NEW APPROACHES FOR OUTLIER DETECTION

BY

CHRISTOPHER ERIC ZWILLING

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Psychology
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2016

Urbana, Illinois

Doctoral Committee:

Associate Professor Michelle Y. Wang, Chair and Director of Research
Professor Carolyn Anderson
Assistant Professor Hans-Friedrich Köhn
Professor John Marden
Professor Fritz Drasgow

ABSTRACT

Outlier detection has relevance in many modern day contexts, including health care, engineering, data processing and analysis, credit card fraud, monitoring computer and internet intrusions and wearable personal health sensors. Outlier detection once represented a single pre-processing step, completed prior to the analysis of data proper. Today it has importance in all stages of the data analysis pipeline, from initial processing to defining data points of interest, such as when a sensor detects an anomaly. Moreover, as data sets have grown to encompass millions and billions of observations and variables, it is imperative to have outlier detection methods capable of effectively and automatically winnowing through large amounts of data with few or no inputs from a data analyst. Many existing outlier detection methods are constrained in certain ways which might limit their utility and efficacy. For instance, it is not uncommon for outlier detection methods to require some knowledge about the data under study or require the analyst to specify information about the number of outliers in the data. Another possible constraint of many outlier detection methods is the use of the raw data. Sometimes outliers can readily be detected in the raw data; but sometimes not, in which case one can achieve greater sensitivity and accuracy from features derived from data. This study uses feature extraction on multivariate time series data and demonstrates the efficacy of a set of features and their potential for aggregation through the use of Voronoi diagrams. Voronoi diagrams are constructed from the data to create tessellations which satisfy certain geometric properties. The covariance based outlier detection is proposed and demonstrated to addresses both of these challenges. It utilizes covariance information in the data and its efficacy lies in its ability to take a set of features constructed from the data and determine which feature is best at detecting outliers. The method is shown to work effectively on time series data; but it is general and can be applied or extended to other types of data objects and data sets.

ACKNOWLEDGMENTS

Completing this thesis would not have been possible without assistance and support from many individuals. I would like to express gratitude to my thesis advisor, Dr. Michelle Wang, for the continuous support of my PhD during the various phases of research, writing and revisions. Her knowledge and guidance were instrumental in helping me to complete my thesis. Just as importantly, the skills and knowledge I have gained working with her will be invaluable in my future research.

In addition to my advisor, I also want to thank the rest of my thesis committee for their insightful comments and suggestions to broaden my thinking on my thesis research: Dr. Carolyn Anderson, Dr. Fritz Drasgow, Dr. Hans-Friedrich Köhn and Dr. John Marden. Drs. Anderson and Marden also provided helpful feedback on drafts of my thesis manuscript.

Last but certainly not least, my wife and family have supported me unconditionally during the dissertation process. Thank you.

Table of Contents

1. Introduction

Chapter 1 provides an introduction to outliers. Section 1.1 introduces outliers by way of some prominent outlier definitions and then proceeding to consider different lenses through which one might view outliers, including statistical and machine learning approaches. Section 1.2 provides a more formal introduction to understanding outliers, including some standard outlier models, important terminology and a case study.

1.1 Outlier definitions

Outliers seem easy to understand. But they are challenging to define rigorously. For instance:

a) …an outlier being an observation which is suspected to be partially or wholly irrelevant because it is not generated by the stochastic model assumed (Box & Tiao, 1968).

b) An outlying observation, or outlier, is one that appears to deviate markedly from other members of the sample in which it occurs (Grubbs, 1969).

c) …any observation that has not been generated by the mechanism that generated the majority of observations in the data set (Freeman, 1980).

d) An observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data (Barnett & Lewis, 1994).

e) …surprising veridical data, a point belonging to class A but actually situated inside class B so the true (veridical) classification of the point is surprising to the observer (John, 1995).

f) …noise points lying outside a set of defined clusters which behave differently from the norm (Aggarwal & Yu, 2001).

Regardless of the definition, outlier detection in data analysis has steadily grown from traditionally serving as the first step in a larger data processing workflow—removing outliers before fitting a model—to an end goal in its own right with enormous economic, personal and national security consequences. Consider the economic costs of detecting fraudulent credit card usage to determine anomalous purchases, for instance, or the national security implications in identifying hackers attempting to access a computer server. Outlier detection plays a key role in many fields, and is growing in importance in many others. These include loan application processing, intrusion detection, activity monitoring network performance, fault diagnosis, structural defect detection in manufacturing, satellite image analysis, novelties in images, motion segmentation (especially moving features that are independent of the background), time series monitoring, medical condition monitoring, pharmaceutical research, identification of mischievous responders in survey research, text or language novelty, detecting database anomalies, mislabeled data in training data sets and many others (Hodge & Austin, 2004). Outliers can also arise from errors caused by humans, incorrect recording of data, instrument or sensors errors, natural variation, fraud and changes or faults in systems.

Detecting outliers represents a complex interplay among several factors:

a) How critical is the response time once an outlier has been identified? The consequences of identifying and finding a recording error from a consumer insights survey are much lower than anomalous readings for a sensor measuring vital indicators of a patient in a hospital.
b) How accurate does the outlier detection algorithm need to be? Are false positives less desired or do false negative carry less weight? The consequences of being right in a life or death medical situation might require different outlier detection methods than a situation where the sensitivity or specificity is less critical.
c) What financial costs are incurred if the outlier is not detected and dealt with properly and quickly? Failing to flag fraudulent credit card charges can lead to hundreds or thousands of dollars for an individual and can cost credit card companies billions of dollars over the course of a year.
d) How complex is the outlier detection algorithm and how quickly can it yield results? An algorithm may have 100% accuracy detecting outliers but take months to produce the result.

Each of these factors trade-off and interact. In a financial setting, one can respond slower; but the need for accuracy is higher because of the financial cost of not detecting fraud. But in a health care environment, immediate detection may be required and that might mean an algorithm which is fast and perhaps yields a lower hit rate than an algorithm which takes weeks to yield a solution but is perfectly accurate.

Classical approaches to outlier detection come from statistics; but more recently advances in computer science, machine learning and neural networks have made important contributions to key problems in outlier detection. Statistical approaches, which have parametric, non-parametric or semi-parametric forms, leverage distributional assumptions and asymptotic properties to identify outliers. Statisticians have also developed robust statistics, methods that yield reliable and accurate results in the presence of outliers. If one has a good understanding of the distribution underlying their data, statistical approaches are excellent; but the classical approaches have shortcomings when this distributional information is unknown. In these cases, machine learning approaches use information in the data to identify outliers. But even these machine learning approaches to outlier detection have shortcomings because they usually require the user to specify values for key input parameters, parameters which usually depend on the structure of the data. Additionally, a certain amount of data is required for some of these approaches to work effectively.

Hodge and Austin (2004) conceptualize three approaches for identifying outliers from a machine learning perspective. The first approach finds outliers in the data but one does not know anything about the structure of the data, similar to an unsupervised learning framework. Outlying points are more distant than the normal data and the algorithm identifies these points. This way is necessarily static because it requires the full dataset. The second approach is more akin to a supervised learning approach, requiring *a priori* knowledge of outliers. Classifiers are good examples of this method. In the third approach a classifier learns what is normal (from pre-labeled data) and then will label anything not normal as an outlier.

The foregoing distinctions for outlier detection across disciplinary contributions are somewhat conceptual, as hybrid approaches that adopt strengths from these disciplines are gaining traction. For instance, statistical and machine learning communities make different assumptions regarding models and have operated mostly independently of each other (Breiman, 2001). But in recent years, these communities have started to hybridize, leading to new insights and overcoming problems that neither perspective alone could address. In a similar way, the outlier detection methodology that follows is best construed as a hybrid between statistics and machine learning.

1.2 An introduction to outliers

Anomalous data can be classified as either an outlier or inlier. The definition of outlier commonly used by researchers (Barnett & Lewis, 1994) states, "An outlier is a data value x_i in a dataset $\boldsymbol{\mathcal{D}}$ that is inconsistent with the nominal behavior exhibited by most of the data values in $\boldsymbol{\mathcal{D}}$." The underlying assumption to this definition is that most data points are homogeneous, in the sense of being generated from the same underlying process of interest to the observer. In statistical parlance, this could mean a particular statistical distribution, such as the normal distribution. When all observed data is generated from the same underlying process, outliers will typically arise from error measurements or due to chance. But some of the observed data can be generated by a mechanism fundamentally different than that under consideration, such as a different statistical distribution. Typically, the fraction of outliers is small (i.e. less than 1%) though it can be as large as 20%.

Inliers are observations that fall within the range of nominal behavior of the entire data set but are not part of the data generating mechanism or are an error. The error may result, for instance, from duplicate records, disguised missing data or file merge errors. Or they could even be generated from another statistical distribution. Detecting outliers is difficult enough. Inliers are even tougher to diagnose. One common manifestation of inliers is the frequent recurrence of a single data value in a dataset. For instance, missing data might be coded as a 0 but, when a statistical analysis is performed, these 0-coded missing data points are included as part of the real data, thereby yielding misleading results.

When anomalous data is suspected, a researcher needs to first detect these anomalies. Once detected, one might opt to remove, replace or set aside and analyze those data points separately. Alternatively, one could use robust statistical methods, which are more resistant to outliers. Irrespective of the strategy adopted, one must detect the presence of an outlier first.

To better appreciate the importance of outlier detection, consider the influence of a single outlier in a dataset on the first 4 statistical moments—the mean, standard deviation, skew and kurtosis.

Estimators of these moments are incredibly sensitive to a *single* outlier, let alone multiple anomalous points. To illustrate this suppose we generate 4 data sets from a normal distribution with $\mu = 0$ and $\sigma = 1$. Two of the data sets have 1,024 observations (large sample) and two have 128 (small sample). One data set each from the larger and smaller sample contained no outliers while the other larger and smaller sample had a single outlier, of magnitude 8, added to one observation. The estimates of the first four moments of these four data sets are shown in Table 1.1.

Table 1.1 reveals at least two important facts. First, sample size has an important influence on the measures of moments. Moments computed from larger sample sizes are less sensitive to outliers; but bear in mind the relativity implicit with this simple example in Table 1.1. If one had many outliers even in a large dataset, or a few outliers with large magnitudes, these could bias even large samples. Second, the influence of the single outlier across the four moments is not equally distributed. The mean and standard deviation of the small sample size with a single outlier are influenced much less than the skew and kurtosis.

Masking is “the failure of an outlier detection rule to detect outliers in the presence of outliers themselves” (Pearson, 2011). Figure 1.1, which has been reproduced from Pearson, illustrates masking with flow rate data from a physical system with a lower bound of 0. The outlier detection rule implemented for the data in this figure is a simple but commonly used one: flagging any observation beyond three standard deviations from the mean as an outlier. This is sometimes called the 3 sigma rule. Figure 1.1 has 3 horizontal lines with intercepts at -150, 315 and 780. The horizontal line with an intercept of 780 is three standard deviations above the mean and the horizontal line at -150 is three standard deviations below the mean. The mean is the horizontal line with a y-intercept of 315. The true outliers of the system are roughly between 0 and 200 (which reflect system shut down processes) while normal system functioning is represented by all points above the mean (y=315). Because the discrepancy between the true and outlying data points is so extreme, the mean is pulled in the direction of the anomalous values, yielding a standard deviation of greater than 150 and thus creating a very wide band of allowable values under the outlier detection rule of plus or minus three standard deviations from the mean. None of the outliers are correctly identified by this method. This example illustrates the masking effect because the simple (and commonly used) method failed to detect the outliers.

The swamping effect is the opposite of masking: outliers present in the data cause data points that are *not* outliers to be misclassified as outliers. Assume we have four sets of data with 100 observations each from a Gaussian normal. Outliers of magnitude 4, 8, 16 or 32 replace observations in the simulated data. In the first data set, one outlier of magnitude 4 replaced one observation. In the second data set, one outlier of magnitude 8 replaced one observation. In the third data set, one outlier of magnitude 16 replaced one observation. And in the fourth data set, one outlier of magnitude 32 replaced one observation. Then this process was repeated on four new sets of 100 observations, except this time 2 outliers at each of the four magnitudes (4, 8, 16 and 32) replaced original observations. Then the process was repeated for 4, 8, 9 and 10 outliers. The number of outliers is called the contamination level, often expressed as a percentage. For instance: when 10 outliers are present in a data set of 100 observations, the contamination level is 10%.

How does the swamping effect manifest in this example? Consider the 4 data sets with a 1% contamination level (a single outlier). When the magnitude of the outlier is 4, and assuming the simple 3-sigma rule, 3 observations are identified as outliers. So even though only 1 observation is truly an outlier, two more observations were flagged as outliers but were not actually outliers. Still considering the 1% contamination level, only when the magnitude of the outlier is 16 or 32 does the swamping effect go away. With 2 or 4 outliers present, and when the magnitude is 8, 16 or 32, the 3-sigma rule works perfectly. But when the magnitude is 4, some observations are flagged as outliers that are not outliers. Once the contamination level reaches 10%--a value often cited as conservative or typical of actual data—swamping no longer exists. But masking rears its head again: in this 10% contamination case, no outliers are identified, even though 10 out of the 100 observations are outliers. And this is true whether the magnitude of the outlier is 4, 8, 16 or 32. So, one key issue to develop an effective outlier detection method is balancing the competing effects between swamping and masking.

One formal approach to modeling outliers uses mixture models. A contaminated normal mixture model assumes most observations are well represented by an i.i.d. sequence of Gaussian random variables with a mean and standard deviation; but some fraction of the observations ε are drawn from a different distribution. Formally we can express the overall data distribution as

$$p(x) = (1 - \varepsilon)\phi(\mu, \sigma; x) + \varepsilon\psi(x), \tag{1.1}$$

where $\phi(\mu, \sigma; x)$ denotes the density $N(\mu, \sigma^2)$ and $\psi(x)$ represents the contaminating distribution.

Technically any contaminating distribution can be used; but a popular outlier model assumes the contaminating distribution is identical to the data generating normal distribution, except the variance of the outlier model is greater than that of the data generating model. A common way to express this type of contaminated normal models is

$$CN(\mu_1, \mu_2, \sigma_1, \sigma_2, \varepsilon). \tag{1.2}$$

Equation 1.2 corresponds to the normal mixture density

$$p(x) = (1 - \varepsilon)\phi(\mu_1, \sigma_1; x) + \varepsilon\phi(\mu_2, \sigma_2; x). \tag{1.3}$$

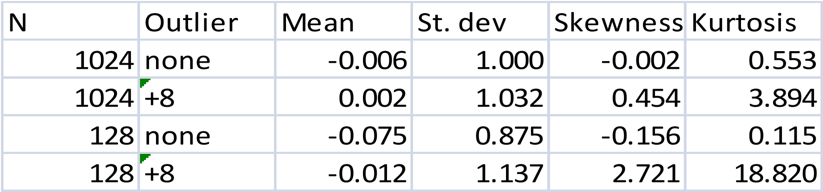
There is also terminology commonly used to describe outliers. The contamination level represents the percentage of outliers. In a multivariable context, the contamination level could happen at the same observation for all variables. Alternatively, there might be differential variable contamination where an outlier influences some variables, but not all, for a given observation. Outlier magnitude refers to the numerical magnitude of the outlier. Depending on the outlier type, this has different interpretations. But generally we can think of this as the value added (or subtracted, with negative-valued observation) for a given observation, yielding the outlier. So, for instance, if a data generating process returns a true value of 2.356, but a sensor error records a value of 3.356, we would say the outlier has magnitude 1. Additive outliers add some magnitude to an observation. Correlation outliers manifest in multivariate settings where a correlation structure controls the behavior of outliers.

There are many excellent surveys published which provide more details of outlier detection algorithms, including Barnett and Lewis (1994). Rousseeuw and Leroy (1987) describe and analyze a broad range of statistical outlier techniques. Marsland (2001) analyzes a wide range of neural network methods while Hodge and Austin (2004) consider many popular machine learning techniques.

The outline for the present work is as follows. Chapter 2 reviews some outlier detection methods, establishes a simulation framework for generating data with outliers and provides the criteria for assessing the efficacy of an outlier detection method. Chapter 3 presents the results from a simple outlier heuristic. These results are reasonable, but not great, and this is the rationale for considering features in outlier detection: can greater sensitivity be achieved with features? Chapter 3 also describes the construction of 13 features for outlier detection which are further tested. Chapter 4 develops a framework for testing features using a multivariate algorithm based on a 2-dimensional Voronoi diagram. The results are presented for all pairs of the 13 features. While the results of the Voronoi diagram represent an advance over simple heuristics, this method does not allow a user to screen candidate features and provide a clear way to assess the total number of outliers. Chapter 5 addresses this issue by proposing a general framework to test any set of candidate features to a) determine if those features are effective or not and b) use the deemed good candidates for outlier detection, including determining the number of outliers in the dataset. Chapter 6 is the application of the outlier detection method. And Chapter 7 is the conclusion and summary.

1.3 *Figure and Table*

Table 1.1. First four moments of four data scenarios with and without outliers.



N	Outlier	Mean	St. dev	Skewness	Kurtosis
1024	none	-0.006	1.000	-0.002	0.553
1024	+8	0.002	1.032	0.454	3.894
128	none	-0.075	0.875	-0.156	0.115
128	+8	-0.012	1.137	2.721	18.820

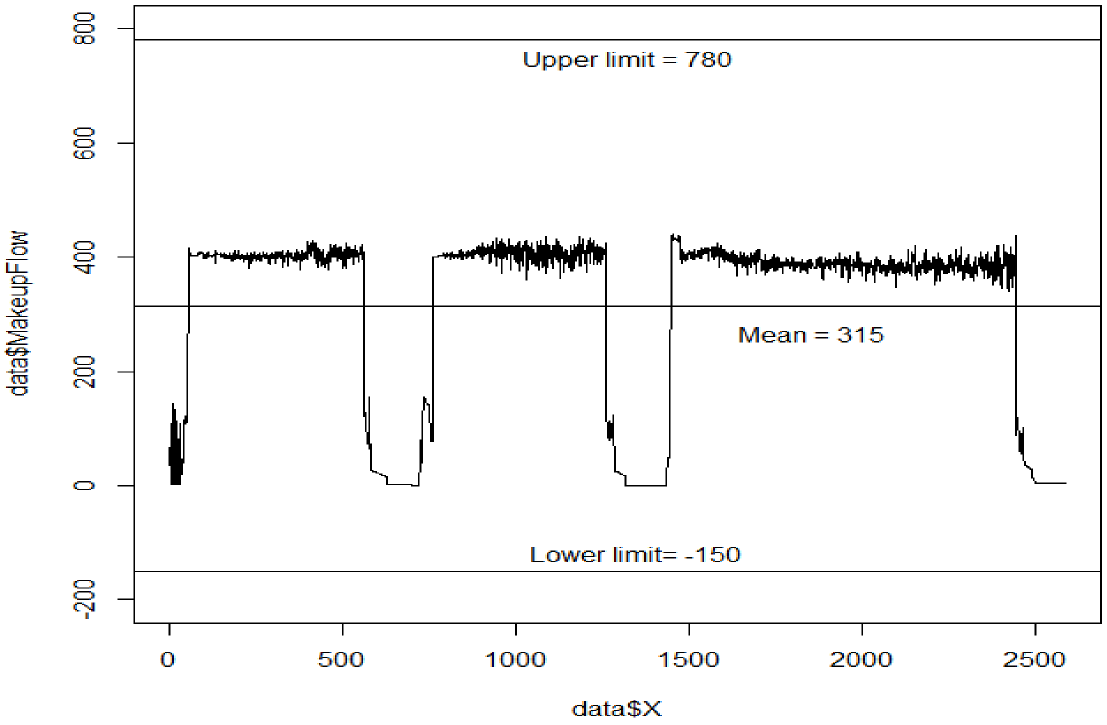


Figure 1.1. Illustration of the masking effect on outliers.

2. Background, Data Model and Simulation

Chapter 2 provides a more in-depth look at outliers and sets the stage for the simulation and modeling in chapters that follow. Section 2.1 covers several established and popular outlier detection methods, including the 3-sigma rule, the Minimum Covariance Determinant, the Local Outlier Factor and a Voronoi diagram based approach. These particular methods are covered in some detail because they provide a foundation on which later chapters will build. Section 2.2 describes the time series model used to generate the data for the simulations. Section 2.3 provides details of the simulation which are common throughout this work. And Section 2.4 discusses the criteria used for evaluation and validation.

2.1 Literature review

Many outlier detection algorithms rely upon a handful of common approaches. These include statistical based, depth-based, deviation, distance-based, set-based, model-based, graph-based, density-based and high dimensional outlier detection (Kriegel et al., 2009). Several popular approaches, which provide a foundation for the proposed outlier detection method, are reviewed first. Additionally, several methods for detecting outliers in time series data are also reviewed.

The first method is a heuristic that uses the number of standard deviations from the mean to set a threshold for outliers. This heuristic is frequently employed when a researcher believes their data is specified by a symmetrical distribution (like the standard normal) and they want an easy and simple way to classify outliers. Consider the standard normal distribution, with a mean of μ and standard deviation of σ. The cumulative distribution function of the normal distribution specifies that 68% of observations fall within 1 standard deviation of the mean, 95% of the observations fall within 2 standard deviations of the mean, 99.7% of the observations fall within 3 standard deviations of the mean, etc. An outlier rule based on the number of standard deviations from the mean is sometimes called a #-sigma rule, where '#' is a place holder that represents the number of standard deviations from the mean. Later on, the 3-sigma rule is explicitly tested and, for purposes of this paper, called the Simple Testing Method, or STM for short.

The Multivariate Least Trimmed Squares (MLTS) algorithm (Rousseeuw et al., 2004; Croux & Joossens, 2008) is a statistical type of outlier detection method. It is a statistical approach because it assumes the underlying probability distribution is symmetrical (i.e. multivariate normal) for the data and also because it seeks to estimate the parameters of a regression model while handling outliers in the data. It relies on the Minimum Covariance Determinant (MCD) that performs fast and efficient statistical outlier detection (Rousseeuw & Driessen, 1999; Hubert & Debruyne, 2010). MCD iteratively samples a subset of *h* observations (out of *n*) and attempts to find the subset whose covariance matrix has the lowest determinant. This subset is then retained as outlier free and used to estimate parameters for the regression model. Further technical details of this algorithm are provided in Chapter 3. This method requires the data follows a symmetrical distribution. It determines outlying points by comparing a score assigned to each observation to a chi-square distribution, again with the underlying assumption that outliers for this type of data should follow a chi-square distribution. Nearly all outlier detection algorithms require the data analyst to specify values for one or more input parameters and the MLTS is no exception. The MLTS algorithm requires specification of the contamination level of the data set under analysis. If one works with data where this information is known and consistent this is not problematic; but there are many instances of data analysis where one cannot be sure outliers are even present and these cases present challenges for the MLTS algorithm.

Another approach to outlier detection is called the Local Outlier Factor, or LOF (Breunig et al., 2000). Many outlier detection algorithms (like the MLTS) use a binary classification to determine outliers; but the LOF assigns a numeric value to each observation, with larger values indicating the observation is more likely to be an outlier. The 'L' in LOF stands for local, in that the degree of localness depends on how isolated the observation is with respect to the surrounding neighborhood of observations. As with the MLTS method discussed previously, the algorithm requires a user input parameter—the number of local neighbors to include.

LOF builds on distribution-based, depth-based and distance-based approaches. In a distribution-based approach a standard distribution is sought that best fits the data. Outliers are defined relative to this probability distribution. One could use a discordancy test, for instance. However, most of these tests are for univariate distributions only. In a depth-based approach, each data object is represented as a point in a k-dimensional space and assigned a depth. Outliers have smaller depths. Distance-based outlier algorithms consider the distances between all data points and flags as outliers those which exceed a user defined distance.

For global outliers (observations outlying relative to the entire data set) distribution-, depth- and distance-based approaches work well. But outliers might have a more complex structure. For instance, an observation can be outlying relative to a local neighborhood, particularly with respect to the density of its neighborhood. These are called ‘local’ outliers, which are problematic for approaches that identify global outliers.

For the data in Figure 2.1, the LOF computes the distance between all points within cluster C_2. Because all of these points are close together, this creates a local density, resulting in small distances with similar magnitudes. But the distance from all points in C_2 to o_1 and o_2 is large. Moreover, and more importantly from the point of view of the LOF, $o_1$and o_2 are the only points with these large distances. A cluster of points does not exist around these two points so the local density is non-existent. So the key to detecting outliers in the LOF framework is the distance between points, conditioned on the information in the surrounding neighborhood. A purely distance based outlier metric would classify all points in C_1 as outliers, in addition to o_1 and o_2. The LOF is better than a distance based metric for outlier detection in this case. The value returned from the LOF algorithm depends on how closely the data points are packed in the local neighborhood. And this neighborhood is defined by the distance to the minimum points (*MinPts*) nearest neighbor, where *MinPts* is the minimum number of points of the nearest neighbors.

An alternative, but related, method is the Voronoi neighbor outlier factor (VNOF) which uses geometric principles to define the neighborhood (Qu, 2008). Instead of a fixed parameter determined by the user, VNOF uses the Voronoi nearest neighbor geometry information to calculate the outlier factor for each data point. VNOF is non-parametric, a definite advantage for users of an outlier detection algorithm who have little understanding of reasonable choices of parameter values in parametric methods. The VNOF algorithm is also computationally efficient.

Let $V(p_i)$ denote a Voronoi cell. Assume we have a set *S* of *n* points, $p_1, p_2, \dots, p_n$, in a plane. Then a Voronoi diagram (Preparata & Shamos, 1985), *Vor(S)*, is a subdivision of the plane into Voronoi cells, with the latter being defined as the set of points q that are closer or as close to p_i than to any other point in *S*. Formally we can express this as

$$V(p_i) = \{q | dist(p_i, q) \leq dist(p_j, q), \forall\, j \neq i\}, \tag{2.1}$$

where *dist* is the Euclidean distance function. Figure 2.2 is an example of a Voronoi diagram where the plane is decomposed into *n*=6 convex polygonal regions, one for each p_i. Vertices are called *Voronoi vertices* and *Voronoi edges* are defined as the boundaries between two Voronoi cells. The boundaries of a Voronoi cell $V(p_i)$ cannot exceed *n*-1 edges. Three theorems (see Preparata & Shamos, 1985) are also needed to apply Voronoi diagrams to outlier detection.

Theorem 1: Every nearest neighbor of p_i *defines an edge of the Voronoi polygon* $V(p_i)$.
Theorem 2: Every edge of the Voronoi polygon $V(p_i)$ *defines a nearest neighbor of* p_i.
Theorem 3: For $n \geq 3$*, a Voronoi diagram on n points has at most 2n-5 vertices and 3n-6 edges.*

Assume we have a data set *S*. For a point $p_i \in S$ each edge of the Voronoi polygon $V(p_i)$ defines a nearest neighbor p_i. The numbers of nearest neighbor vary for different points, some have more and some have less. Once the polygons are formed, a periphery of the immediate neighborhood is created. The Voronoi neighborhood is now defined more precisely.

Voronoi nearest neighbor. For a point p_i of set *S*, the nearest neighbors of p_i defined by the Voronoi polygon $V(p_i)$ are the Voronoi nearest neighbor of p_i, denoted as $V_{NN}(p_i)$. In the figure above, the nearest Voronoi neighbors to point p_1 are p_2, p_3, p_4, p_5 and p_6.

Voronoi reachability density. The Voronoi reachability distance of point p_i is defined as

$$V_{RD}(p_i) = \frac{1}{\sum_{O \in V_{NN}(p_i)} \frac{dist(p_i, o)}{|V_{NN}(p_i)|}}, \tag{2.2}$$

where $|V_{NN}(p_i)|$ is the number of points in $V_{NN}(p_i)$. This means that the reachability distance is an inverse average of the distance determined by the Voronoi nearest neighbors of p_i.

Voronoi neighbor outlier factor. The Voronoi neighbor outlier factor of p_i is defined as

$$V_{NOF}(p_i) = \frac{1}{|V_{NN}(p_i)|} \sum_{O \in V_{NN}(p_i)} \frac{V_{RD}(o)}{V_{RD}(p_i)}. \tag{2.3}$$

This means the Voronoi neighbor outlier factor of p_i is the average of the ratio of the local Voronoi density of p_i and those of p_i's Voronoi nearest neighbors.

The formal definitions given in Qu (2008) are implemented algorithmically as follows:

```
Input: Data set $S$.
Step 1. Construct a Voronoi diagram of $S$.
Step 2. For each $p_i \in S$, compute the Voronoi reachability density, $V_{RD}(p_i)$.
Step 3. For each $p_i \in S$, compute the Voronoi neighbor outlier factor, $V_{NOF}(p_i)$.
Step 4. Sort the data in descending order by $V_{NOF}(p_i)$.
Output: Outlier factor of the points in $S$.
```

Now we consider some algorithms which have been developed to work optimally on time series data. One method requires fitting an auto-regressive time series model to the data first and subsequently identifies outlying points using residuals and the hat matrix (Hau & Tong, 1989). Burridge and Taylor (2006) developed an outlier detection algorithm for additive outliers using extreme-value theory. Extreme value theory is a branch of statistics that considers events which show extreme deviation from the median of a probability distribution. It makes sense to consider outliers from this perspective, as researchers sometimes conceptualize outlying observations as extreme points that happen less frequently.

Other time series algorithms for identifying outliers include those that leverage the stationarity, or lack thereof, of the time series. For instance, Choy (2001) developed a method for identifying outliers in stationary time series by iteratively estimating a model, detecting outliers and removing outlying points. Other algorithms rely on violations in stationarity, such as the algorithm that identifies outliers in autoregressive time series data relies using a change detection paradigm

(Gombay, 2007). If the order of the model, the mean or variance changes significantly over time, this reflects a disturbance which can be identified using an efficient score vector. Another outlier detection paradigm leverages the local violations of stationarity in the time series to determine the presence of outliers (Last & Shumway, 2008).

A final important class of outlier detection algorithms for time series data concerns those which require automatic or online detection, which not only have the function of identifying an outlier in the present moment but also facilitate the prediction of future states and/or allow an analyst to take immediate action based on outliers. One such method uses the median from a neighborhood of data points in the time series and compares that value to a threshold, a method that is fast and works well for online or streaming data (Basu & Meckesheimer, 2007). Time series forecasting also falls under this purview because, in a regression-based forecast model, outliers are identified based on their deviation from expected (or forecasted) values (Aggarwal, 2013). But forecasting can also be the goal, where one would like to make optimal future predictions. In order to do this accurately and effectively it is important that outliers are removed from the data or else the model fit will be incorrect, leading to biased predictions.

This concludes the review of several key outlier detection algorithms which are representative of broader classes of outlier detection methods. The MLTS is a statistical approach to outlier detection; but it requires the user to input the percent of outliers in the data. It also makes assumptions about the distribution of the underlying data generating mechanism. The LOF is a density-based outlier detection tool that requires the user to specify the number of *MinPts* and the algorithm is sensitive to this parameter. Since LOF ranks points only considering the neighborhood density of the points (determined by the parameter *MinPts*) it may miss potential outliers whose densities are close to their neighbors. The Voronoi diagram approach overcomes some shortcomings of the LOF algorithm, as it does not require the user to define the minimum number of points to define the neighborhood since those neighborhoods are created by Voronoi tessellations. However, just as with LOF, the results are in the form of order statistics and one must still set a threshold to determine which observations are outliers. The Voronoi diagram method reviewed here has only been developed for univariate data. In general, there are not many outlier detection algorithms designed for multivariate data but the proposed outlier detection method is designed for such case. Both the LOF and VNOF represent a machine learning approach to outlier detection. There are several outlier detection algorithms specifically designed for time series data and time series models (such as the AR model). Moreover, some of these algorithms are effective at identifying outliers in real time.

The foregoing review raises some important challenges in developing an effective outlier detection algorithm. First, the user of the algorithm should not even need to assume that outliers are present and, even if they are present, have any foreknowledge of their structure in the data. A robust and sensitive algorithm should have the capacity to determine the presence of outliers first and, if present, determine the number of outliers in the dataset without the analyst supplying an input parameter. In cases where one is unsure if outliers are present, what good is an outlier detection algorithm that requires the user to provide information about outliers that may not exist? Additionally, a good outlier detection algorithm should flag outliers automatically, if they are present. Before specifying the algorithm in more detail, it is necessary to describe the data simulation and evaluation methods.

2.2 Time series model

Let $u_v \in \mathcal{R}^m$ be the realizations of a stationary time series. If these realizations are generated from a multivariate auto-regressive (AR) model with order *p*, then define the auto-regressive model as

$$\boldsymbol{u}_t = \boldsymbol{w} + \sum_{l=1}^{p} \boldsymbol{A}_l \boldsymbol{u}_{t-l} + \boldsymbol{\varepsilon}_t, \tag{2.4}$$

where $\boldsymbol{\varepsilon}_t$ is white noise (uncorrelated random variables with zero mean and finite variance), t designates the observation and l specifies the lag. The coefficient matrices of the AR(*p*) model are represented by $A_1, \ . \ . \ . \ , A_p \in \mathcal{R}^{m \times m}$ and $\boldsymbol{w} \in \mathcal{R}^m$ is an intercept vector which allows the time series to have a nonzero mean (Lutkepohl, 2005). The AR coefficients used in the simulation studies of this thesis are provided in Table 2.1. This is like a multiple regression but with lagged values of $\boldsymbol{u}_t$ as predictors. At each lag, each time series has 3 predictors because there are 3 variables in the system.

The outlier model builds upon Equation 2.4 and is specified in Equation 2.5 as

$$\boldsymbol{z}_t = \boldsymbol{u}_t \pm \sum_{v=1}^{n} I_t \boldsymbol{x}\,. \tag{2.5}$$

In Equation 2.5, $\boldsymbol{z}_t$ is the observed time series value, $\boldsymbol{u}_t$ are the realizations from the AR model in equation 2.1, I is an indicator variable which takes the value of 1 if an outlier is added to that time point and 0 otherwise and $\boldsymbol{x}$ is the magnitude of the outlier added to the time series.

2.3 Simulation setup and data generation

For each simulation experiment conducted, 100 multiple time series observations were generated from a Vector AR(2) model, which generalizes the univariate AR(2) model as defined in Equation 2.4. The simulations were implemented in Matlab, using published Matlab code to generate the AR realizations (Schneider, 2001). In all simulation studies, an uncorrelated variance/covariance error term was specified so as to have data generated from a model with all variables having variance 1 and covariance 0. The number of variables for all simulations was 3.

Outliers were always additive, meaning they were introduced after the time series was generated and that the magnitude of the outlier was added to the simulated observation value. However, because the underlying data generation process was Gaussian normal, negative observations could occur as frequently as positive observations. So for observations with a negative value the outlier was subtracted. Additive outliers were always introduced for the same observation for all variables in the time series. Table 2.2 shows an example of 5 time series observations as generated by Equation 2.4 and then, for observation 3, with an additive outlier of magnitude 3.

For all simulation studies, 15 unique additive outlier conditions were examined. 5, 10 or 15 outliers were introduced. These are the contamination levels. Each contamination level has outliers of magnitude 1, 2, 3, 4 or 5. Fully crossing three contamination levels with five magnitudes yields 15 outlier conditions. These were chosen partly based on prior literature reviews; but also to demonstrate the strength of an outlier detection method. Because the data were generated from a Gaussian standard normal (where we expect values greater than approximately 3 very infrequent), a magnitude of 4 or 5 is quite extreme and even weak outlier detection methods can do well in these conditions, especially when the contamination level of the time series is high. But a more critical test of an outlier detection algorithm is made at magnitudes of 1 or 2 (and small contamination levels) because, in these cases, it is often times difficult to separate the underlying signal from noise. A time series observation with a small initial magnitude, say 1.2, that has an additive outlier magnitude of 1 added to it results in an observation of 2.2, which is well within the bounds of where approximately 99% of observations generated under a normal Gaussian distribution are expected to fall within three standard deviations of the mean. Hence, observed values between -3 and 3, if they have an additive outlier added to them, are challenging to separate from expected observations. These are examples of inliers, as they lie within the normal range of the rest of the data.

For each of the 15 conditions 25 different time series with 100 observations were generated. Multiple time series within the same condition were generated to average out the random fluctuations expected in simulation studies and provide a more stable estimation of the outlier detection technique. Within a condition, only the specified contamination level and outlier magnitude were used. The observations selected for the introduction of additive outliers were randomly determined. And across each of the 25 different time series generated for each condition, the observations (i.e. rows) selected were always randomly selected.

Analyses were conducted and results were obtained on multivariate time series data in a multivariate and a univariate fashion. For the multivariate approach, all variables were analyzed together. In a univariate approach, the same multivariate data were used; but each variable was analyzed separately, as though it was independent. This was done because many outlier detection methods are specified for univariate and it was important to see the change in performance and outcomes that come from a univariate versus a multivariate analysis. Moreover, there are some researchers who advocate for the use of univariate analyses, even if multivariate approaches are available.

2.4 Validation criteria

Outlier detection efficacy was assessed with True and False Positive Rates (TPR and FPR, respectively), as defined in Table 2.3. To compute the true and false positive rates for the first set of simulation studies a sliding threshold was employed. The TPR and FPR were calculated assuming there was only a single outlier in the dataset. Then they were computed assuming there were two outliers in the dataset. This process was repeated up to 20 outliers. For each outlier detection method this resulted in 20 TPRs and FPRs, one for each threshold between 1 and 20. For outlier detection, this approach makes sense because, in data practice, one often does not know *a priori* the outlier contamination level of the dataset. Then, depending on the contamination level for the given condition (either 5, 10 or 15 outliers in a time series with 100 observations), a subset of TPRs were averaged. If there were 5 outliers in the condition, then the TPR was computed by averaging the TPR for the 5 thresholds from 1 to 5; if there were 10 outliers the 5 TPR thresholds from 6 to 10 were averaged; and if there were 15 outliers the thresholds from 11 to 15 were averaged. Other averaging possibilities also make sense, such as averaging across both sides of the contamination level. But irrespective of the averaging approach used, the ordinal results were always consistent within an averaging scheme and, moreover, were close to the result at 5, 10 or 15 outliers. Averaging was used to produce a result not influenced by statistical fluctuations across the multiple simulations. Another possibility that does not rely on a moving threshold is to use a threshold based on a statistical distribution. Observations beyond that threshold are considered outliers. Some existing outlier detection methods leverage such a threshold. Thresholding based on a statistical distribution makes some sense, as it gives a user of an outlier detection technique a different way to decide if observations are outliers or not. But at the same time, the statistical distribution may or may not fit the data and/or outliers of the particular data application so, while it could facilitate ease of use, this might also lead to biased results.

A TPR for an outlier detection algorithm is only effective if one knows the location of outliers ahead of time. One of the key aims of this thesis is to develop an outlier detection algorithm that does not require this knowledge (nor a parameter which relates to this knowledge). Hence, the proposed method can identify outliers without using a true positive rate and will be useful and effective in applied research; but even to demonstrate that the method is valid, TPR and FPR are still used.

The final chapter presents an application of the proposed method for forecasting time series with outliers. To evaluate the efficacy of the method, the mean square error (MSE) is also used. The mean square error is the sum of the squared deviations between the true values and biased values. The MSE provides a sense of how far away the statistical estimate is from the true value, with values of 0 meaning the estimate is very accurate.

2.5 Summary

This chapter introduced in some detail several outlier detection algorithms. Each represented a different class of outlier detection methods. For instance, the Minimum Covariance Determinant/Multivariate Least Trimmed Squares is representative of a statistical approach to outlier detection. One goal for introducing outlier detection methods form different backgrounds is to show the diversity of approaches to outliers. But this diversity also quickly reveals a fundamental challenge underlying many existing outlier techniques, irrespective of the disciplinary grounding from which they come: that users are almost always required to provide some prior information about their data and/or outliers in order to effectively apply the algorithm. This assumption works fine if this information is known beforehand, as it might for instance in a quality control setting for an industrial factory process. Or in some cases, such as when base rates are known, specifying the number of outliers in advance might be possible. But many times a researcher may not have prior expectations and these necessary constraints and assumptions that are required to use the methods could induce artifacts in the data, especially if the assumptions are not true. So an important goal going forward is to leverage the strengths and outlier detection ability of established methods but to try and find a way to use those methods, or pieces of them, by peeling away the prior assumptions and constraints. To this end, feature construction is proposed in Chapter 3 as a way to meet these goals.

2.6 Figures and Tables

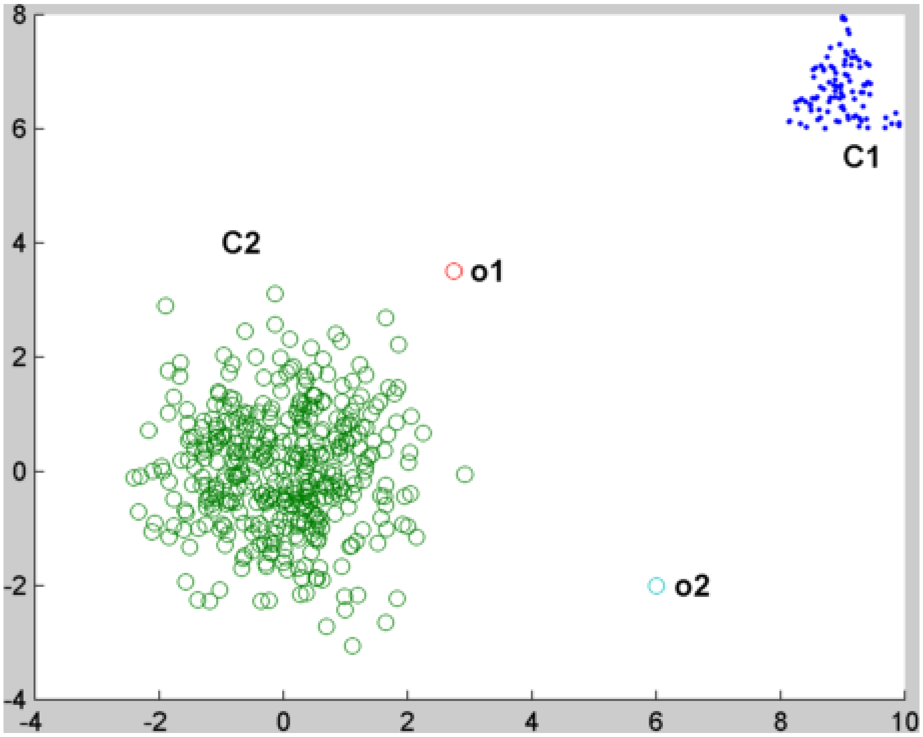


Figure 2.1. Two data clusters (C_1 and C_2) with two outliers (o_1 and o_2).

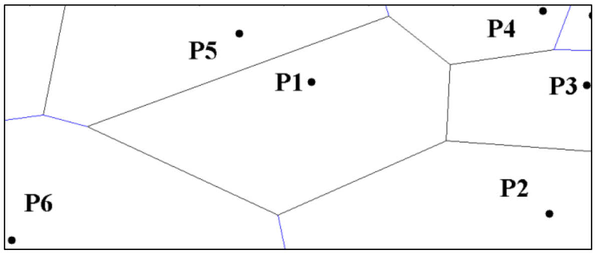


Figure 2.2. A Voronoi diagram with 6 cells.

Table 2.1. AR(2) coefficients for time series model used to generate simulation data. The eigenvalues of the coefficients are all less than 1, indicating the system is stable and stationary (Johnston & DiNardo, 2001; Glaister 1991).

	Lag 1			Lag 2		
Variable 1	0.0	0.1	0.2	0.0	0.3	0.0
Variable 2	0.0	0.4	0.2	0.4	0.0	0.1
Variable 3	0.1	0.0	0.0	0.3	0.4	0.0

Table 2.2. Example time series with and without outliers. Row 3 is bolded because, in the 'Additive Outlier' columns, an outlier of magnitude 3 has been added or subtracted (if the observation was negative) to the observations in the corresponding 'No Outliers' columns.

Observation		No Outliers				Additive Outlier		
1		2.1	3.6	4.9		2.1	3.6	4.9
2		-3.2	-.9	-1.2		-3.2	-.9	-1.2
3		**-1.2**	**.75**	**2.1**		**-4.2**	**3.75**	**5.1**
4		-.5	-1.9	-.03		-.5	-1.9	-.03
5		1.7	3.0	.25		1.7	3.0	.25

Table 2.3. Definition of True and False Positive Rate (TPR and FPR, respectively). The Definitions in the right column of the table are based on the cells on the left side of the table. TP means True Positive; FP means False Positive; FN means False Negative; and TN means True Negative.

		Outlier in data? (Gold Standard)			Definitions
		Yes	*No*		***True Positive Rate*** TP / (TP + FN)
Detected Outlier?	*Yes*	TP	FP		***False Positive Rate*** FP / (FP +TN)
	No	FN	TN		

3. Feature Extraction

Chapter 3 provides the first set of simulation studies. Section 3.1 considers a very simple, but commonly used, outlier heuristic. Despite its popularity by researchers, the simulation shows it limitations. Sections 3.2 through 3.5 contain principles of feature construction, a technical review of an existing outlier detection algorithm (which is used as a feature in the current study) and an overview of all the features used. Section 3.6 tests via simulation all the features. We will see that many of the features are much better at detecting outliers than the Simple Testing Method heuristic. This will also set up an important question by the end of this chapter: how can one determine which feature will yield the best outlier detection results?

3.1 Simple Testing Method

Researchers often employ simple outlier detection heuristics. While they are easy to implement and understand, they also have shortcomings. Perhaps the simplest approach to outlier detection is a visual inspection of the plotted data. With visual inspections, a rule of thumb for outlier identification is adopted, such as identifying points far from the others or identifying points that lay beyond some threshold. However, what is 'far' and how to determine the threshold is mostly arbitrary and varies from one researcher to another and one research context to the next, not exactly a solid basis for reproducible and accurate research.

Figure 3.1 is a simple univariate plot of 100 observations of time series data generated from a Normal distribution with a mean of 0 and a standard deviation of 1 ($\sim Normal(\mu = 0, \sigma^2 = 1)$). The x-axis is time while the y-axis represents the observed values at each time point. The red horizontal lines (intercept 3 and -3) are arbitrary reference lines that one might use to adjudicate outliers from non-outliers. This approach is not without foundation because if one reasonably expects their data to follow a standard normal distribution, approximately 99% of all observations should take values between (-3, 3). This rule is sometimes called the 3-sigma rule. If we adopt the heuristic that observations taking values larger than 3 or less than -3 are outliers for Figure 3.1, then 4 observations are labeled outliers.

One criticism of a heuristic like the 3-sigma rule is that there is no guarantee the points beyond -3 or 3 are in fact outliers. For standard normally distributed data, we expect some observations to take values more extreme than 3 or -3. Additionally, some observations which fall within the range of most data (between 3 and -3) can still be an outlier. As we shall soon see, a very important diagnostic criterion for comparing outlier detection methods is their performance when outlier magnitudes are small and within the range of the 'normal' data. These are examples of so-called inliers.

Multivariate data also presents problems for a heuristic like the 3-sigma rule. Figure 3.2 is a multivariate time series plot with three variables. As with Figure 3.1, assume the 3-sigma outlier detection rule and call observations outliers if their magnitudes are beyond -3 or 3. The observation at time point 78 (which is labeled 'outlier' in Figure 3.2) is clearly classified as an outlier because its magnitude is less than -3. But this multivariate data set has three variables (each represented by a different colored line in Figure 3.2) which could co-vary because they are related to the same behavior. At the very least, they are linked by virtue of being measured at the same point in time. If we classify the observation measured at time point 78 for the blue line as an outlier, should the other two observations at time point 78—represented by the orange and yellow lines—also be labeled as outliers, even though they take values between -3 and 3? One possibility is that only the observation at time point 78 represented by the blue line is an outlier. Another possibility is that all three variables measured at time point 78 are outliers because an environmental variable influenced the outcome of these variables and made them more extreme than they otherwise would have been, such as 3 different sensors subjected to the same sudden movement. If one only identified outliers based on a plot with a 3-sigma rule, it would be challenging to adjudicate outliers in multivariate datasets where some observations for some variables exceed the outlier threshold while other observations at the same time point for other variables do not exceed the threshold.

Figure 3.3 and Table 3.1 present the results from the Simple Testing Method based on 3 standard deviations from the mean using a simulation study. 25 sets of 3 variable multivariate time series with 100 observations each were generated at random from a standard normal distribution with a mean of 0 and standard deviation of 1 for each of the 15 outlier conditions (5, 10 or 15 outliers at each of 5 magnitudes of outliers—1, 2, 3, 4 or 5). The TPR, as defined in Chapter 2, was calculated for each univariate time series and these were averaged together across variables to obtain a single TPR for each multivariate data set. Figure 3.2 shows a plot of one of these time series. The TPR for the Simple Testing Method for each of these 15 outlier conditions is plotted in Figure 3.3 and the numerical values underlying this figure are in Table 3.1.

An outlier magnitude of 1 yields the lowest TPR for all three contamination levels and the value is relatively similar: .31 for 5 outliers, .28 for 10 outliers and .33 for 15 outliers. The best TPR is .82 for 5 outliers of magnitude 3 and the worst TPR is for outlier magnitude 1 with 10 outliers. The average TPR across all 15 outlier conditions is .60. One important trend that emerges from Figure 3.3 is a larger TPR with larger outlier magnitudes. Specifically, we see that for magnitudes 3, 4 and 5, the 3-sigma rule has nearly identical performance. This makes sense for this rule because it is only flagging values that are more extreme than 3 standard deviations from the mean as outliers. Outliers of magnitude 3 (or more) added to the original observations make those observations more extreme than the plus or minus 3 cutoffs for outliers. So it matters little how much larger than 3 the magnitude of the outlier is for the 3-sigma rule. Overall, the Simple Testing Method with a cutoff of 3 standard deviations from the mean as a cutoff method gives some accurate information; but given the mediocre TPR results, there is much room for improvement. Feature construction is proposed to improve these true positive rates.

3.2 Feature construction

Features are numerical vectors which represent some object. Feature construction (Liu & Motoda, 2012) is the process or method by which one defines a new feature from the original object for better or more desirable next stage data analysis. It is commonly employed in pattern recognition, social networks and machine learning.

Define a data structure $\mathcal{D}$ as a $n \times p$ matrix where n represents observations and is equivalent to the number of rows in $\mathcal{D}$. p is the number of response variables and is equivalent to the number of columns in $\mathcal{D}$. Since all-time series data used for the simulation studies that follow contained 100 observations and 5 variables, we can set $n = 100$ and $p = 5$. 13 unique features were constructed and the details of these features are presented in the next section. All 13 features are $n \times 1$, which means they have the same number of observations as the underlying data object. But the number of variables has been compressed from 5 in the original data to 1 in the feature vector.

The 13 features are representative of the types of features one could construct; but otherwise there is nothing special about these particular features. Some are based on a regression model while some are model free. Some are based on statistics while others are based on a to-be described leave one out covariance algorithm. Table 3.2 overviews these features.

9 of the 13 features implement a so-called ‘leave one out’ method, which algorithmically is the same as the jackknife (Quenouille, 1949; Quenouille, 1956; Tukey, 1958; Efron & Stein, 1981), though the typical goal of the jackknife procedure is the estimation of variance and bias. Whereas here we are only concerned with the computation of a feature. Denote the number of observations in a time series as $i = 1, \dots, n$. The first observation is removed from the time series, which now has length $n - 1$, and a statistic is computed on this remaining set of $n - 1$ observations. The first observation is then returned and the second observation is removed and the statistic is computed again on this new set of $n - 1$ observations. This sequence of removing and replacing each of the i observations in order is repeated n times, cycling through the total set of observations exactly once. The particular statistical operation required for the feature is computed n times for each feature. F2, F11, F12 and F13 do not require the leave one out method.

This ‘leave one out’ method, while algorithmically equivalent to the jackknife, is perhaps more similar in its goal to computing Cook’s distance (or other influence measures) in ordinary least squares. Cook’s distance assesses the influence of a single observation by determining the difference between the statistic (usually the residual in the classic Cook’s distance measure) on an entire data set when the observation is included as compared to when it is removed. Each data point is then assigned an index, with larger values reflecting more influential points (possibly outliers) which might affect the regression models in unintended ways.

Several of the features require the computation of the determinant of the covariance matrix based on data while another subset of features require the computation of the determinant of the covariance matrix after a model has been fit. This determinant is known as Generalized Sample Variance (Wilks, 1932) and is a 1-dimensional scalar measure of multivariate scatter (Johnson & Wichern, 2007).

Before the features are presented, the Minimum Covariance Determinant (MCD) and Multivariate Least Trimmed Squares (MLTS) methods are reviewed because three of the features depend on these.

3.3 MCD and MLTS

The Multivariate Least Trimmed Squares (MLTS) (Rousseeuw, 2004) is a robust approach for estimating the vector autoregressive model while handling outliers in the data. It relies on a popular statistical procedure, called the Minimum Covariance Determinant (MCD) that performs fast and efficient statistical outlier detection (Rousseeuw, 1999; Hubert, 2010). MCD finds *h* observations (out of *n*) whose covariance matrix has the smallest determinant. Incidentally, this process embodies the same algorithmic underpinning as D-optimality experimental design (Fedorov, 1972) in that both the MCD procedure and D-optimality seek to optimize the determinant of the covariance matrix. MCD seeks the smallest determinant whereas D-optimality desires the largest determinant. MCD and D-optimality also differ in their aims, where the former uses the minimum determinant to identify a set of observations that are outlier free whereas the latter aims to define the parameter values that will yield the most optimal experimental design. Geometrically, since the determinant is inversely related to the volume of an ellipsoid, MCD aims to find the ellipsoid with the smallest volume, as points that are outlying or extreme, when they are included in the set of observations used to compute the determinant; however, D-optimality will have the effect of making the volume of this ellipsoid large.

Assume we have a data set with *p* variables with $i = 1 \ldots n$ observations. Take a subset of these *n* observations, *h*, where *h* is chosen by the user and constrained by $[(n+p+1)/2] \leq h \leq n$. The MCD algorithm selects a subset of randomly selected observations of size *h*, computes the mean $\boldsymbol{T_1}$, the variance/covariance matrix $\boldsymbol{S_1}$ and then determines the statistical distance *d* for each data point $\boldsymbol{x_i}$ in *n* according to

$$d(i) = \sqrt{(\boldsymbol{x}_i - \boldsymbol{T}_1)^t \boldsymbol{S}_1^{-1} (\boldsymbol{x}_i - \boldsymbol{T}_1)}. \tag{3.1}$$

The obtained distances from (3.1) are next sorted from smallest to largest, and the h smallest are retained as h_2. A new mean $\boldsymbol{T_2}$ and variance/covariance matrix $\boldsymbol{S_2}$ are computed from h_2. The relationship in Equation 3.2 between the determinant (*det*) of the two variance/covariance matrices $\boldsymbol{S_1}$ and $\boldsymbol{S_2}$ is defined by

$$\det(\boldsymbol{S2}) \leq \det(\boldsymbol{S1})\,. \tag{3.2}$$

These steps are repeated 500 times (when *n* is less than 500), where a different subset h is chosen for each iteration. The subset yielding the smallest overall determinant is then used for further statistical analysis.

The MCD framework has been extended to a regression framework for time series data (Croux & Joossens, 2008), which is called the multivariate least trimmed squares, or MLTS. The MLTS algorithm leverages the residuals from least squares regression. Then, instead of determining the *h* from the raw data as in the MCD, MLTS selects the *h* observations with the smallest determinant of the covariance matrix of the residuals. So in the MLTS the joint variability (of the predictor and response variables) is modeled by using residuals. Also, the MLTS returns a binary output vector where a 0 indicates the observation is an outlier.

Define h as the size of the subset. Let $\mathcal{H}$ denote the superset of all samples H of size h in time series data $\boldsymbol{Z} = \{(x_t, y_t), t = k + 1, \dots T\}$ where k is the AR model order, x_t is the predictor and y_t is the response. For any $H \in \mathcal{H}$, define the classical least squares regression fit for the estimates of beta and the covariance matrix of the error as

$$\hat{\beta}_{OLS}(H) = (X_H' X_H)' X_H' Y_H \tag{3.3}$$

and

$$\widehat{\Sigma}_{OLS}(H) = \frac{1}{h-p}\Big(Y_H - X_H\hat{\beta}_{OLS}(H)\Big)' \Big(Y_H - X_H\hat{\beta}_{OLS}(H)\Big), \tag{3.4}$$

where p is the number of variables. Let $\widehat{H}$ be the subset of size h which has the smallest determinant of all iterations in the MCD framework after computing the covariance matrix of the least squares regression:

$$\widehat{H} = \begin{matrix} argmin \\ H \in \mathcal{H} \end{matrix} \, det\left(\hat{\Sigma}_{OLS}(H)\right). \tag{3.5}$$

Then the MLTS estimators of beta and the covariance matrix of the error are

$$\hat{\beta}_{MLTS}(Z) = \hat{\beta}_{OLS}\big(\widehat{H}\big) \tag{3.6}$$

and

$$\widehat{\Sigma}_{MLTS}(H) = c_\alpha \hat{\Sigma}_{OLS}\big(\widehat{H}\big). \tag{3.7}$$

c_α is a correction factor to consistently estimate the variance/covariance error term of the model where α is a user-defined trimming proportion, $\alpha = 1 - \frac{h}{n}$, and

$$c_\alpha = \frac{1-\alpha}{F_{\chi^2_{p+2}}(q_\alpha)}, \tag{3.8}$$

where p is the number of variables, $q_\alpha = \chi^2_{q,1-\alpha}$ is the $1-\alpha$ quantile of a chi-square distribution with q degrees of freedom (where q=p*k + 1; k is the AR model order) and F is the CDF of this chi-square distribution. Once the model parameters have been obtained, the residuals are computed. The set of h sorted residuals which yield the smallest determinant of the covariance matrix (as in MCD) are considered outlier free.

The authors (Croux & Joossens, 2008) propose a reweighting to improve statistical efficiency and improve its performance in their simulation setup. Here, the residuals are compared to q_δ, which is the chi-square distribution

$$q_\delta = \chi^2_{q,1-\delta}, \tag{3.9}$$

where δ is a user defined parameter. If the residual is larger than this critical threshold, then that observation is flagged as an outlier (coded as a 1) and the result is a vector of 1's and 0's.

3.4 Features

Feature 1: MLTS-based covariance

Feature 1 is identical to the non-weighted MLTS result except it implements the leave one out (or jackknife) algorithm. So the MLTS is iterated *n* times, where *n* is the number of observations in the data. The result is a vector of values representing the smallest determinant of the covariance of the residuals. All else being equal, determinants of larger magnitude reflect the presence of an outlier.

Feature 2: Reweighted MLTS

Feature 2 is the reweighted result of the MLTS. The output of this feature is a vector of binary outputs of 1's and 0's that codes observations as 'good' and outliers, respectively. This feature requires setting two parameters, the subset of observations believed to be outliers (h) and the parameter δ used in the reweighting step for the chi-square distribution. h was chosen to optimize the algorithm (i.e. to equal the exact number of outliers) so the reweighting could have a negative effect for this simulation by excluding observations which were already selected as outliers in the final subset h.

Feature 3: Determinant of covariance (model-based)

On each iteration of leave one out, a vector AR(p) model is fit and the estimate of the population error variance/covariance matrix is computed from the sum of squared residuals divided by the degrees of freedom. Calculation of the residuals requires the estimates of the intercept and beta parameters. The inverse determinant of this variance/covariance matrix is then taken,

$$\frac{1}{\det(\boldsymbol{S})}, \tag{3.10}$$

where the resulting scalar values with larger magnitude represent observations more likely to be outliers. The denominator of this feature is the same as the product of the eigenvalues of this covariance matrix.

Feature 4: Trace of sigma (model-based)

Using the same procedure as in Feature 3, the estimate of the population error variance/covariance matrix is computed. The trace of this variance/covariance matrix is then taken, which is the sum of the values of the diagonal. This is equivalent to the sum of the eigenvalues. The inverse of the trace is taken, where the resulting scalar values with larger magnitude represent observations more likely to be outliers, defined as

$$\frac{1}{tr(\boldsymbol{S})}. \tag{3.11}$$

Feature 5: Determinant of the correlation matrix (model-based)

As in Feature 3 and Feature 4, the estimate of the population error variance/covariance matrix is first calculated. The correlation matrix is computed from the estimated variance/covariance matrix (this is equivalent to the product of the variance multiplied by the product of the eigenvalues). The inverse determinant is taken, where the resulting scalar values with larger magnitude represent observations more likely to be outliers, yielding the equation

$$\frac{1}{\det(\boldsymbol{R})}. \tag{3.12}$$

Feature 6: Product of variances (model-based)

As in other model-based features (Feature 3 to Feature 5), the estimate of the population error variance/covariance matrix is computed. The variances of this matrix, denoted by σ_i for $i = 1, \dots, p$ comprise the diagonal. The inverse of the product of these diagonals is taken for this statistic, where the resulting scalar values with larger magnitude represent observations more likely to be outliers. The result is defined as

$$\frac{1}{\prod \sigma_i} \; i = 1, \dots, p. \tag{3.13}$$

Feature 7: Determinant of covariance (model-free)

Feature 7 is computed identically as in Feature 3 except with one difference: no AR model is fit. This feature is model free, in that the inverse determinant of the covariance matrix is based exclusively on the raw data. The resulting scalar values with larger magnitude represent observations more likely to be outliers, defined as

$$\frac{1}{\det(\boldsymbol{S}_{data})}. \tag{3.14}$$

Feature 8: Trace of sigma (model-free)

Feature 8 is computed identically as in Feature 4 except the covariance matrix is computed from the raw data on each iteration of the leave one out algorithm. The resulting scalar values with larger magnitude represent observations more likely to be outliers, defined as

$$\frac{1}{\operatorname{tr}(\boldsymbol{S}_{data})}. \tag{3.15}$$

Feature 9: Determinant of the correlation matrix (model-free)

Feature 9 is computed identically as in Feature 5 except the correlation matrix is computed from the raw data on each iteration of the leave one out algorithm. The resulting scalar values with larger magnitude represent observations more likely to be outliers, defined as

$$\frac{1}{\det(\boldsymbol{R}_{\mathrm{data}})}. \tag{3.16}$$

Feature 10: Product of variances (model-free)

Feature 10 is computed identically as in Feature 6 except the covariance matrix is computed from the raw data on each iteration of the leave one out algorithm. The variances of Σ, denoted by σ_i for $i = 1, \dots, p$ comprise the diagonal. The inverse of the product of these diagonals is taken for this statistic. The resulting scalar values with larger magnitude represent observations more likely to be outliers, defined as

$$\frac{1}{\prod \sigma_{i\ data}} \; i = 1, \dots, p. \tag{3.17}$$

Feature 11: Sum of time series magnitude

Take the absolute value of all time series points for each variable, which provides some information about the total magnitude each time series contributes. These magnitudes are summed. The resulting scalar values with larger magnitude represent observations more likely to be outliers: Let $\boldsymbol{y}$ denote a vector of time series variables of length n. Then the sum of the absolute value of all the time series realizations across all variables for a single observation y is defined as

$$\sum abs(y_i)\,, i = 1 \dots p. \tag{3.18}$$

Feature 12: Sum of squared residuals

After fitting a AR(p) model, the residuals are obtained. These are squared and then summed across all variables (in the multivariate case) for a given observation. Larger summed residuals reflect a greater propensity to be an outlier. Let $\boldsymbol{r}$ denote a vector of residuals for all variables after fitting the model. Then the residuals r for each observation with p variables are

$$\sum r_i^2\,, i = 1, \dots, p. \tag{3.19}$$

Feature 13: Literature based Multivariate Least Trimmed Squares (MLTS)

This is the exact result from the non-reweighted MLTS method in the literature (Croux & Joossens, 2008). It should be noted in the simulation studies that the subset of size *h* selected was designed to give the algorithm the best chance at succeeding, namely the exact number of outliers introduced. Normally the true value of this parameter is not known and it is up to the analyst to choose this. So a choice which is quite discrepant from the true number of outliers can lead to incorrect outlier identification. The output is a feature vector of 1's and 0's, where a 1 is an outlier and this set of outliers is based on the subset chosen which yields the smallest determinant.

One reasonable criticism is that the leave one out approach, when applied to a model fit to time series data, might disrupt the pattern of residuals and/or autocorrelation when the observation is removed. However, the effect is inconsequential for two reasons. First, disrupting the ordering of the time series is only done to identify the possible outliers. Corrections of those outliers are done on the original, ordered time series data (which we will see in the next chapter). Second, this disruption happens randomly, with a large number of iterations (500), thereby making this disruption something that is held equal. The idea is to identify outliers or extreme points and, as we will soon see, in spite of the problem that might be introduced by un-ordering the time series, this technique is quite effective. If a strong conclusion were being drawn based on the unordered series alone, that could be problematic. But this is only an intermediate step towards the goal of outlier identification. Finally, and if one felt this procedure of reordering the time series for some features was completely untenable, it should be borne in mind that the features developed in this study are not required to be used for outlier detection. The broader goal is an algorithm that will winnow the best feature from a set of features. As has been shown for some of the 13 features, the leave one out approach that disrupts the ordering of the time series is not necessary for feature construction.

3.5 Feature evaluation studies

These thirteen features were tested with two simulation studies. These studies aim to demonstrate the efficacy of features in identifying outliers and to test if their TPR performance can exceed a raw data based approach such as the Simple Testing Method. To avoid a proliferation of figures and tables if results were presented for each of 15 outlier conditions for each feature for each of these studies, results are only presented for each feature averaged across all 15 conditions and for the maximum value obtained across any one of these 15 conditions. The patterns in these summary level figures mimic the results for the individual outlier conditions, however. (Chapter 2.4 details the evaluation criteria for these studies.)

Multivariate Study (Study A)

Study A presents results from all thirteen features. Non-parametric features have two desirable properties over parametric features. First, they require less knowledge and fewer working assumptions to implement. Second, because a model is not fit, there is an increase in computational efficiency. In outlier detection frameworks where speed is critical, these non-parametric features offer definite advantages. For the results in Study A and Study B, the time series realizations were generated from a model with no covariance and unit variance in the error term (i.e. the variance/covariance matrix has 1's on the diagonal and 0's on the off-diagonal). This simulation study is considered multivariate because the feature was obtained by operating on all variables together.

Figure 3.4 shows the TPR averaged across all 15 conditions for each of the 13 features. Features 7, 8, 9 and 10 are the non-parametric analog of Features 3, 4, 5 and 6. Based on the TPR, the non-parametric features are as effective (or more) at identifying outliers, as compared to the parametric features. Most features have TPR's at .9, though a few features like 2, 5 and 9, do much worse. Figure 3.5 shows the outlier condition on the x-axis (see Table 3.3 for a mapping of these labels to the particular outlier condition on the x-axis) and the maximum TPR achieved for any of the features. This demonstrates the best that one might be able to achieve with this set of features. Outlier conditions 1, 6 and 11 show the worst TPR and these correspond with outlier magnitudes of 1, irrespective of the number of outliers. As the outlier magnitude increases, the TPR also increases. This illustrates how inliers—points within the range of the data—can be difficult to detect. Table 3.4 presents all results from which Figure 3.5 was derived. Rows in that table represent outlier conditions (per Table 3.3) while columns are feature labels. The cells contain the maximum TPR for a given outlier condition and features.

Overall, Feature 11 dominates, with the maximum TPR for almost all outlier conditions. But there are a lot of other features—especially for outlier magnitudes of 3, 4 and 5—which also achieve a TPR of 1.0. And even for smaller magnitudes, Feature 11 is only better by one or two hundredths of a percentage point. So the take home message is that many features do well—and a few like Feature 2 perform quite poorly—but the outlier condition has an important impact in determining how well a feature can ultimately detect outliers.

Univariate Study (*Study B*)

The results in Figures 3.6 and 3.7 are the univariate analog of the multivariate results in Figures 3.4 and 3.5. This means the TPR rate for each feature was computed on each time series individually. Then, to get an overall TPR, the univariate TPRs were averaged into a single TPR. The univariate results show the same pattern as the multivariate result in that non-parametric features (see the TPR of features 7 through 10 in Figure 3.6) yield nearly identical results as the parametric features (see the TPR of features 3 through 6 in Figure 3.6). One important difference between the multivariate and univariate results can be seen in the magnitudes of the TPRs for Figure 3.6 as compared to Figure 3.4. The former is on average about .10 TPR points above the TPR for the latter; but aside from the magnitude difference, the patterns across features in the two figures are consistent. One plausible suggestion for the differential performance lies in how the features aggregate information across the variables and how this is advantageous for the multivariate method. For these simulation studies the outliers appear at the same observation for all variables. In some cases, those outlier magnitudes are more correctly classified as inliers and, as a result, the univariate method has more trouble picking these points out.

But the multivariate method is stronger here because it can leverage information across the variables for a given observation to 'decide' if the data has been contaminated by an outlier. Figure 3.7 is the univariate analog of Figure 3.5 and we see in the former a similar pattern as we do in the latter, but with lower magnitudes. Outlier conditions with outliers of magnitude one have the lowest TPRs and, as the magnitude increases, the TPRs increase to the maximum of 1. But an important difference between Figure 3.5 and Figure 3.7 can be seen in a more gradual 'stair stepping' up to the maximum for the univariate case (Figure 3.7) whereas the rise is much steeper and the leveling off at 1 occurs sooner for the multivariate case (Figure 3.5). For instance, comparing outlier conditions 1 through 5 (5 outliers of magnitude 1, 2, 3, 4 and 5) for the univariate and multivariate analysis, we see values of 0.29, 0.64, 0.92, 1.00 and 1.00 for outlier conditions 1, 2, 3, 4 and 5 for the univariate analysis but see values of 0.58, 0.95, 1.00, 1.00 and 1.00 for the multivariate analysis. Clearly, the information from all variables gives the multivariate analysis a boost. The full set of TPR values underlying Figure 3.7 are in Table 3.5.

3.6 Summary

At the beginning of Chapter 3, the efficacy of the Simple Testing Method (which classifies outliers based on the number of standard deviations from the mean) was evaluated. It performed respectably, but also left a lot to be desired, as it never achieved close to a TPR of 1.0. Features, which were built from some principles of existing outlier detection methods, were constructed and tested in simulation studies to see if TPRs greater than those obtained by the Simple Testing Method could be achieved. Section 3.3 showed that, indeed, many features achieved TPRs much greater than the Simple Testing Method, providing evidence that features might provide a reasonable path forward for outlier detection. Another insight from this chapter is that multivariate approaches have larger TPRs than univariate approaches. For the particular type of outliers introduced—outliers contaminate all variables at the same observation—this makes sense because the multivariate approach can leverage information across the variables to more effectively identify outliers, even in the presence of inliers. Univariate approaches do not benefit from looking at other variables and, especially for inliers, can yield misleading results.

Within the multivariate and univariate results, we see that some features do extremely well while others have very poor performance. Poor performing features include F2, F5 and F9. F2 probably fails because it identifies a further subset (of the *h* observations) which is outlier free. In the simulation studies, *h* was set to the contamination level so it is quite expected that performance would be optimal. The reweighting step for F2 requires comparing the results to a chi-square distribution with a critical value determined by a user specified parameter. The results here are for a single value of this parameter, which resulted in a smaller set of outliers being identified than the original *h*, thus leading to a decrement in performance. One may choose to use different values of this parameter and it could lead to better performance. F5 and F9 are identical in that they represent the determinant of the correlation matrix. In results not included in this thesis, results were obtained for features by first standardizing the variables and this yielded very poor performance for all features. The reasoning is that the standardization procedure removes the variance, which is critical for these features to work effectively. Correlation matrices are standardized versions of the variance/covariance matrix so, from this perspective, it is perhaps not surprising they fail.

However, at this juncture and given the evidence of this chapter, it is evident that not all features are equal. Some are better than others and it seems those that do better effectively leverage information in the covariance matrix. More importantly, how can one winnow a set of features to find those that are best or most able to predict outliers in their data set? In the current chapter we could leverage prior knowledge of the dataset---the so-called ground truth—to assess the efficacy of the features. But if progress is to be made on the larger goal—namely that of developing a general outlier detection algorithm which does not require prior assumptions of the data or outliers—then it will be necessary to develop a method capable of teasing out the effective features from those that are ineffective at identifying outliers. Before addressing that larger goal, Chapter 4 continues the work of Chapter 3 by continuing the feature construction and evaluation process; but Chapter 4 also extends this work by suggesting a framework that might help select features that are better at outlier detection.

3.7 Figures and Tables

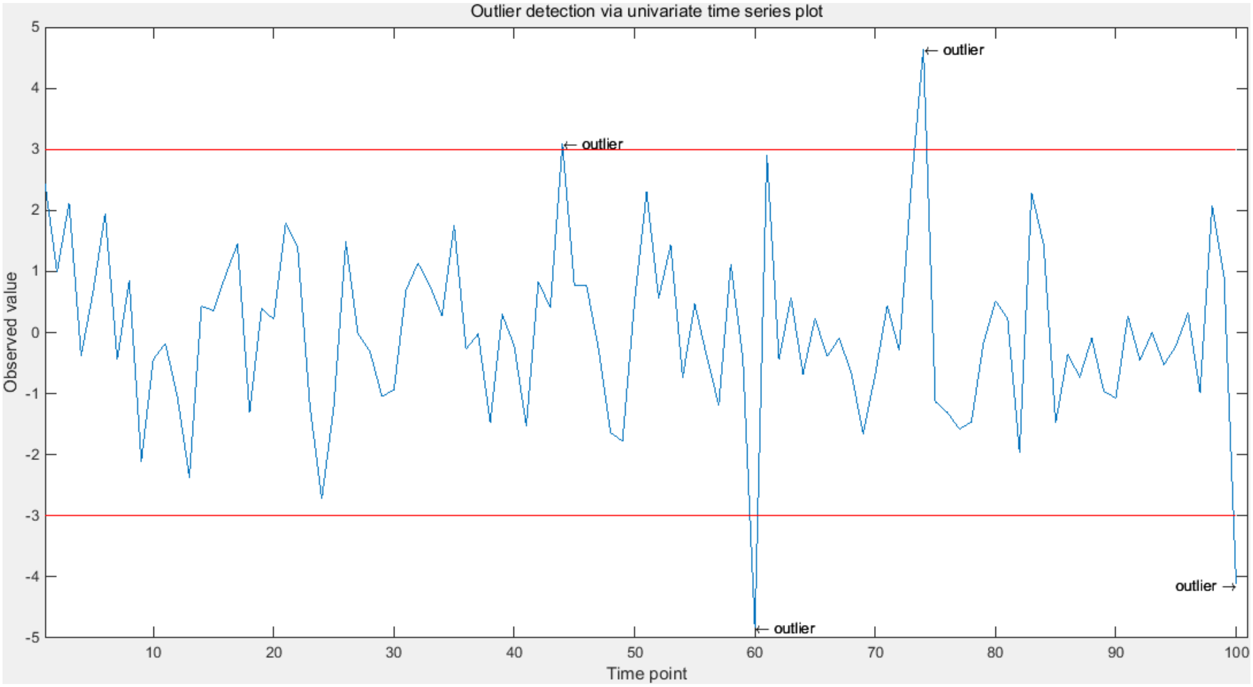


Figure 3.1. Univariate time series with 4 outliers, as defined by a simple heuristic. The x-axis is time and the y-axis is the magnitude of the observation. The red horizontal lines with intercepts at -3 and 3 represent three standard deviations from the mean. Points beyond these red lines are labeled 'outliers' under the 3-sigma outlier detection algorithm (also called the Simple Testing Method in Chapter 3).

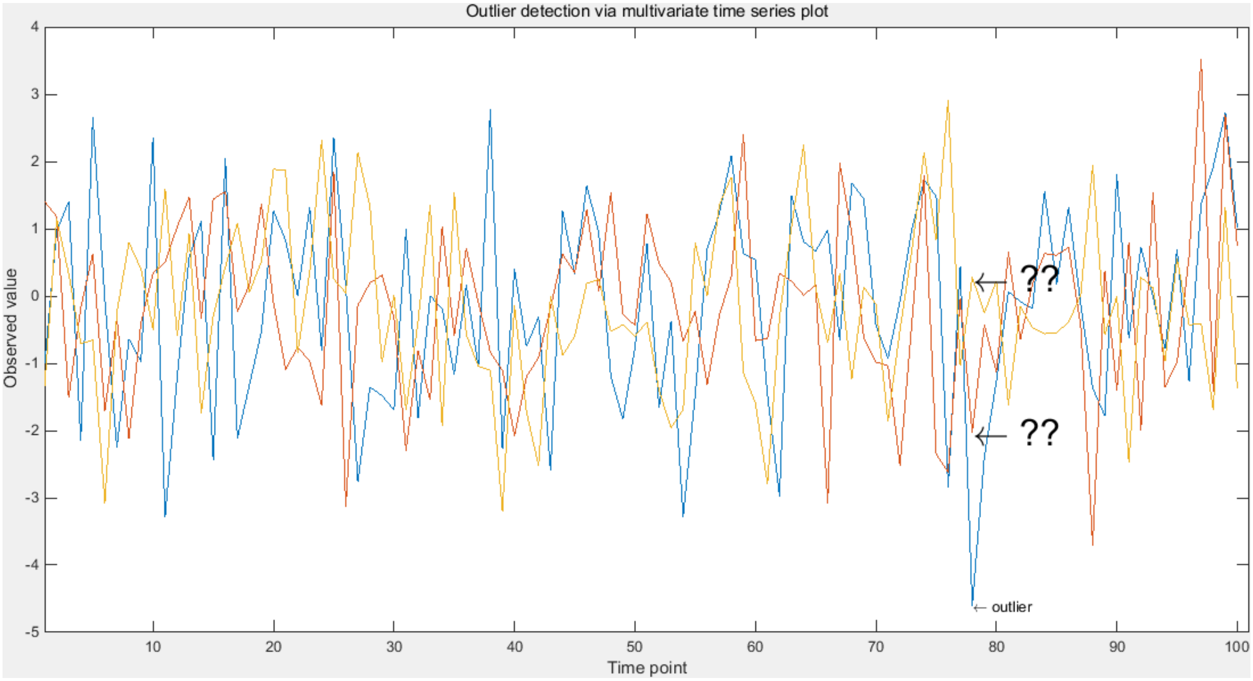


Figure 3.2. Multivariate time series plot with one outlier when adopting the 3-sigma outlier detection rule. The point labeled ‘outlier’ for the blue line exceeds the -3 threshold and is identified as an outlier by the 3-sigma rule. The points for the red and yellow lines at the same observation are labeled ‘??’ in this figure because the 3-sigma outlier rule would not identify these observations as outliers. But if this time series data were generated from 3 sensors measuring the same behavior, for instance, and a disturbance impacted all 3 sensors, then technically all three variables at observation 78 should be classified as outliers.

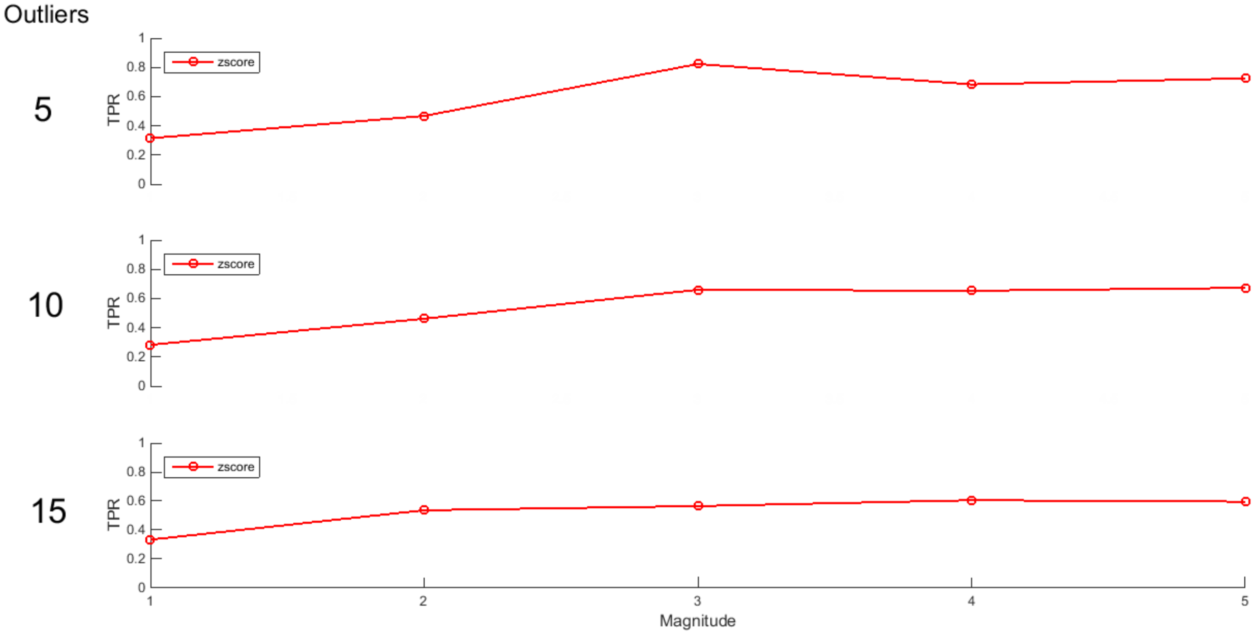


Figure 3.3. TPR for each of 15 outlier conditions from the Simple Testing Method. Red lines represent an average across 25 simulated time series for each of 15 outlier conditions. x-axis is the magnitude of the outlier introduced and they y-axis is the contamination level, 5, 10 or 15 outliers.

Table 3.1. TPR for each of 15 outlier conditions from the Simple Testing Method. Rows of the table represent outlier magnitudes (1, 2, 3, 4 or 5) and columns represent contamination levels (5, 10 or 15).

True Positive Rates			*Contamination*		
			5	*10*	*15*
Magnitude	*1*	3 sigma	0.31	0.28	0.33
	2	3 sigma	0.47	0.46	0.54
	3	3 sigma	0.82	0.66	0.57
	4	3 sigma	0.68	0.65	0.61
	5	3 sigma	0.73	0.67	0.60

Table 3.2. Overview and labeling of 13 Features. The number in the left column represents the feature label number.

	Feature Description
1	Multivariate Least Trimmed Squares
2	Reweighted Multivariate Least Trimmed Squares
3	Model based determinant of covariance matrix
4	Model based trace of covariance matrix (i.e. sum of variances)
5	Model based determinant of correlation matrix
6	Model based product of variances
7	Model free determinant of covariance matrix
8	Model free trace of covariance matrix (i.e. sum of variances)
9	Model free determinant of correlation matrix
10	Model free product of variances
11	Sum of absolute value of time series observations
12	Model based sum of squared residuals
13	Literature based Multivariate Least Trimmed Squares (Agullo, et al. 2008; Croux & Joossens, 2008)

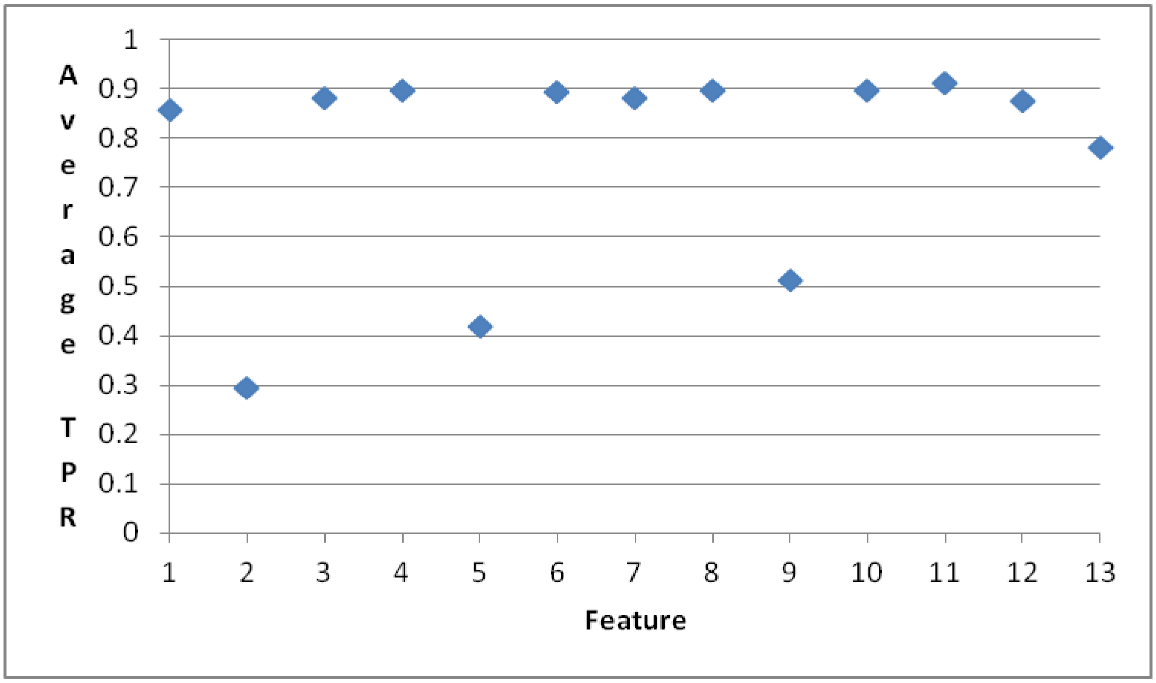


Figure 3.4. Multivariate Study A results. TPR averaged across 15 outlier conditions for 13 features using a multivariate approach.

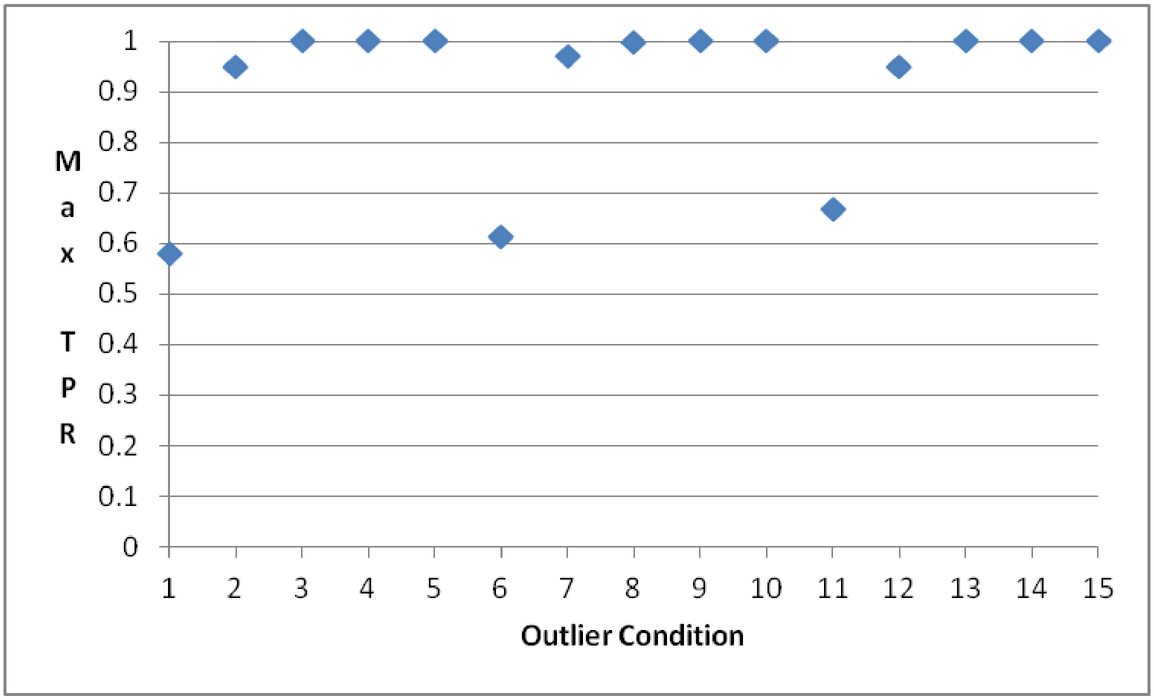


Figure 3.5. Multivariate Study A results. Max TPR for any of 13 features for each outlier condition using a multivariate approach. The x-axis represents the outlier condition in Table 3.3.

Table 3.3. Labels of the 15 outlier conditions in Figures 3.5 and 3.7.

	Outlier Description
1	5 outliers, magnitude 1
2	5 outliers, magnitude 2
3	5 outliers, magnitude 3
4	5 outliers, magnitude 4
5	5 outliers, magnitude 5
6	10 outliers, magnitude 1
7	10 outliers, magnitude 2
8	10 outliers, magnitude 3
9	10 outliers, magnitude 4
10	10 outliers, magnitude 5
11	15 outliers, magnitude 1
12	15 outliers, magnitude 2
13	15 outliers, magnitude 3
14	15 outliers, magnitude 4
15	15 outliers, magnitude 5

Table 3.4. Multivariate study A maximum TPRs (cell entries) for each outlier condition (rows, as defined in Table 3.3) and each feature (columns). The maximum of each row is highlighted in bold and italicized.

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13
5-1	0.39	0.23	0.54	0.55	0.20	0.54	0.50	0.55	0.29	0.55	***0.58***	0.46	0.31
5-2	0.87	0.40	0.94	0.95	0.46	0.94	0.91	0.91	0.62	0.91	***0.95***	0.91	0.78
5-3	0.98	0.20	0.97	0.98	0.46	0.98	0.98	***1.00***	0.60	***1.00***	***1.00***	0.98	0.94
5-4	0.99	0.19	0.99	0.99	0.54	0.99	***1.00***	***1.00***	0.68	***1.00***	***1.00***	0.99	0.97
5-5	0.98	0.13	0.99	***1.00***	0.52	***1.00***	0.99	***1.00***	0.59	***1.00***	***1.00***	***1.00***	0.99
10-1	0.52	0.31	0.58	***0.61***	0.30	0.60	0.55	0.53	0.42	0.53	0.59	0.50	0.27
10-2	0.87	0.51	0.89	0.93	0.38	0.93	0.89	0.94	0.49	0.95	***0.97***	0.92	0.76
10-3	0.97	0.29	0.97	0.99	0.43	0.99	0.97	0.99	0.49	0.99	***1.00***	0.98	0.93
10-4	***1.00***	0.33	0.99	0.99	0.44	0.99	0.99	***1.00***	0.49	***1.00***	***1.00***	0.99	0.96
10-5	***1.00***	0.16	***1.00***	***1.00***	0.47	***1.00***	***1.00***	***1.00***	0.53	***1.00***	***1.00***	***1.00***	0.99
15-1	0.51	0.31	0.58	0.61	0.31	0.60	0.60	0.61	0.45	0.62	***0.67***	0.56	0.27
15-2	0.87	0.43	0.88	0.91	0.37	0.91	0.91	0.93	0.47	0.94	***0.95***	0.90	0.72
15-3	0.94	0.53	0.93	0.96	0.38	0.96	0.95	***1.00***	0.47	***1.00***	***1.00***	0.96	0.89
15-4	0.99	0.30	0.99	***1.00***	0.51	***1.00***	0.99	***1.00***	0.57	***1.00***	***1.00***	***1.00***	0.97
15-5	***1.00***	0.08	0.98	0.99	0.51	0.99	***1.00***	***1.00***	0.53	***1.00***	***1.00***	***1.00***	0.99

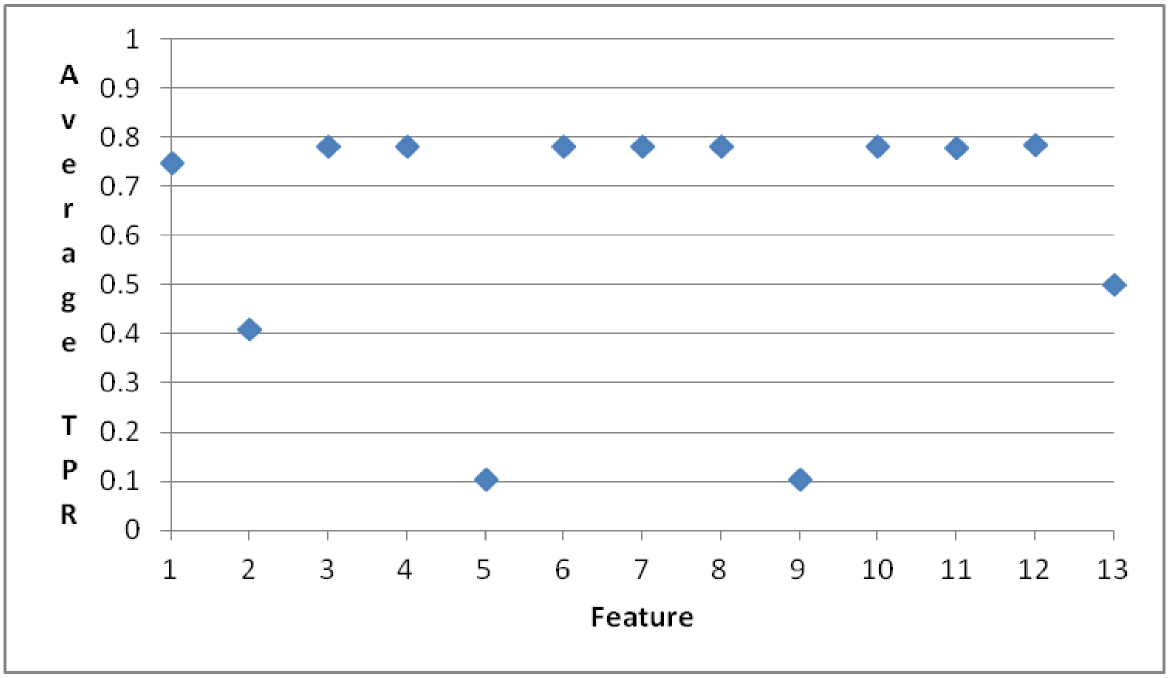


Figure 3.6. Univariate Study B results. TPR averaged across 15 outlier conditions for 13 features using a univariate approach.

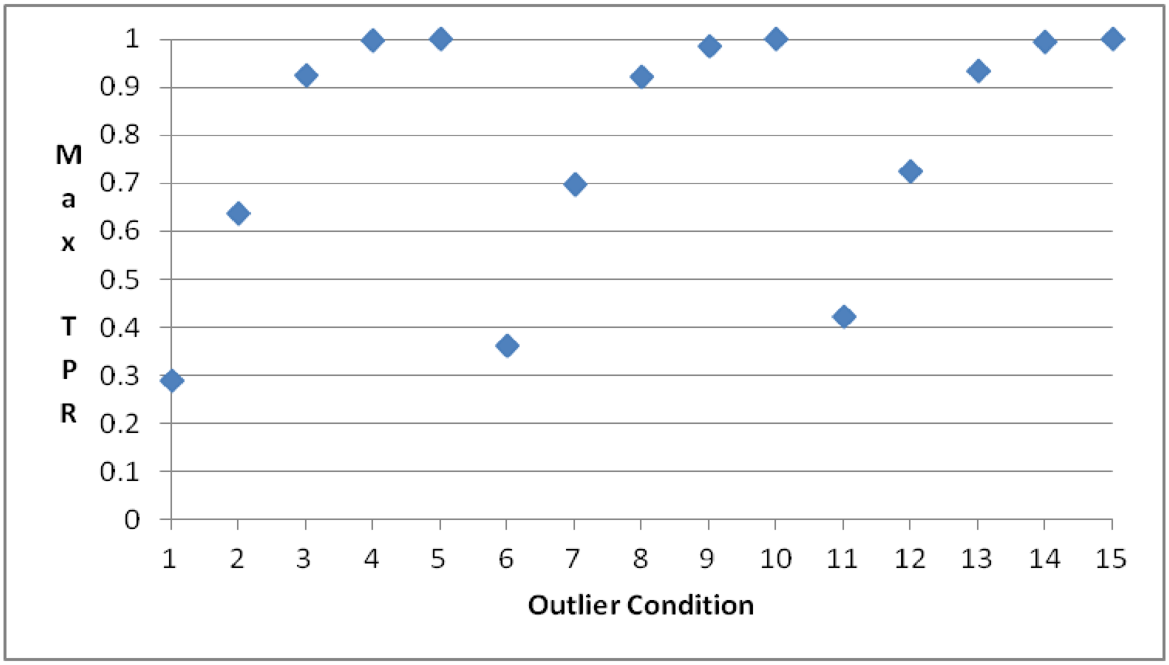


Figure 3.7. Univariate Study B results. Max TPR for any of 13 features for each outlier condition using a univariate approach. The x-axis represents the outlier condition in Table 3.3.

Table 3.5. Univariate study B maximum TPRs (cell entries) for each outlier condition (rows, as defined in Table 3.3) and each feature (columns). The maximum for each row is highlighted in bold and italicized.

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0.25	0.28	***0.29***	***0.29***	0.05	***0.29***	0.28	0.28	0.05	0.28	0.25	0.28	0.05
2	0.50	0.39	***0.64***	***0.64***	0.06	***0.64***	0.61	0.61	0.06	0.61	0.60	0.63	0.26
3	0.82	0.43	***0.92***	***0.92***	0.07	***0.92***	0.91	0.91	0.07	0.91	0.90	***0.92***	0.63
4	0.95	0.19	0.98	0.98	0.02	0.98	0.99	0.99	0.02	0.99	***1.00***	0.99	0.93
5	0.99	0.13	0.99	0.99	0.03	0.99	***1.00***	***1.00***	0.03	***1.00***	***1.00***	***1.00***	0.99
6	0.33	0.29	***0.36***	***0.36***	0.07	***0.36***	0.34	0.34	0.07	0.34	0.34	0.35	0.06
7	0.64	0.51	0.69	0.69	0.09	0.69	***0.70***	***0.70***	0.09	***0.70***	0.67	0.68	0.22
8	0.84	0.57	0.90	0.90	0.11	0.90	0.91	0.91	0.11	0.91	***0.92***	0.91	0.47
9	0.96	0.50	0.97	0.97	0.13	0.97	0.98	0.98	0.13	0.98	***0.99***	0.98	0.79
10	***1.00***	0.20	0.99	0.99	0.11	0.99	***1.00***	***1.00***	0.11	***1.00***	***1.00***	***1.00***	0.96
11	0.39	0.38	***0.42***	***0.42***	0.13	***0.42***	***0.42***	***0.42***	0.13	***0.42***	***0.42***	0.41	0.04
12	0.69	0.56	0.72	0.72	0.19	0.72	***0.73***	***0.73***	0.19	***0.73***	0.71	0.71	0.15
13	0.89	0.66	0.91	0.91	0.16	0.91	0.92	0.92	0.16	0.92	***0.94***	0.92	0.34
14	0.98	0.65	0.98	0.98	0.16	0.98	0.98	0.98	0.16	0.98	***0.99***	***0.99***	0.72
15	***1.00***	0.41	0.99	0.99	0.16	0.99	***1.00***	***1.00***	0.16	***1.00***	***1.00***	***1.00***	0.92

4. Voronoi Diagram Outlier Detection

This chapter builds on the success of Chapter 3 by aggregating individual features. In Section 4.1, Voronoi diagrams are considered. Section 4.2 generalizes the work of Section 4.1. These approaches are nonparametric, straightforward to implement and computationally efficient. The goal is to test whether even greater TPRs can be achieved than with individual features, which themselves represented an important advance over the Simple Testing Method.

4.1 Multivariate Voronoi Outlier Detection (MVOD)

A Voronoi outlier detection algorithm was reviewed in Chapter 2; but this method was only designed for univariate time series. The univariate method was extended to multivariate time series and generalized in recent work (Zwilling & Wang, 2014). This extension is briefly reviewed here, with the results presented.

The Multivariate Voronoi Outlier Detection (MVOD) method is based upon Voronoi nearest neighbors. For a point p_i of set S, the nearest neighbors of p_i defined by the Voronoi polygon $V(p_i)$ are the Voronoi nearest neighbor of p_i, denoted as $V_{NN}(p_i)$. In Figure 2.2 the nearest Voronoi neighbors to point p_1 are p_2, p_3, p_4, p_5 and p_6. For each point in the data set, the MVOD uses the nearest neighbors to compute a Voronoi outlier index of how likely that point is an outlier. It is multivariate because it aggregates information across all individual time series, thus retaining features which might be common to the entire interlocking set of variables.

The method is based upon the geometric principles of Voronoi diagrams for defining the neighborhood relationship of the data points and this facilitates the assignment of outliers and non-outliers. Construction of a two dimensional Voronoi diagram requires two coordinates for each data point; but Voronoi diagrams can have as many dimensions as desired. The present work only considered 2-dimensional Voronoi spaces. Figure 4.1 overviews the process used by Zwilling and Wang (2014).

The 2-dimensional vector fed into the Voronoi diagram had 2 features. The feature value for the x-coordinate in Figure 4.1 was the same as Feature 1. The feature value for the y-coordinate in Figure 4.1 was computed by multiplying Feature 11 (sum of the absolute value of the time series) and Feature 12 (the sum of the residuals after fitting a MVAR model) together, both of which were already described in Chapter 3. These x- and y-coordinates were fed into a Voronoi diagram from which a Voronoi Outlier Index (VOInd) was computed for each time series observation. The VOInd for point p_i has as its numerator the sum of the Euclidean distance (*dist*) between each point and all its neighbors. This is divided by the denominator term which is the number of neighbors, yielding an average density

$$VOInd(p_i) = \frac{\sum_{O \in V_{NN}(p_i)} dist(p_i, o)}{|V_{NN}(p_i)|}. \tag{4.1}$$

Results of this method (Zwilling & Wang, 2014) are displayed in Figure 4.2 and Table 4.1. These results compare the MVOD method with the popular MLTS method, showing a clear advantage of the MVOD method when the magnitude of outliers is small, which is the most difficult case to identify outliers. For 5 outliers of magnitude 1, the MVOD has a TPR of 0.52 whereas the MLTS is 0.21. For 5 outliers of magnitude 2, the MVOD has a TPR of 0.91 and the MLTS has a TPR of 0.63. The same pattern is true for 10 and 15 outliers with magnitudes 1 and 2. The TPR of the two methods track similarly for magnitudes of 3, 4 and 5 for all outlier conditions.

4.2 MVOD Extension

The MVOD developed in Zwilling and Wang (2014) was applied to all features. Since there are 13 features, there are theoretically 78 (13 C 2=78) unique feature pairs which can serve as input coordinates for the 2-dimensions Voronoi diagram. Voronoi diagrams will not yield unique solutions under certain cases, such as when there are degenerate point sets—at least the algorithm implemented in Matlab. In the features constructed for outlier detection, there were two features which yielded redundancy, preventing the algorithm from finding a unique solution for all points. Features 2 and 13 were dropped here, leaving 11 features. A Voronoi Outlier Index was computed for these 55 pairs of input coordinates (11 C 2=55). The results from just these 55 pairings are discussed now.

Figure 4.3 and Figure 4.4 contains the results from the feature pairs that were input as x- and y-coordinates into the MVOD method. Figure 4.3 presents the maximum TPR (for all 15 outlier conditions) while Figure 4.4 presents the average TPR (across the 15 outlier conditions); the full set of tabled results from which these figures were created can be found in the Appendix (Table A.1). In both figures there are several input coordinates which do quite poorly: 8-9, 8-10, 8-11, 8-12, 9-10, 9-11, 9-12, 10-11, 10-12 and 11-12 all have TPRs less than 0.10, no matter whether considering the average or maximum. But there are also features which have TPRs close to 1 (for the maximum) and values around 0.8 for the average. Overall, especially in comparison to the results of the features by themselves, the Voronoi diagram results here may or may not offer many gains. However, the Voronoi diagram is capable of testing multidimensional spaces. Additionally, the results of the features tested here might be different for other data sets.

4.3 Summary

At this point in analyzing the role of features in outlier detection, one could proceed laterally or vertically. The lateral path means that one could search hundreds or thousands or millions or billions of individual features or pairs of features or triples and find the best one at predicting outliers. The vertical approach would say, of features that have been defined, can we theoretically motivate why that feature does well and why it might be the best universal outlier algorithm?

An honest reading of the literature will reveal that outlier detection research has yielded a multitude of unique methods that work better (or worse) depending on the data situation. Given this reality, it is probably unlikely that one could identify a single best outlier detection rule, even if that was the goal. So perhaps a smarter approach is to develop a method that screens a candidate set of features and finds the best feature that is most optimized for identifying outliers for the particular data set at hand. Ideally, this procedure can be applied to many other novel situations and is not constrained to special properties of the data. This is the approach that is developed in the next chapter.

4.4 Figures and Table

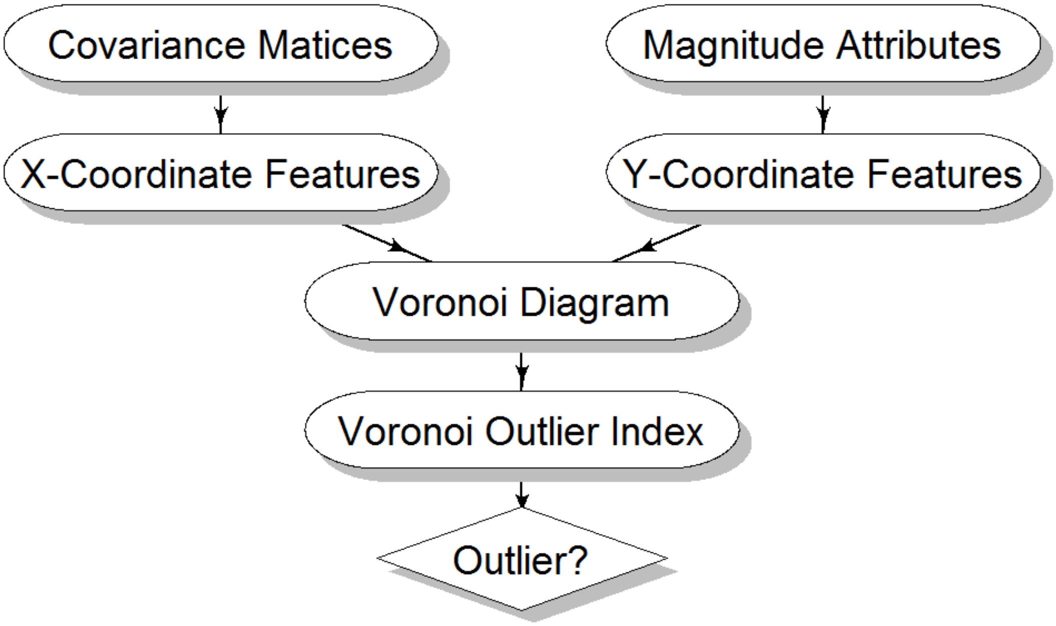


Figure 4.1. Flowchart of information processing steps for 2-dimensional Voronoi outlier detection.

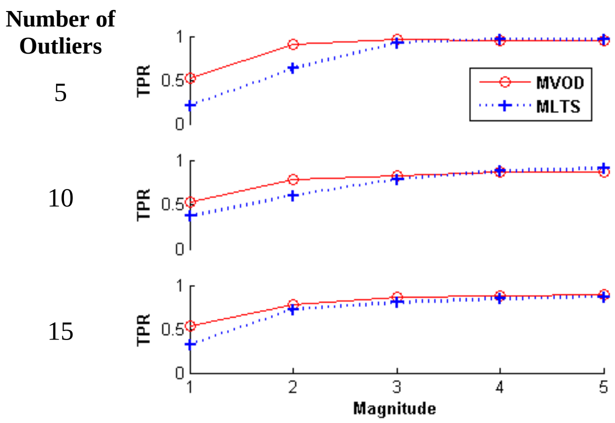


Figure 4.2. True positive rate (TPR, y-axis) for MVOD and MLTS for 5 outliers (top panel), 10 outliers (middle panel) and 15 outliers (bottom panel) with outlier magnitudes of 1, 2, 3, 4, or 5 (x-axis).

Table 4.1. True and false positive rates for the MVOD and MLTS methods with 5, 10 or 15 outliers of magnitude 1, 2, 3, 4 or 5.

			True Positive Rate			**False Positive Rate**		
			Number of Outliers					
			5	*10*	*15*	*5*	*10*	*15*
Magnitude	*1*	MVOD	0.52	0.52	0.54	0.037	0.065	0.094
		MLTS	0.21	0.37	0.32	0.012	0.028	0.047
	2	MVOD	0.91	0.79	0.78	0.025	0.041	0.056
		MLTS	0.63	0.61	0.73	0.002	0.012	0.011
	3	MVOD	0.96	0.83	0.86	0.023	0.037	0.045
		MLTS	0.93	0.78	0.80	0.004	0.006	0.005
	4	MVOD	0.96	0.86	0.88	0.023	0.034	0.042
		MLTS	0.97	0.87	0.85	0.002	0.002	0.003
	5	MVOD	0.96	0.86	0.90	0.023	0.034	0.039
		MLTS	0.96	0.90	0.87	0.002	0.002	0.002

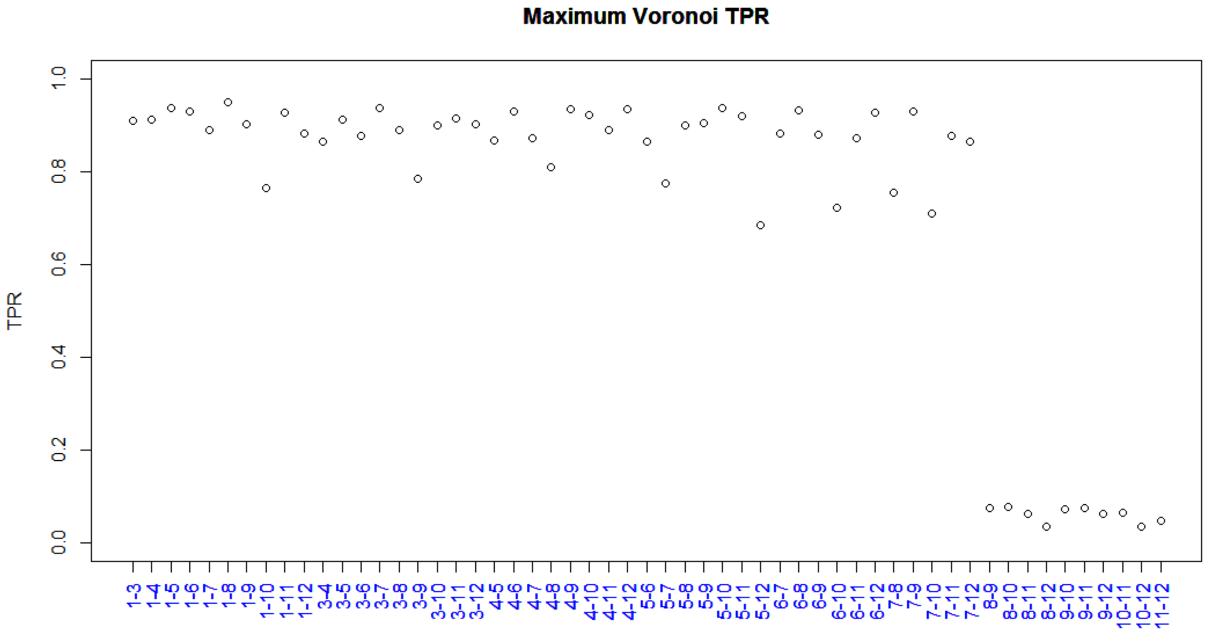


Figure 4.3. Maximum TPR (of any of 15 outlier conditions) of all 2-dimensional input coordinates to Voronoi diagram. y-axis is accuracy and x-axis is the label of features, where the number preceding the dash is the first feature and the number after the dash is the second feature.

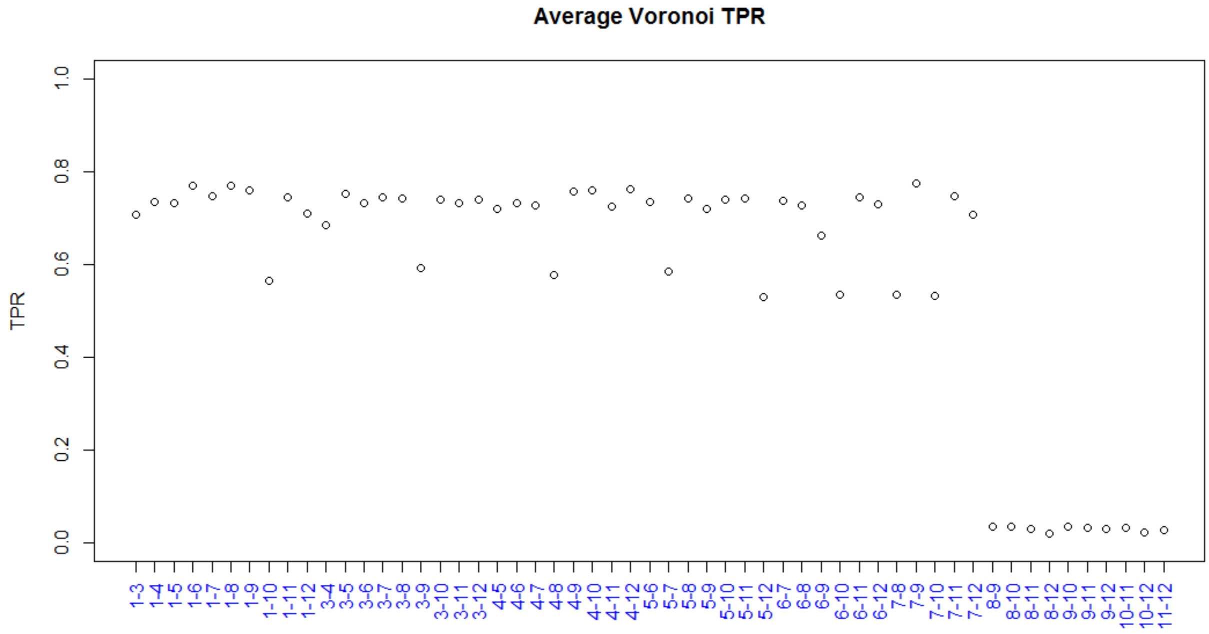


Figure 4.4. Average TPR (across all 15 outlier conditions) of all 2-dimensional input coordinates to Voronoi diagram. y-axis is accuracy and x-axis is the label of features, where the number preceding the dash is the first feature and the number after the dash is the second feature.

5. Covariance Based Outlier Detection

In searching for outliers in real datasets, one rarely has the luxury of knowing which observations are outliers and which are not. Indeed, if this information is known, then using an outlier detection algorithm becomes superfluous. Hence, using a simulation approach and looking at the TPR is good in a simulation setting; but this has very limited practical utility. One shortcoming addressed in the literature review of this thesis, and a criticism leveled against many existing outlier detection methods, is that most outlier detection algorithms require users to input a parameter value that reflects their belief about the number and/or nature of the outliers in the data. But this logic is circular: if one knew about the outliers in their data, why use an outlier detection algorithm? Granted, there are cases where on might have good guesses about the number of outliers in their data. But even in those cases, one is not precluded from using this method. It provides a basis for comparing the algorithm results to the actual results. But, as a further qualification, even if one knows the number of outliers, they still may not know their location so an algorithm like this could still be of utility.

This chapter presents a covariance based outlier detection algorithm that selects from a candidate set of feature vectors those that are best at identifying outliers. While this chapter only considers the 13 features introduced in Chapter 3, there are no restrictions on the number of features that can be tested. An important challenge for an algorithm operating on a set of features is for it to winnow the effective features from the ineffective features. The algorithm leverages covariance information from the feature vectors to identify those that are best at outlier identification. Covariance matrices communicate variability and outliers show different patterns of variability than the normal data. So when this variability is examined over a time stream, it is possible to identify which observations are signal and which are noise. The work in this chapter demonstrates a method that accomplishes this challenging but important goal (Zwilling and Wang, 2014, 2015).

5.1 Algorithm description

The covariance based outlier detection algorithm, diagrammatically shown in Figure 5.1, generalizes and extends the multivariate Voronoi outlier detection approach (Zwilling & Wang, 2014 and Chapter 4 of this work) through a powerful feature selection procedure. Each of these steps is now discussed in more detail.

Step 1 — Feature extraction: Features have the capacity to be more informative than the data itself (Liu & Motoda, 2013). Without loss of generality, suppose the original data are multivariate time series with *n* observations and *p* columns of variables. Univariate feature vectors of length *n* can be computed using the multivariate data.

The 13 feature vectors are reviewed again in Table 5.1. Features 3, 4, 5 and 6 require first fitting a parametric autoregressive (AR) time series model whereas Features 7, 8, 9 and 10 are the model free analogues that operate just on the raw data. Features 11 and 12 are closely related to the time series data. Features 1, 2 and 13 are binary indicator vectors derived based on the Multivariate Least Trimmed Squares estimator (Agullo et al., 2008), an important classical statistical method for outlier detection. All features except F11, F12, and F13 implement a leave one out, or jackknife, approach. For a given feature, its statistic is first computed on all data except the first observation. The first observation is added back, the second observation is removed and the statistic is computed again. This process is repeated for all observations, yielding a *n* x 1 vector for each feature. If an observation is influential (i.e. outlier), removing it will lead to a less extreme feature value than leaving the observation in the data. At each time point, the features are calculated based on the descriptions in Chapter 3.

Step 2 — Order statistics computation: For each feature vector, order statistics are computed. The sorting operation happens on the feature vector, so the maximum value is listed first and the minimum value is last. The observations corresponding to each feature value are shuffled according to the order statistics of the feature value. Once the order statistics are computed for all feature vectors, the order sorted feature vectors now encode outlier predictions. The largest feature value is most likely to be an outlier. Two features can theoretically have different statistical or mathematical underpinnings but could still make identical predictions. These features are redundant. Steps 3, 4 and 5 proceed iteratively.

Step 3 — Fixing outliers in order: For a given feature vector, the observation under consideration that is predicted to be an outlier is corrected by interpolating with the adjacent observations in the un-sorted data. Interpolation is not the only or required procedure for fixing the outlier. One could use multiple imputation, for instance, and treat it as missing data. Or there are many Bayesian algorithms which would allow one to generate a distribution of what the values should look like. These, and other approaches, would be equally acceptable for this step.

Step 4 — Log ratio of covariance determinants: After the predicted outlier has been corrected through interpolation, the determinant of the covariance matrix of the data is computed. A determinant can be geometrically interpreted as a volume, where a larger relative volume reflects data with more extreme values. A log of the ratio between the current determinant of the covariance matrix and the determinant of the covariance matrix from the 1-step back interpolation is computed. The determinant of the covariance matrix at each step of iteration, and across all iterations, will obtain values unique to the data set at hand. For instance, a time series with outliers of larger magnitudes will initially have a determinant of the covariance matrix that is larger than the same time series with outliers that are smaller in magnitude. Similarly, even in the case where no outliers are present, different sequences of time series will yield determinants of the covariance matrix with different values. So relying on the value alone of individual determinant of a covariance matrix will not prove general enough for an outlier detection algorithm that can adapt to the data at hand. What is required is a measure which does not depend on the dataset at hand, or the number and size of outliers. The log ratio of two consecutive determinants of the covariance matrices is a measure that does not depend on the data and it measures the rate of change. If this ratio is unchanging (up to some small tolerance), this suggests no further outliers are present. But as long as the ratio is decreasing, this suggests the feature is identifying outliers.

Step 5 — Convergence check: After each interpolation, and computing the log ratio of the covariance determinant described in Step 4, convergence is checked. If the log ratio of the determinant approaches 0 with some small tolerance like .05, the feature has identified all of its predicted outliers. If the log ratio of the determinant is not close to within a tolerance close to 0, then the algorithm repeats steps 3, 4 and 5. The number of iterations to reach convergence (excluding the current iteration) determines the number of outliers predicted by that feature.

Step 6 — Outlier detection with feature selection: Steps one through five are applied to each feature individually. One will typically have a set of features for detecting outliers and so it will be necessary to identify the best feature(s). The plot of the determinant of the covariance for each feature provides visual and analytical evidence towards this goal when compared on the same data

Step 6a – Convergence plot: Plotting the convergence (i.e. the determinant of the covariance after fixing the candidate outlier) for all features across all iterations will yield different patterns. As we will soon see with the results, certain patterns are more reflective of good features than poor features.

Step 6b – Sum of area under convergence plot: The best feature can be identified by a single value derived analytically from the convergence plot: the sum of the area under the curve.

5.2 Algorithm illustration

Before looking at the experimental results, this section provides more details of the algorithms inner workings by way of an example. Table 5.2 has features as columns. The first two rows represent the value of the determinant of the time series before any outliers were added and the second row represents the determinant once all outliers were added. Starting with the 3rd row, the values in each cell represent the determinant of the covariance matrix from the time series after each interpolation. The first row, labeled 'None' in Table 5.2, is the determinant of the covariance matrix of the time series before any outliers were added. The second row, labeled 'All' in Table 5.2, is the initial value of the covariance of the time series with outliers added but before interpolation. The row labeled '1' represents the first interpolation step. Different features yield different results down each column. An important trend that emerges for these features is the quick decline in magnitude of the values in each cell when reading down a column, at least up to a point. For instance, feature 1, labeled column '1', has an initial value of 17.87. After the 1st 'outlier' identified by feature 1 is corrected, the determinant drops to 11.18. After the 2nd 'outlier' is corrected, the determinant drops to 6.202. Continuing down the column, we see that the magnitudes begin to level off starting with the row labeled '6'. Indeed, this should be expected for a good feature. So this provides evidence that there are outliers and how many outliers exist in the data set, without requiring special input or knowledge about the data.

Once the algorithm has computed the table of values for the convergence check as illustrated in Table 5.2, the log ratio of adjacent points in each column of Table 5.2 is taken by moving down the column one cell at a time. Doing this for rows 2 through 20 of Table 5.2 yields Table 5.3, which is ***Step 5***.

In Table 5.3 it is apparent that all of these features do a decent job of identifying that outliers exist, as well as the exact number of outliers. The very last row in the Table 5.3 indicated the number of observations flagged as an outlier for that feature. From Table 5.3 however, it is not known whether the outliers identified are actually correct. So in order to validate these results, the experimental results presented next will show how the algorithm works on a simulation study, along with an assessment of the true positive rate to determine if the algorithm is accurate.

5.3 Experimental results

25 multivariate time series data sets of 5 variables and 100 observations were generated for each of the 15 outlier conditions. The 15 conditions were all combinations of 5, 10 or 15 outliers with magnitudes of 1, 2, 3, 4 or 5. Each multivariate time series was simulated from an AR(2) process with standard normal Gaussian noise (see Chapter 2 for further details). For each outlier condition and for each feature, a receiver operating characteristic (ROC) curve was constructed by using convergence thresholds ranging from 0 to 1, with a step size of .01. Keep in mind that this ROC curve would not be available to a researcher who did not have prior knowledge of the number of outliers, as this plot was constructed with that information. This plot is one way to establish the validity of the features, independent of their feature performance through the convergence plots.

Table 5.4 presents a summary of these ROC results. The entries of Table 5.4 were computed by taking the maximum and average of the ratio of the true positive rate (TPR) divided by the false positive rate (FPR). Larger values are better. Within a condition, we see variability across features. For instance, F2 has a max of 26 whereas F1 has a max of 1097. We also see variability within a feature, as we consider 5, 10 or 15 outliers. Generally speaking, good features will have large values within a column, relative to other columns and demonstrate consistency across outlier conditions. Magnitudes differ within a column because different outlier conditions have differing levels of detection difficulty, for example, it is much easier to identify 10 outliers with magnitude

5 as compared to 5 outliers of magnitude 1. Tables A.2, A.3 and A.4 in the Appendix provide further evidence demonstrating the difficulty most features have in identifying outliers of magnitude 1, no matter whether there are 5, 10 or 15 outliers in the data.

Figure 5.2 plots the log of the results in Table 5.4, but adds two lines for the overall average across all 15 outlier conditions for the maximum and average. The lines representing the max values always have larger values than the lines representing the averages; but generally the max and average track similarly for all feature vectors. Features with larger values (whether the maximum or average) are better at detecting outliers—like F1 and F3—whereas features with smaller values—like F2 and F11—do a poor job of identifying outliers. This figure also shows that it is more difficult for any feature—good or poor—to identify more outliers. We see that higher maximum or average values were obtained for 5 outliers and smaller values for the 15 outlier condition.

Now we transition to results that would be available to researchers if they were using the algorithm in practice and trying to determine the number of outliers in the data. Figure 5.3 shows the number of outliers identified by each feature vector for 5, 10 and 15 outlier conditions. If a feature predicted 5 outliers, then it should have its plot symbol at 5. For F4 and F6, we see accurate predictions of 5, 10 or 15 outliers for each of those conditions. But F2 fails for instance because it predicts 5 outliers (or fewer) for all 3 outlier conditions. In Figure 5.3 the lines are averaged for all 5, 10 and 15 outlier conditions. Figure 5.3 is based on a convergence threshold of a log ratio of .05, meaning the number of outliers identified by each feature was determined once the value of the log ratio between two adjacent values drops below .05. Zero is the theoretical convergence of the log ratio and one could construct many such graphs as Figure 5.3, where each figure would be based on a different convergence threshold. For instance, one could construct a figure similar to Figure 5.3, except use a log ratio threshold of .04. For good features, the choice of a threshold does not seem to have important consequences for the number of outliers detected but for poor features this value can make a difference. The best recommendation is to pick a threshold based on the location of the sharpest bend in the convergence plot, which is Figure 5.4.

Figure 5.4 shows the covariance of the determinant at each iteration for four features only: F1, F2, F5 and F6. Four features were selected to allow for better visual discrimination; the full set of analogous results to Figure 5.4 for all 13 features is included in the Appendix. These features also showcase the range of variability for poor versus good feature vectors. Figure 5.5 shows complimentary information in the form of the value of the integral, which is derived from the convergence plot of Figure 5.4. These two figures provide consistent, yet different, information.

In Figure 5.4, good features like F1 and F6 reach a bend quickly and level off. F1 has a pattern of convergence most like the ideal ‘L’, because it demonstrates a negative slope fastest and has the sharpest bend and, once it levels off, maintains a horizontal line. Poor features, like F2 and F5 have a different pattern. F2 does not drop as quickly and only reaches the floor about halfway through the dataset, which would indicate that feature predicted half the observations as outliers. F5 seems to start as a good feature because it drops off fairly quickly; but notice that it levels off at a higher point on the y-axis than F1 and F6, which implies it did not identify all the outlier points.

Figure 5.5 is a bar chart which displays the value of the integral (averaged, across all 15 outlier conditions) and these results confirm what we see in the convergence plot in Figure 5.4. F1 has the smallest value (237) while F2 has the largest value (938), which means it is the worst feature. F5 has a value of 521 while F6 has a value of 258, making it 2nd best. Again, the information in Figure 5.4 and Figure 5.5 would be available to a researcher trying to identify outliers in the data and the result does not depend on knowledge of the outliers beforehand.

Figure 5.6 is the ROC plot for the four features. This ROC plot would not be a diagnostic tool for a researcher using this algorithm. Figure 5.6 is a different way of presenting the same results as in Figure 5.2 and Table 5.4 (at least for the four features presented). The good features (F1 and F6) have high TPRs whereas the poor features (F2 and F5) have low TPRs when they are compared at the same FPRs. F6 performs well most likely due to the leveraged covariance information in the data. F5 is a standardized version of the covariance matrix. For outlier detection, variance may be critical. Suppressing it makes the feature unlikely or unable to predict outliers effectively.

The pattern of results in Figure 5.4, Figure 5.5 and Figure 5.6 demonstrates that the covariance based method can simultaneously select a good set of candidate feature vectors and, from that candidate or effective set, accurately predict outliers.

Figure B.1, Figure B.2, Figure B.3 and Figure B.4 in the Appendix each have three panels. One figure shows the covariance of the determinant. Another panel shows the corresponding ROC curves. And the final panel shows the value of the integral for the determinant of the covariance across all observations/iterations. These figures present the results for 5, 10 or 15 outliers for all 13 features for outlier magnitudes of 1, 2, 3, 4 and 5. (Also see Comment A.1 preceding those figures in the appendix for a general description of those figures.) A subset of these results from the Appendix were presented in the current chapter to highlight the efficacy of the method (and reduce visual clutter); but the pattern of results and conclusions drawn for the features examined in Figure 5.4, Figure 5.5 and Figure 5.6 apply to all 13 features in the Appendix.

5.4 Algorithm usage

Having seen the performance of the covariance based outlier detection algorithm on simulated data, further details of its usage in practice are now offered.

Step 1 — Feature extraction: The set of candidate features are proposed or constructed. Each feature vector is univariate and its length must match the number of observations of the original data. Additionally, the values of the feature vector must make unique ordinal predictions in accordance with the extent that a given observation is an outlier. Redundant feature values mean those corresponding time series observations are equally likely to be an outlier. As a general rule, better features will make unique predictions

Step 2 — Order statistics computation: For each feature vector, order statistics are computed. A small or large magnitude could encode observations that are more likely to be outliers. This decision needs to be made by the user on a per feature basis, in accordance with the properties of each feature selected. But even if one was not sure of whether smaller or larger values were more predictive of outliers, they could create two features, one which ordered from small to large and a second feature which ordered large to small. Two features can theoretically have different statistical or mathematical underpinnings but could still make identical predictions. These features are redundant.

Step 3 — Fixing outliers in order: For a given feature vector, the observation under consideration that is predicted to be an outlier is corrected by interpolating with the adjacent observations in the un-sorted time series data. Alternative interpolation schemes could also be implemented, such as using two observations ahead and behind from the current one, which may lead to more data smoothing effect.

Step 4 — Log ratio of covariance determinants: After the predicted outlier has been corrected through interpolation, the determinant of the covariance matrix of the data is computed. A log of the ratio between the current determinant and the determinant from the 1-step back interpolation is computed. This log ratio represents a rate of change.

Step 5 — Convergence check: After each interpolation, and computing the log ratio of the covariance determinant described in Step 4, convergence is checked. If the log ratio of the determinant approaches 0 with some small tolerance, such as .05, the feature has identified all of its predicted outliers. If the log ratio of the determinant is not close to 0, then the algorithm repeats steps 3, 4 and 5. The number of iterations to reach convergence (excluding the current iteration) determines the number of outliers predicted by that feature.

Step 6 — Outlier detection with feature selection: Steps one through five are applied to each feature individually. Ideally, in most cases, one will have a set of features for detecting outliers and so it will be necessary to identify the best feature or subset of features. The plot of the determinant of the covariance for each feature provides two complimentary pieces of evidence to identify the best feature(s), one visual and one analytical.

Step 6a – Convergence plot: Plotting the convergence (i.e. the determinant of the covariance after fixing the candidate outlier) for all features over the iterations yields different patterns. There are several key properties of this plot that will discriminate good features from poor ones. All of the following points assume the data has outliers. However, it is possible for a dataset not to contain outliers, and this is also addressed below.

- First, we expect a negative slope between points in the convergence plot, at least until convergence is achieved.
- Second, assuming the outliers are additive in nature and the feature correctly identifies all of those outliers correctly, there should be a piecewise continuous negative slope between consecutive points, at least until convergence is achieved.
- Third, a steeper overall slope in the overall piecewise continuous descent is better. This means the feature is identifying the observations which contribute the most variability, first. All else being equal, one could have two features which both identify all of a set of 10 outliers. But the order in which each of those features predict the outliers is different. So while both features will arrive at the same result (in terms of predicting the number of outliers), the feature that predicts the more extreme observations to be outliers first will have a steeper overall slope.
- Fourth, there will be a sharp bend (less than 180-degrees, but more than 90). Where this bend occurs is the features' prediction for the total number of outliers. This bend is also where the log ratio should, for the first time, reach 0 (or a value very close to 0). A poor feature might never have such a bend or it might occur prematurely.
- Fifth, after the sharp bend, there will be a leveling off or the remainder of the observations up to minor oscillatory noise. The theoretical lower limit is the determinant of the covariance matrix without any outliers. However, this is not useful to know in practice because one may not know whether the data has outliers or not. So a better indicator, which is independent of the underlying data, is the rate of change and this is accounted for by the log ratio. When the rate of change goes to 0 within some tolerance--.05 was used in the current work—we can say the feature has converged. Figure 5.7 shows a plot of a time series with no outliers and it shows the value of the determinant of the covariance matrix as successive observations of the time series are corrected by interpolation. Notice that the starting value of about .5 is arbitrary and that each time series would have its own different starting value. However, for the same time series in which the log ratio of adjacent determinants of the covariance matrix are taken in Figure 5.8, we see the value stays close

to 0. Since we expect some small variation in the rate of change in a time series without outliers, setting the threshold to some small value such as .05 (which is the threshold used in the current work) says that reduction in the rates of change greater than .05 are not due to expected fluctuations but, instead the presence of outliers.

- Overall, the pattern of convergence should approximate an L-shape.

Step 6b – Sum of area under convergence plot: The best feature can be identified by a single value derived analytically from the convergence plot: the sum of the area under the curve. In order to compute this integral, we proceed with the following steps.

- First, the x-axis is always identical for each feature because the x-axis represents the number of observations in the time series, which never changes.
- Second, the starting values of the y-axis (when the x-axis equals 1) are equivalent for all features, as these represent the value of the determinant of the covariance matrix without any corrections.
- Third, the final values of the y-axis (when the x-axis equals 100) are reflective of how close to (or far away from) the theoretically true value.
- Fourth, the points in between these starting and ending values represent the particular path taken by a given feature. We can recast this feature path as a polynomial function by using cubic splines, based on which we can compute the integral and determine the area under the curve.
- Finally, for a given dataset, the area obtained for each feature can be used to determine the best and worst feature. In fact, one can obtain a rank ordering of the features from best to worst with this single quantitative index, where smaller areas reflect better features. Building on our previous intuition regarding the properties of the convergence plot, the ideal L shaped function will have a much smaller area than a function which does not decrease as quickly or which does not level off or which demonstrates more oscillation, for instance.

Step 7 – Feature adaptation: It should be noted that a feature that works really well for one dataset might do poorly in a different data set. This outlier detection algorithm is adaptive to the data because it does not claim any one feature is always best. It also does not require the user to assume or specify the number of outliers in the data, which is a common limitation with many existing outlier techniques. One could have 0 outliers in the data and the log ratio of the determinant of the covariance matrix of a good feature would oscillate around 0, up to some tolerance, as we see in Figure 5.8.

5.5 Summary

In contrast to the simulation only studies, where the number and location of the outliers is known beforehand and allows for the computation of the true positive rate to measure an outlier detection method's efficacy, experimental data is more challenging because one does not even know the number and location of outliers, or if they are present at all. Most existing outlier detection methods require user defined input parameters, which make assumptions about the nature, number or type of outliers and/or the data itself. But the circular logic implemented in these methods—that one should know something about the outliers in the data beforehand—makes some methods ineffective in some situations.

The current chapter showed how the covariance based outlier detection algorithm can winnow a set of features which make predictions about outliers and determine which features yield the most effective outlier detection results. The result was shown to be effective both from a data point of view (where the outliers are not known beforehand) and from the point of view of a sensitivity analysis where the TPR and FPR are calculated because the outliers are known beforehand. Remarkable consistency is seen between the features that perform best under each scenario.

The key to achieving this difficult goal is the covariance matrix of the error term of the time series. The covariance based outlier detection algorithm is predicated on the assumption that outliers in the data perturb the error terms in a discernible and systematic way and that, by monitoring the rate of change in the error term, one can determine the number of outliers in the data. As the rate of change converges to some small tolerance level, one has evidence that the outliers have successfully been detected and corrected. Another powerful property of this proposed method is that, unlike many existing outlier detection algorithms, one need not know if the data has outliers or not. If outliers do not exist, the rate of change of the error term will start small and stay small, which stands in contrast to data with outliers where the rate of change will start large but then decrease until all outliers have been detected. A final important property of this proposed covariance based solution is that it can help differentiate features that are better or worse at detecting outliers. One can compare the convergence rate for a set of features. Features ineffective at identifying outliers will show a slow rate of change from the beginning (indicating that features inability to detect outliers) whereas features that are better predictors of outliers will have a rate of change that drops quickly as outliers are identified and corrected.

5.6 Figures and Tables

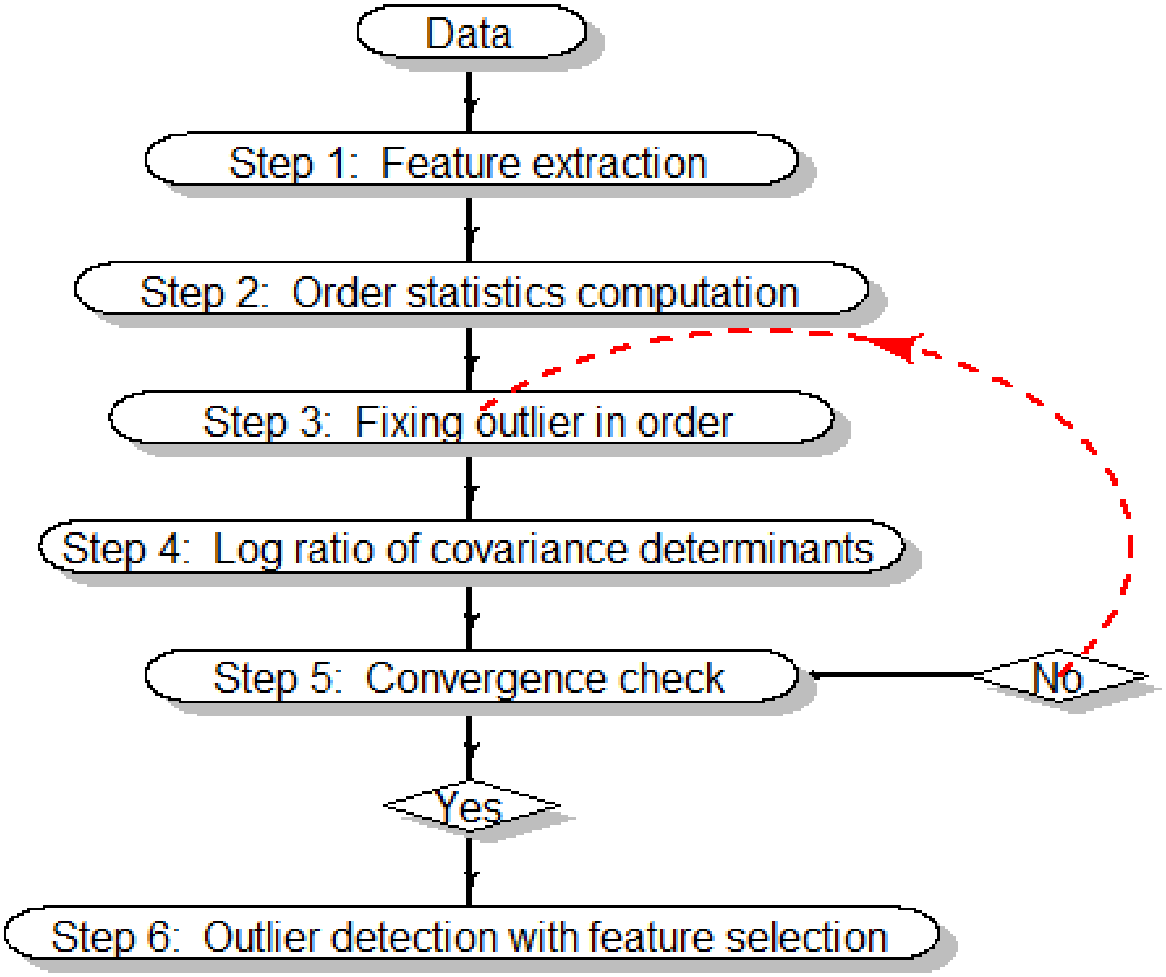

Data
Step 1: Feature extraction
Step 2: Order statistics computation
Step 3: Fixing outlier in order
Step 4: Log ratio of covariance determinants
Step 5: Convergence check
No
Yes
Step 6: Outlier detection with feature selection

Figure 5.1. Workflow of the Covariance Based Outlier Detection Algorithm.

Table 5.1. Summary of 13 features presented in Chapter 3.

	Feature Description
F1	Multivariate Least Trimmed Squares
F2	Reweighted Multivariate Least Trimmed Squares
F3	Model based determinant of covariance matrix
F4	Model based trace of covariance matrix (i.e. sum of variances)
F5	Model based determinant of correlation matrix
F6	Model based product of variances
F7	Model free determinant of covariance matrix
F8	Model free trace of covariance matrix (i.e. sum of variances)
F9	Model free determinant of correlation matrix
F10	Model free product of variances
F11	Sum of absolute value of time series observations
F12	Model based sum of squared residuals
F13	Literature based Multivariate Least Trimmed Squares (Agullo, et al. 2008; Croux & Joossens, 2008)

Table 5.2. Determinants of the covariance matrix for different features (columns) before outliers were added (row labeled ‘None’), after all outliers were added but no corrections (row labeled ‘All’) and sequential corrections. The values in the row labeled ’None’ are identical because this is the determinant of the covariance matrix before outliers were added so it is the same for all features. The same logic is true for the row labeled ‘All’. Starting with the row labeled ‘1’, interpolation was implemented based on the outlier predictions of each feature. These results are for the outlier condition with 5 outliers of magnitude 3.

Outliers	F1	F3	F4	F6	F7	F8	F10	F11	F12	F13
None	0.961	0.961	0.961	0.961	0.961	0.961	0.961	0.961	0.961	0.961
All	17.87	17.87	17.87	17.87	17.87	17.87	17.87	17.87	17.87	17.87
1	11.18	10.18	12.48	12.27	9.757	12.09	12.35	14.09	12.12	12.92
2	6.202	5.942	7.883	8.014	6.009	8.004	8.069	10.49	8.499	7.724
3	3.68	3.924	4.449	4.318	3.724	4.334	4.407	6.589	4.267	4.573
4	1.852	2.413	2.27	2.265	2.31	2.215	2.187	3.422	2.155	2.317
5	1.149	1.245	1.175	1.175	1.151	1.082	1.082	2.089	1.082	1.149
6	1.04	1.072	1.145	1.162	1.027	1.076	1.007	1.767	1.054	1.127
7	0.999	1.157	1.197	1.166	1.04	1.076	1.072	1.765	1.141	1.068
8	1.051	1.169	1.326	1.417	1.037	1.063	1.05	1.755	1.107	1.232
9	1.183	1.215	1.375	1.381	1.026	1.025	0.999	1.739	1.133	1.223
10	1.177	1.201	1.362	1.35	1.005	1.04	0.996	1.739	1.085	1.222
11	1.13	1.242	1.475	1.41	0.985	1.038	1.031	1.722	1.122	1.199
12	1.121	1.287	1.391	1.372	0.975	1.037	1.051	1.69	1.174	1.178
13	1.137	1.277	1.388	1.413	0.988	0.998	1.034	1.689	1.191	1.178
14	1.119	1.25	1.477	1.482	0.96	0.996	0.984	1.679	1.174	1.205
15	1.106	1.223	1.448	1.43	0.928	0.953	0.94	1.673	1.149	1.231
16	1.164	1.198	1.444	1.427	0.893	0.904	0.907	1.646	1.152	1.208
17	1.099	1.174	1.414	1.376	0.851	0.962	0.972	1.642	1.17	1.18
18	1.142	1.196	1.397	1.354	0.824	0.962	0.966	1.64	1.162	1.122

Table 5.3. Log ratio of adjacent values from columns of Table 5.2. Row 1 of Table 5.3 is the log ratio of rows 2 and 3 in Table 5.2. Row 2 of Table 5.3 is the log ratio of rows 3 and 4 in Table 5.2. Highlighting shows the location where the threshold is less than .05. Columns are features, rows are observations and the bottom row is the number of observations flagged by that feature as outliers. These results are for the outlier condition with 5 outliers of magnitude 3.

	1	3	4	6	7	8	10	11	12	13
1	0.47	0.56	0.36	0.38	0.61	0.39	0.37	0.24	0.39	0.32
2	0.59	0.54	0.46	0.43	0.48	0.41	0.43	0.30	0.35	0.51
3	0.52	0.41	0.57	0.62	0.48	0.61	0.60	0.48	0.69	0.52
4	0.69	0.49	0.67	0.65	0.48	0.67	0.70	0.64	0.68	0.68
5	0.48	0.66	0.66	0.66	0.70	0.72	0.70	0.49	0.69	0.70
6	0.10	0.15	0.03	0.01	0.11	0.01	0.07	0.17	0.03	0.02
7	0.04	-0.08	-0.04	0.00	-0.01	0.00	-0.06	0.00	-0.08	0.05
8	-0.05	-0.01	-0.10	-0.19	0.00	0.01	0.02	0.01	0.03	-0.14
9	-0.12	-0.04	-0.04	0.03	0.01	0.04	0.05	0.01	-0.02	0.01
10	0.01	0.01	0.01	0.02	0.02	-0.01	0.00	0.00	0.04	0.00
	6	6	5	5	6	5	6	6	5	5

Table 5.4. Maximum and Average Ratios of TPR over FPR for all 13 Features. Larger values are better.

		F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13
5 outliers, all magnitudes	Maximum	1097	26	1079	1023	107	1042	1116	911	233	930	251	967	949
	Average	118	5	108	87	24	86	123	119	39	122	53	89	125
10 outliers, all magnitudes	Maximum	1804	40	695	810	246	818	880	792	414	810	128	739	713
	Average	93	5	48	67	54	67	71	49	75	52	36	50	42
15 outliers, all magnitudes	Maximum	1129	12	592	575	365	299	288	631	387	620	80	625	598
	Average	44	3	38	35	30	27	22	20	29	30	20	20	27

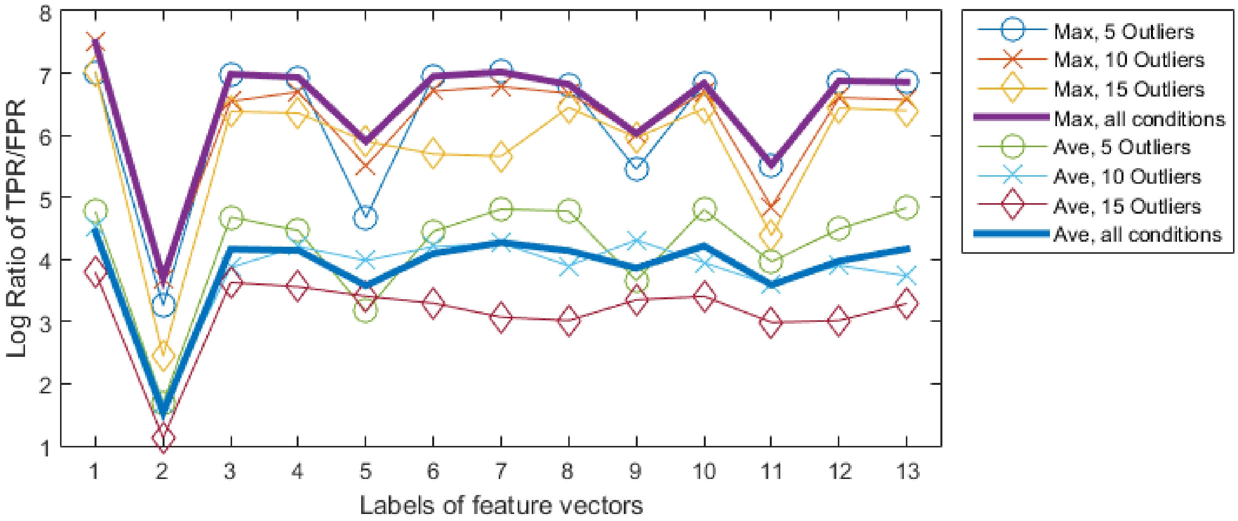


Figure 5.2. Maximum and average values for 3 outlier conditions (5, 10 or 15 outliers) and all outlier conditions (thick line). x-axis is the labels of features vectors. y-axis is the log of the ratio of the TPR over the FPR. (This figure plots the log of the values in Table 5.4.)

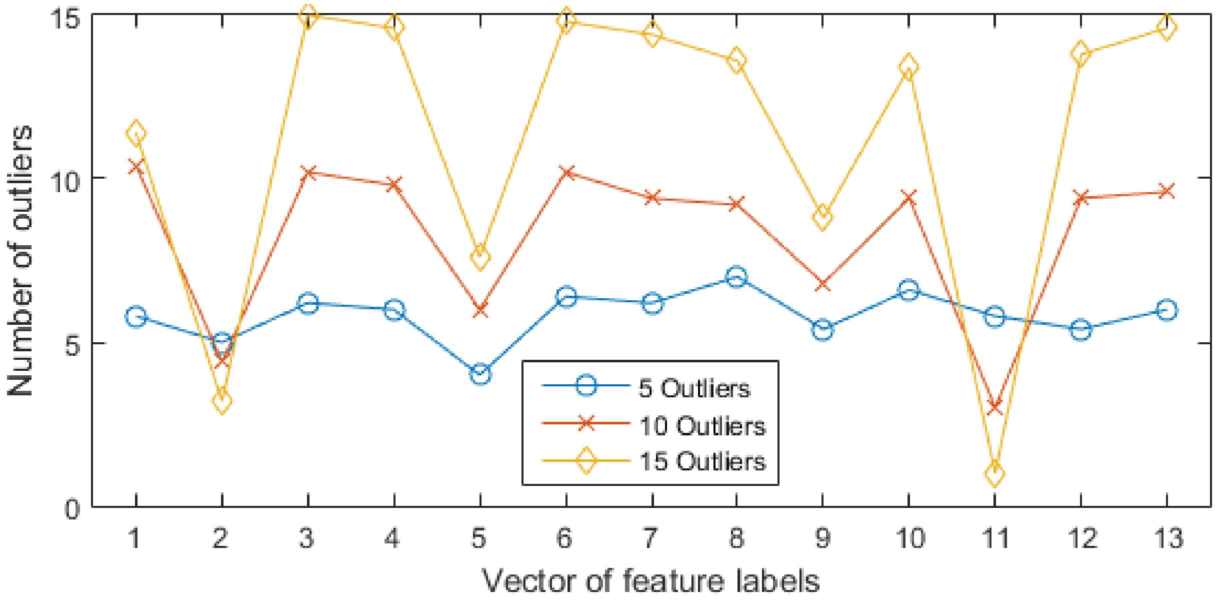


Figure 5.3. Number of outliers identified by each feature vector. x-axis is the label of feature vectors and y-axis is the expected number of outliers. The lines representing the 5, 10 and 15 outlier conditions are each averaged across the five magnitudes of 1, 2, 3, 4 and 5.

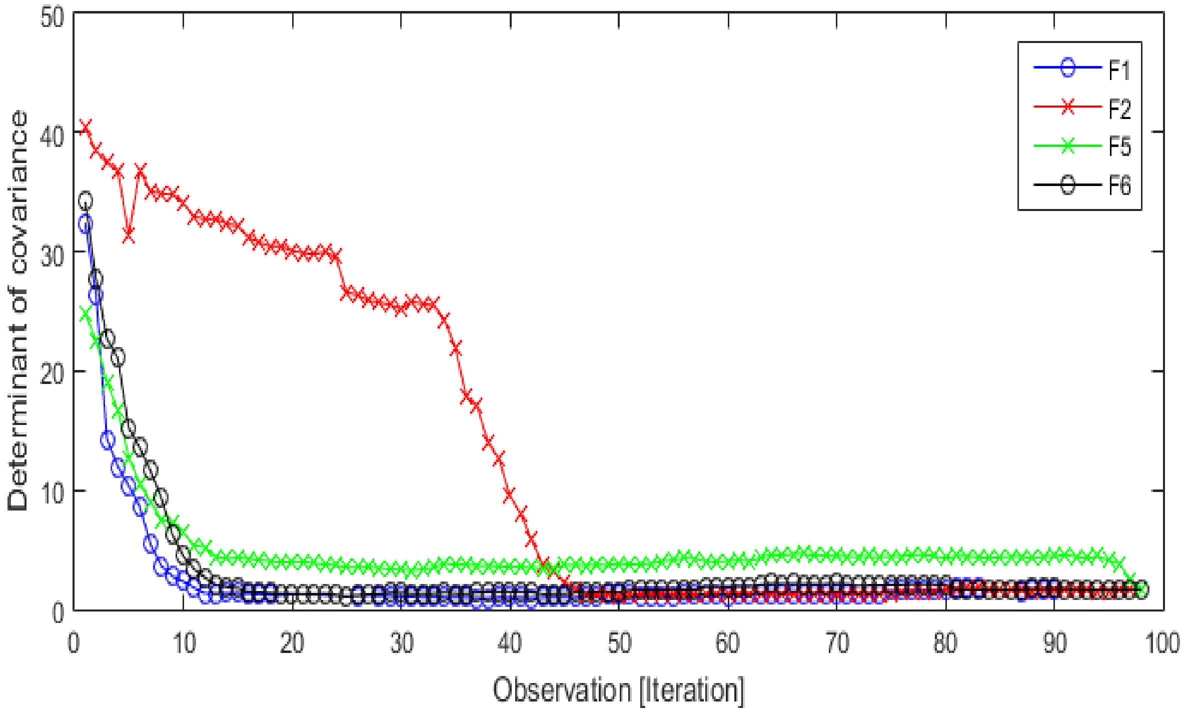


Figure 5.4. Convergence plot for F1, F2, F5 and F6. x-axis is observation (or iteration) and the y-axis is the determinant of the covariance matrix. Results are averaged across all 15 outlier conditions (5, 10 and 15 outliers for magnitudes of 1, 2, 3, 4 and 5).

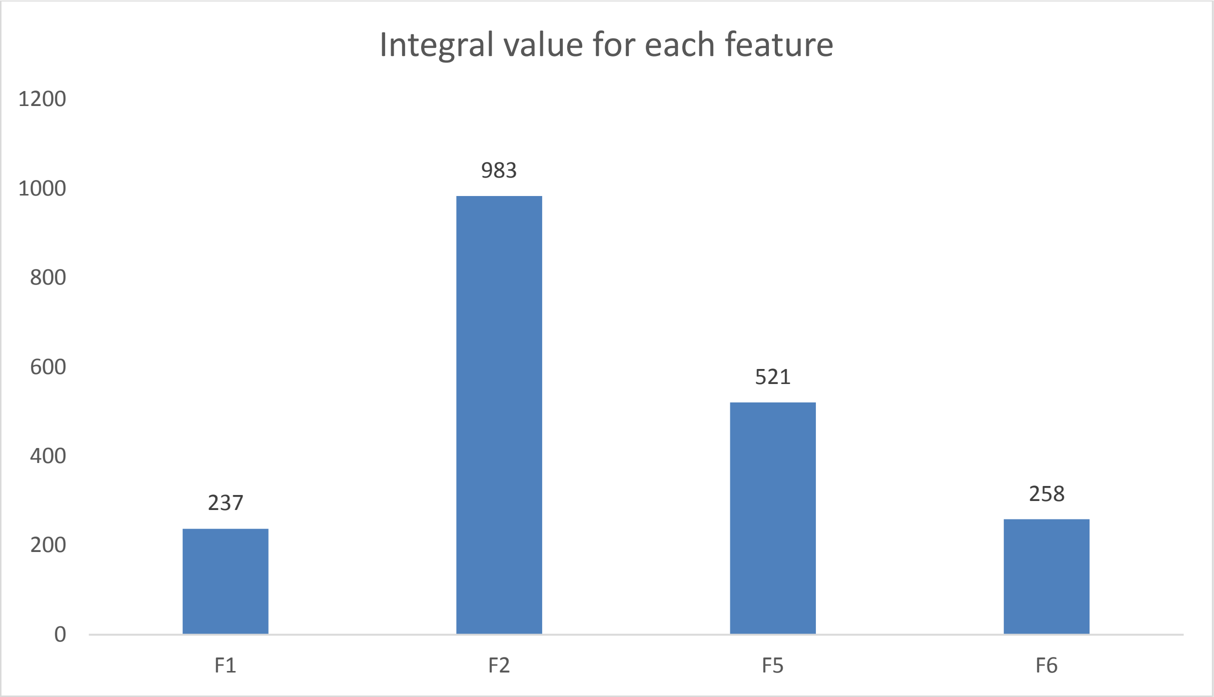


Figure 5.5. Integral value of the convergence plot for four features (F1, F2, F5 and F6). y-axis represents the value of the integral and x-axis is the feature label. Results are from the average of 15 outlier conditions (5, 10 or 15 outliers and 1, 2, 3 4 or 5 magnitudes).

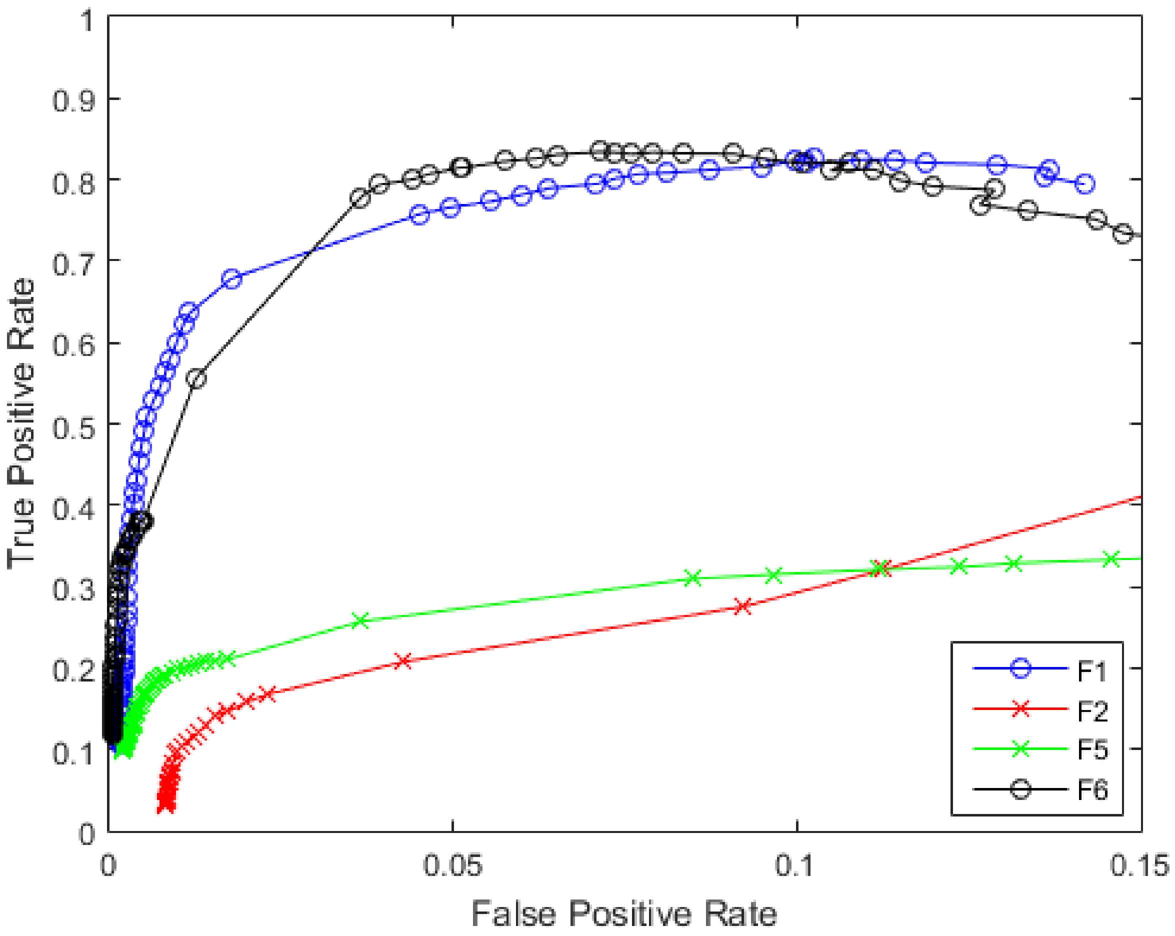


Figure 5.6. ROC plot for F1, F2, F5 and F6. x-axis is FPR and y-axis is TPR. Results are from the average of 15 outlier conditions (5, 10 or 15 outliers and 1, 2, 3 4 or 5 magnitudes).

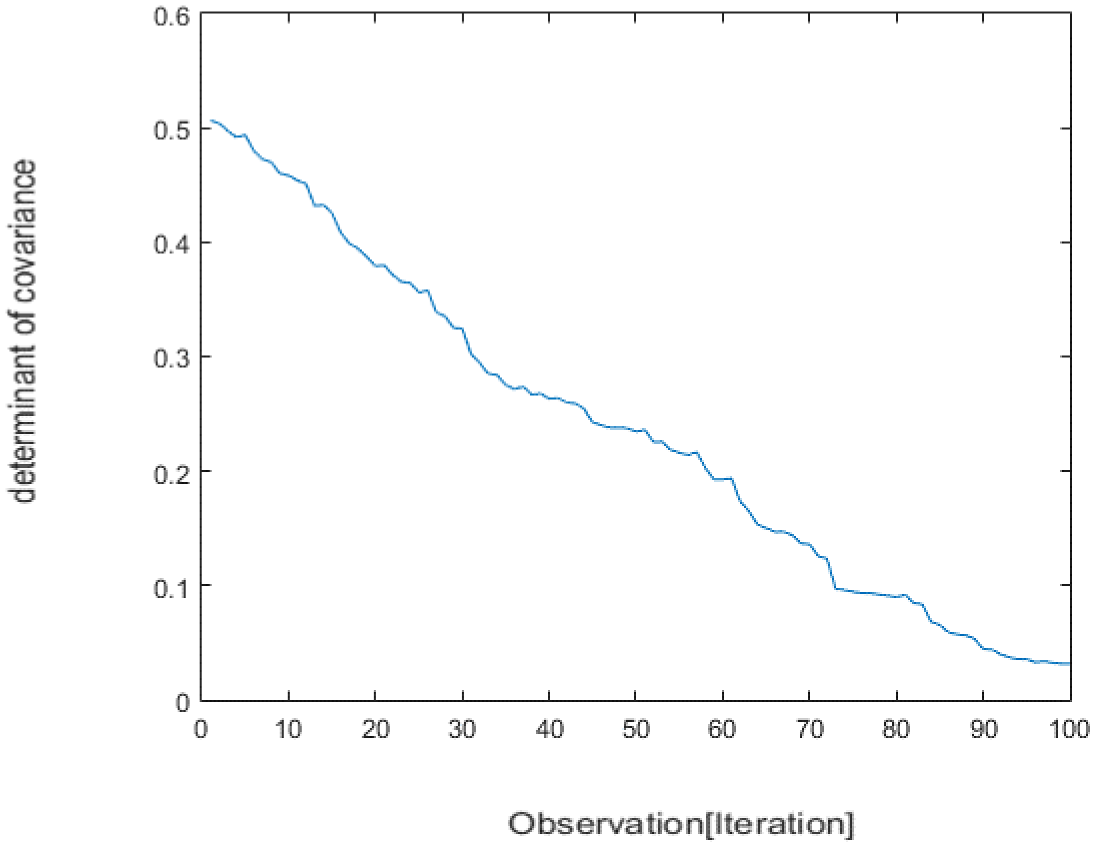


Figure 5.7. Plot of the determinant of the covariance matrix of a time series with no outliers.

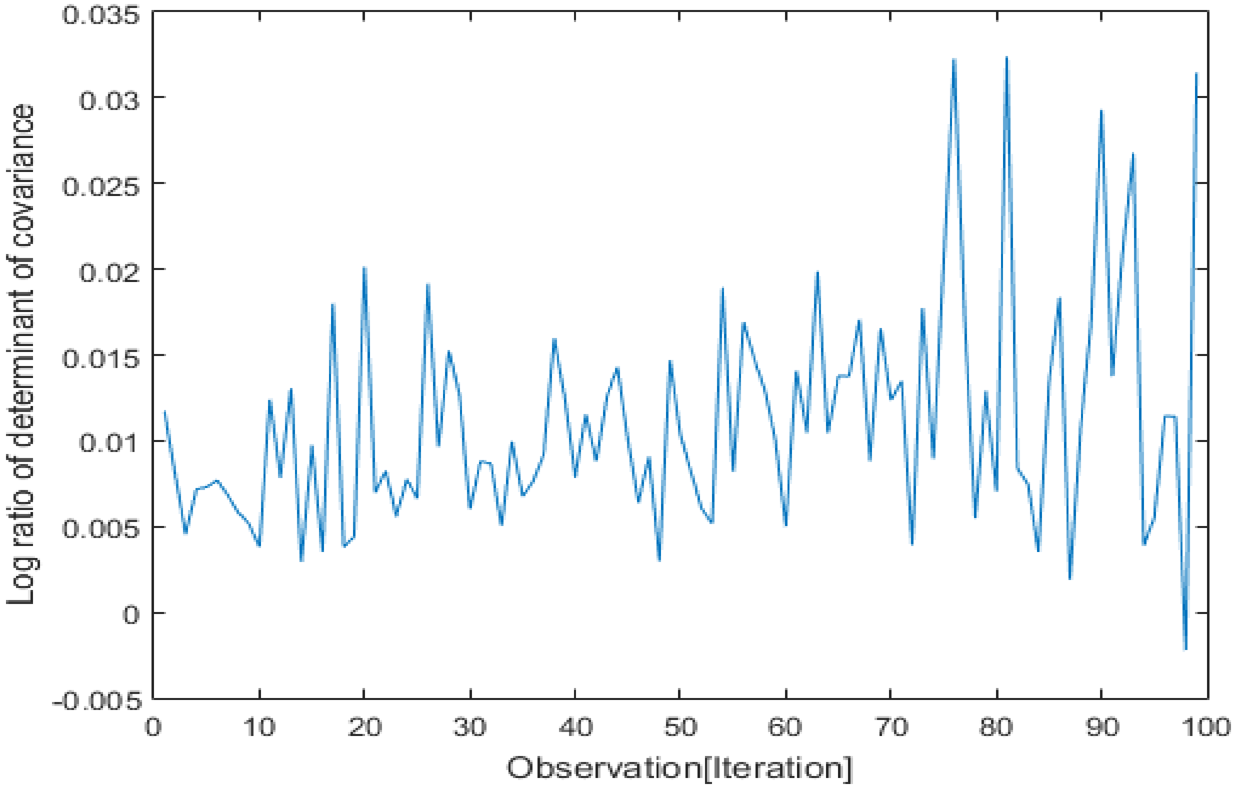


Figure 5.8. x-axis is the observation number of the time series and y-axis is the log ratio of the determinant of the covariance matrix after successive observations of the time series were corrected by interpolation. This time series was outlier free. Notice the scale of the y-axis ranges from -.005 to .035 and that the true/theoretical value of the time series oscillates around 0.

6. Forecasting Time Series with Outliers

In this chapter, the covariance based outlier detection method is brought to bear on a process of outlier detection, model parameter estimation and data forecasting. The rationale and workflow for forecasting is first overviewed and the results are presented.

6.1 Time series forecasting

Time series forecasting is employed to predict future observations based on known or existing observations (Tsay, 2002; Zivot & Wang, 2006). To better understand forecasting, we present an alternative formulation of the AR(p) time series model presented in Equation 2.4. Equation 6.1 expresses Equation 2.4 in the form of the p-lag vector autoregressive model,

$$\boldsymbol{u}_t = \boldsymbol{w} + \boldsymbol{A}_1\boldsymbol{u}_{t-1} + \boldsymbol{A}_2\boldsymbol{u}_{t-2} + \cdots + \boldsymbol{A}_p\boldsymbol{u}_{t-p} + \boldsymbol{\varepsilon}_t, \qquad t = 1, \dots, T, \tag{6.1}$$

where $\boldsymbol{u}_t = (u_{1t}, u_{2t}, \dots, u_{nt})'$ is a $(n\ x\ 1)$ vector of time series variables where n is the number of variables, $\boldsymbol{w}$ is an intercept vector which allows the time series to have a nonzero mean, $\boldsymbol{A}_i$ are $(n\ x\ n)$ coefficient matrices and $\boldsymbol{\epsilon}_t$ is a $(n\ x\ 1)$ unobservable 0 mean independent white noise vector process with time invariant covariance matrix Σ.

In Equation 6.1, each row of the vector $\boldsymbol{u}_t$ represents one equation, where the number of equations equals the number of variables. Now let $\boldsymbol{u}_i$ denote the i^{th} equation from Equation 6.1. Under the assumption of covariance stationarity and no parameter restrictions and for purposes of parameter estimation, Equation 6.1 can be recast as

$$\boldsymbol{u}_i = \boldsymbol{Z}A_i + \boldsymbol{e}_i, \qquad i = 1, \dots, n, \tag{6.2}$$

where $\boldsymbol{u}_i$ is a $(T\ x\ 1)$ vector of observations for the i^{th} equation, $\boldsymbol{Z}$ is a $(T\ x\ k)$ matrix with the t^{th} row determined by $Z_t' = (1, \boldsymbol{u}_{t-1}', \dots, \boldsymbol{u}_{t-p}')$, $k = np + 1$, $\boldsymbol{A}_i$ is a $(k\ x\ 1)$ vector of parameters and $\boldsymbol{\epsilon}_i$ is a $(T\ x\ 1)$ error term that has a covariance matrix defined by $\sigma_i^2\boldsymbol{I}_T$. Because the AR($p$) is in the form of a seemingly unrelated regression, where each equation has the same explanatory variables, each equation may be estimated separately with OLS regression. The variance of this model is estimated as follows.

Define $\widehat{\boldsymbol{A}} = [\widehat{\boldsymbol{A}}_1, \dots, \widehat{\boldsymbol{A}}_n]$ as the $(k\ x\ n)$ matrix of least squares coefficients for the n equations. Then, let $vec(\widehat{\boldsymbol{A}})$ be the operation which stacks the columns of the $(k\ x\ n)$ matrix $\widehat{\boldsymbol{A}}$ into a column vector of length $(nk\ x\ 1)$ which yields

$$vec(\widehat{\boldsymbol{A}}) = \begin{pmatrix} \widehat{\boldsymbol{A}}_1 \\ \vdots \\ \widehat{\boldsymbol{A}}_n \end{pmatrix}. \tag{6.3}$$

Assuming a stationary and ergodic VAR model (Lutkepohl, 2005), $vec(\widehat{\boldsymbol{A}})$ is asymptotically normally distributed with covariance matrix:

$$\widehat{covar}\left(vec(\widehat{\boldsymbol{A}})\right) = \widehat{\boldsymbol{\Sigma}} \otimes (\boldsymbol{Z}'\boldsymbol{Z})^{-1}, \tag{6.4}$$

where

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{T-k}\sum_{t=1}^{T} \hat{\varepsilon}_t \hat{\epsilon}_t', \tag{6.5}$$

and $\hat{\varepsilon}_t = \boldsymbol{u}_t - \hat{A}' Z_t$ is the multivariate least squares residual at time *t*.

Once the parameter estimates are obtained, one can use the VAR model to do *h*-step ahead forecasting according to Equation 6.6,

$$\boldsymbol{u}_{T+h|T} = \boldsymbol{w} + \boldsymbol{A}_1 \boldsymbol{u}_{T+h-1|T} + \cdots + \boldsymbol{A}_p \boldsymbol{u}_{T+h-p|T} \tag{6.6}$$

where *h* is the number of steps ahead on which one desires to make a forecast.

6.2 Forecasting workflow

Step 1. Data was simulated according to the VAR (Vector Auto Regressive) model in Equation 2.4 (or, equivalently, Equation 6.1). The parameters of this model were estimated for both a VAR and a VARMA (Vector Auto Regressive Moving Average) model because, in real practice, one would usually fit multiple models. The estimated parameters were used to generate 10-step ahead forecasting of the data. This model represents the ground truth because it is not contaminated by outliers.

Step 2. Outliers were added to the data generated in Step 1. Because the objective of this study is to demonstrate the efficacy of the covariance based outlier detection algorithm, only three outlier conditions are considered: 5, 10 or 15 outliers with an outlier magnitude of 3. These conditions represent the middle ground of the outlier conditions considered. Once the data had been contaminated with outliers, the parameters of the model were estimated for both a VAR and VARMA model and the parameters were used to predict observations that are 10-steps (i.e. observations) ahead. This would represent the situation of a data analyst who has contaminated data, but does not know it, and uses this contaminated data as though it does not have outliers. Parameter estimation and forecasting with contaminated data would lead to spurious or erroneous predictions. The mean square errors (MSE) are computed for the model parameter estimates, where the sum of the squared differences for the estimates is compared to the ground truth (Step 1).

Step 3. A feature that has performed well on time series data—Feature 1—and a feature that has performed poorly—Feature 2—were used to identify the outliers in the data and correct them. Then, once the corrections were completed, the model parameters were fit and the 10-step ahead forecasting was executed. If the feature is good, then the MSE for this step should be really small, whereas a poor feature will show a MSE that is larger the MSE of the good feature. The results for these steps are presented for each component of the model fitting and forecasting.

6.3 Performance evaluation

Estimation of A coefficients

As outlined in Section 6.1, the coefficients for the model were estimated. The MSE for these coefficients was computed by summing the difference between the coefficients in Step 1 (the ground truth) and Steps 2 and 3. Table 6.1 shows the results. It is clear that the 'Good Feature' has the lowest MSE values, as compared to the 'Poor Feature' and 'Outlier' time series. It is also apparent that the results are not influenced by fitting a VAR or VARMA model. This makes sense because the data was generated from a VAR process. Since the VAR is a special case of the VARMA, the model in both cases converged on the same solution, at least for the parameter coefficients.

Model covariance estimate

Equation 6.5 shows the covariance matrix estimated and Table 6.2 displays the results for those covariance matrix estimate. Again the 'Good Feature' yields results near 0 whereas the 'Outlier' and 'Poor Feature' time series yielded estimates very far away from the ground truth. Also, the two model fits track identically, with neither showing any difference over the other. The results were computed by calculating the MSE for each cell of the variance/covariance matrix and the MSE's were summed across all cells.

An alternative way to assess the impact of outliers on the error term is to use the determinant of the variance/covariance matrix of the error term. The determinant is nice because it uses a single number to represent the entire matrix (whereas the MSE requires multiple computations to arrive a single number). This result is presented in Table 6.3.

In Table 6.3, the true value of the determinant for the ground truth time series is in the column labeled 'Truth'. The 'Good' feature, which correctly identified the outliers, yields values of the determinant that are nearly identical to the determinant. The untreated case, 'Outlier' has values for the determinant which are quite far from the ground truth and this is also the case for the 'Poor' feature.

10-step ahead prediction

Using the parameter values obtained, one can then forecast observations according to Equation 6.6. Table 6.4 displays the MSE results for this forecasting step. The VAR and VARMA yield different results for some conditions. This is expected because the VARMA model fits an additional component—the moving average part of the model—and uses this information in the forecasting algorithm. The VAR model does not have a moving average component. But to the main point of this thesis, we see the ‘Good Feature’ fares better than the untreated outcome (‘Outlier’) and the ‘Poor Feature’.

In conclusion, it is readily apparent that using a good feature for outlier detection with the covariance based convergence check method offers an accurate result. The result is much better than using outlier contaminated data or even a poor feature. After a VAR and VARMA model was fit to a set of data, the covariance based convergence method accurately detected the outliers and allowed for the correct estimation of the model parameters and yielded a forecast very close to that of the raw data which was not contaminated by outliers.

6.4 Tables

Table 6.1: MSE for the A coefficients for outliers of magnitude of 3 and contamination levels of 5, 10 or 15 outliers for an outlier time series, for time series data that has been corrected using a good feature and time series data that has been corrected using a poor feature. MSE values closer to 0 are better.

	Outlier		Good Feature		Poor Feature	
	VAR	VARMA	VAR	VARMA	VAR	VARMA
5-3	0.02	0.02	0.01	0.01	0.04	0.04
10-3	0.02	0.02	0.01	0.01	0.03	0.03
15-3	0.05	0.05	0.03	0.03	0.08	0.08

Table 6.2: MSE for the variance/covariance matrix for outlier conditions of magnitude equal to 3 and contamination levels of 5, 10 or 15 outliers for uncorrected outlier time series, time series that has been corrected using a good feature and time series that has been corrected using a poor feature.

	Outlier		Good Feature		Poor Feature	
	VAR	VARMA	VAR	VARMA	VAR	VARMA
5-3	0.67	0.67	0.00	0.00	0.48	0.48
10-3	2.12	2.12	0.01	0.01	1.92	1.92
15-3	5.87	5.87	0.02	0.02	5.22	5.22

Table 6.3: MSE for the determinant of the variance/covariance matrix for outlier conditions of magnitude equal to 3 and contamination levels of 5, 10 or 15 outliers for uncorrected outlier time series, time series that has been corrected using a good feature and time series that has been corrected using a poor feature.

	Truth	Outlier	Good	Poor
5-3	0.35	2.58	0.32	2.01
10-3	0.87	13.19	0.82	10.53
15-3	0.96	33.22	1.15	26.79

Table 6.4: MSE for the 10-steps ahead predictions for outlier conditions with magnitude 3 and contamination levels of 5, 10 or 15 outliers for the untreated case (‘Outlier’), the case treated with a ‘Good Feature’ and the one treated with a ‘Poor Feature’.

	Outlier		Good Feature		Poor Feature	
	VAR	VARMA	VAR	VARMA	VAR	VARMA
5-3	0.05	0.04	0.01	0.01	0.08	0.26
10-3	0.07	0.28	0.02	0.01	0.09	0.42
15-3	0.58	0.16	0.11	0.06	0.89	0.56

7. Conclusion

One important weakness of nearly all outlier detection methods is that a researcher is required to express some *a priori* knowledge about the underlying statistical distribution. Or that he or she knows something about the outliers themselves. A more general method which can accommodate data that does not require special knowledge of the data is more desirable in these cases and the outlier detection method proposed here aimed to meet this requirement. A second important weakness of many methods is the specification of some input parameter, such as the number of outliers that might be present or the choice of a threshold, either of which can make an important difference in the number of outliers the algorithm identifies. The proposed method does not have any such input parameter required to be specified by the user.

Commonly used heuristic (such as plots and the so called 3-sigma rule) were introduced to provide baseline performance for algorithms for what many researchers commonly use. These methods are not wrong per se; but they have severe limitations. Plots are fine in very small dimensional spaces with a limited number of data points; but visual inspection of millions or billions of data points and hundreds of thousands of variables quickly push the limits of cognitive and perceptual processing, thereby making visual heuristics ineffective. These situations demand analytical based methods, which have the added advantages of being more principled and reproducible. The 3-sigma rule (or Simple Testing Method) showed an upper true positive rate of around .6 for outlier conditions where outliers should be easy to detect and showed lower bounds around .2.

After constructing a set of features (some of which were based on existing methods in the literature, others were novel, some were parametric, some were non-parametric, some lead to fast computations and other features take longer to compute), these features were tested using the gold standard true positive rate. The TPR is only effective if one knows in advance the location and number of outliers so it is not a realistic measure in practice; but in simulation studies it provides an excellent benchmark to assess the efficacy of features in outlier detection. In these studies, which considered both univariate and multivariate time series and different error levels, a few conclusions became evident. First, certain features tended to do well across all these conditions, providing support for the hypothesis that certain features might have more general utility across a wide range of conditions. For instance, features 3, 7 and 12 tended to do well across all studies (as well as several other features) whereas feature 2 did not fare well at all. Feature 3 and feature 7 were the determinant of the covariance matrix, with the former being a parametric version and the latter being a non-parametric version. Feature 12 was the sum of the magnitudes of the time series observations. Feature 2 did not do well in most studies.

Another important conclusion from the feature evaluation studies in Chapter 3 is that parametric features do not yield a clear advantage over non-parametric features. This is interesting and important because parametric features generally take longer to compute and, all else being equal, an algorithm that yields a quicker result is preferred to one that takes longer. Another conclusion was that univariate analyses generally had lower outcomes than multivariate analyses. This was expected because multivariate approaches can aggregate information across variables, thereby yielding a more accurate pattern of results and outliers. Many of the features do really well, with some having an average TPR of .9—and for outlier conditions where detecting outliers is straightforward, many of these features had TPR of 1. It would be interesting to see how the error level used to generate the time series might impact these conclusions in future research.

Chapter 4 showed results from higher order feature construction, which were obtained by using a Voronoi diagram. The diagram takes as inputs individual features to construct a new feature. A 2-dimensional input vector was considered in the present work; but larger dimensional inputs are indeed possible. For this set of data, aggregating features led to some advantages over the features individually; but in other cases the individual features did better. We see in the original MVOD result (Zwilling & Wang, 2014)—which compared a pair of features multiplied together to the literature based MLTS—an advantage for the features, demonstrating an improvement over a more established method. It is also important to keep in mind the number of individual features which were used as inputs to the Voronoi diagram was quite small. In machine learning applications, feature construction can lead to hundreds or thousands of features for testing. So it may be the case that different inputs might yield different and better results also. The Voronoi diagram is capable of testing multidimensional spaces. Additionally, the results of the features tested here might be different for other data sets.

Chapter 5 demonstrated the efficacy of the covariance based convergence method. This method leverages the error term of the time series to identify outliers. The results from the simulation studies show a remarkable correspondence between the results using just a TPR approach (as in Chapter 3) and the covariance based convergence method (as in Chapter 5). For instance, in the simulation study from Chapter 3.3, we see that 3 features—F2, F5 and F9 all perform poorly. These are also 3 of the features identified by the covariance based convergence method that do poorly as well. Moreover, the features that do really well with the simulation approach based on the TPR—F1 and F6—also do really well with the covariance approach. A key motivation for developing the covariance based approach is to discriminate among a candidate set of features proposed or used by the analyst and with the set of features proposed here this has been demonstrated. This approach has potential application for so-called Big Data because it has the capacity to detect outliers in huge multivariate datasets. While time series data was tested here, the method is general and could be used on any multivariate data.

Chapter 6 demonstrated how the covariance based outlier detection method can be used in service of forecasting data. The parameters were estimated and 10-steps ahead were forecasted for the contaminated time series for 3 outlier conditions. Then, a good and poor feature were used to identify outliers and then 10-steps ahead were forecasted. When outliers are effectively identified by the good feature, not only are the parameter estimates of a time series model (i.e. VAR) improved, but the resulting data forecasting is much more closely aligned to the results of the uncontaminated time series. However, a poor feature does not have the same effect, as it cannot effectively identify the outlier and so the performance using this feature was more similar than the result obtained with the contaminated time series.

Table 7.1 presents a summary of some important characteristics of each of the 13 features, which would be useful both for identifying a good feature but also a feature that may or may not work for the constraints of a data analyst. A key point raised in the opening chapter was that certain outlier detection situations might trade speed for accuracy; but in other situations, such as a medical setting, being correct is probably more important than speed (though clearly there are times where speed and accuracy are important). For each feature, this table summarizes three pieces of information. First, the Model Based column indicates whether the feature is based on a model (such as fitting a regression model) or whether the feature was derived directly from the data. ‘Yes’ means a model was fit whereas ‘No’ means the feature was derived directly from the data. The Speed column, measured in seconds, represents the amount of time required to complete one iteration of the feature for one time series with 100 observations for a single outlier condition on a Windows 8 computer with an Intel Core i7-5500U CPU @ 2.40 GHz, 16.0 GB RAM and 64-bit operating system. The Average Integral column is the area under the curve for the determinant of the covariance matrix and this value was averaged across all 15 outlier conditions.

If one had constructed a feature that was fast (small time needed to compute the feature) with a small integral (it converged correctly) this would mean this feature was an excellent choice for the data set at hand. For the 13 feature considered in this study, F7 is really good on all dimensions. It is the fastest features (0.0227 seconds), has the smallest average integral (209) and is not model based. By contrast, we can see that a poor feature, such as F2, is slow (0.5514 seconds) and has a very large average integral (983).

In F1, we see a good example of a feature that takes a very long time to complete (33.145 seconds) and has a good integral (209). However, because this feature is very slow to compute, it is probably not a good choice to use, especially since we do not see this increased speed gaining anything over some of the better features, as measured by the area under the curve. Overall, in practice, it is important to balance situational constraints with feature properties, like those provided in Table 7.1.

7.1 Table

Table 7.1. Summary table of 13 Features. The first column is the feature label. Model Based refers to whether a feature required fitting a model (Yes) or whether the feature was computed directly from the data (No). The speed listed in seconds is the amount of time required to complete one iteration of the feature for a single time series of 100 observations for a single outlier condition. The Integral represents the area under the curve of the determinant of the covariance matrix, averaged across all 15 outlier conditions. Smaller values are better.

-

	Model	**Speed**	**Average**
	Based	**(seconds)**	**Integral**
F1	Yes	33.145	237
F2	Yes	0.5514	983
F3	Yes	0.0284	232
F4	Yes	0.0263	259
F5	Yes	0.0364	521
F6	Yes	0.0256	259
F7	No	0.0227	209
F8	No	0.0201	245
F9	No	0.0211	458
F10	No	0.0165	244
F11	No	0.0067	573
F12	Yes	0.5565	257
F13	Yes	0.5514	255

References

Aggarawal, C. (2013). Outlier Analysis. New York: Springer.

Aggarwal, C. C., & Yu, P. S. (2001). Outlier detection for high dimensional data. *Proceedings of the ACM SIGMOD*. doi:10.1145/376284.375668

Agullo, J., Croux, C., & Aelst, S. V. (2008). The multivariate least-trimmed squares estimator. *Journal of Multivariate Analysis*, 99, 311-338.

Barnett, V., & Lewis, T. (1994). *Outliers in statistical data* (3rd ed.). Chichester: Wiley.

Box, G. E., & Tiao, G. C. (1968). A Bayesian approach to some outlier problems. *Biometrika*, *55*(1), 119-129. doi:10.1093/biomet/54.1.119

Becker, C., Fried, R. & Kuhnt, S. (2013). Robustness and Complex Data Structures. Springer-Verlag.

Basu, S. & Meckesheimer, M. (2007). Automatic outlier detection for time series: an application to sensor data. *Knowledge and Information Systems, 11*(2), 137-154. doi:10.1007/s10115-006-0026-6

Breiman, L. (2001). Statistical Modeling: The Two Cultures. *Statistical Science*, *16*(3), 199-231.

Breunig, M. M., Kriegel, H., Ng, R. T., & Sander, J. (2000). LOF: identifying density-based local outliers. *ACM SIGMOD*. doi:10.1145/335191.335388

Burridge, P. & Taylor, A. M. R. (2006). Additive outlier detection via extreme value theory. *Journal of Time Series Analysis*, *27*(5), 685-701. doi:10.1111/j.1467-9892.2006.00483.x

Choy, K. (2001). Outlier detection for stationary time series. *Journal of Statistical Planning and Inference*, *99*, 111-127.

Croux, C., & Joossens, K. (2008). Robust estimation of the vector autoregressive model by a least trimmed squares procedure. In *COMPSTAT 2008* (pp. 489-501). Physica-Verlag HD.

Efron, B. & Stein, C. (1981). The Jackknife Estimate of Variance. *The Annals of Statistics* **9** (3): 586–596. doi:10.1214/aos/1176345462

Fedorov, V. V. (1972). Theory of Optimal Experiments. Academic Press.

Glaister, S. (1991). *Mathematical methods for economists.* Wiley.

Gombay, E. (2008). Change detection in autoregressive time series. *Journal of Multivariate Analysis*, *99*, 451-464. doi:10.1016/j.jmva.2007.01.003

Grubbs, F. E. (1969). Procedures for Detecting Outlying Observations in Samples. *Technometrics*, *11*(1), 1-21. doi:10.1080/00401706.1969.10490657

Hau, M. C. & Tong, H. (1989). A practical method for outlier detection in autoregressive time series modelling. *Stochastic Hydrology and Hydraulics, 3*, 241-260.

Hodge, V. J., & Austin, J. (2004). A Survey of Outlier Detection Methodologies. *Artificial Intelligence Review*, *22*, 85-126. Retrieved from http://dx.doi.org/10.1023/B:AIRE.0000045502.10941.a9

Hubert, M., & Debruyne, M. (2010). Minimum covariance determinant. *Wiley Interdisciplinary Reviews: Computational Statistics, 2*, 36-43. doi:10.1002/wics.61

John, G. H. (1995). Robust Decision Trees: Removing Outliers from Databases. *Proceedings of the 1st International Conference on Knowledge Discovery and Data Mining*, 174-179.

Johnston, J. & DiNardo, J. (2001). Econometric Methods. McGraw-Hill.

Kriegel, H., Kröger, P., & Zimek, A. (2009). Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery From Data*. doi:10.1145/1497577.1497578

Kumar, V. (2008). *Computational Methods of Feature Selection*. Chapman & Hall/CRC.

Last, M. & Shumway, R. (2008). Detecting abrupt changes in a piecewise locally stationary time series. *Journal of Multivariate Analysis, 99*, 191-214. doi:10.1016.j.jmva.2007.06.010

Liu, H., & Motoda, H. (2012). *Feature selection for knowledge discovery and data mining* (Vol. 454). Springer Science & Business Media.

Lütkepohl, H. (2005). *New introduction to multiple time series analysis*. Springer Science & Business Media.

Maddala, G. S. & Rao, C. R., eds., Handbook of Statistics, Vol. 15: Robust Inference. 1997. Elsevier.

Marsland, S. (2001). Online novelty detection through self organization, with application to inspection robotics. Ph.D. thesis, Faculty of Science and Engineering, University of Manchester, UK.

Nouira, K. & Trabelsi, A. (2006). Time series analysis and outlier detection in intensive care data. IEEE ICSP Proceedings.

Pearson, R. K. (2011). *Exploring data in engineering, the sciences, and medicine*. New York: Oxford University Press.

Penny, K. I. & Joliffe, I. T. (2001). A comparison of multivariate outlier detection methods for clinical laboratory safety data. *The Statistician, 50, Part 3,* 295-308.

Penzer, J. & Jong P. (2004). The ARMA model in state space form. *Statistics and Probability Letters, 70*, 119-125.

Preparata, F.P. & Shamos, M.I. (1985). *Computational Geometry-An Introduction*. Springer, Heidelberg.

Qu, J. (2008). Outlier detection based on Voronoi diagram. In *Advanced Data Mining and Applications* (pp. 516-523). Springer Berlin Heidelberg.

Quenouille, M. H. (1949). Problems in Plane Sampling. *The Annals of Mathematical Statistics* **20** (3): 355–375. doi:10.1214/aoms/117772998

Quenouille, M. H. (1956). Notes on Bias in Estimation. *Biometrika* **43** (3-4): 353–360. doi:10.1093/biomet/43.3-4.353

Rousseeuw, P. J., Aelst, S. V., Driessen, K. V., & Gulló, J. A. (2004). Robust Multivariate Regression. *Technometrics*, *46*, 293-305. doi:10.1198/004017004000000329

Rousseeuw, P. J., & Driessen, K. V. (1999). A Fast Algorithm for the Minimum Covariance Determinant Estimator. *Technometrics*, *41*(3), 212-223. doi:10.1080/00401706.1999.10485670

Rousseeuw, P. J., & Leroy, A. M. (1987). *Robust regression and outlier detection*. New York: Wiley.

Schneider, T., & Neumaier, A. (2001). Algorithm 808: ARfit—A Matlab package for the estimation of parameters and eigenmodes of multivariate autoregressive models. *ACM Transactions on Mathematical Software (TOMS), 27*(1), 58-65. doi:10.1145/382043.382316

Tsay, R. S. (2002). *Analysis of financial time series*. John Wiley & Sons.

Tukey, J. W. (1958). Bias and confidence in not quite large samples. *The Annals of Mathematical Statistics* **29**: 614–623. doi:10.1214/aoms/1177706647

Wichern, D. W. & Johnson, R. A. (2007). *Applied Multivariate Statistical Analysis*. Pearson.

Wilks, S. S. (1932). Certain generalizations in the analysis of variance. *Biometrika, 26*, 471-494.

Zivot, E. & Wang, J. (2006). *Modeling financial time series with S-plus*. New York: Springer.

Zwilling, C. E., & Wang, M. Y. (2014). Multivariate Voronoi outlier detection for time series. In *IEEE Healthcare Innovation and Point of Care Technologies Conference*, pp. 300-303.

Zwilling, C. E., & Wang, M. Y. (2015). Achieve precision through feature selection in covariance-based outlier detection. In *IEEE Healthcare Innovation Point of Care Technologies Conference.*

Zwilling, C. E., & Wang, M. Y. (2016). Covariance based outlier detection with feature selection. In *IEEE Engineering in Medicine and Biology Conference,* 4 pages.

Appendix

Appendix A. Feature pairs performance in MVOD extension

Table A.1. Average, maximum and minimum TPR for the set of 55 feature pairs tested with a Voronoi diagram.

Average				Maximum				Minimum		
Label	TPR	Pairs		Label	TPR	Pairs		Label	TPR	Pairs
42	0.78	7 9		6	0.95	1 8		51	0.01	9 11
4	0.77	1 6		32	0.94	5 10		52	0.01	9 12
6	0.77	1 8		3	0.94	1 5		48	0.01	8 11
27	0.76	4 12		14	0.94	3 7		49	0.01	8 12
25	0.76	4 10		24	0.94	4 9		47	0.01	8 10
7	0.76	1 9		27	0.93	4 12		53	0.01	10 11
24	0.76	4 9		36	0.93	6 8		46	0.01	8 9
12	0.75	3 5		21	0.93	4 6		55	0.02	11 12
5	0.75	1 7		4	0.93	1 6		54	0.02	10 12
44	0.75	7 11		42	0.93	7 9		50	0.02	9 10
39	0.74	6 11		40	0.93	6 12		38	0.35	6 10
9	0.74	1 11		9	0.93	1 11		43	0.36	7 10
14	0.74	3 7		25	0.92	4 10		16	0.37	3 9
33	0.74	5 11		33	0.92	5 11		8	0.37	1 10
15	0.74	3 8		18	0.91	3 11		37	0.37	6 9
30	0.74	5 8		12	0.91	3 5		29	0.38	5 7
19	0.74	3 12		2	0.91	1 4		34	0.39	5 12
17	0.74	3 10		1	0.91	1 3		41	0.39	7 8
32	0.74	5 10		31	0.91	5 9		21	0.40	4 6
35	0.74	6 7		19	0.90	3 12		22	0.43	4 7
28	0.74	5 6		7	0.90	1 9		23	0.43	4 8
2	0.73	1 4		17	0.90	3 10		20	0.44	4 5
13	0.73	3 6		30	0.90	5 8		35	0.44	6 7
21	0.73	4 6		15	0.89	3 8		45	0.44	7 12
18	0.73	3 11		5	0.89	1 7		44	0.45	7 11
3	0.73	1 5		26	0.89	4 11		40	0.45	6 12
40	0.73	6 12		35	0.88	6 7		11	0.45	3 4
36	0.73	6 8		10	0.88	1 12		31	0.46	5 9
22	0.73	4 7		37	0.88	6 9		1	0.46	1 3
26	0.73	4 11		13	0.88	3 6		10	0.46	1 12
31	0.72	5 9		44	0.88	7 11		30	0.47	5 8

20	0.72	4 5		22	0.87	4 7		36	0.47	6 8
10	0.71	1 12		39	0.87	6 11		19	0.47	3 12
1	0.71	1 3		20	0.87	4 5		32	0.47	5 10
45	0.71	7 12		45	0.87	7 12		4	0.47	1 6
11	0.69	3 4		11	0.86	3 4		39	0.47	6 11
37	0.66	6 9		28	0.86	5 6		9	0.48	1 11
16	0.59	3 9		23	0.81	4 8		5	0.48	1 7
29	0.59	5 7		16	0.79	3 9		12	0.48	3 5
23	0.58	4 8		29	0.77	5 7		2	0.49	1 4
8	0.56	1 10		8	0.77	1 10		14	0.49	3 7
38	0.53	6 10		41	0.76	7 8		17	0.49	3 10
41	0.53	7 8		38	0.72	6 10		26	0.49	4 11
43	0.53	7 10		43	0.71	7 10		7	0.50	1 9
34	0.53	5 12		34	0.68	5 12		25	0.50	4 10
47	0.04	8 10		47	0.08	8 10		3	0.51	1 5
50	0.04	9 10		51	0.07	9 11		28	0.51	5 6
46	0.03	8 9		46	0.07	8 9		24	0.51	4 9
51	0.03	9 11		50	0.07	9 10		33	0.52	5 11
53	0.03	10 11		53	0.06	10 11		15	0.52	3 8
48	0.03	8 11		52	0.06	9 12		13	0.52	3 6
52	0.03	9 12		48	0.06	8 11		27	0.52	4 12
55	0.03	11 12		55	0.05	11 12		42	0.53	7 9
54	0.02	10 12		49	0.03	8 12		6	0.53	1 8
49	0.02	8 12		54	0.03	10 12		18	0.54	3 11

Table A.1 (cont.). Average, maximum and minimum TPR for the set of 55 feature pairs tested with a Voronoi diagram.

Appendix B. Further evaluation of covariance based method

The following description of figure layouts applies to Figure B.1, Figure B.2, Figure B.3 and Figure B.4, below. Each figure contains an upper, middle and lower panel.

Figure B.1 presents all five outlier conditions with 5 outliers; Figure B.2 presents all five outlier conditions with 10 outliers; Figure B.3 presents all five outlier conditions with 15 outliers; and Figure B.4 is all outlier conditions—5, 10 and 15 outliers. All of these figures are averaged across all magnitudes—1, 2, 3, 4 and 5.

The upper panel shows the pattern of convergence for the determinant of the covariance matrix for each feature. The x-axis is the number of observations. While the time series had 100 observations, these figures are truncated to show greater discrimination among the plotted features. The pattern in the figures presented extends across the full set of observations. The maximum value of the x-axis varies for each figure because the optimal location of the bend (the point at which the plots level out horizontally) depends on the outlier condition. So for Figure B.1, there were only 5 outliers so if a feature does a good job of detecting outliers the bend should occur at 5 outliers. The y-axis is the determinant of the covariance of the time series. Notice that the range varies across Figure B.1, Figure B.2, Figure B.3 and Figure B.4 (upper panels). The range of the y-axis varies systematically as a function of the number of outliers. Figures with more outliers (5 versus 10 versus 15) have a correspondingly larger value on the y-axis initially. However, irrespective of the number of outliers, and their magnitudes, good features always converge to the same value.

The middle panel is the ROC plot for all features for the same outlier condition as the upper panel. The x-axis is the False Positive Rate (always constrained between 0 and .1) and the True Positive Rate (always constrained between 0 and 1).

The lower panel is a bar plot of the average value of the integral under the curve for the determinant of the covariance. The x-axis represents feature labels and the values on top of the bars are the areas.

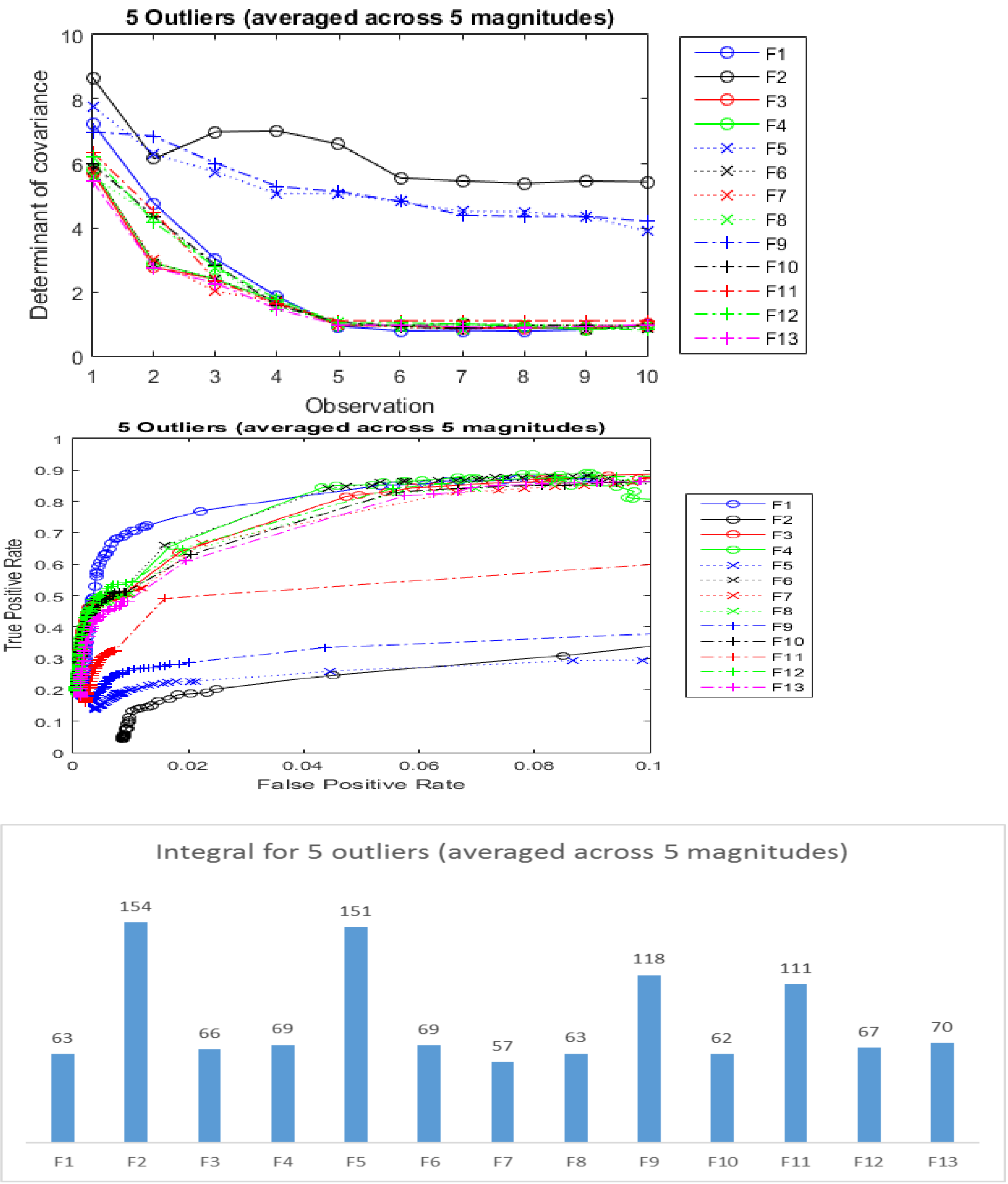


Figure B.1. Convergence plot (upper panel), ROC plot (middle panel) and integral (lower panel) for outlier conditions with 5 outliers averaged across magnitudes 1, 2, 3, 4, and 5.

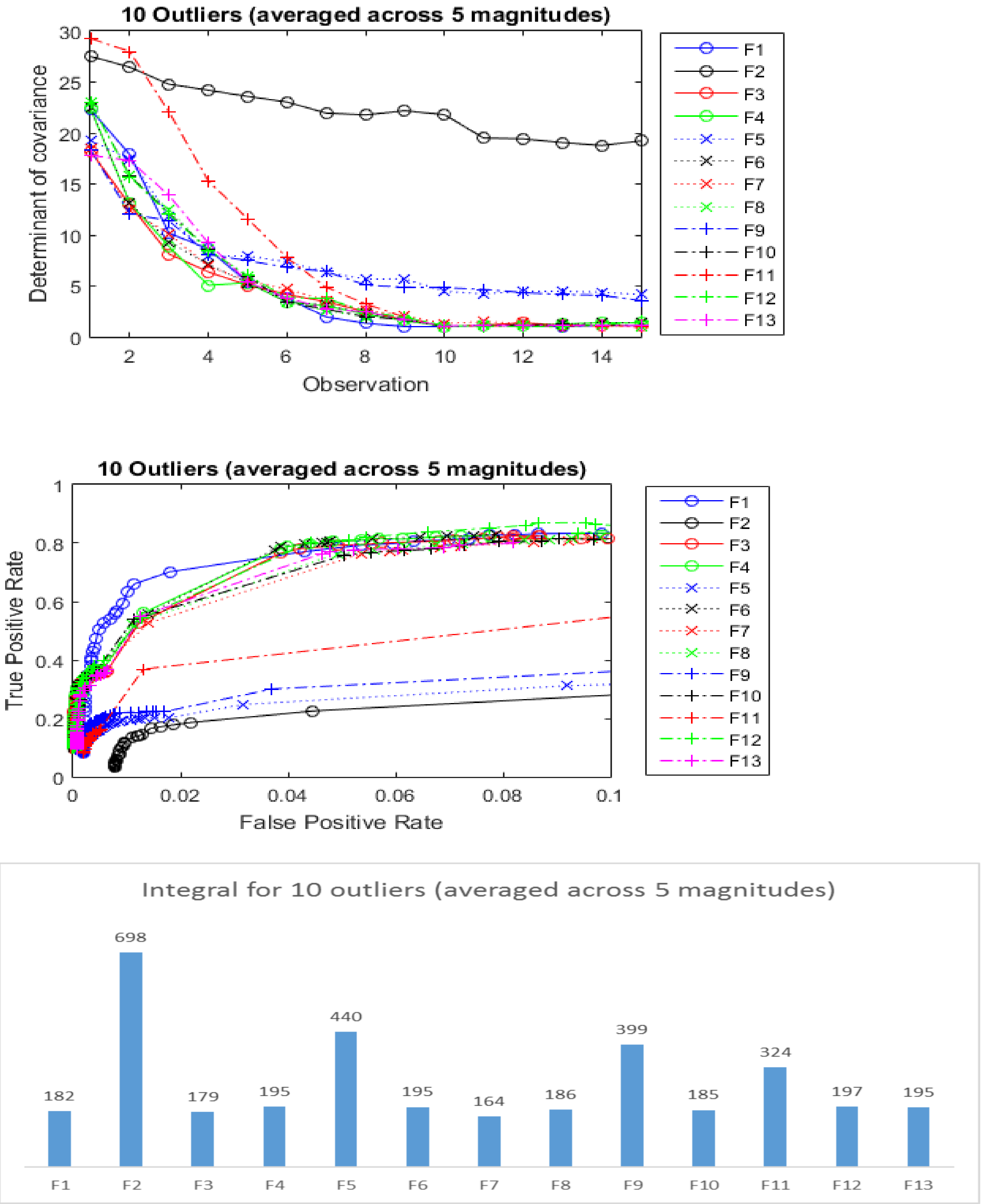


Figure B.2. Convergence plot (upper panel), ROC plot (middle panel) and integral (lower panel) with 10 outliers averaged across magnitudes 1, 2, 3, 4, and 5.

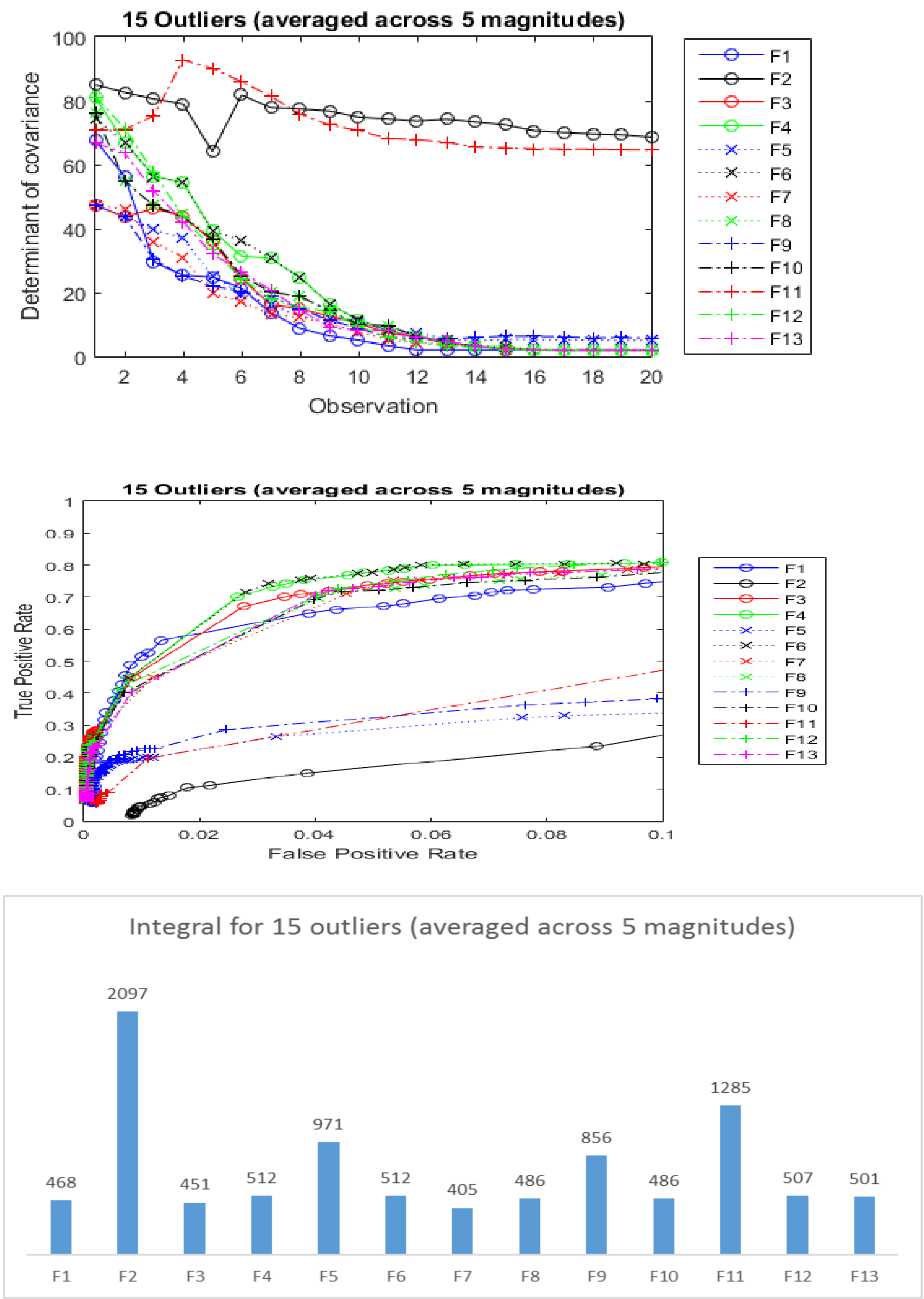


Figure B.3. Convergence plot (upper panel), ROC plot (middle panel) and integral (lower panel)) for outlier conditions with 15 outliers averaged across magnitudes 1, 2, 3, 4, and 5.

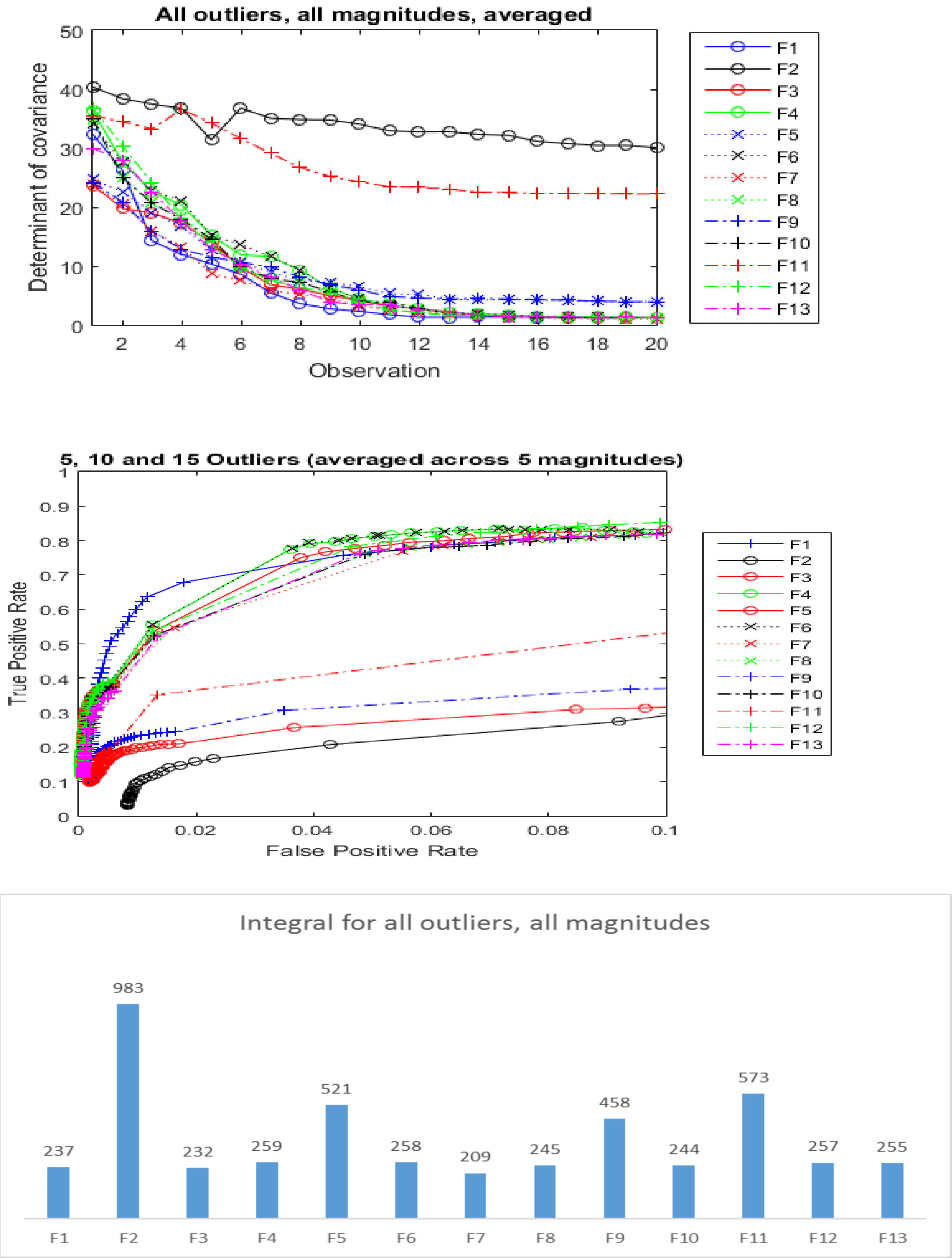


Figure B.4. Convergence plot (upper panel), ROC plot (middle panel) and integral (lower panel) for all outlier conditions –5, 10 or 15 outliers and magnitudes 1, 2, 3, 4, and 5—averaged.

Table B.1. Log ratios of first 10 values for 5 outliers with magnitude 1.

1	3	4	6	7	8	10	11	12	13
0.10	0.13	0.13	0.13	0.10	0.14	0.15	0.09	0.16	0.15
0.05	0.08	0.06	0.05	0.14	0.09	0.08	0.07	0.04	0.02
0.09	0.06	0.08	0.09	0.04	0.07	0.06	0.08	0.07	0.04
0.11	0.07	0.06	0.06	0.05	0.05	0.08	0.02	0.04	0.08
0.04	0.06	0.05	0.08	0.07	0.04	0.05	0.00	0.04	0.08
0.02	0.01	0.01	0.02	0.10	0.01	-0.03	0.00	0.05	0.06
0.08	0.07	0.09	0.06	0.00	0.05	0.05	0.00	0.04	0.04
0.06	0.02	0.07	0.05	0.00	0.01	0.08	0.00	0.03	0.06
0.05	0.09	0.05	0.03	0.05	0.08	0.03	0.00	0.05	0.01
0.06	0.10	0.05	0.06	0.08	0.01	0.00	0.00	0.05	0.03

Table B.2. Log ratios of first 10 values for 10 outliers with magnitude 1.

1	3	4	6	7	8	10	11	12	13
0.10	0.09	0.10	0.09	0.01	-0.01	0.01	0.04	0.01	0.06
0.08	0.11	0.11	0.11	0.17	0.08	0.10	0.04	0.10	0.06
0.11	0.07	0.09	0.05	0.05	0.09	0.02	0.08	0.04	0.05
0.09	0.13	0.07	0.11	0.01	0.04	0.06	0.06	0.05	0.09
0.10	0.06	0.02	0.09	0.06	-0.01	0.00	0.05	0.04	0.05
0.07	0.07	0.12	0.06	0.04	0.05	0.05	0.01	0.06	0.04
0.08	0.00	0.05	0.03	0.08	0.08	0.06	0.00	0.02	0.06
0.06	0.16	0.07	0.08	0.03	0.07	0.08	0.01	0.07	0.10
0.05	0.03	0.04	0.06	0.07	0.03	0.03	0.00	0.03	0.07
0.02	0.07	0.06	0.07	0.06	0.05	0.07	-0.02	0.06	0.01

Table B.3. Log ratios of first 15 values for 5 outliers with magnitude 1.

1	3	4	6	7	8	10	11	12	13
0.03	0.10	0.07	0.07	0.08	0.05	0.03	0.02	0.00	0.06
0.15	0.10	0.09	0.10	0.08	0.04	0.04	0.05	0.08	0.14
0.05	0.06	0.11	0.11	0.06	0.05	0.09	0.05	0.02	0.10
0.00	0.12	0.05	0.07	0.05	0.06	0.08	0.07	0.04	0.02
0.15	0.09	0.12	0.07	0.04	0.08	0.09	0.05	0.08	0.00
0.02	0.06	0.03	0.08	0.08	0.10	0.03	0.04	0.06	0.06
0.10	0.07	0.02	0.01	0.03	-0.05	-0.04	0.03	0.03	0.10
0.04	0.01	0.01	0.01	0.03	0.12	0.09	0.01	0.06	0.06
0.03	0.06	0.03	0.04	0.08	0.03	0.04	0.00	0.01	0.01
0.06	0.01	0.03	0.04	0.08	0.01	0.08	0.02	0.10	0.07

Appendix C. Additional features in covariance based method

While there is no upper limit to the number of features one could implement, some results provided thus far strongly suggest that certain operations on the covariance matrix yield features that are quite effective at detecting outliers. Principal component analyses also operate on the covariance matrix, and given that one of the prerequisites for a feature in this study is that it compresses the multivariate time series data observations into a univariate feature vector (i.e. a data reduction), it makes sense to see how some features built from the steps of PCA would perform.

Principal component analysis aims to identify a set of linearly uncorrelated variables (Wichern & Johnson, 2007) through an orthogonal transformation of the originally correlated observations. This transformation proceeds sequentially such that the first principal component accounts for the most variance, the second component accounts for the next largest variance, etc. Geometrically, we can think of PCA as fitting a multi-dimensional ellipsoid to the data, where each axis represents one principal component and the respective axis length of each dimension reflects the amount of variance accounted for in that component. We can leverage this logic for the identification of outliers. For the specific features then, the three eigenvalues were computed (because the data is 3-dimensional). These are listed in Table C.1.

All of the same steps for the previous thirteen features (F1 to F13) were carried out on these new features derived from eigenvalues. So the feature was computed for each time series data set (replicated twenty-five times) for each of fifteen outlier condition. The original ordered time series was interpolated according to the prediction of each feature vector. After each correction the determinant of the covariance matrix (and the associated log ratio) were computed, yield information and curves identical to those presented in Figure 5.4 and Figure 5.6. Figure C.1 shows the plot of the determinant of the covariance matrix (averaged across all 15 outlier conditions) and we see that the features show good convergence, like we see of the good features in Figure 5.4. Figure C.2 provides further supporting information that these new features do well. The bar graph represents the area under the curve for Figure 5.5. A smaller value is better because it means the convergence plot drops more quickly. In comparing the values of Figure C.2 with Figure 5.5, the best features among the set of 13 are F1 and F6 with integral values of 237 and 258. The individual eigenvalues (F14, F15 and F16) have decent integral values (344, 340 and 410, respectively), and they are not as small as the really good features which have even smaller values.

Table C.1. List of five new features that relate to eigenvalues.

	Feature Description
F14	Largest eigenvalue
F15	Second largest eigenvalue
F16	Smallest eigenvalue

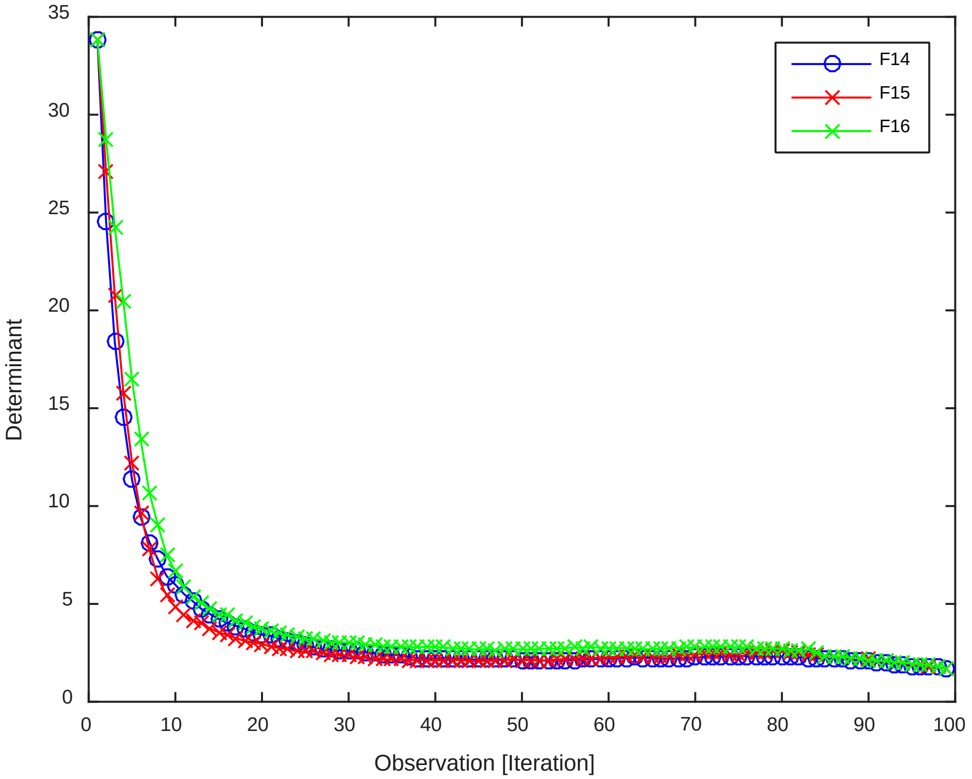


Figure C.1. Convergence plot for F14, F15 and F16. x-axis is observation (or iteration) and the y-axis is the determinant of the covariance matrix. Results are from average across all 15 outlier conditions.

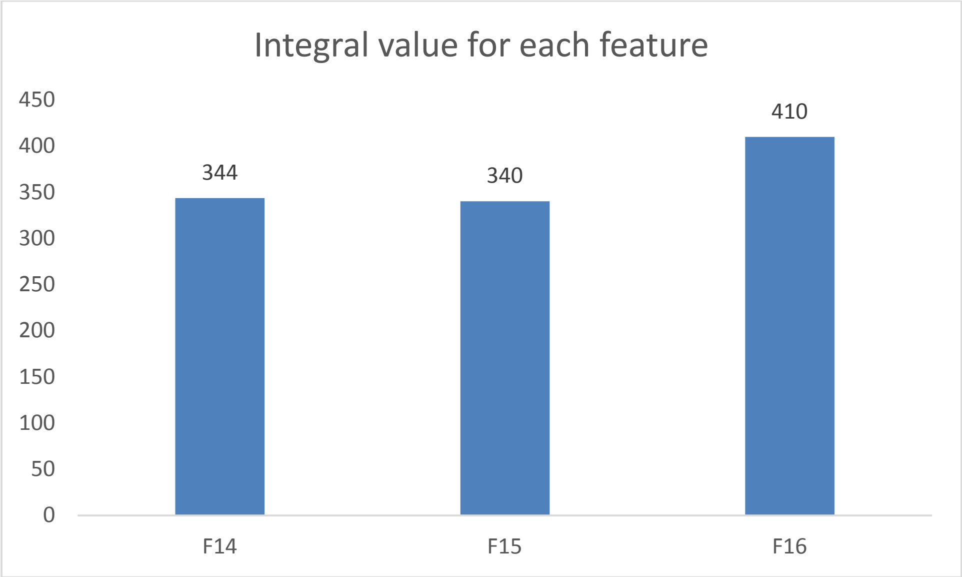


Figure C.2. Integral value averaged across 15 outlier conditions for F14, F15 and F16. y-axis represents the value of the integral and x-axis is the feature label. Results are from the average of 15 outlier conditions.

Appendix D. Performance of all features in covariance based method

Most of the results presented in this thesis are averaged across all 15 outlier conditions. In looking at each individual condition separately, sometimes features that do as well on the average overall actually do quite well for some cases and the best overall features don't necessarily do best for each individual condition. This reiterates the important point that seeking to find a so-called best feature is probably not a wise pursuit. Rather, what is more desirable is an algorithm that can assist one in finding a feature which is optimal for the data set at hand.

As a specific example, F7 in Table D.1 (with values representing the areas under the curves for the covariance of the determinant plots) is probably the best feature for almost all individual conditions and the best overall. This feature even outperforms the product of the eigenvalues. F7 is very simple and fast to compute: the determinant of the covariance matrix of the data. So from this point of view, we see that the proposed method for detecting outliers is effective at allowing a head to head contest among different features.

Table D.1. Integral values for 18 features for all outlier conditions separately in covariance based method. The minimum and maximum for each outlier condition are listed at the bottom of the table and the average across all outlier conditions for each feature is listed in the right-most column.

	5-1	5-2	5-3	5-4	5-5	10-1	10-2	10-3	10-4	10-5	15-1	15-2	15-3	15-4	15-5	Ave
F1	33	37	49	76	118	39	69	130	205	466	52	101	243	631	1313	237
F2	38	50	91	197	396	53	110	349	579	2401	73	172	522	2289	7431	983
F3	33	37	52	82	124	40	71	136	214	436	50	100	239	613	1255	232
F4	33	38	53	84	135	40	72	141	231	493	51	102	263	688	1456	259
F5	52	69	116	196	324	65	124	290	536	1186	80	219	533	1287	2735	521
F6	33	38	53	84	135	40	72	141	231	492	51	102	263	688	1454	258
F7	30	37	48	72	98	39	65	122	204	391	52	95	223	545	1110	209
F8	33	40	52	78	109	44	68	134	223	459	58	106	258	642	1366	245
F9	43	53	86	145	260	54	109	244	482	1105	66	169	401	1085	2561	458
F10	33	40	52	78	109	43	68	133	224	458	58	106	257	641	1368	244
F11	66	72	107	130	181	87	133	208	516	675	105	219	537	1335	4227	573
F12	36	42	55	82	119	46	77	141	233	488	61	107	273	677	1418	257
F13	37	41	54	84	135	45	75	139	228	486	57	106	265	672	1404	255
F14	40	52	75	131	261	50	100	198	378	688	66	157	370	870	1716	344
F15	40	55	72	121	171	50	98	213	309	638	65	142	369	842	1918	340
F16	51	71	92	129	176	69	121	209	345	782	91	197	483	1083	2246	410
Min	30	37	48	72	98	39	65	122	204	391	50	95	223	545	1110	
Max	66	72	116	197	396	87	133	349	579	2401	105	219	537	2289	7431	