maximum likelihood estimator of σ^2 is

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n}(x_i-\mu)^2}{n}$$

The maximum likelihood is then

$$\log L_0(\hat{\sigma}^2) = -\frac{n}{2}\log 2\pi - \frac{n}{2}$$

Under H_1, the log likelihood function is

$$\log L_1(\sigma_1^2, \sigma_n^2) = -\frac{n}{2}log2\pi - \frac{k}{2}\log\sigma_1^2 - \frac{n-k}{2}\log\sigma_n^2 - \frac{\sum_{i=1}^{k}(x_i-\mu)^2}{2\sigma_1^2} - \frac{\sum_{i=k+1}^{n}(x_i-\mu)^2}{2\sigma_n^2}$$

The maximum likelihood estimators of σ_1^2 and σ_n^2 are

$$\hat{\sigma}_1^2 = \frac{\sum_{i=1}^{k}(x_i - \mu)^2}{k} \text{ and } \hat{\sigma}_n^2 = \frac{\sum_{i=k+1}^{n}(x_i - \mu)^2}{n-k}$$

Note that these maximum likelihood estimates require that $2 \leq k \leq n-2$. The maximum likelihood is

$$\log L_1(\hat{\sigma}_1^2, \hat{\sigma}_n^2) = -\frac{n}{2}\log 2\pi - \frac{k}{2}\log \hat{\sigma}_1^2 - \frac{n-k}{2}\log \hat{\sigma}_n^2 - \frac{n}{2}$$

The final likelihood ratio test statistic is

$$\lambda_n = \left\{ \max_{1<k<n-1} \left[n \log \hat{\sigma}^2 - k \log \hat{\sigma}_1^2 - (n-k)\log \hat{\sigma}_n^2 \right] \right\}^{1/2}$$

Then, under H_0, as $n \to \infty$ and $k \to \infty$ such that $(k/n) \to \infty$, for all $x \in \mathbb{R}$

$$\lim_{n\to\infty} P\left[a(\log n)\lambda_n - b(\log n) \leq x\right] = exp\{-2e^{-x}\}$$

where $a(\log n) = (2\log\log n)^{1/2}$ and $b(\log n) = 2\log\log n + \frac{1}{2}\log\log\log n - \log\Gamma(\frac{1}{2})$

If the mean is unknown, through the same maximum likelihood procedure, we have the maximum likelihood under H_0

$$\log L_0(\hat{\sigma}^2, \hat{\mu}) = -\frac{n}{2}\log 2\pi - \frac{n}{2}\log \hat{\sigma}^2 - \frac{n}{2}$$

where the maximum likelihood estimators of σ^2 and μ are

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n} \text{ and } \hat{\mu} = \bar{x}$$

Under H_1,

$$\log L_1(\mu, \sigma_1^2, \sigma_n^2) = -\frac{n}{2}\log 2\pi - \frac{k}{2}\log \sigma_1^2 - \frac{n-k}{2}\log \sigma_n^2 - \frac{\sum_{i=1}^{k}(x_i-\mu)^2}{2\sigma_1^2} - \frac{\sum_{i=k+1}^{n}(x_i-\mu)^2}{2\sigma_n^2}$$

with likelihood equations

$$\begin{cases} \sigma_n^2 \sum_{i=1}^{k}(x_i-\mu)^2 + \sigma_i^2 \sum_{i=k+1}^{n}(x_i-\mu)^2 = 0 \\ \sigma_1^2 = \frac{1}{k}\sum_{i=1}^{k}(x_i-\mu)^2 \\ \sigma_n^2 = \frac{1}{n-k}\sum_{i=k+1}^{n}(x_i-\mu)^2 \end{cases}$$

Solving for μ, σ_1^2,and σ_n^2 would yield the maximum likelihood estimators $\hat{\mu}, \hat{\sigma}_1^2$, and $\hat{\sigma}_n^2$, but this does not yield a closed form solution.We must therefore use an iterative method to obtain an approximate solution. Under regularity conditions we will get a unique solution and once again, as $n \to \infty$

$$\lim_{n\to\infty} P\left[a(\log n)\lambda_n - b(\log n) \leq x\right] = exp\left\{-2e^{-x}\right\}$$

for $x \in \mathbb{R}$, where $a(\log n)$ is defined as $(2\log\log n)^{1/2}$ and $b(\log n) = 2\log\log n + \log\log\log n$

3.1.2 Parametric Alternatives to Maximum Likelihood Method - Informational and Bayesian Approaches

Now that we have seen the maximum likelihood method in general and applied specifically to the univariate normal model, we can look at some alternative approaches.

One such alternative is the informational approach. We can view changepoint hypothesis testing as a kind of model selection problem and use some kind of information criterion, perhaps the Akaike Information Criterion in a first approach. If we use the AIC, we consider the alternative hypothesis of q changepoints as equivalent to a model where

$$\begin{aligned} &X_1, ..., X_{k_1} \sim \text{ iid } f(\theta_1), \\ &X_{k_1+1}, ..., X_{k_2} \sim \text{ iid } f(\theta_2), \\ &..., X_{k_{q-1}+1}, ..., X_{k_q} \sim \text{ iid } f(\theta_{q-1}), \\ &X_{k_q+1}, ..., X_{k_n} \sim \text{ iid } f(\theta_q) \end{aligned}$$

where $1 < k_1 < k_2 < ... < k_q < n$, q is the unknown number of changepoints, and $k_1, k_2, ..., k_q$ are the unknown changepoint positions.

We choose the model(k) that minimizes

$$AIC(k) = -2 \log L(\hat{\theta}_k) + 2k$$

for $k = 1, 2, ..., K$. $L(\hat{\theta}_k)$ is the maximum likelihood for model(k).

We denote the "full" model, that is the model with K free parameters as

$$Model(K) : \{f(\cdot|\theta) : \theta = (\theta_1, \theta_2, ..., \theta_k), \theta \in \Theta_K\}$$

where the parameter space Θ_k is restricted so that k parameters are free

$$\Theta_k = \{\theta \in \Theta_K | \theta_{k+1} = \theta_{k+2} = ... = \theta_K = 0\}$$

The model with the restricted parameter space Θ_k is denoted as $model(k)$.

One weakness of the Minimum AIC estimate is that it is not asymptotically consistent for model order. Other information criterions have been proposed to correct this and other weaknesses, for example the Schwarz Information Criterion

$$SIC(k) = -2 \log L(\hat{\theta}_k) + k \log n$$

for $k = 1, 2, ..., K$. The Minimum SIC is asymptotically consistent for model order. In practice, we perform the hypothesis test by accepting H_0 if

$$SIC(n) < \min_{2 \leq k \leq n-2} SIC(k)$$

and accept H_1 if

$$SIC(n) > SIC(k)$$

for some k. We can then estimate $\hat{k}$ so

$$SIC(\hat{k}) = \min_{2 \leq k \leq n-2} SIC(k)$$

This basic informational approach has the advantage of not requiring any knowledge about the distribution of the information criterion that we are using or of any test statistic.

However, to make more formal conclusions, we can also introduce a critical value $c_\alpha \geq 0$ which is based on a significance level α and accept H_0 if

$$SIC(n) < \min_{2 \leq k \leq n-2} SIC(k) + c_\alpha$$

where c_α is such that

$$1 - \alpha = P\left[SIC(n) < \min_{2 \leq k \leq n-2} SIC(k) + c_\alpha | H_0 \text{ is true}\right]$$

Critical value tables for the Gaussian and other common models are available for the SIC-based informational approach in Chen and Gupta (2012).

Corrections to make the SIC unbiased have also been proposed. Again, these have been derived for many common testing problems, including detecting the changes we have discussed within the Gaussian model.

Bayesian approaches to changepoint detection are also very popular. The basic idea is as follows: We let $K_{r,m}$ be the posterior odds of r changes versus m changes. We first check whether there is at least one changepoint by finding $K_{1,0}$. If $K_{1,0} > 1$, we conclude that there is at least one change. Then, if $K_{2,1} > 1$ we conclude that there are at least two changes. We stop this procedure when $K_{r+1,r} \leq 1$ and conclude that there are r changepoints. More information on finding an expression for $K_{\hat{r},m}$ can be found in Inclan (1993) or PSCPA. Once the number of changepoints $\hat{q}$ has been estimated, the locations of those changepoints are found via the posterior pdf of $\mathbf{k}$, the vector of all changepoint locations. Then, using the marginal distribution of each k_j, $j = 1, 2, ..., \hat{q}$, the estimated locations of the changepoints are given by the joint mode $(mode(k_1), mode(k_2), ..., mode(k_{\hat{q}}))$.

We have now seen a Bayesian approach, an informational approach, and a maximum likelihood ratio based approach to offline changepoint detection. We looked at the basic maximum likelihood approach in general and then looked at its application to the univariate normal model example. We will now discuss available results for other models. Most are based on one of the three methods already discussed (particularly the "standard" maximum likelihood procedure) and will therefore be presented in brief.

3.1.3 Multivariate Normal Model

We now have $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$, a sequence of independent m-dimensional normally distributed random vectors with parameters $(\mu_1, \mathbf{\Sigma}_1), (\mu_2, \mathbf{\Sigma}_2), ..., (\mu_n, \mathbf{\Sigma}_n)$. Analogously to the univariate case, we might be interested in testing for changes in mean or covariance, either individually or simultaneously. We will state the maximum likelihood ratio based test statistics and results about their null distribution for the multivariate versions of the testing problems already covered for the univariate normal model.

To test for a **change in the mean vector**, it is possible to use Hotelling's T^2 test. Let

$$\mathbf{y}_k = \sqrt{\frac{k(n-k)}{n}}(\bar{\mathbf{x}}_k - \bar{\mathbf{x}}_{n-k})$$

represent the standardized difference between two "samples": The parts of the sample before and after the changepoint. Let

$$W_k = \frac{1}{n-2}\left[\sum_{i=1}^{k}(\mathbf{x}_i - \bar{\mathbf{x}}_k)(\mathbf{x}_i - \bar{\mathbf{x}}_k)' + \sum_{i=k+1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}}_{n-k})(\mathbf{x}_i - \bar{\mathbf{x}}_{n-k})'\right]$$

The Hotelling's T^2 test statistic is

$$T_k^2 = \mathbf{y}_k' W_k^{-1} \mathbf{y}_k$$

and H_0 is rejected for

$$\max_{1 \le k \le n-1} T_k^2 > c$$

where c is chosen based on the null distribution of $\max_{1 \le k \le n-1} T_k^2$, for which there are many available approximations. Once a changepoint is assumed, its location $\hat{k}$ is estimated as the position that maximizes the test statistic.

Srivastava and Worsley (1986) propose an approximation of the null distribution of a function of $\max_{1 \le k \le n} T_k^2$. First we define

$$S_k = \mathbf{y}_k' V^{-1} \mathbf{y}_k \text{ with } V = \sum_{i=1}^{n} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})'$$

for $k = 1, ..., n-1$. Then

$$S_k = \frac{T_k^2}{n - 2 + T_k^2}$$

and, under H_0

$$P(S_{\hat{k}} > c) \lessapprox 1 - G_{m,\nu}(c) + q_1 \sum_{k=1}^{n-2} t_k - q_2 \sum_{k=1}^{n-2} t_k^3$$

where

$$\begin{aligned}
&\nu = \tfrac{n-m-1}{2} \\
&q_1 = g_{m,nu}\{2c(1-c)/\pi\}^{1/2} \Gamma\{(m+\nu-1)/2\}/\Gamma\{(m+\nu)/2\} \\
&q_2 = q_1\{(m^2-1)/c + (\nu^2-1)/(1-c) - (m+\nu)(m+\nu-1)\}/\{12(m+\nu)\}
\end{aligned}$$

$g_{m,\nu}(\cdot)$ is the pdf of beta$(m/2, \nu/2)$.

To test for a **change in covariance with mean assumed to be known**, the maximum likelihood based test statistic is

$$\lambda_n = \max_{m<k<n-m} \left(\log \frac{|\frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i'|^n}{|\frac{1}{k} \sum_{i=1}^{k} \mathbf{x}_i \mathbf{x}_i'|^k |\frac{1}{n-k} \sum_{i=k+1}^{n} \mathbf{x}_i \mathbf{x}_i'|^{n-k}} \right)^{1/2}$$

The asymptotic null distribution of λ_n, as presented in Chen and Gupta (2012), is as follows. Under H_0, as $n \to \infty$ and $k \to \infty$ such that $(k/n) \to 0$

$$\lim_{n \to \infty} Pa(\log n)\lambda_n - b_{2m}(\log n) \le x = exp\{-2e^{-x}\}$$

for all $x \in \mathbb{R}$, where $a(\log n) = (2 \log\log n)^{1/2}$ and $b_m(\log n) = 2\log\log n + (m/2)\log\log\log n - \log\Gamma(m/2)$.

If we wish to test for a **change in covariance when the mean is constant but unknown**, direct application of the maximum likelihood ratio procedure will not give closed form solutions for $\hat{\mu}, \hat{\mathbf{\Sigma}}_1, \hat{\mathbf{\Sigma}}_n$. Just as in the univariate case, we must use approximate solutions.

To test for a **simultaneous mean and covariance change**, the maximum log likelihood test statistic is

$$\lambda_n = \left(\max_{m<k<n-m} \log \frac{|\hat{\mathbf{\Sigma}}|^n}{|\hat{\mathbf{\Sigma}}_1|^k |\hat{\mathbf{\Sigma}}_n|^{n-k}} \right)^{1/2}.$$

Under H_0, as $n \to \infty$ and $k \to \infty$ such that $(k/n) \to 0$

$$\lim_{n\to\infty} P\{a(\log n)\lambda_n - b_{2m}(\log n) \leq x\} = exp\{-2e^{-x}\}$$

for $x \in \mathbb{R}$. $a(\log n) = (2 \log\log n)^{1/2}$ and $b_{2m}(\log n) = 2\log\log n + m\log\log\log n - \log\Gamma(m)$.

An informational approach to the multivariate normal model is also well covered in the literature, with expressions for the SIC under H_0 and H_1 readily available, along with approximate critical values. Derivations of corrected unbiased SICs are also readily available for the multivariate normal model.

3.1.4 Change Points in the Regression Model

Change point detection methods can also be used in regression analysis. Here, the standard basic approach is based on minimizing information criteria. We again use the notation and approach from Chen and Gupta (2012). The first approach to likelihood-ratio based test for a changepoint in linear regression was developed in Quandt (1958) and Quandt (1960) and then furthered in Kim (1994). To describe the general procedure, we look at simple linear regression, where $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$ are a sequence of observations that we assume follow the underlying model

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

for $i = 1, ..., n$, where x is nonrandom , β_0 and β_1 are unknown regression coefficients, and $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ and are uncorrelated for $i = 1, ..., n$. y_i is therefore a random variable and is $\mathcal{N}(\beta_0 + \beta_1 x_i, \sigma^2)$ distributed.If we want to test whether the data structure is not constant, we test the null hypothesis

$$H_0 : \mu_{y_i} = \beta_0 + \beta_1 x_i \text{ for } i = 1, ..., n$$

versus

$$H_1 : \mu_{y_i} = \beta_0' + \beta_1' x_i \text{ for } i = 1, ..., k \text{ and } \mu_{y_i} = \beta_0^* + \beta_1^* x_i \text{ for } i = k+1, ..., n$$

where $k = 2, ..., n-2$ is the location of the changepoint, and $\beta_0, \beta_1, \beta_0', \beta_1', \beta_0^*, \beta_1^*$ are unknown regression coefficients.

From the likelihood functions and maximum likelihood estimators of the unknown parameters, we can find the SIC under H_0 and H_1. Under H_0

$$SIC(n) = -2\log L_0(\hat{\beta}_0, \hat{\beta}_1, \hat{\sigma}^2) + 3\log n$$

and under H_1

$$SIC(k) = -2\log L_1(\hat{\beta}_0', \hat{\beta}_1', \hat{\beta}_0^*, \hat{\beta}_1^*, \hat{\sigma}^2)) + 5\log n$$

where $k = 2, ..., n-2$. Similar to the previously discussed information criterion minimization procedure, we accept H_0 if $SIC(n) < SIC(k)$ for all k. We select a regression model with a change at $\hat{k}$ if $SIC(\hat{k}) = \min_{1<k<n-1} SIC(k) < SIC(n)$.

It is also possible to adopt a Bayesian approach here. Generally, this means assuming that a changepoint exists. The unknown parameters are assigned a very general prior, for example the changepoint can be assumed to be uniformly distributed over all time points between 2 and $n-2$ and the variance between 0 and ∞. The distribution of the regression coefficients conditioned on the values of the changepoint location and the variance can be said to be proportional to some constant. Then we find the joint posterior density of all the parameters. Integrating over this with respect to the parameters gives us the posterior density of the changepoint location k. Then, we estimate a changepoint $\hat{k}$ at the point where the posterior distribution of k is maximized. The foundations of this Bayesian approach were developed in Ferreira (1975).

Expressions for the SICs and derivations of the posterior distribution of the changepoint location are available for both simple linear regression and multiple regression. For more on multiple regression, see Brown, Durbin, and Evans (1975).

3.1.5 Other Models

Many models beyond the Gaussian are well-studied in terms of the methods covered above, with existing relevant test statistics and corresponding (at least approximate) null distributions. Many are available in Chen and Gupta (2012). Parameter changes in the gamma and exponential models are particularly well-covered, as well as discrete models like the binomial and Poisson distributions to a lesser extent. Kander and Zacks (1966) and Hsu (1979) are fundamental references for the exponential and gamma models, respectively.

3.2 One Changepoint Versus Multiple Change Points

Methods of searching for the optimum number of changepoints fall into two classes: Exact and approximate methods. In general, finding the optimum number of changepoints means minimizing a cost function, perhaps the penalized negative log likelihood. Amongst the approximate methods is the binary segmentation algorithm, proposed by Vostrikova (1981). The basic idea is that we first test for a single changepoint versus no changepoint. If we decide in favor of a changepoint, we segment the data at the point that achieves minimal value of the cost function in the two resulting segments. We repeat this process on the two data subsequences until we have subsequences for which we decide against the presence of a changepoint. This is an popular algorithm for estimating the optimal number of changepoints.

A popular exact method is the "segment neighborhood" algorithm, proposed by Auger and Lawrence (1989). Given a maximum number of possible changepoints Q, this dynamic programming based algorithm searches the entire segmentation space to find the optimum partitioning of the data.

Both the binary segmentation and segment neighborhood algorithms can be computationally expensive, $\mathcal{O}(n \log n)$ and $\mathcal{O}(Qn^2)$ respectively. Obviously the binary segmentation method is relatively efficient but it only provides an approximate result. Killick, Fearnhead, and Eckley (2011)developed the Pruned Exact Linear Time, or PELT, algorithm. The PELT algorithm prunes the full segmentation space, removing segments that could not

possibly be minimizers of the cost function. Under mild conditions, the PELT algorithm achieves exact results in $\mathcal{O}(n)$ time.

Estimating the number of optimal number of changepoints is a minimization problem that we will not go into in detail. Binary segmentation is done by applying techniques for a single potential changepoint repeatedly. Exact searches of the segmentation space look for a segmentation that minimizes functions of the type we have discussed.

3.3 Online Change Point Detection

We now look at changepoint detection in the online setting. That is, we wish to detect a change in the structure of streaming data. Usually, the primary objective is to sound an alarm as soon as possible after a change while minimizing the incidence of false alarms. Estimating the actual location of the changepoint is built into some online changepoint detection algorithms, but it is generally not the main goal. To make the development of several basic algorithms coherent we follow the exposition and, partially, the notation in Basseville and Nikiforov (1993).

As in offline detection, the standard online changepoint detection methods are built on the log likelihood ratio. We have $x_1, x_2, ..., x_n$, a series of independent observations with associated probability densities $p_{\theta_1}(x), p_{\theta_2}(x), ..., p_{\theta_n}(x)$. We will denote the location of an unknown changepoint as k, where before the changepoint, $\theta_i = \theta_1$ and afterward, $\theta_i = \theta_n$. The basic idea is that the log likelihood ratio

$$s(x) = \ln \frac{p_{\theta_n}(x)}{p_{\theta_1}(x)}$$

has positive expectation under p_{θ_n} and negative expectation under p_{θ_1}:

$$\mathbf{E}_{\theta_n}(s) > 0 \text{ and } \mathbf{E}_{\theta_1}(s) < 0$$

Therefore, a change in the parameter θ implies a change in sign of the mean log likelihood ratio.

We will now summarize some of the classical approaches to online changepoint detection. Most of these were developed within the context of statistical quality control and may be referred to as quality control charts. They have been historically presented as a graph of the data with "control limits", upper and/or lower bounds on the data (or some statistic), outside of which the process is considered "out of control." (A process being out of control means that a change has occurred).

We begin with algorithms that are developed under the assumption that we know the parameters before and after the changepoint. We will later address the more realistic case of not knowing θ_n. We will also generally assume that θ_1 is known. In practice, we often obtain a preliminary estimate of this value from training data or else we use the maximum likelihood estimate of θ_1, maximized over θ_1, θ_n, and the changepoint location.

3.3.1 Basic Methods

We begin by taking samples of size N. For each sample, we test the hypotheses

$$H_0 : \theta = \theta_1$$

versus

$$H_1 : \theta = \theta_n$$

We continue taking samples until the first sample for which we reject H_0. Let

$$S_j^k = \sum_{i=j}^{k} s_i$$

where

$$s_i = \ln \frac{p_{\theta_n}(x_i)}{p_{\theta_1}(x_i)}$$

This statistic will be used throughout our discussion of online changepoint detection. The decision rule for each sample is then

$$d = \begin{cases} 0 \text{ if } S_1^N < h \\ 1 \text{ if } S_1^N \geq h \end{cases}$$

where h is a chosen threshold. To formalize when we will end the sampling procedure, we define a stopping rule

$$t_a = N \cdot min\{K : d_k = 1\}$$

where we call t_a the alarm time, and d_k is the decision rule for the sample number K (of size N). The alarm time is essentially the end of the first sample for which the decision indicates a parameter change.

The **Shewhart control chart** (Shewhart (1931)) is directly based on this method, and is a classic way for detecting changes in the mean of a process with constant variance. Its most basic derivation is for a one-sided change in mean, where the decision function is

$$S_1^N = \frac{b}{\sigma} \sum_{i=1}^{N} (x_i - \mu_1 - \frac{\nu}{2})$$

where

$$\nu = \mu_n - \mu_1$$

which is the magnitude of the change we hope to detect, and

$$b = \frac{\mu_n - \mu_1}{\sigma}$$

is the signal-to-noise ratio. Then

$$d = \begin{cases} 0 \text{ if } S_1^N(K) < h \\ 1 \text{ if } S_1^N(K) \geq h \end{cases}$$

where

$$S_1^N(K) = S_{N(K-1)+1}^{NK}$$

This means that the alarm will be triggered when

$$\bar{x}(K) \geq \mu_1 + \kappa \frac{\sigma}{\sqrt{N}}$$

where

$$\bar{x}(K) = \frac{1}{N} \sum_{i=N(K-1)+1}^{NK} (x_i)$$

and κ (along with the sample size N) is a tuning parameter.

To find a two-sided change in mean(where μ_n, the mean after change, is either $\mu_n^+ = \mu_1 + \nu$ or $\mu_n^- = \mu_1 - \nu$), an alarm is sounded for the first sample for which

$$|\bar{x}(K) - \mu_1| \geq \kappa \frac{\sigma}{\sqrt{N}}$$

The **geometric moving average algorithm** (Roberts (1959)) is a simple elaboration of the previous log-likelihood ratio procedure that allows for weighting recent observations more heavily than old observations. We define the decision function

$$g_k = \sum_{i=0}^{\infty} \gamma_i s_{k-i}$$

where γ_i are exponential weights

$$\gamma_i = \alpha(1-\alpha)^i$$

for $0 < \alpha \leq 1$. α controls how quickly past observations are underweighted. It can be convenient to express the decision function recursively. (It is often very helpful in terms of computational ease in an online setting if the decision function can be expressed recursively). Then

$$g_k = (1-\alpha)g_{k-1} + \alpha s_k \text{ with } g_0 = 0$$

The alarm time is

$$t_a = min\{k : g_k \geq h\}$$

We could also use a finite set of weights, in which case the method is referred to as the finite moving average algorithm.

The **filtered derivative algorithm** (see Basseville and Nikiforov (1993)) is another method for detecting changes in mean, under the assumption that a change in mean should correspond to a local high magnitude of the discrete derivatives of the observations. The data is first filtered for noise before taking a derivative to improve performance. Again define

$$g_k = \sum_{i=0}^{N-1} \gamma_i \ln \frac{p_{\theta_n}(x_{k-i})}{p_{\theta_1}(x_{k-i})}$$

where γ_i are any set of filtering weights. Then

$$\nabla g_k = g_k - g_{k-1}$$

can serve as the discrete derivative of g_k. We will sound an alarm if the derivative exceeds a threshold, but we can adjust sensitivity to noise by also keeping a count of how many times this derivative exceeds its threshold. Then the alarm time is

$$t_a = \min\{k : \sum_{i=0}^{N-1} \mathbf{1}_{\{\nabla_{g_{k-i}} \geq h\}} \geq \eta\}$$

3.3.2 CUSUM

The cumulative sum, or CUSUM, algorithm is also based on the log-likelihood ratio, but is a more standard method for modern usage. It was first developed in Page (1954). The CUSUM algorithm relies on the fact that a change in parameter corresponds to a change in sign in the expected value of the log-likelihood ratio, so it tends to become more negative before the change time and more positive afterward. So, it theoretically should reach a minimum at the change time. Of course, it is subject to random fluctuations. The CUSUM method tracks the difference between the log-likelihood ratio and its past minimum value. We therefore define the decision function

$$g_k = S_k - m_k$$

where

$$S_k = \sum_{i=1}^{k} s_i$$

and, as before

$$s_i = \ln \frac{p_{\theta_n}(x_i)}{p_{\theta_1}(x_i)}$$

and

$$m_k = \min_{1 \leq j \leq k} S_j$$

Then the alarm time is

$$t_a = \min\{k : g_k \geq h\}$$

We are comparing the log-likelihood ratio to an adaptive threshold in the sense that the alarm time can equivalently be expressed as

$$t_a = \min\{k : S_k \geq m_k + h\}$$

The CUSUM algorithm can also be derived from an offline perspective, with the same end result. We again test the hypothesis

$$H_0 = \theta_1 = \theta_2 = ... = \theta_n$$

versus

$$H_1 : \theta_1 = ... = \theta_k \neq \theta_{k+1} = ... = \theta_n$$

Then the likelihood ratio between H_0 and H_1 is

$$\Lambda_1^n(k) = \frac{\Pi_{i=1}^{k} p_{\theta_1}(x_i) \Pi_{i=k+1}^{n} p_{\theta_n}(x_i)}{\Pi_{i=1}^{n} p_{\theta_1}(x_i)}$$

Then the log-likelihood ratio is

$$S_j^k = \sum_{i=j}^{k} \ln \frac{p_{\theta_n}(x_i)}{p_{\theta_1}(x_i)}$$

Using the standard maximum likelihood approach means that we will base the testing decision on the maximum of this value with decision function

$$g_k = \max_{1 \leq j \leq k} S_j^k$$

and so

$$t_a = \min\{k : \max_{1 \le j \le k} S_j^k \ge h\}$$

The CUSUM algorithm, unlike the more basic online changepoint detection algorithms mentioned previously, provides an estimate for the change time:

$$\hat{k} = t_a - N_{t_a}$$

We now look at the CUSUM method for the specific case of detecting a change in the mean of a series of independent Gaussian random variables. We assume the variance σ^2 is known. We first assume that we only want to detect an increase in the mean, that is, $\mu_n > \mu_1$. Then

$$g_k = S_1^k - \min_{1 \le j \le k} S_1^j$$

where

$$S_1^j = \frac{\mu_n - \mu_1}{\sigma^2} \sum_{i=1}^{j} \left(x_i - \frac{\mu_n + \mu_1}{2} \right)$$

To do a two-sided test, where $\mu_n^+ = \mu_1 + \nu$ or $\mu_n^- = \mu_1 - \nu$, with ν assumed to be known, we can use two "simultaneous" CUSUM algorithms so that

$$t_a = \min\{k : (g_k^+ \ge \bar{h}) \cup (g_k^- \ge \bar{h})\}$$

where, in this (Gaussian) case,

$$g_k^+ = \left(g_{k-1}^+ + x_k - \mu_1 - \frac{\nu}{2} \right)^+$$

and

$$g_k^- = \left(g_{k-1}^- - x_k + \mu_1 - \frac{\nu}{2} \right)^+$$

If we do not know anything about the parameter after the changepoint, we can view ν as the magnitude of the smallest change we would like to detect. However, the algorithm is optimized to detect changes of exactly magnitude ν, so if we have prior knowledge of ν we should choose the most probable value. If we need to detect any change, we set $\nu = 0$. For $\nu = 0$, the two-sided CUSUM for mean changes in the normal case is characterized by

$$t_a = \min\{k : R_k \ge \bar{h}\}$$

where

$$R_k = \max_{j \le k} \sum_{i=1}^{j} (x_i - \mu_1) - \min_{j \le k} \sum_{i=1}^{j} (x_i - \mu_1)$$

3.3.3 Bayesian Approach

We can also take a Bayesian approach to online changepoint detection.Here, taking a Bayesian approach generally refers to incorporating prior knowledge about the distribution of the change time. The decision function is then based on the a posteriori probability of a change. It is often hard in practice to get specific information about the distribution of the changepoint. It is common to assume that the change time follows a geometric distribution, but finding the parameter of this distribution poses a preliminary estimation problem.

3.3.4 Unknown Parameter after Change - Weighted CUSUM and GLR

Now that we have discussed algorithms developed under the assumptions that we know the post-change parameter, we will address what can be done when we have no such knowledge. Two major techniques in this category are the weighted CUSUM and the generalized likelihood ratio algorithm.

The weighted CUSUM algorithm uses a weighting function, $dF(\theta_n)$, to weight the likelihood ratio for all possible values of θ_n. $F(\theta_n)$ is essentially the cumulative distribution function for θ_n. The decision function is based on the statistic

$$\tilde{\Lambda}_j^k = \int_{-\infty}^{\infty} \frac{p_{\theta_n}(x_j, ..., x_k)}{p_{\theta_1}(x_j, ..., x_k)}$$

Then,

$$t_a = \min\{k : \max_{1 \le j \le k} \ln \tilde{\Lambda}_j^k \ge h\}$$

Of course, we seldom have very specific information about $F(\theta_n)$. We may simply take θ_n to be uniformly distributed over some interval, but we can take other distributions, for example the Gaussian. If we have absolutely no information about the distribution of θ_n, we must use its maximum likelihood estimate. The generalized likelihood ratio algorithm does this. This means that, just like in the offline setting with unknown parameters, we have to maximize over both the change time and the possible values of θ_n, so that the decision is based on

$$\max_{1 \le j \le k} \ln \tilde{\Lambda}_j^k = \max_{1 \le j \le k} \sup_{\theta_n} S_j^k(\theta_n)$$

Lorden(1971) has more information on which densities the observations must have in order to perform this maximization.

3.4 Evaluating the Quality of Change Point Algorithms

Now that we have covered the basics of online and online changepoint detection methods, we will cover common ways to evaluate the quality of these methods.

3.4.1 Offline Methods

Offline changepoint detection is a hypothesis testing problem and methods can therefore be characterized by the classic measures of the size and power of a test

$$\alpha_0(g) = \mathbf{E}_0 \left[g(x_1, ..., x_N) \right]$$

and

$$\beta(g) = 1 - \alpha_1(g) = \mathbf{E}_1 \left[g(x_1, ..., x_N) \right],$$

the expectations of the decision function under H_0 and H_1.

There are many measures of optimality that have been developed for hypothesis testing in general that also apply here: Bayesian, minimax, etc. The testing problem involves

a nuisance parameter, the change time k. Sometimes change detection algorithms are evaluated via their minimum power

$$\beta_{min} = \min_{1 \leq k \leq N} \beta(k)$$

or the mean power

$$\bar{\beta} = \sum_{i=1}^{N} \gamma_i \beta(i)$$

rather than at the power for a specific change time.

We also need a way to characterize the estimated change time $\hat{k}$. It is possible to use classical properties of estimators to evaluate $\hat{k}$. Deshayes and Picard (1986) and Henkly(1970) discuss the consistency of $\hat{k}$. It is more popular to look at the distribution of $\hat{k} - k$

$$\mathbf{P}(\hat{k} = k \pm n)$$

for $n = 0, 1, 2, ...$ or

$$\mathbf{P}(|\hat{k} - k| \leq n)$$

for fixed n.

3.4.2 Online Methods

We now discuss ways to assess online change detection methods. We define the mean time between false alarms as

$$\bar{T} = \mathbf{E}_{\theta_1}(t_a)$$

Sometimes it is also useful to use the conditional mean detection delay to take into account the behavior of the process before the change time. The conditional mean delay for detection is

$$\mathbf{E}_{\theta_n}(t_a - k | t_a \geq k + 1; x_1, ..., x_k)$$

Other similar measures are the worst mean delay

$$\bar{\tau}^* = \sup_{k \geq 1} \operatorname{ess\,sup} \mathbf{E}_{\theta_n}(t_a - k | t_a \geq k + 1; x_1, ..., x_k)$$

(where ess sup refers to the essential supremum) or the mean delay

$$\bar{\tau} = \mathbf{E}_{\theta_n}(t_a)$$

The online "parallel" to the power of a test is the average run length, or ARL, defined as

$$L(\theta) = \mathbf{E}_{\theta}(t_a)$$

The ARL is a function of θ, where$L(\theta_1)$ is the mean time between false alarms and $L(\theta_n)$ is the mean delay for detection. The primary measure of optimality for online change detection is minimizing the mean delay for detection when the mean time between false alarms is fixed. A detection method can also be asymptotically optimal if it achieves this optimality measure asymptotically as $\bar{T} \to \infty$.

3.5 A Word on Nonparametric Approaches

Most classical changepoint analysis methods assume we know the distribution of the observations up to some parameter. Many assume that the pre- and post- change distributions belong to the same exponential family. Of course, the most common assumption is that the data are normally distributed.

One basic example of a nonparametric hypothesis testing method was proposed by Pettitt (1979) and is based on the Mann-Whitney two sample test. Let

$$D_{ij} = sgn(x_i - x_j)$$

where $sgn(x) = 1$ if $x > 0$, 0 if $x = 0$,-1 if $x < 0$. Let

$$U_{k,n} = \sum_{i=1}^{k} \sum_{j=k+1}^{n} D_{ij}$$

Then we can base a decision for or against the hypothesis of no change on the statistic

$$\lambda_n = \max_{1 \leq k \leq n} |U_{k,n}|$$

Many nonparametric online changepoint detection methods are based on ranks. One popular such method is the nonparametric CUSUM proposed in McDonald (1990). The basic idea is to calculate the rank R_{n+1} of observation x_{n+1} among observations $x_1, ..., x_n$. Then, let

$$U_{n+1} = \frac{R_{n+1}}{(n+1)}$$

Then, if

$$C_{n+1} = \max(0, C_n + U_{n+1} - k)$$

(where k serves as a tuning parameter) exceeds a threshold, an alarm is sounded.

The usual way to assess an online changepoint detection method's sensitivity to deviations from the assumed distribution is to compare the expected average run length under θ_1 for the assumed distribution to the same value under other distributions. If a method's pre-change run length distribution is the same for every continuous distribution of the data, it is considered to be fully nonparametric. In the case of online change detection with control charts, if the sample size is one (we are evaluating each observation individually), we can not rely on the central limit theorem to overcome deviations from normality as we might usually do. It is especially important to consider the robustness of the method in this case.

Borror, Montgomery, and Runger (1999) compare the performance of the Shewhart and EWMA charts in the case of skewed or heavy-tailed distributions. While the Shewhart control chart with small sample size is very sensitive to deviations from normality, the EWMA chart is shown to be quite robust, and is therefore a recommended approach for many situations with suspect non-normalities.

3.6 R Implementations

We now take a look at which methods for general changepoint detection are already implemented in R. We will then explore some of the capabilities of a major changepoint detection package (**changepoint**).

3.6.1 changepoint

The **changepoint** (Killick and Eckley (2012)) package is one of the broadest packages for basic changepoint detection. Its primary focus is on detecting changes in either mean or variance, or simultaneous mean and variance changes. To test for mean or variance changes, the package allows us to use the assumption that the data is normally distributed and therefore use a likelihood ratio based test statistic. We can also refrain from making any distributional assumptions about the data and use CUSUM test statistics. For changes in mean and variance, the package works for normal, exponential, and gamma distributional assumptions. The package allows us to choose from a range of penalizations: AIC, SIC, BIC, Asymptotic, Hannan-Quinn, and Manual. Manual penalties are often used when using the CUSUM test statistics under no distributional assumptions, because these statistics tend to be smaller than the likelihood ratio statistics and would fail to detect changepoints of penalized in the same way. The three search methods for multiple changepoints that were previously mentioned (binary segmentation, segment neighborhoods, and PELT) are implemented in this package. The package can therefore be used to test for the presence of a change and estimate the most probable changepoint location in the at-most-one-changepoint setting, as well as to estimate the optimum number of changepoints and their positions in the multiple changepoint setting. It can also return estimates for parameter values within the identified sequences between changepoints.

3.6.2 A Small Look at the changepoint Package

We now take a practical look at the **changepoint** package. We will attempt to detect a single change in mean in an independent series of normally distributed data. We will do this using the assumption of normality and then without any distributional assumptions. We will do this for several change magnitudes. (We will fix the variance at 1, so that the magnitude of the mean change is equivalent to the signal-to-noise ratio.) We will then look at the distribution of the difference between the estimated change time and the actual change time.

We first generate a series of normal observations with at most one changepoint. We create 1000 datasets, each with 500 observations. We will decide randomly whether each has a changepoint, with equal chance of having a changepoint or not. The mean before change will be 0 with post-change means of 0.5, 1, 2, 3, 4, and 10. To illustrate the overall function of the package, we will use the cpt.mean function to plot an example of one dataset for each change magnitude. These plots can be found in the first section of the appendix. This is the default plot type for the output of the basic cpt.mean and cpt.var functions.

We now use the cpt.mean function to find the most likely position of a changepoint. Using the test statistic for that point, we generate ROC curves for the decision of changepoint versus no changepoint. We do this for the decision based on assuming the data is normally

distributed and for the decision based on no distributional assumption. The first method uses test statistics based on the assumption of normality, as in Section 3.1.1. The second method uses CUSUM statistics that do not require a distributional assumption. They are built on a foundation of the ideas presented in Section 3.2.2. Plots of the two ROC curves (created using Robin, Turck, Hainard, Tiberti, Lisacek, Sanchez, and M§ller (2011)) for all change magnitudes can be found in the second section of the appendix. Below we plot the AUC for the decision based on both procedures for each change magnitude.

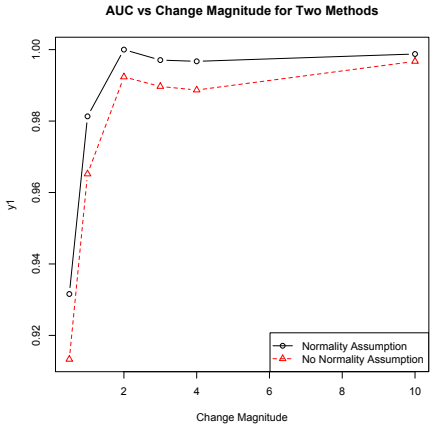

Figure 3.1: AUC for both methods versus change magnitude

We see that both decisions perform very well, but, as we might expect, using the knowledge of the distribution improves performance for smaller change magnitudes.

We will now look at the distribution of $\hat{k} - k$, again for several change magnitudes and for both distributional assumptions. We "force" a changepoint estimate by not penalizing for a changepoint, taking the most likely changepoint even if that likelihood is low. First we show histograms for $\hat{k} - k$ for each change magnitude. There are 1000 datasets for each change magnitude. Plots of the superimposed histograms for the two estimation methods are shown in the third section of the appendix.To characterize these distributions more succinctly, we show the mean squared error of the estimate $\hat{k}$ versus the change magnitude for both methods, with bars showing $\pm$ one standard error:

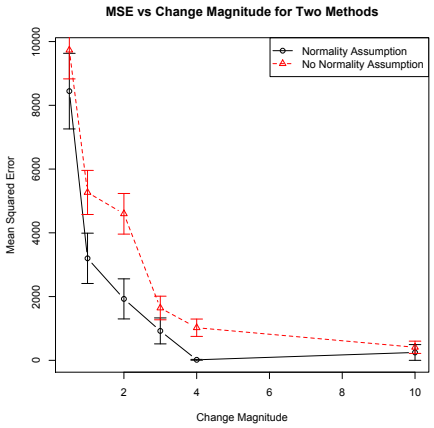

Figure 3.2: Mean Squared Error of change time estimate versus change magnitude

3.6.3 Other Possibilities in R

We will now mention some other more specialized or advanced R implementations of methods that we have discussed.

The **qcc** (Scrucca (2004)) package can calculate and plot quality control charts for continuous data, such as the Shewhart, CUSUM and Exponential Weighted Moving Average charts.

The **strucchange** (Zeileis, Leisch, Hornik, and Kleiber (2002)) package detects changepoints in linear regression models, estimates the location of structural changes, and provides an associated confidence interval.

The package **bcp** (Erdman and Emerson (2007)) implements a Bayesian approach for detecting changes in mean for normal data with a weakened independence assumption, with priors on the mean and change time as described in Barry and Hartigan (1993). The package provides the best partitioning of the data, the posterior means, and posterior probability of a changepoint at each data position.

The package **surveillance** (Hoehle (2007)) was primarily developed to detect outbreaks in health-related time series count data. However, it can be used more generally for non-continuous data. It implements many methods for count, binary, and categorical data, including a categorial version of the CUSUM for online change detection in categorical data.

Chapter 4

Summary

We have discussed many different approaches to anomaly detection and some of the fundamentals of changepoint analysis. We have described the problem setting necessary for the application of these methods in terms of data structure and the nature of the anomalies. One important area for ongoing research is the development of online anomaly detection algorithms. The importance of good online methods is increasing, especially in the fields of network security and financial fraud. Even for many existing well-defined algorithms, like incremental LOF, open source R implementations are lacking.

Bibliography

Auger, I. E. and C. E. Lawrence (1989). Algorithms for the optimal identification of segment neighborhoods. *Bulletin of Mathematical Biology 51*, 39–54.

Basseville, M. and I. V. Nikiforov (1993). *Detection of Abrupt Changes: Theory and Application*. Prentice Hall.

Bianco, A. M., M. Garcia Ben, E. J. Martinez, and V. J. Yohai (2001). Outlier detection in regression models with arima errors using robust estimates. *Journal of Forecasting 20*(8), 565–579.

Borror, C. M., D. C. Montgomery, and G. C. Runger (1999). Robustness of the ewma control chart to non-normality. *Journal of Quality Technology 31*(3).

Breunig, M., H.-P. Kriegel, R. T. Ng, and J. Sander (2000). Lof: Identifying density-based local outliers. In *PROCEEDINGS OF THE 2000 ACM SIGMOD INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA*, pp. 93–104. ACM.

Brown, R., J. Durbin, and J. Evans (1975). Techniques for testing the constancy of regression relationships over time (with discussion). *Journal of the Royal Statistical Society. Series B*, 149–192.

Chalmers, P. (2012). *faoutlier: Influential case detection methods for factor analysis and SEM*. R package version 0.2.2.

Chandola, V., A. Banerjee, and V. Kumar (2009). Anomaly detection: A survey. *ACM Comput. Surv. 41*(3), Article 15.

Chawla, N. V., K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research 16*, 321–357.

Chen, D., X. Shao, B. Hu, and Q. Su (2005). Simultaneous wavelength selection and outlier detection in multivariate regression of near-infrared spectra. *Analytical Sciences 21*(2), 161–167.

Chen, J. and A. K. Gupta (2012). *Parametric Statistical Change Point Analysis: With Applications to Genetics, Medicine, and Finance*. Birkhauser.

Cortez, P. (2011). *rminer: Simpler use of data mining methods (e.g. NN and SVM) in classification and regression*. R package version 1.1.

Erdman, C. and J. W. Emerson (2007). bcp: An R package for performing a bayesian analysis of change point problems. *Journal of Statistical Software 23*(3), 1–13.

Eskin, E. (2000). Anomaly detection over noisy data using learned probability distributions. In *In Proceedings of the International Conference on Machine Learning*, pp. 255–262. Morgan Kaufmann.

Eskin, E., A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo (2002). A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. In *Applications of Data Mining in Computer Security*. Kluwer.

Ferreira, P. (1975). A bayesian analysis of a switching regression model: Known number of regimes. *Journal of the American Statistical Association 70*, 370–374.

Filzmoser, P. and M. Gschwandtner (2012). *mvoutlier: Multivariate outlier detection based on robust methods*. R package version 1.9.7.

Fraley, C. and A. E. Raftery (2006). Mclust version 3 for r: Normal mixture modeling and model-based clustering. *Technical Report No. 504, Department of Statistics, University of Washington*.

Fu, Y., J. L. Zhou, and Y. Wu (2008, Dec). Online Reactive Anomaly Detection over Stream Data. In *2008 International Conference on Apperceiving Computing and Intelligence Analysis*, pp. 291–294. IEEE.

Grubbs, F. E. (1969, February). Procedures for detecting outlying observations in samples. *Technometrics 11*(1), 1–21.

Hahsler, M., B. Gruen, and K. Hornik (2005). a rules – a computational environment for mining association rules and frequent item sets. *Journal of Statistical Software 14/15*.

Hawkins, D. (1977). Testing a sequence of observations for a shift in location. *Journal of the American Statistical Association 72*(357).

Hawkins, S., H. He, G. Williams, and R. Baxter (2002). Outlier detection using replicator neural networks. In *In Proc. of the Fifth Int. Conf. and Data Warehousing and Knowledge Discovery (DaWaK02)*, pp. 170–180.

He, Z., X. Xu, and S. Deng (2003). Discovering cluster based local outliers. *Pattern Recognition Letters 2003*, 9–10.

Hoehle, M. (2007). surveillance: An R package for the surveillance of infectious diseases. *Computational Statistics 22*(4), 571–582.

Hsu, D. (1979). Detecting shifts of parameter in gamma sequences with applications of stock price and air traffic flow analysis. *Journal of the American Statistical Association 74*, 31–40.

Hu, Y., W. Murray, Y. S. with Strategic Data Mining Team, D. of Human Services, and Australian. (2011). *Rlof: R parallel implementation of Local Outlier Factor(LOF)*. R package version 1.0.0.

Joshi, M. V., R. C. Agarwal, and V. Kumar (2001). Mining needle in a haystack: classifying rare classes via two-phase rule induction. In *Proceedings of the 2001 ACM SIGMOD international conference on Management of data*, SIGMOD '01, New York, NY, USA, pp. 91–102. ACM.

Joshi, M. V. and V. Kumar (2004). Credos: Classification using ripple down structure

(a case for rare classes). In M. W. Berry, U. Dayal, C. Kamath, and D. B. Skillicorn (Eds.), *SDM*. SIAM.

Kander, Z. and S. Zacks (1966). Test procedures for possible changes in parameters of statistical distributions occurring at unknown time points. *Annals of Mathematical Statistics 37*, 1196 – 1210.

Killick, R. and I. A. Eckley (2012). *changepoint: An R package for changepoint analysis*. R package version 0.6.1.

Killick, R., P. Fearnhead, and I. Eckley (2011). Optimal detection of changepoints with a linear computational cost. *Submitted*.

Kim, D. (1994). Tests for a change-point in linear regression. *IMS Lecture Notes - Monograph Series 23*, 170–176.

Laurikkala, J., M. Juhola, and E. Kentala (2000). Informal identification of outliers in medical data. In *The Fifth Workshop on Intelligent Data Analysis in Medicine and Pharmacology*, pp. 20–24.

Maechler, M., P. Rousseeuw, A. Struyf, M. Hubert, and K. Hornik (2-02). *cluster: Cluster Analysis Basics and Extensions*. R package version 1.14.2 — For new features, see the 'Changelog' file (in the package source).

McDonald, D. (1990). A cusum procedure based on sequential ranks. *Naval Research Logistics (NRL) 37*(5), 627–646.

Page, E. S. (1954). Continuous inspection schemes. *Biometrika 41*, 100–115.

Papadimitriou, S., H. Kitagawa, P. B. Gibbons, and C. Faloutsos (2003). Loci: Fast outlier detection using the local correlation integral. In *ICDE'03*, pp. 315–315.

Parra, L., G. Deco, and S. Miesbach (1995). Statistical independence and novelty detection with information preserving nonlinear maps. *Neural Computation 8*, 260–269.

Pettitt, A. N. (1979). A non-parametric approach to the change-point problem. *Journal of the Royal Statistical Society. Series C (Applied Statistics) 28*(2), 126–135.

Pokrajac, D. (2007). Incremental local outlier detection for data streams. In *In Proceedings of IEEE Symposium on Computational Intelligence and Data Mining*, pp. 504–515.

Pokrajac, D., N. Reljin, N. Pejcic, and A. Lazarevic (2008). Incremental connectivity-based outlier factor algorithm. In *Proceedings of the 2008 international conference on Visions of Computer Science: BCS International Academic Conference*, VoCS'08, Swinton, UK, UK, pp. 211–223. British Computer Society.

Quandt, R. (1958). The estimation of the parameters of a linear regression system obeying two separate regimes. *Journal of the American Statistical Association 53*, 873–880.

Quandt, R. (1960). Tests of the hypothesis that a linear regression system obeys two separate regimes. *Journal of the American Statistical Association 55*, 324–330.

Roberts, S. W. (1959). Control charts based on geometric moving averages. *Technometrics 1*, 239–250.

Robin, X., N. Turck, A. Hainard, N. Tiberti, F. Lisacek, J.-C. Sanchez, and M. M§ller

(2011). proc: an open-source package for r and s+ to analyze and compare roc curves. *BMC Bioinformatics 12*, 77.

Scrucca, L. (2004). qcc: an r package for quality control charting and statistical process control. *R News 4/1*, 11–17.

Shewhart, W. A. (1931). *Economic Control of Quality Manufactured Product*. D. Van Nostrand Reinhold.

Shyu, M., S. Chen, K. Sarinnapakorn, and L. Chang (2003). A novel anomaly detection scheme based on principal component classifier. In *in Proceedings of the IEEE Foundations and New Directions of Data Mining Workshop, in conjunction with the Third IEEE International Conference on Data Mining (ICDMÕ03*, pp. 172–179.

Srivastava, M. S. and K. J. Worsley (1986). Likelihood ratio tests for a change in the multivariate normal mean. *Journal of the American Statistical Association 81*(393).

Steinwart, I., D. Hush, and C. Scovel (2005). A classification framework for anomaly detection. *J. Machine Learning Research 6*, 211–232.

Tamayo, P. and A. Mozes (2011). *RODM: R interface to Oracle Data Mining*. R package version 1.1.

Tang, J., Z. Chen, A. Fu, and D. Cheung (2002). Enhancing effectiveness of outlier detections for low density patterns. In M.-S. Chen, P. Yu, and B. Liu (Eds.), *Advances in Knowledge Discovery and Data Mining*, Volume 2336 of *Lecture Notes in Computer Science*, pp. 535–548. Springer Berlin / Heidelberg.

Torgo, L. (2010). *Data Mining with R, learning with case studies*. Chapman and Hall/CRC.

Tsay, R. S., D. Pena, and A. E. Pankratz (2000). Outliers in multivariate time series. *Biometrika 87*(4), 789–804.

Vostrikova, L. J. (1981). Detecting "disorder" in multidimesional random processes. *Soviet Mathematics Doklady 24*, 55–59.

Worsley, K. J. (1977). On the likelihood ratio test for a shift in location of normal populations. *Journal of the American Statistical Association 74*(366).

Yamanishi, K., J. Takeuchi, and G. Williams (2000). On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. In *Proceedings of the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 320–324. ACM Press.

Yu, D., G. Sheikholeslami, and A. Zhang (1999). Findout: Finding outliers in very large datasets. Technical report, Department of Computer Science and Engineering State University of New York at Buffalo Buffalo.

Zeileis, A., F. Leisch, K. Hornik, and C. Kleiber (2002). strucchange: An r package for testing for structural change in linear regression models. *Journal of Statistical Software 7*(2), 1–38.

Zhang, J. and H. Wang (2006, October). Detecting outlying subspaces for high-dimensional data: The new task, algorithms, and performance. *Knowl. Inf. Syst. 10*(3), 333–355.

Appendix A

Appendix

A.1 Default Mean Changepoint Plots

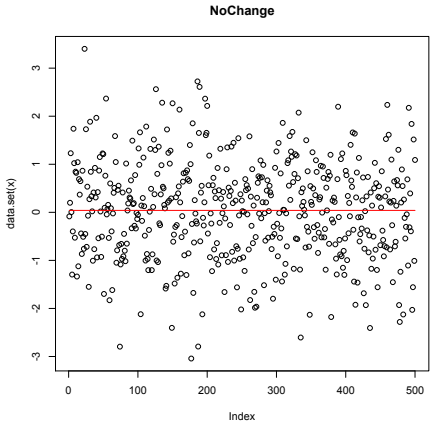

Figure A.1: Data with no change

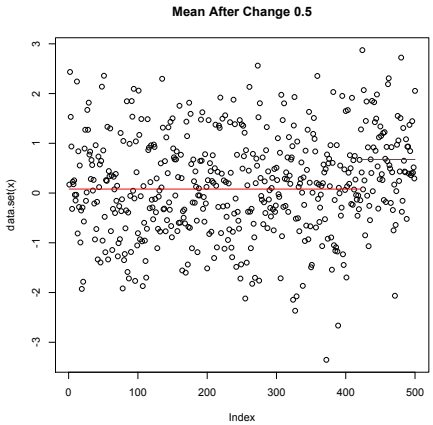

Figure A.2: Data with mean 0.5 after change

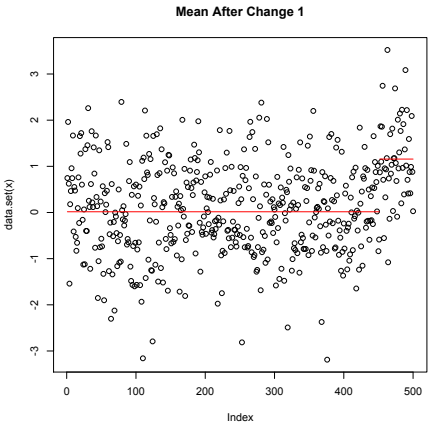

Figure A.3: Data with mean 1 after change

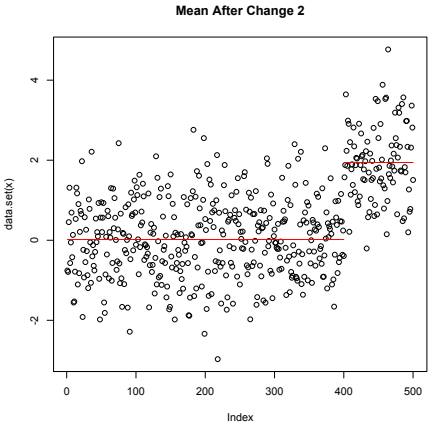

Figure A.4: Data with mean 2 after change

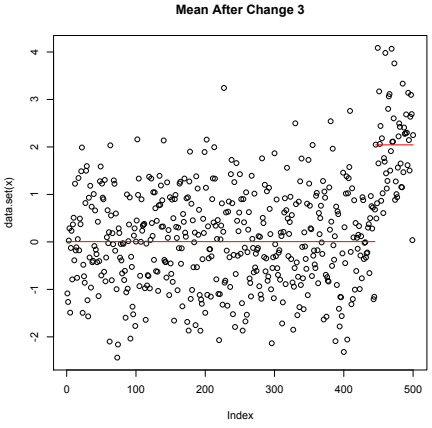

Figure A.5: Data with mean 3 after change

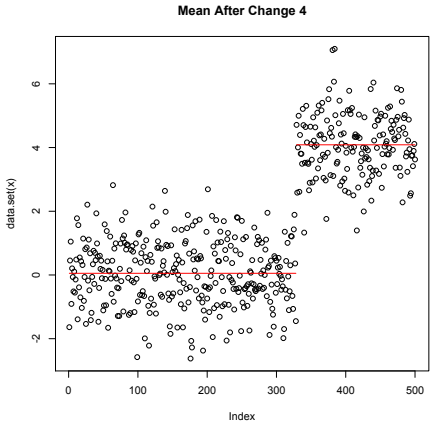

Figure A.6: Data with mean 4 after change

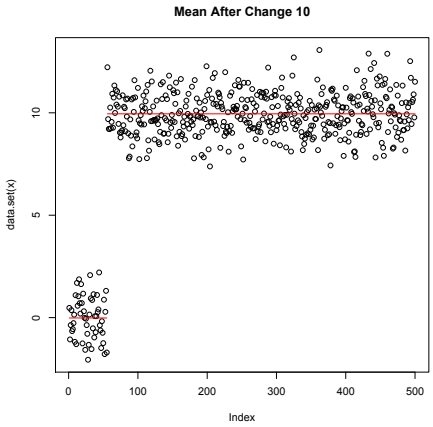

Figure A.7: Data with mean 10 after change

A.2 ROC Curves for Two Methods in changepoint Package

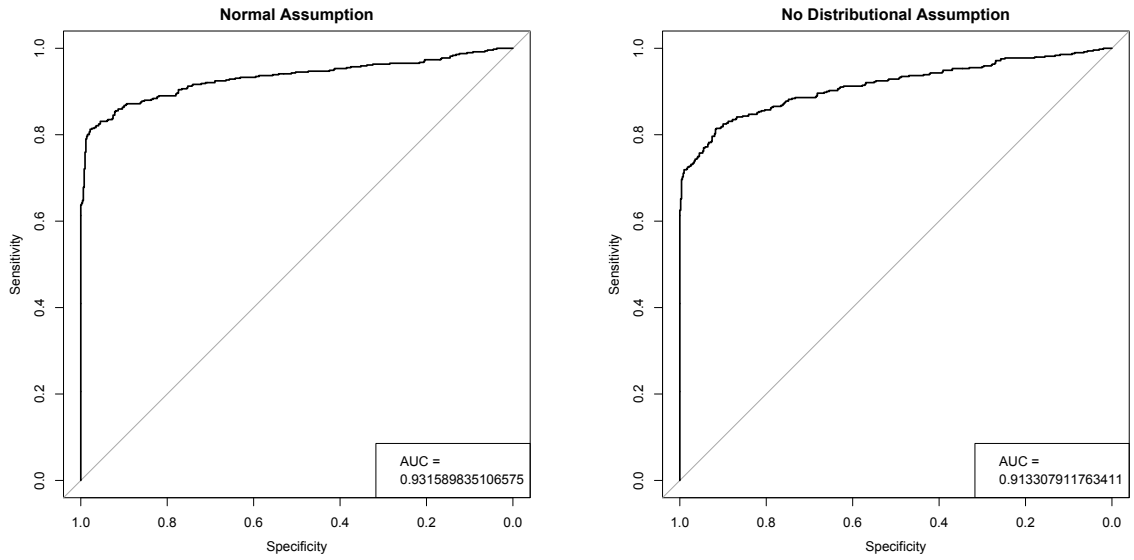

Figure A.8: Mean after change 0.5

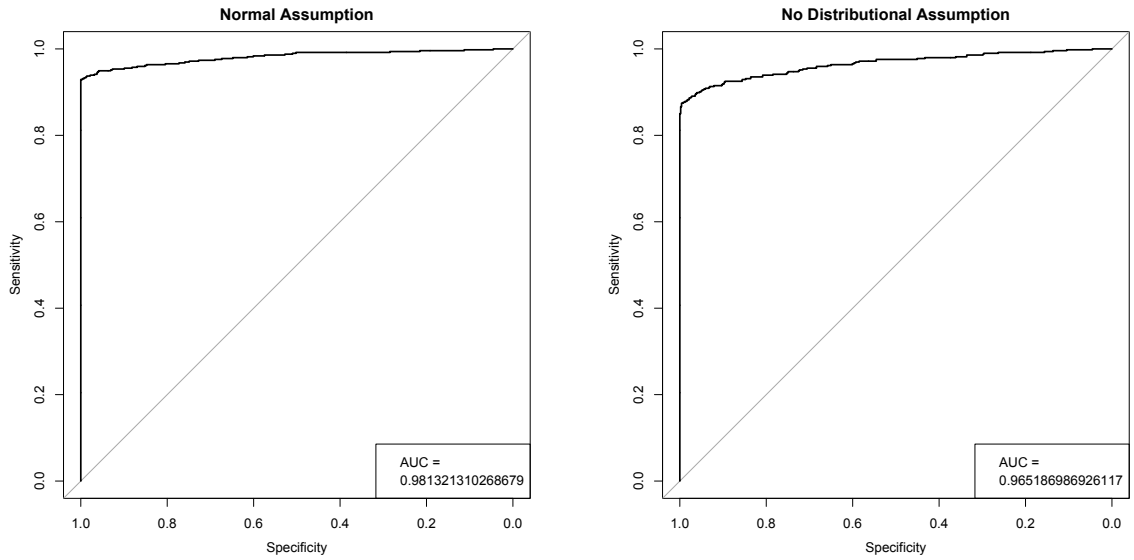

Figure A.9: Mean after change 1

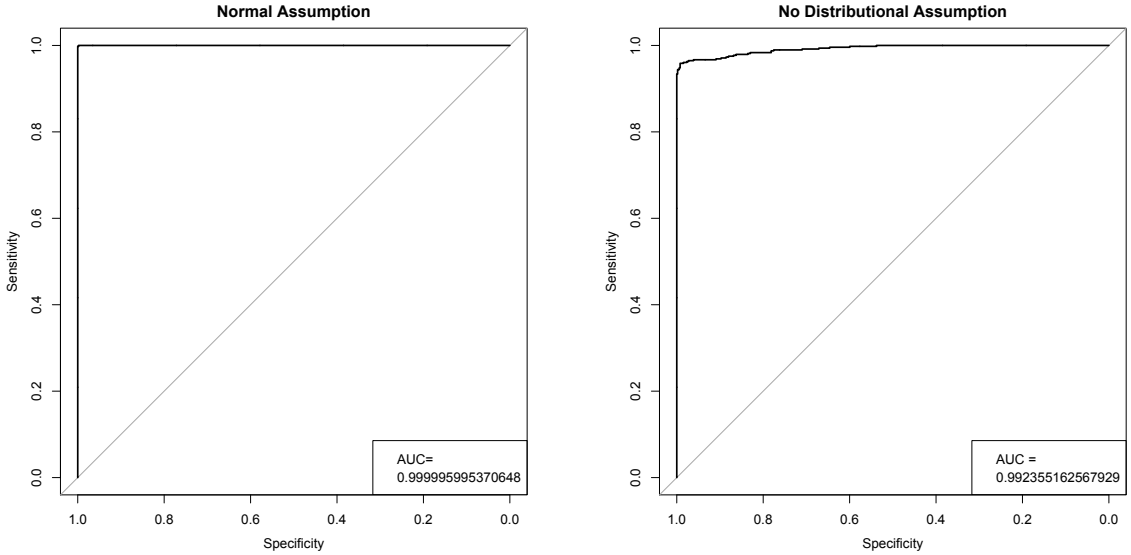

Figure A.10: Mean after change 2

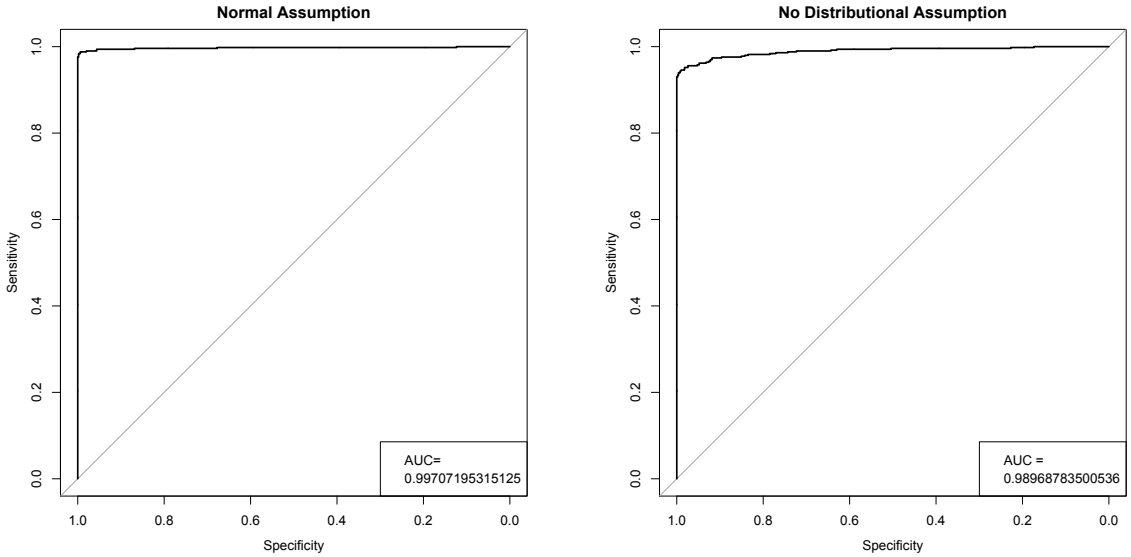

Figure A.11: Mean after change 3

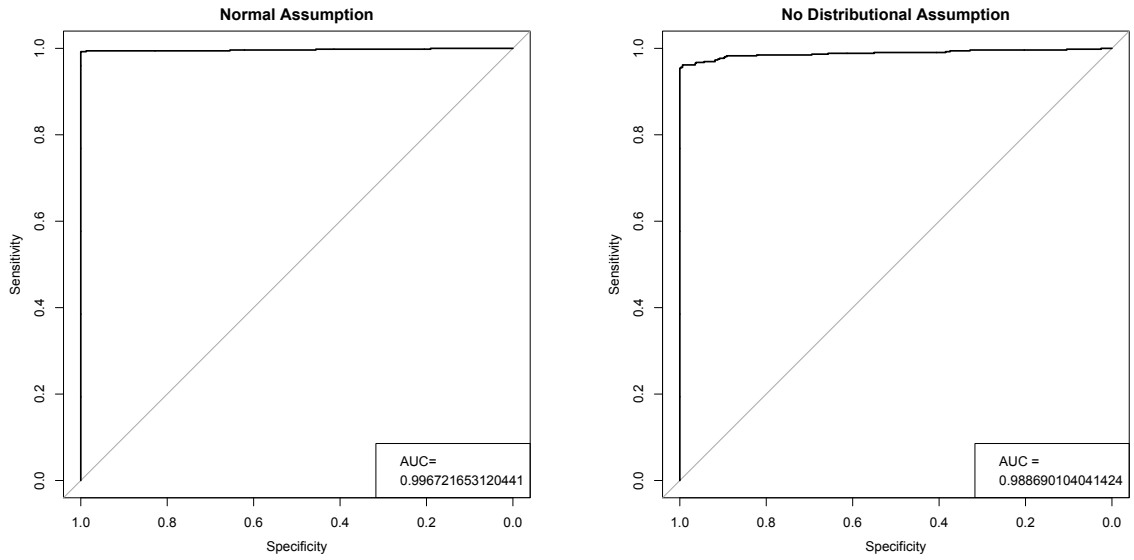

Figure A.12: Mean after change 4

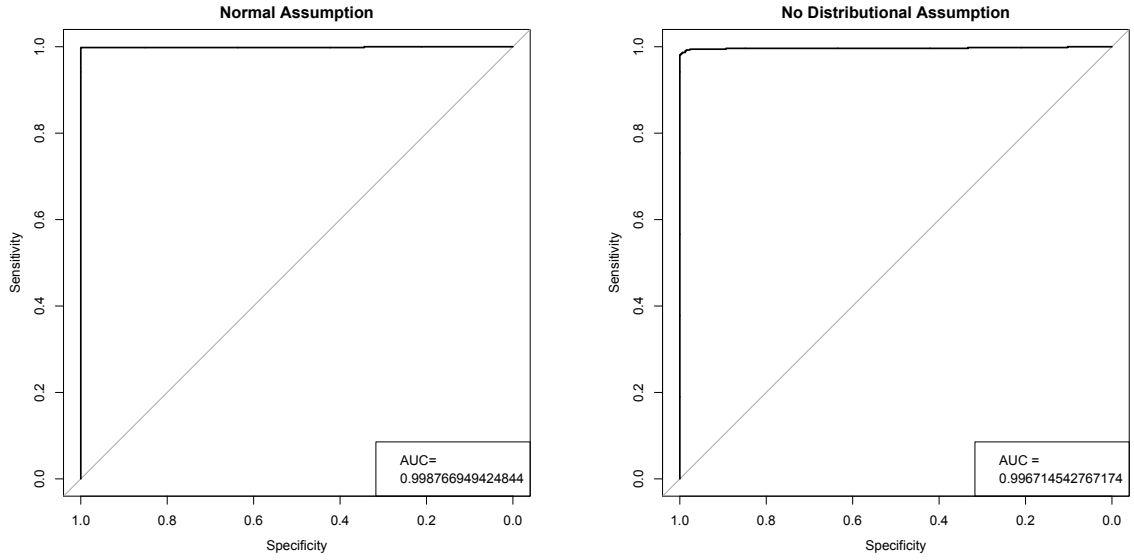

Figure A.13: Mean after change 10

A.3 Histograms of the Difference Between Estimated and Actual Change Time

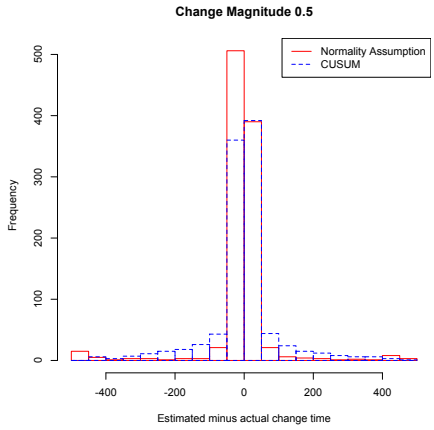

Figure A.14: Histogram of difference between estimated and actual change time: Mean after change 0.5

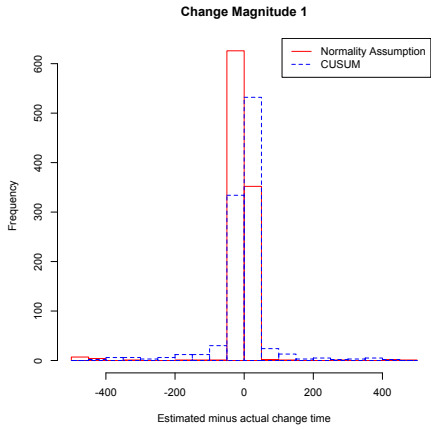

Figure A.15: Histogram of difference between estimated and actual change time: Mean after change 1

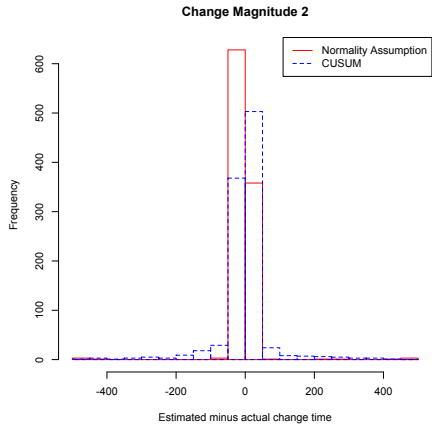

Figure A.16: Histogram of difference between estimated and actual change time: Mean after change 2

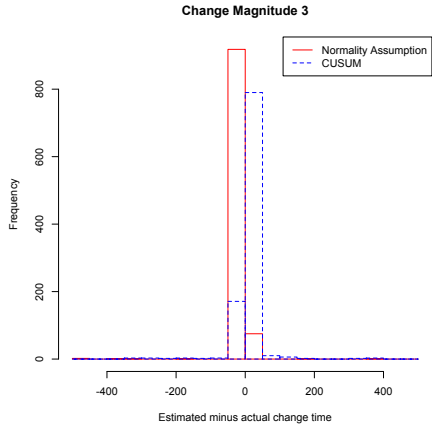

Figure A.17: Histogram of difference between estimated and actual change time: Mean after change 3

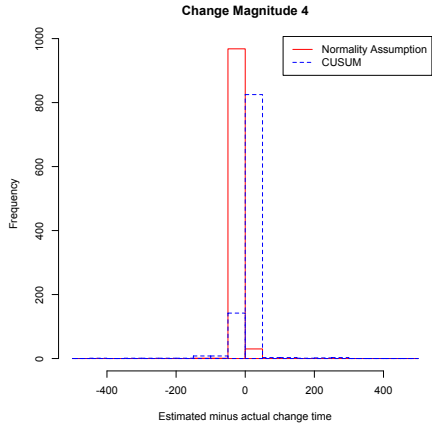

Figure A.18: Histogram of difference between estimated and actual change time: Mean after change 4

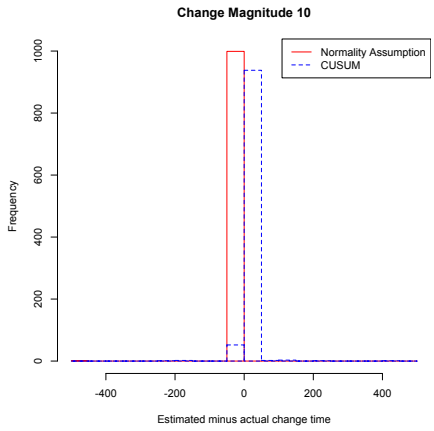

Figure A.19: Histogram of difference between estimated and actual change time: Mean after change 10