Unsupervised Detection of Anomalous Text

by

David Guthrie

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

Department of Computer Science
University of Sheffield

July 2008

Thesis advisor

Dr. Robert Gaizauskas

Author

David Guthrie

Unsupervised Detection of Anomalous Text

Abstract

This thesis describes work on the detection of anomalous material in text without the use of training data. We use the term *anomalous* to refer to text that is irregular, or deviates significantly from its surrounding context. In this thesis we show that identifying such abnormalities in text can be viewed as a type of outlier detection because these anomalies will differ significantly from the writing style in the majority of the data. We consider segments of text which are anomalous with respect to topic (i.e. about a different subject), author (written by a different person), or genre (written for a different audience or from a different source) and experiment with whether it is possible to identify these anomalous segments automatically. Five different innovative approaches to this problem are introduced and assessed using many experiments over large document collections, created to contain randomly inserted anomalous segments. In order to identify anomalies in text successfully, we investigate and evaluate 166 stylistic and linguistic features used to characterize writing, some of which are well-established stylistic determiners, but many of which are original. Using these features with each of our methods, we examine the effect of segment size on our ability to detect anomaly, allowing segments of size 100 words, 500 words and 1000 words. We show substantial improvements over a baseline in all cases for all methods, and identify a novel method which performs consistently better than others and the features that contribute most to unsupervised anomaly detection.

Contents

List of Figures

List of Tables

Dedicated to my father Joe,
my mother Louise,
and my brother Chris.

Chapter 1

Introduction

For he that knows the ways of nature will more easily observe her deviations; and on the other hand, he that knows her deviations will more accurately describe her ways.

-Sir Francis Bacon in *Novum Organum* (1620)

1.1 Defining Anomaly

This thesis is about identifying the unusual use of language. Obvious questions that follow from this are "What makes language use unusual?" and "how can we get a consensus on what kind of language use is unusual?" After all, language that is unfamiliar to one person may be completely commonplace to others. Likewise, writing that is unconventional in one context can seem normal in another. Language use can be domain-centric and subjective and often is not unusual in and of itself, but because it is different or stands out from the language around it. In fact, when viewed this way, any language or writing can be considered unusual in the right circumstances

or 'context'. The central idea behind this thesis is that unusual language use can be viewed as a type of anomaly because it is out of place in its context and as such we can approach the problem of identifying unusual language as a type of anomaly detection.

Anomaly is typically defined as: "something that deviates from what is standard, normal, or expected" [Burchfield, 1971] and the word is often used to express incongruity, irregularity, or inconsistency. We use the term to describe exactly this notion of abnormality, but within text. Text that is unexpected, irregular, or inconsistent is for us anomalous because it breaks the pattern of its context. This thesis investigates whether it is possible to identify anomalous language automatically because it deviates from its context.

During the course of this research, it became clear that, depending on how broad we made our definition of anomalous text, almost any text could be seen as an anomaly for one reason or another in the proper context. We have attempted throughout the research to keep our anomaly detection techniques as broad as possible, but this work specifically addresses the detection of four types of textual anomalies:

- Authorship
- Genre
- Style of writing
- Emotional tone of writing

We use the term *authorship* anomalies to describe pieces of text that deviate from their contexts because they were written by a different author. A *genre* anomaly is defined by text that has a different external use or purpose (for instance narrative

versus argumentative writing). Closely related to genre is the notion of *style*, where text is anomalous because of the way it is written, so the word choice, grammar, voice, register, etc. play a part in determining the style of the text. Lastly the *emotional tone* of writing can be anomalous because it encapsulates a different sentiment or feeling, for instance, positive versus negative writing or angry vs good-humored. A detailed discussion of how these text types are defined and of related work in these areas is presented in Chapter 3 of this thesis.

1.2 Motivation

In everyday life, situations abound that rely on the ability of computers to detect differences from what is normal or expected. Credit card companies identify possible fraud by detecting spending patterns that differ from what is 'normal' for a given cardholder [Burge and Shawe-Taylor, 1997; Bolton and Hand, 2002; Fawcett and Provost, 1997] and network analysts detect possible attacks by spotting network traffic that is out of the ordinary [Denning, 1987; Kruegel and Vigna, 2003]. The principal motivation for this research was the development of technologies to similarly detect anomalies in text. This is an interesting problem because anomalies in text can be of many different types and extremely varied. For text we would like techniques that make no assumptions about how many anomalies will be present or what type of anomaly will be present. Professors, for example, can regularly identify plagiarism in a paper by noticing writing that is 'abnormal' compared to the rest of the paper, even if there are multiple plagiarized sections from different sources. An automatic technique for detecting unusual writing (like plagiarism) that could similarly spot

several different dimensions of anomaly and multiple occurrences of different types would be extremely useful.

Finding these textual anomalies is important for a range of practical applications from detecting paragraphs in a single document that are plagiarized to improving the quality and integrity of data sets by finding textual data in collections that is inconsistent with the rest of the collection. An important aspect of our research was also that these tasks should be able to be performed without having to gather large collections of corpora and train a technique to detect a single type of anomaly, but would be able to detect a wide range of anomalies in any new domain encountered.

A list of applications where automatic detection of textual anomaly would be beneficial includes:

Plagiarism, Disputed Authorship, and Text Reuse: Anomaly detection for text would be useful to automatically identify documents where an author uses sections of work which are not his own, as in the case of plagiarism. This is an obvious case for the use of anomaly detection because plagiarized passages should be unusual when compared to the writing in the rest of the document. Anomaly detection methods would be beneficial to spot plagiarized passages in a document because the writing is odd and not, as it the case with most current methods for detecting plagiarism [Maurer et al., 2006; Bull et al., 2001], by using outside resources to try and actually find the source material that was copied. Similarly, it would be helpful where collaborative writing has taken place to identify areas in the text that appear incongruent and should possibly be rewritten.

Anomaly detection would also be beneficial to the problem of identifying *text reuse* [Gaizauskas et al., 2001; Clough et al., 2002; Wilks, 2004], where text is directly copied from one source for use in another. Text reuse has mostly been defined in the context of newspapers copying newswire for use in production articles (either verbatim or slightly adapted) and it is desirable to be able to automatically identify text in this scenario that has been reused. Given the capability to detect anomalies we would not need to have a copy of the newswire that has been copied from to compare newspapers against, but rather would attempt to identify reused text because it does not fit in with the writing in the rest of the article.

Forensic linguists is another area that lends it self to the application of textual anomaly detection because it deals with a related problem of attributing authorship to texts [Coulthard, 1992, 1994; McMenamin, 2002]. Sometimes sections of a text have disputed authorship as is the case when defendants claim that their statements, taken down by police, have be altered [Coulthard, 2004]. The language in these altered portions of the statement can be viewed as a type of authorship anomaly and it would be useful if these could be identified automatically.

Improving the Homogeneity of Corpora: Another goal of this research is to improve the quality and integrity of corpora by automatically identifying pieces of text that should not be present, so that they may be removed from these collections of text and thus make them more homogeneous.

The availability of a wide range of electronic corpora and lexical resources has

had a dramatic impact on the study of languages and led to many of the most exciting advances in natural language processing and computational linguistics. Question answering, language modeling [Goodman, 2001], automatic speech recognition, text classification, information extraction [Gaizauskas and Wilks, 1998], machine translation and many other research areas have benefited greatly from availability of large reliable corpora. Corpora play such an important role in these fields that the selection, quality, and size of corpora can have much more impact on system performance than the choice of a machine learning technique or the method used to perform that task.

The creation and validation of corpora has generally relied on humans, but this can be a very expensive process and it is becoming increasingly common, in research, to use more automated methods for corpus generation. Many automated techniques [Hassel, 2001; Chen and Dumais, 2000; Sato and Sato, 1999] make use of the vast amount of text accessible on the World Wide Web to construct corpora that specifically meet the needs of an application. For instance, it is now possible to construct a corpus of editorials from newspapers, a corpus of Swedish news stories, a corpus about infectious diseases, or a corpus of movie reviews relatively quickly and cheaply. The construction of these corpora usually involves some form of information retrieval or automated scraping of web pages to gather relevant data, which can lead to errors in precision; where documents are gathered that should not have been. It is difficult to validate these corpora, because this usually involves some form of human interaction, but automatic techniques for this type of validation or the identification of irrelevant pages, or

outliers, are immediately useful.

Anomalous text in these corpora may not have been introduced in the gathering stage, but at an earlier time. It is possible that because the corpus is taken from the Web it may naturally contain anomalies. A corpus that has been gathered from an online bulletin board or wiki (such as the collaborative encyclopedia Wikipedia[1]) may contain undesirable information or anomalies because text may typically be added or edited by anyone on the Web. While this collaborative editing is the strength of these sites, allowing information to be continually checked for factuality by a large number of people, the corpus is constantly changing and at any time can contain entries that might be considered spam, such as advertising or gibberish messages, or even, more subtly, information that is an opinion rather than a fact, such as rants posted about political figures. It would be very helpful if these intrusions could be identified automatically and removed from corpora so that applications that make use of them (a question answering system for example) do not propagate these errors.

Kilgarriff [1997, 2001], Kilgarriff and Rose [1998] and Sahlgren and Karlgren [2005] have explored methods to measure this notion of homogeneity within a corpus and these may be important for determining in what circumstances anomaly detection is appropriate. It is likely that anomaly techniques would only apply to corpora that have a high level of homogeneity.

Atypical or Intentionally Deceptive Textual Data: This includes identifying *personal* email versus *work* email (and other textual spam that might occur in

[1] `http://www.wikipedia.org`

email) as well as detecting ranting or subversive language on websites. A growing problem that would particularly benefit from anomaly detection is the identification of machine generated spam emails. This type of unsolicited email spam often contains sentences or sequences of words that have been randomly drawn from a corpus and strung together into something resembling sentences and paragraphs to elude spam filters[2] (the actual advertisement or 'hook' is often an image). These messages can be generated at great volume (and can all be unique) so they get past most spam filters, which are trained on a collection of spam and so can not possibly have seen these unique random sentences before. These emails would, however, have stylistic qualities (apart from the vocabulary used) that make them stand out from other emails you have received and so we could view them as a type of anomaly to be detected. (As similar problem has been discovered on the web where machine generated spam webpages are set up which contain grammatically well formed sentences [Fetterly et al., 2004, 2005], but exist solely for the purpose of linking to other websites and thus increasing their rank within search engines.)

It is also possible that anomaly detection may be useful for the detection of deceptive or hidden messages. These might take the form of a short message hidden in the middle of another longer message, or a message encoded to look like English, but which is nonsensical. Text could appear strange or nonsensical because, for instance, the ordering of the words was used to encode the message. (This is a type of *steganography*, from the Greek for "hidden writing"; for a

[2] `http://www.process.com/techsupport/spamtricks.html`

discussion of this and other techniques used to hide messages in text see Wayner [2000]). Another kind of deception might be passages in documents which are deliberately misleading or where opinion or speculation is asserted as though it were fact. These tasks may be difficult, even for humans (as in the case of classifying business and personal email [Jabbari et al., 2006]), but we believe that they can be aided by unsupervised anomaly detection.

Some of these problems have been investigated previously (never in the context of anomaly detection), but they have always used either large segments (i.e. greater than 1000 words) or *a priori* knowledge that allows for more traditional classification using training data. Our motivation was to develop techniques that could be used for these tasks on much shorter segments of text and across different domains, without the need to gather corpora or other domain specific resources.

1.3 Focus and Contribution of Thesis

1.3.1 Unsupervised Methods

A strategy for attempting to identify whether things are abnormal or anomalous, might start by gathering instances of things that are "anomalies‘" and also instances of data that are considered to be "normal". A classification approach could then be used to decide if previously unseen instances should be labeled normal or anomalous. Often, however, one does not have a reliable definition of what it means to be normal or anomalous and rather must look at other ways to spot anomaly.

Some work on anomaly detection (sometimes called *novelty* detection) has as-

sumed only the existence of a collection of data that defines "normal", which is used to model the normal population; methods are then developed to identify data that differs significantly from this model [Markou and Sing, 2003; Song et al., 2007]. It is possible to take a similar approach to detecting anomalies in text (a form of one-class classification [Koppel and Schler, 2004; Koppel et al., 2006]), but this requires building up a large corpus of "normal" data for any task (or domain) encountered. In order for this model to be useful, the "normal" collection should be representative and thus should not contain anomalies. This process can be expensive and time consuming for new domains and often impossible for some tasks due to the nature of the data. We chose a different way to attack this problem and in the rest of this thesis we focus on the challenging anomaly detection scenario where we assume that we have no *a priori* knowledge of what it means to be "normal" language. The techniques we investigate for this task do not make use of any training data for either the normal or the anomalous populations and so are referred to as unsupervised.

In this scenario, the task is to find which parts of a collection or document are most anomalous with respect to the rest of the collection. For instance, if we had a collection of news stories with one fictional story inserted, we would want to identity this fictional story as anomalous, because its language is anomalous with respect to the rest of the documents in the collection. In this example we have no prior knowledge or training data informing us of what it means to be "normal", nor what it means to be news or fiction. As such, if the collection were switched to be fiction stories and one inserted news story then we would hope to identify the news story as anomalous with respect to the rest of the collection, because its language use differs

from the bulk of the collection.

We approach the unsupervised anomaly detection task slightly differently than we would if we were carrying out unsupervised classification of text [Willett, 1988; Manning and Schütze, 1999; Oakes, 1998]. In unsupervised classification (or clustering) the goal is to group similar objects into subsets; but in unsupervised anomaly detection we are interested in determining which segments are most different from the majority of the document. The techniques used here do not assume anomalous segments will be similar to each other: therefore we have not directly used clustering techniques, but rather developed methods that allow many different types of anomalous segments within one document or collection to be detected[3]. Our approach to this problem is more closely related to the task of statistical outlier detection (discussed in detail in Chapter 2) where we would like to identify data that does not fit with the rest of the population.

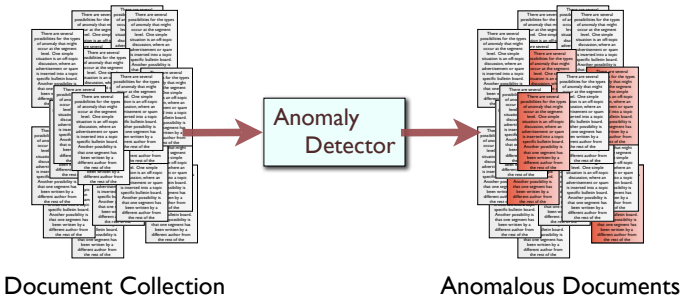


Figure 1.1: Detecting Anomalous Documents

[3]While clustering is not directly appropriate for anomaly detection in text was useful as an exploratory tool for determining how to best characterize text and choose features, see Appendix A where we performed some initial experiments using clustering on different types of data.

1.3.2 Segment level focus

A task well suited to unsupervised anomaly detection, and the focus of the experimental part of this work, is the identification of segments (or paragraphs) in documents that are anomalous (with respect to the rest of the document). While we focus exclusively on segments of text, it should be pointed out that all techniques, methods, and analysis in this work also apply to larger pieces of text (such as identifying an anonymous document in a collection of documents). Identifying anomalous segments is typically more difficult than anomalous documents because in collections of documents text length is normally longer and you will see more repetition of phenomena and thus have a much better representation of a text. This segment level concentration steered us to make choices and develop techniques that are appropriate for characterizing and comparing smaller segments.

There are several possibilities for the types of anomaly that might occur at the segment level. One simple situation is an off-topic discussion, where an advertisement or spam is inserted into a topic-specific bulletin board. Another possibility is that one segment has been written by a different author from the rest of the document, as in the case of plagiarism. Plagiarism is notoriously difficult to detect automatically when the source of the plagiarism cannot be found [Woolls and Coulthard, 1998] (using a search engine like Google or by comparison to the work of other students or writers.) In addition, the plagiarized segments are likely to be on the same topic as the rest of the document, so lexical choice often does not help to differentiate them. It is also possible for a segment to be anomalous because of a change in tone or attitude of the writing. The goal of this work is to develop a technique that will detect an

anomalous segment in text without knowing in advance the kind of anomaly that is present.

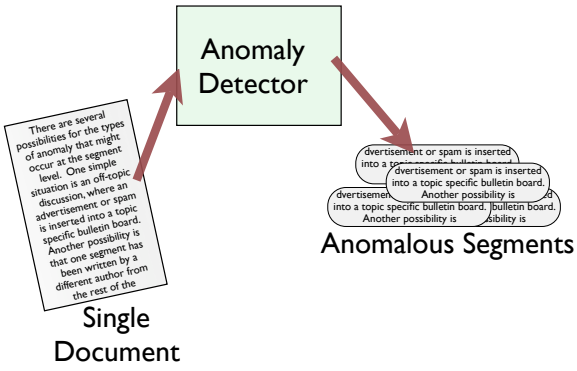


Figure 1.2: Detecting Anomalous Segments

Unsupervised detection of small anomalous segments cannot depend on the strategies for modeling language that are employed when training data is available. With a large amount of training data, we can build up an accurate characterization of the words in a document. These language-modeling techniques make use of the distribution of the vocabulary in a document and, because language use and vocabulary are so diverse, it is necessary to train on a considerable amount of data to see the majority of cases (of any specific phenomenon) that might occur in a new document. If we have a more limited amount of data available, as in the segments of a document, it is necessary to characterize the language using techniques that are less dependent on the actual distribution of words in a document and thus less affected by the sparseness of language. In this unsupervised anomaly detection scenario we make use of techniques that employ some level of abstraction from words and focus on characterizing style, tone, and classes of lexical items.

1.3.3 Contribution of Work

The central question investigated in this thesis is whether it is possible to automatically identify text that deviates from its context. This is an exciting and interesting research question that has many practical applications. The work presented here shows that we can view these deviations in text as a type of anomaly and that these can be successfully detected using automatic unsupervised techniques.

We introduce the notion of outliers to the analysis of anomalies in text and explore and test five different approaches to the problem, some of which are related to methods used in statistical outlier detection and unsupervised learning and some of which are novel. We demonstrate that a novel technique which measures a piece of text's distance from its complement (the union of all other prices of text) can most accurately identify anomalous language.

Several thousand experiments were preformed with all methods and we measured their ability to detect different types of anomalies in various kinds of corpora using test sets automatically created by taking documents and artificially inserting text that differs because of authorship, tone, topic, or style. We show that all techniques for detecting anomalies are most accurate when there is large variation in the writing style or genre of the anomalous text (as opposed to the topic, tone, or authorship). Additionally this work studies the impact that the size of anomalies has on our ability to detect them and shows a substantial improvement as the size of anomalies increases.

This thesis also investigates and evaluates 166 stylistic and linguistic features based on their ability to characterize and differentiate between different types of text. This set of features represents a union of the most popular features used in related

research and a few novel variations of our own. We rank all these features by their usefulness in anomaly detection and show a subset of these features are an excellent choice for identifying a wide range of anomalies.

1.4 Thesis Outline

This thesis can be divided into three parts. Chapters 2 and 3 discuss essential background for understanding the problem and techniques employed and how they compare to previous work. These chapters give a comprehensive review of outlier detection in statistics as well as research in natural language processing and linguistics related to our work. Chapters 4 and 5 cover the methods and techniques we have introduced for characterizing language and identifying anomalies, some of which are novel and some are based on procedures from statistical outlier detection. Chapter 6 and 7 give extensive experimental results over different collections, using different methods, and look at how techniques can be optimized and what are the most important aspects of the techniques to different types of anomaly detection. The final concluding chapter summarizes the result and contributions of the thesis. The appendices contain additional material including further results and a review of the corpora used for experiments.

Chapter 2

Outlier Detection in Statistics

2.1 Overview

The term *outlier* is used in statistics to describe any observation in a data set that differs significantly form the other observations in that data set. These outlying observations are often studied because they are far from, or in some way inconsistent with, the majority of the data and so might be indicative of unusual phenomena or errors. The fact that outliers do not fit with the rest of the data also makes their identification crucial in many areas of statistical analysis where models are constructed over data and used to make predictions [Mosteller and Tukey, 1977]. It is important that these models are not unduly influenced by possible mistakes or errors. This has lead to a wealth of research in statistics into the problem of spotting outliers and how best to model data that might contain outliers. In this chapter we present a review of this research and the techniques used. We begin with a review of univariate outlier detection and then look at the more active research field of

multivariate outlier detection.

Finding unusual, or anomalous text can be viewed as a type of outlier identification, where we are attempting to determine if pieces of text are outliers with respect to the rest of the textual data. A problem comes in how best to characterize text, so that we can measure the differences between texts and use these differences to determine if some texts are outliers (we would also like these measurements to have some linguistic relevance). How to characterize text is intertwined with the problem of how to detect 'outliers' in text, but we will hold off on the discussion of how to deal with text until Chapter 4, so that in this chapter we can concentrate on a review of techniques used for the detection of outliers in statistical data.

This chapter starts with an introduction to the problem of outlier detection by looking at identifying outliers in observations of one variable (univariate outliers). This problem is relatively simple to understand and we present an overview of approaches and illustrate the problems that can arise in outlier detection and how they are dealt with in the univariate case. The next section focuses on outlier detection in data where each observation is made up of more than one variable (multivariate outliers). The basic idea is similar to univariate outlier detection, but this problem also requires understanding the interaction of the multiple variables. We describe in detail the common approaches to this problem and their limitations. In the last section of this chapter we focus on the detection of outliers where the number of variables for each observation is very large, so large that the number of variables could be greater than the number of observations. This is a special case of multivariate outlier detection and requires different methods than are used in typical multivariate outlier

detection (where there are many more observations than variables). We are particularly interested in this last type of outlier detection because these methods are well suited to the detection of anomalies in text. We describe several methods in detail and give two that will be compared with other approaches for detecting anomalies in text in Chapters 5 and 6.

2.2 Univariate Outliers

2.2.1 Outlier detection assuming Gaussian Data

The most basic type of outlier detection is so called univariate outlier detection, where observations are of a single variable. The setup is simple, we have set of observations (that are single values) and would like to identify any of these observations that are very far away from the other observations. In this section we will refer to a vector of n such observations as $\mathbf{x} = (x_1, x_2, ..., x_n)^T$. In order to measure how "far away" observations are from the rest of the data, it is useful to have estimates of the center of a data set and some indication of how spread out the values are. These are often referred to, respectively, as the *location* and *scale* estimates of a population and these estimates form the basis for most outlier detection techniques. The classical method of estimating the location (or center) of a set of data is the sample *mean*, $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$. This is the average value in the data and we would expect outliers to be far from this value, where far is determined by the scale of the data. The classical method of estimating scale is the sample *standard deviation* $\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$. A simple method of identifying outlying observations using

these estimates, if we make the assumption that our data follows a normal (Gaussian) distribution, Φ, is to measure how many standard deviations each observation is from the mean. This is an observations z-score.

$$z_i = \frac{|x_i - \bar{x}|}{\sigma}$$

When this simple measurement is used for the detection of outliers by finding the observation with the higest z-score, it is often called the *maximum normalized residual test* or *Grubbs' Test*, after Frank Grubbs, who used it [Grubbs, 1960] to identify outliers and calculated critical values of z for different sample sizes above which an observation should be labeled an outlier. Grubbs' method can be used to detect more than one outlier in a data set, by applying it iteratively, removing one outlying observation every time, but it is unreliable if the data contains many outliers or very large outliers because this will distort the sample mean and standard deviation size (as we explain later).

For Gaussian data, we know that 68.27% of observations should lie within one standard deviation, 95.45% in two standard deviations, and 99.73% lie within 3 standard deviations of the mean, so we might make a rule that all observations that are farther than 3 standard deviations from the mean should be classed as outliers. This would label, on average, only 0.27% of our *good* points as outliers (false positives) in normal data and thus seems fairly reliable. Unfortunately, if the data has more than one outlier this rule will likely not be able to detect all of the outliers (or possibly any outliers). Consider, for example, 10 observations taken from a normal distribution with a mean of zero and a standard deviation size of 1, $\mathcal{N}(0, 1)$, and two large outlying

observations which should clearly be marked as outliers (the numbers 30 and 60).

$$-0.91 \quad -0.64 \quad -0.52 \quad -0.43 \quad 0.06 \quad 0.75 \quad 0.81 \quad 1.19 \quad 1.34 \quad 2.22 \quad 30 \quad 60$$

This sample has a mean of 7.8 and a standard deviation of 18.5, which means that in fact the largest observation, 60, has a z-score of 2.8 and therefore is within 3 standard deviations and would not be labeled an outlier. Even worse is that the other outlier, 30, is within 2 standard deviations and so will not even be labeled an outlier with a cutoff at 2 standard deviations. We see that the use of the standard z-score in this example is not very effective for outlier detection as it will fail to see these observations as extreme. The problem is due to the large estimates of the mean and standard deviation caused by the outlying observations (if we instead had estimated the mean and standard deviation using only the *good* data we would have had no problem detecting the outliers). This inability to see outliers, by virtue of the fact that the presence of outliers can shift the estimates of location and scale, is called the *masking effect* [Bendre and Kale, 1987].

These classical estimators of location and scale, a sample's mean and standard deviation (or variance σ^2), are deemed *non-robust* measurements because they are extremely sensitive to outliers and even a single outlying observation can distort them. For example, if we replace any one of the n observations in some data, $\mathbf{x}$, with an extremely large value it will completely change the estimate of the mean (and if this value were near infinity the mean would also approach infinity). This property of estimators is called the *finite sample breakdown point* and is defined as the proportion of observations in a sample that can be replaced before the estimator fails to describe the data accurately [Rousseeuw and Leroy, 2003]. The mean and the

standard deviation estimates, given above, both have a breakdown point of 0% for large n because even replacing one point can render the estimator useless.

Robust estimators on the other hand are so named because they have higher breakdown points and thus are more resistant to outliers. The *median* is a well-known robust estimator of location and has a breakdown point of 50%. It is defined as the middle observation in a finite list of observations arranged from least to greatest (if there are an even number of observations then typically it is the mean of the two middle observations). If we denote the i^{th} ordered observation as $x_{(i)}$ then we can write the median as $x_{(n/2)}$ for an odd number of observations and $\frac{1}{2}(x_{(n/2)} + x_{(n/2+1)})$ for an even number of observations. The median's 50% breakdown point means that we could replace up to half of the n observations in $\mathbf{x}$ with ones that are arbitrarily far away before the estimate becomes unusable. Returning to the previous case of replacing one observation above the median in $\mathbf{x}$ with a value close to infinity, we can see that the estimate of the median will not be affected. A breakdown point of 50% is actually the maximum possible for any estimator because after half of the points have been replaced with outliers, the majority of the distribution is made up of outliers.

The *median absolute deviation (mad)* is an example of a robust estimate of scale. The *mad* gives an indication of the "median distance from the median" and is defined as $median_{i=1,\ldots,n}|x_i - median_{j=1,\ldots,n}(x_j)|$, this value is often is multiplied by a constant (1.4826) to make it consistent with the standard deviation for normal distributions.

$$\text{mad}(\mathbf{x}) = 1.4826 \times \underset{i=1,\ldots,n}{\text{median}} \, |x_i - \text{median}(\mathbf{x})|$$

We can make the z-score more resistant to outliers, and thus more robust, by using the robust estimates of location and scale instead of the mean and standard deviation

used previously.

$$z_i = \frac{|x_i - \text{median}(\mathbf{x})|}{\text{mad}(\mathbf{x})} \tag{2.1}$$

Using this robust z-score measure we have no trouble detecting the outliers in the data given above (in Section 2.2.1) using the same rule which classifies observations as outliers if they are farther than 3 standard deviations from the mean. Our two outlying points, 30 and 60, have robust z-scores of 15.7 and 31.8 (well above 3), while the other 10 points all have robust z-scores less than 1. This is a good method for the detection of outliers in univariate data, but it relies on the assumption that the data is normally distributed.

Often in statistical analysis the underlying distribution is not known [Barnett and Lewis, 1998] and using outlier identification rules that make the assumption that the population is Gaussian, when it isn't, will lead to errors (detecting too many or too few outliers). If the data comes from a distribution like the Students-t distribution, for example, which has fatter tails than a normal distribution, then using the rule above will result in many many more points being labeled outliers. For a Student-t distribution with 3 degrees of freedom only 98.6% of observations fall within ± 3 standard deviations so 5 times more observations, 1.4%, will be classed as outliers (false positives) than would occur if the distribution were normal. These distributions are shown in Figure 2.1 along with the percent of the population that falls within their respective standard deviations.

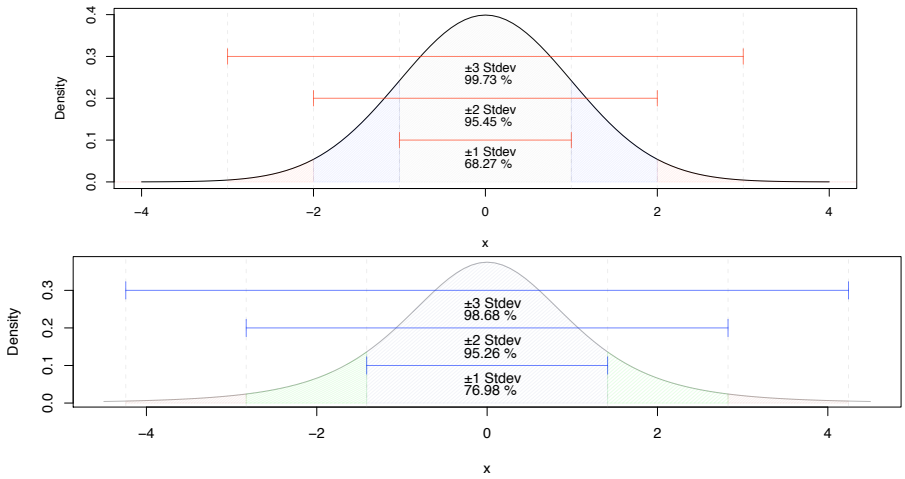


Figure 2.1: Normal distribution with a mean of 0 and a standard deviation size of 1 (above) and a Students-t distribution with 3 degrees of freedom (below). They are marked with the percent of the population falling within their respective standard deviations.

2.2.2 Data with unknown distribution

The previous section illustrates the challenge of detecting univariate outliers in observations that come from an unknown distribution. Most outlier detection rules make the assumption that the data follow a normal distribution and, if the distribution of the data is not normal, these methods are likely to perform poorly. However, if we know the exact distribution the data comes from, even if it is non-Gaussian, then it is possible to construct an appropriate test for outliers based on that particular distribution, in the example above, we could use a test for outliers for a t-distribution (many such tests for various distributions are given by Barnett and Lewis [1998]). We face a tougher problem when the underlying distribution is completely unknown (and we refuse to make any assumptions about how it is distributed).

Some outlier detection methods attempt to be more general and so are suitable

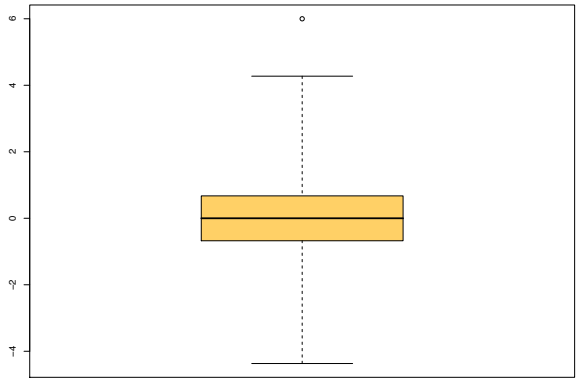


Figure 2.2: An example of Tukey's boxplot where the whiskers extend to the observations nearest the outer fence. The data was generated from a normal distribution $\mathcal{N}(0, 1)$ with one outlier.

for detecting outliers in many different types of distributions, but this is difficult to achieve, not least because these distributions might be asymmetric. There are some basic properties of all distributions that are sometimes used in outlier detection to mitigate these problems. One such property is given by Chebyshev's inequality theorem (sometime called the Bienaymé-Chebyshev theorem), which states that, for any finite distribution, a minimum of $1 - \frac{1}{k^2}$ of the data will lie within k standard deviations of the mean [Grimmett, 2001; Papoulis, 2002]. So, for any distribution, at least 93.75% of the observations will lie within 4 standard deviations from the mean and 96% lie within 5 standard deviations. (Conversely, Chebyshev's theorem with $k = 5$, gives $\frac{1}{5^2} = 0.04$, so no more than 4% of the population lies outside 5 standard deviations.) This can aid in outlier detection by calculating the maximum probability that a point lies some number of standard deviations from the mean or calculating the maximum number of false positives that will be returned for any population. It should be noted that to use Chebyshev's theorem it is important to have a reliable

estimate of the population's standard deviation.

The most widely used methods to analyze data and detect outliers for unknown distributions are due to Tukey [1977], who invented the boxplot as a way to visualize and explore data. An example boxplot is shown in Figure 2.2 showing normal data with an outlier. Tukey recommended visualizing data as the best way to identify outliers in univariate data, but as part of the boxplot he defined what he called the inner and outer fences for data, outside of which observations were likely to be outliers. These fences can be used as cutoffs to identify outliers automatically. These fences make use of *quartiles* which are extensions of the idea of the median (which is called the second quartile, Q_2) to other fractions of the data. $x_{(1/4)}$ is the first quartile, often written Q_1, or the observation that marks the point where one-fourth of the observations in the data are smaller and three-fourths of the observations are larger. Similarly the third quartile, Q_3 marks the $x_{(3/4)}$ observation in the data. (Tukey actually recommends calculating these values using the median of the upper half of the data and median of the lower half of the data.)

$$\begin{array}{lllll} \text{Inner Fences:} & x_{(1/4)} - 1.5 \times IQR(\mathbf{x}) & \text{and} & x_{(3/4)} + 1.5 \times IQR(\mathbf{x}) \\ \text{Outer Fences:} & x_{(1/4)} - 3 \times IQR(\mathbf{x}) & \text{and} & x_{(3/4)} + 3 \times IQR(\mathbf{x}) \end{array}$$

These fences are also defined using the *inter quartile range (IQR)*, which is the difference between the third and fourth quartiles (this is also the height of the box in a boxplot).

$$IQR(\mathbf{x}) = x_{(3/4)} - x_{(1/4)}$$

Tukey's inner and outer fences are for approximately normal data and mark boundaries in the data where any observations lying outside these boundaries are likely to be outliers. The inner fence marks the boundary for *mild* outliers and the

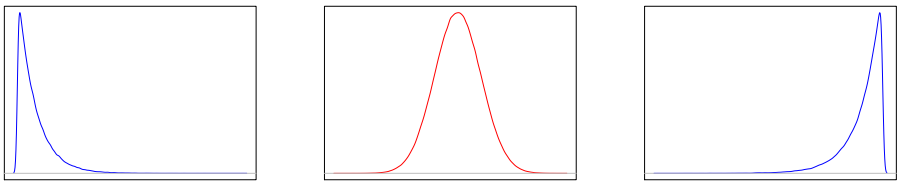


Figure 2.3: Right, symmetric, and left skewed distributions

outer fence marks the boundary of *extreme* outliers. If the data is non-Gaussian then the data can either be transformed to look more normal, or, the constants in the fences can be changed to allow for that distribution. The interval $[Q_1 - k \times IQR, Q_3 + k \times IQR]$ can be defined to give reliable fences for any symmetric distribution if we choose k to correctly allow for the distribution shape. For asymmetric distributions, these k's will differ and are calculated for each half of the distribution to take into account the direction the distribution is skewed (left or right tailed, see Figure 2.3).

Automatic methods to estimate these constants for skewed data have been investigated by Kimber [1990] and Vandervieren and Hubert [2004]. Vandervieren and Hubert specifically test the use of *quartile skewness* (this measure has also been used by Hinkley [1975], who called it the *tilt factor*) and the *medcouple* [Brys et al., 2004].

The *Quartile Skewness* (Qn) estimates the skewness of a distribution and is also resistant to outliers (robust). It is zero for symmetric distributions, positive for right skewed distributions, and negative for left skewed distributions and is calculated as:

$$Qn = \frac{(Q_3 - Q_2) - (Q_2 - Q_1)}{Q_3 - Q_1}$$

The *Medcouple* (MC) is a similar robust measurement of skewness developed more

recently in outlier detection.

$$MC = \underset{x_i \leq Q_2 \leq x_j}{\text{median}} \left(\frac{(x_j - Q_2) - (Q_2 - x_i)}{x_j - x_i} \right)$$

Both of these skewness measures are used to calculate fences in similar way as was done in Tukey's boxplot. These fences are given in Table 2.1, along with similar fences used by Kimber [1990]. These fences are used, just as those of Tukey, as cutoff points above which we say points are outliers, but they are appropriate to detect these outliers even in skewed data. We show the boundaries marked by these fences on data generated from a skewed distribution in Figure 2.4. Notice that the fences which attempt to account for the skewness of the data, like the Medcouple, do not label any points in this figure as outliers and visually seem to be sensible boundaries for identifying outlying points in this data. In contrast symmetric fences like the robust z-score are clearly not appropriate for this data and in Figure 2.4 label nearly 5% of the data as outliers and undesirably set outlier fences far below zero which could lead to genuine outliers being missed.

Measure	Left Fence	Right Fence
Tukey's Fences	$Q_1 - 3 \times IQR$	$Q_3 + 3 \times IQR$
Robust z-score Fences (For 4 std. deviations)	$Q_2 - 4 \times mad$	$Q_2 + 4 \times mad$
Qn Fences	$Q_1 - 3 \times e^{-3.5Qn} IQR$	$Q_3 + 3 \times e^{3.5Qn} IQR$
Medcouple Fences	$Q_1 - 3 \times e^{-3.5MC} IQR$	$Q_3 + 3 \times e^{3.5MC} IQR$
Kimber's Fences	$Q_1 - 3 \times 2(Q_2 - Q_1)$	$Q_3 + 3 \times 2(Q_3 - Q_2)$

Table 2.1: Common fences used for univariate outlier detection.

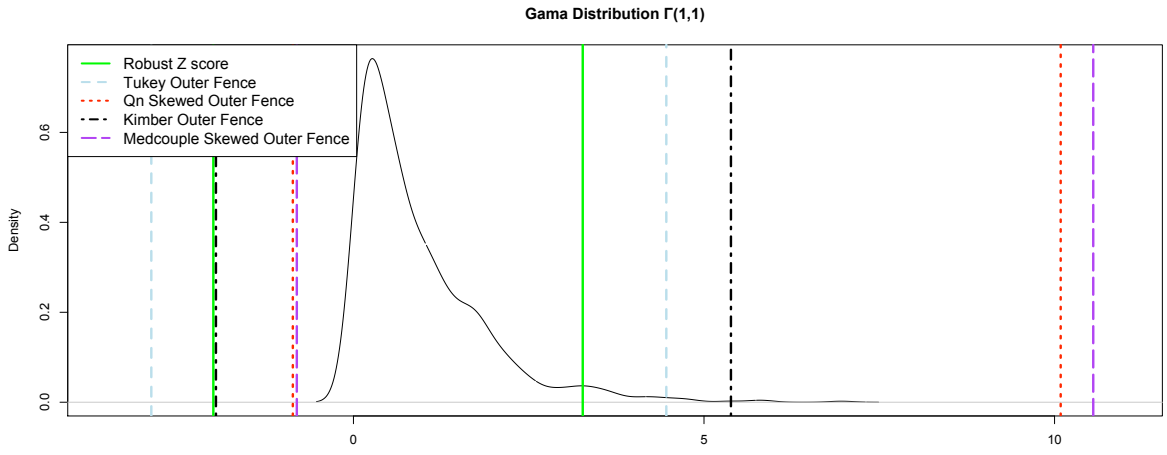


Figure 2.4: Outlier fences for a skewed distribution. This graph shows the fences for 1,000 points randomly generated from a Gamma Distribution with shape and scale equal to one.

2.3 Multivariate Outliers

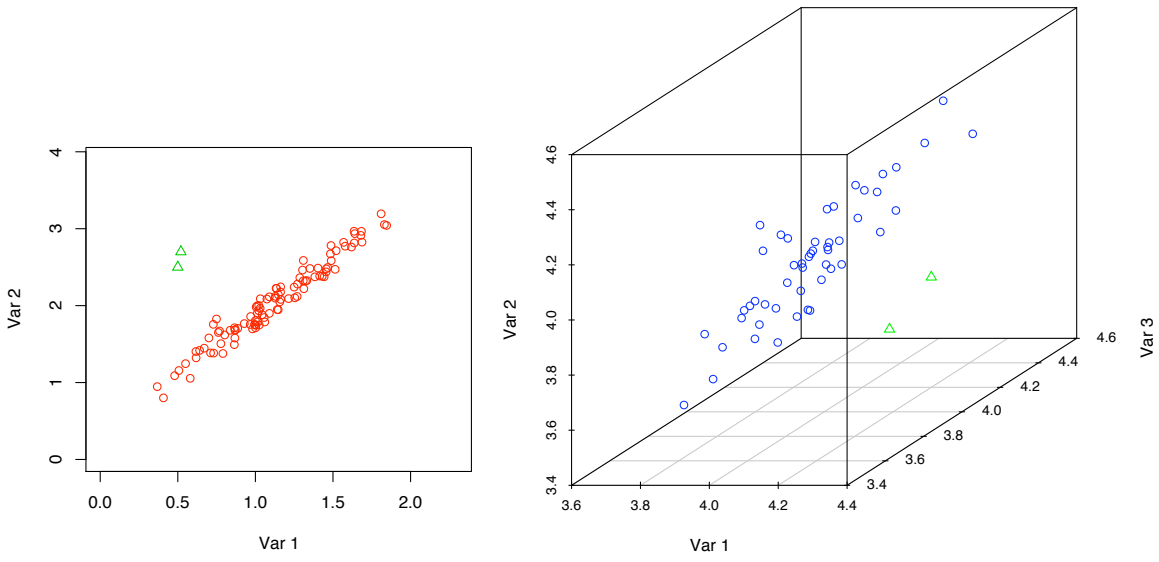
In multivariate data, outlier detection becomes a slightly less intuitive problem because it is not as obvious what is considered "far away" or atypical when observations are composed of more than one variable. It is useful to think of these observations, as points in p-dimensional space $\mathbb{R}^p$ and that we would like to identify points that are far from the center of this cloud of points. Although it is possible to identify outliers in univariate data by plotting the data on a graph and visually detecting the outliers (as advocated by Tukey [1977]), it becomes difficult or impossible when the number of dimensions is greater than 2 [Rousseeuw and van Zomeren, 1990]. It is therefore of greater necessity to have automatic methods to detect outliers in multivariate data than in univariate (or bivariate) data, where there exists the possibility of graphing the data and utilizing a human's ability to spot outliers visually.

Throughout this section we will let $\mathbf{X}$ be an $n \times p$ matrix of observations where the rows in $\mathbf{X}$ are observations and the columns of $\mathbf{X}$ are the variables. (The columns of this matrix might be called *features*, in the machine learning sense.)

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \\ x_{n1} & \cdots & & x_{np} \end{pmatrix}$$

Observations of more than one variable introduce new complexity into the outlier identification problem because in multivariate data it is necessary to take into account not only the individual variables, but also the interactions of these variables. Take, for instance, the data shown in Figure 2.5(a). This figure shows observations consisting

of two variables (Var 1 and Var 2) as points in $\mathbb{R}^2$. There are two clear outliers visible in this figure (the green triangles), yet these points are not outliers in either direction (Var 1 or Var 2). These observations are not univariate outliers with respect to either of their variables individually, but only by virtue of the interaction between them being atypical for the sample. A similar example for observations in three dimensions is shown in Figure 2.5(b). It is clear from this picture that it is not enough to detect outliers based on whether they are outliers in either dimension. Any tests for multivariate outliers must take into account where the points lie in the space defined by these dimensions.



(a) Two Dimensions

(b) Three Dimensions

Figure 2.5: Multivariate outliers 'hidden' in space

2.3.1 Classical Multivariate Distance

Mahalanobis Distance

To detect outliers in multivariate space it is desirable to be able to measure how far points are from the center of the population in this space. If we used simple Euclidean distance, we could think of the distance from the center of a multivariate sample in two dimensions as a series of concentric circles expanding out from the center of the observations and judge outlying points based on the radius of this circle. This, unfortunately, will give a poor notion of distance unless the data is arranged spherically. Figure 2.5 shows that when measuring distance from the center of a mass in multivariate space it is not good enough to simply measure how far away an observation is from the center of the data, the direction of the distance is important as well. Specifically, what is important about the direction is the variance in that direction. Instead of concentric circles, the distance from the center of multivariate data must be allowed to be ellipsoidal, so that it can take into account the spread of the variables in different directions. This spread of the variables is due to their interaction or *covariance* and Mahalanobis [1936] introduced a way to measure distance with regard to this covariance. This measure, known as *Mahalanobis Distance*, is calculated for each observation as:

$$d_i = \sqrt{(\mathbf{x}_i - \bar{\mathbf{x}})^T \mathbf{\Sigma}^{-1} (\mathbf{x}_i - \bar{\mathbf{x}})} \tag{2.2}$$

Where, $\bar{\mathbf{x}}$ is the center (or location) of the data, estimated as a vector whose columns are the means of the individual variables (called the coordinate-wise sample mean). The Σ^{-1} denotes the inverse of the $p \times p$ covariance matrix, the sample covariance

matrix $\hat{\Sigma}$ is calculated for every pair of variables (columns) as:

$$\hat{\mathbf{\Sigma}}_{jk} = cov(j,k) = \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{x}_{ij} - \bar{x}_j)(\mathbf{x}_{ik} - \bar{x}_k)$$

This covariance matrix can also be written in a simpler way as the dot product of the columns of the *centered* matrix. Let $\tilde{\mathbf{X}}$ be a new matrix formed by centering the matrix $\mathbf{X}$, this is achieved by subtracting the column mean from each column in $\mathbf{X}$, so

$$\tilde{\mathbf{X}}_j = \mathbf{x}_j - \bar{x}_j \text{ , for } j = 1, ..., p$$

$$\hat{\mathbf{\Sigma}} = \frac{1}{n-1}\tilde{\mathbf{X}}^{\mathbf{T}}\tilde{\mathbf{X}} \tag{2.3}$$

It is helpful for understanding Mahalanobis distance to look at the special case where all variables in the data are uncorrelated (this means they vary completely independently and thus their covariance is zero). It is obvious that for this case all elements off the diagonal of the covariance matrix will be zero. (Thinking of the columns of our matrix as vectors, the dot products of a column with any other column will be zero, and thus all variables all orthogonal.) In this case the diagonal covariance matrix reduces the Mahalanobis distance in Equation 2.2 to a normalized Euclidean distance:

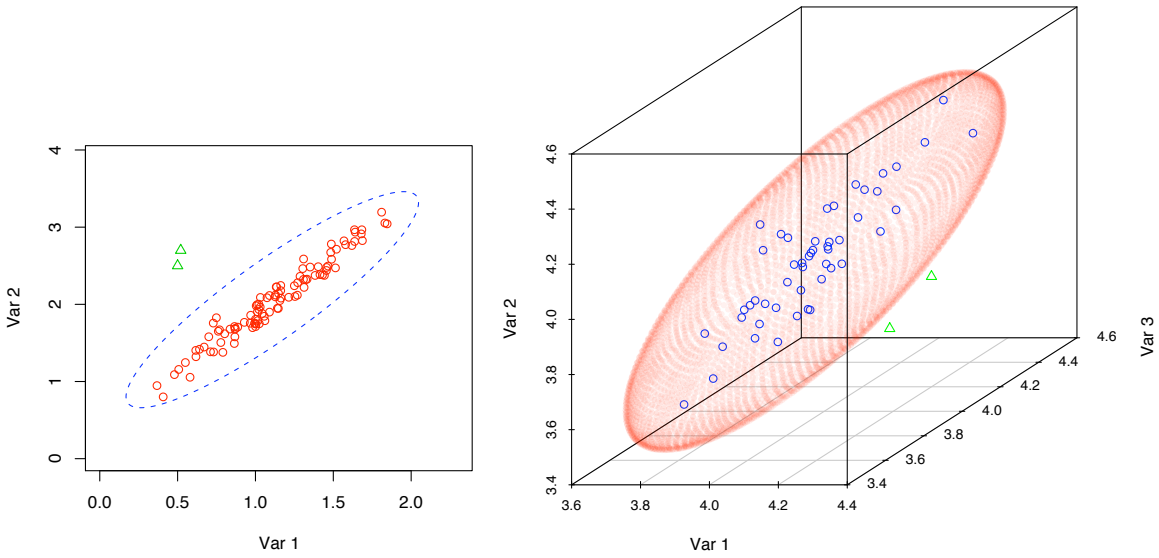
$$d_i = \sqrt{\sum_{j=1}^{p}\left(\frac{(x_{ij} - \bar{x}_j}{\sigma_j}\right)^2} \tag{2.4}$$

From this equation it is clear that if the data were perfectly spherical, with a standard deviation of 1 in all directions (the covariance matrix would be the identity matrix), then the Mahalanobis distance would reduce to the standard Euclidean distance $d_i = \sqrt{\sum_{j=1}^{p}(x_{ij} - \bar{x}_j)^2}$. So, if our data were arranged spherically in space,

it would be equivalent to measure the distances to observations using Mahalanobis distance or simple Euclidean distance. This confirms the intuition behind estimating multivariate distance as described at the beginning of the section. If the data is scattered equally and independently in all directions then we can measure distance equally in all directions.

Distribution of Distances

Observations with a large Mahalanobis distance (using the real mean and covariance of the population) are far from the center of the data (taking into account its variance) and thus are likely to be outliers. Determining where to draw the cutoff above which observations should be labeled outliers relies on the fact that for normal distributions the squared Mahalanobis distances, d^2, will follow a χ^2 distribution with p degrees of freedom [Hardin and Rocke, 2005]. It is common in the literature to label as outliers any observations whose Mahalanobis distance, d_i, is greater than $\sqrt{\chi^2_{p,.975}}$ [Rousseeuw and van Zomeren, 1990], where $\chi_{p,.975}$ is the 0.975 quantile of the χ^2 distribution with p degrees of freedom. For instance, if our multivariate data consists of observations with 3 variables then observations whose distances exceed $\sqrt{\chi_{3,.975}} = 3.06$ will be labeled outliers. The 0.975 quantile insures that for normal data there is only a probability of 2.5 percent that distances chosen from the distribution will fall in this range and be labeled outliers; if we wanted less chance of false positives when labeling outliers, we might use the .99 or even .999 quartiles as our boundaries. Figure 2.6 shows the same examples from Figure 2.5, but with tolerance ellipses drawn for the classic Mahalanobis distance up to the .975 quantile of the χ^2



(a) Two Dimensions

(b) Three Dimensions

Figure 2.6: Classical Mahalanobis Distance with a tolerance of $\sqrt{\chi^2_{2,.975}}$ and $\sqrt{\chi^2_{3,.975}}$ respectively. We see that these 'hidden' outliers will be correctly labeled using this measure.

distribution. Notice that in this example the outliers we identified visually would be labelled correctly.

If the data is not normally distributed then the distances are unlikely to follow a χ^2 distribution and the cutoffs used above will not give meaningful results. A method used to overcome this problem, with good success [Maronna and Yohai, 1995; Maronna and Zamar, 2002; Filzmoser et al., 2008], is to transform the Mahalanobis distances to distances, d^*, that more closely match the χ^2 distribution so that the same cutoffs can be used.

$$d_i^* = d_i \times \frac{\sqrt{\chi^2_{p,.5}}}{median(d_1, ..., d_n)} \tag{2.5}$$

While transforming the distances in this manner has shown to be reliable for

most data, there has been other work in outlier detection that has achieved good results choosing the cutoff for outliers in different ways. Werner [2003] used a more complicated procedure that calculates the density of all Mahalanobis distances and chooses to label as outliers observations that occur after this density has decreased to nearly zero. This makes the assumption that the distances of the non-outliers will group together and there will be some space between the next distances, which are the outliers.

It should be apparent that this last approach depends on what is essentially a univariate outlier detection task, where we are determining which distances are outliers with respect to the other distances. For this reason Brys et al. [2005] and Rousseeuw et al. [2006] make use of the right medcouple fence shown in Table 2.1, to take into account the skew of this distribution and mark outliers as observation appearing outside this fence. Hardin and Rocke [2005] investigate this problem thoroughly and give good insight into the distribution of these distances; they show that distances can also be labeled as outliers accurately through a well-founded procedure that describes the distances using a scaled F-distribution.

2.3.2 Robust Outlier Detection

The classical Mahalanobis distance suffers from the same problem as the z-score in univariate outlier detection. Namely, the masking effect, where multiple outliers or large outliers will distort the estimates for the measure so that outliers do not receive large distances and thus it is not possible to detect them. This is due to the classical estimates of location and scale it uses, which render it unsuitable for use except on data that is "certain to contain no outliers" [Filzmoser et al., 2008]. The Mahalanobis distance can be made more robust if, instead of the column-wise mean and sample covariance matrix used in Equation 2.2, we use more robust multivariate measures of location and scale [Rousseeuw and van Zomeren, 1990].

$$RD_i = \sqrt{(\mathbf{x}_i - T(\mathbf{X}))^T C(\mathbf{X})^{-1}(\mathbf{x}_i - T(\mathbf{X}))} \tag{2.6}$$

Here $T(\mathbf{X})$ and $C(\mathbf{X})$ represent robust estimates of location and scale. A more robust estimate of location of our data, $T(\mathbf{X})$, is the coordinate-wise median, calculated as a vector whose values are the median of every column in $\mathbf{X}$. This measure has a 50% breakdown point, but Croux and Ruiz-Gazen [2005] call it "a crude approximation" of the location. A better estimate of location with the same high breakdown point [Small, 1990] is the L_1 *median* (often called the *spatial median*) defined as the point in space which minimizes the sum of distances to all data points.

$$T(\mathbf{X}) = \underset{\mu}{\operatorname{argmin}} \sum_{i=1}^{n} ||\mathbf{x}_i - \mu|| \tag{2.7}$$

Here $||\cdot||$ denotes the standard Euclidean norm, $\sqrt{\sum x^2}$. Croux and Ruiz-Gazen [2005] give a fast algorithm for its computation and code is freely available[4].

[4]As part of the *PcaPP* package in the R statistical language. This package is available at CRAN (Comprehensive R Archive Network) at `http://cran.r-project.org/web/packages/`

Affine Equivariance and Robustness

A property for robust estimators that is often important is that they are independent of the coordinate system chosen to represent the observations. So, if the data were moved, rotated, or stretched, we would like the estimator to move, rotate, and stretch accordingly. This property is called *affine equivariance* and guarantees that if we transform the data (say by changing the units of measurement) that the estimator will change in exactly the way expected. Thus when using these estimates for outlier detection, we would hope to get the same results before and after the data has been transformed. If we think of observations in two dimensions as dots that have been plotted on a thin piece of rubber, then any stretching or rotating performed on this piece of rubber, by holding the edges and keeping it flat, will not change which points are outliers. The coordinate-wise median does not have this property and the L_1 median is only orthogonally equivariant (only equivariant with respect to rotation and flipping), but later in this chapter we will see examples of robust procedures that are affine equivariant.

Many procedures that would seem to be good for eliminating outliers (and thus for robust estimates of location and scale) have been shown to have relatively poor breakdown points in high dimensions. For example, Donoho and Gasko [1992] show that iteratively removing the point with the largest Mahalanobis distance can never have a breakdown point higher than $\frac{1}{p+1}$. (Where p is still the number of variables (or dimentions) of the data.) This is similar to Grubbs's method in univariate data (pg. 19). They also study a variation of this procedure, where at every

`pcaPP/index.html` [Filzmoser and Fritz, 2007].

iteration the Mahalanobis distance is calculated for observations using the mean and covariance of all observations but that observation (i.e. a 'leave one out' distnace). So, at every step removing the observation with the largest distance where $d_i^2 = (\mathbf{x}_i - \bar{\mathbf{x}}_{(-i)})^T \mathbf{\Sigma}_{(-i)}^{-1} (\mathbf{x}_i - \bar{\mathbf{x}}_{(-i)})$. This procedure, unfortunately, shares the same poor upper bound for the break down point. Donoho and Gasko [1992] and Donoho [1982] give several other procedures for choosing a set of good points, including ellipsoidal trimming, convex hull peeling, and ellipsoidal peeling, that also share this low breakdown point. Lopuhaä and Rousseeuw [1991] give proofs that the maximum possible breakdown point for an estimator that is affine invariant can be, as we would hope, 50%.

Robust estimators for the scale of multivariate data, $C(\mathbf{X})$, by virtue of their methods, often give a robust estimate of the location along with their scale estimate. Outlier detection methods usually make use of these location and scale estimates rather than relying on the location estimates above. Most robust estimators proceed by identifying a small set of *good* observations (non-outliers) and then use only these observations to estimate the classical mean and covariance. If these observations are truly *good* observations then the estimates of the location and scatter of the data will be very accurate and this will lead to the highest possible breakdown point. Finding this set of good data points is essentially the reverse problem of outlier detection and faces many of the same problems, except that you only need to find a fixed fraction of the data which you know to be good (usually between one-half and three-fourths of n). In the next section we examine one such method for choosing *good* points, which leads to a robust method for identifying outliers.

Minimum Covariance Determinant

A popular robust estimator of location and scale with a high breakdown point is the *Minimum Covariance Determinate* (MCD) estimator, which was proposed by Rousseeuw [1984]. This estimator is hard to beat in terms of accuracy and robustness, as it can achieve a 50% breakdown point and has the desirable property of affine equivariance. Its computation in a reasonable amount of time is non-trivial, but a fast algorithm for its computation, called fastMCD[5], was later introduced by Rousseeuw and van Driessen [1999] that greatly widened the applicability and usefulness of the MCD.

The MCD estimator works by finding a subset of the observations, h, whose classical covariance matrix has the minimum determinant. The size of h can be varied to allow for a higher breakdown point or better efficiency, but is usually chosen to be a little more than half the number of observations. An exhaustive search of all possible subsets is obviously not possible, so the fastMCD algorithm makes clever choices about which observations to include and exclude. Recall that in 2 dimensions, the absolute value of the determinant of a matrix gives the area of a parallelogram whose edges are determined by the rows of that matrix (similarly in 3 dimensions, the volume of the parallelepiped). The intuition behind the MCD is that it minimizes the area spanned by the covariance matrix and so is useful in finding a group of observations that are close together and therefore are likely to form the center of the data.

The MCD estimator uses this *good* set of h points to estimate the location and

[5]Source code for the fastMCD algorithm is available in the R statistical language as the function *cov.mcd* in the *MASS* package.

scale using the standard coordinate-wise mean and sample covariance matrix and then re-weights all observations. Let T_h and C_h be the mean and covariance estimates of the h good points, then the MCD calculates the robust Mahalanobis distances for every observation by plugging T_h and C_h into Equation 2.6. These distances, RD_h, are used to re-weight the location and scale estimates so that all observations receive some weight (see Rousseeuw and van Driessen [1999] for this re-weighting step and Lopuhaä and Rousseeuw [1991] for a proof of why it is important to perform this step). Figure 2.7 gives some sample data that is made up of 15% clear outliers. We have drawn the the classical Mahalanobis distance with $\sqrt{\chi^2_{2,.975}}$ tolerance which fails to identify all but a few of these outliers due to the estimations of mean and variance being skewed by presence of the outliers. The MCD estimator's tolerance ellipse, using the same quantile of the chi-squared distribution, clearly gives a good estimation of the mean and variance and thus identifies all outliers.

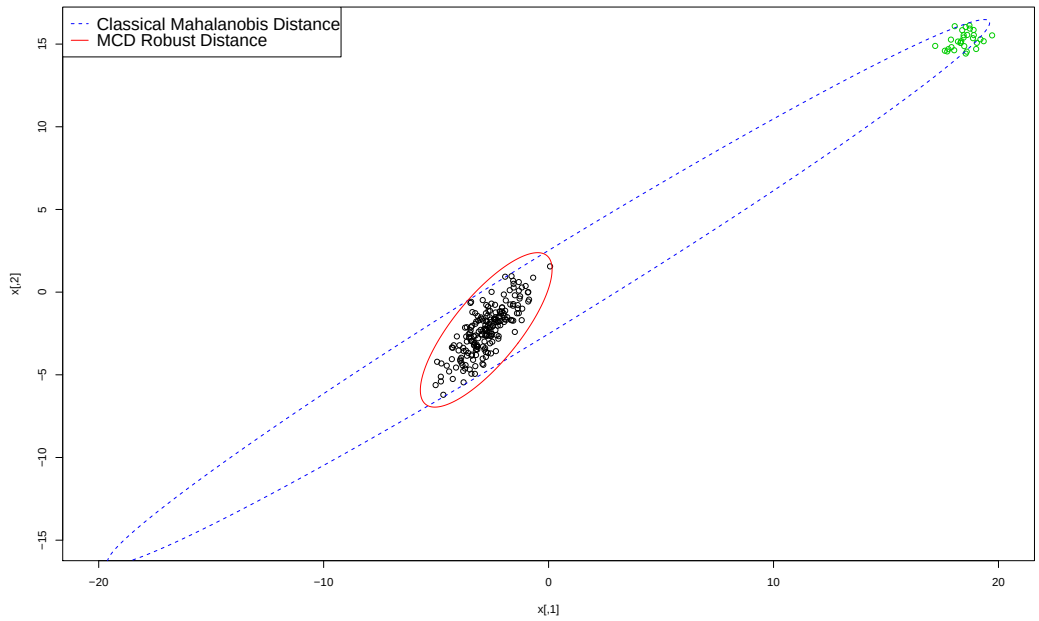


Figure 2.7: The classical Mahalanobis distance versus the robust MCD distance with a tolerance of $\sqrt{\chi^2_{2,.975}}$. The data has been constructed to contain 200 normal observations (with a certain covariance) and 30 outliers. The classical distance, which uses non-robust measures of location and scale, clearly fails to identify most outliers, whereas the MCD method is more robust and easily identifies all outliers.

2.4 Approaches for High Dimensional Data

When observations contain a large number of variables, namely when $p > n$, there are concerns that arise in the detection of outliers. The main problem is that the covariance matrix estimate, $\hat{\mathbf{\Sigma}}$, is guaranteed to be singular (and thus not have an inverse) if the dimension of the data, $\mathbf{X}$, is greater than the number of observations. This is because the rank of the covariance matrix, r (the number of non-zero eigenvalues), will always be less than or equal to n, the number of observations (see Lay [2003] for an full explanation) and the basic fact that in order for a matrix to have an inverse, its dimension must equal its rank. Thus, if $p > n$, the $p \times p$ covariance matrix will have rank $r \leq n$ which is $< p$, so clearly $r \neq p$ and the matrix will be singular.

The singularity of the covariance matrix means that it does not have an inverse and therefore it is not possible to compute the Mahalanobis distance for any observations. Likewise, we can see that this effects the MCD estimate, which relies on finding a subset of h observations whose covariance matrix has the smallest determinant. Recall that the determinant of any singular matrix is zero. Thus we know that any subset of the observations will necessarily have a singular covariance matrix because it will have dimension $p > h$, so all covariance matrices will have a determinant of zero, rendering the estimator useless (as there is nothing to minimize). So, if the number of variables is greater than the number of observations then neither the classical Mahalanobis distance nor the MCD will yield meaningful results. In the rest of this section we explore some methods for dealing with data in high dimensions.

2.4.1 Projection Pursuit Approach

Stahel [1981] and Donoho [1982] independently introduced an estimator that is useful for high dimensional data and works by the simple idea of projecting the data down to one dimension in space and measuring the "outlyingness" of observations in that dimension. This was inspired by the *Projection Pursuit* techniques of Friedman and Tukey [1974] that looked at finding "interesting" projections of high dimensional data (see Huber [1985] for a detailed look at PP). The goal is to find a projection of the data onto a direction that maximizes an observation's robust z-score in that direction. This has become known as the *Stahel-Donoho Estimator* (SDE), and it is affine equivariant and can achieve the highest breakdown point possible ($\frac{1}{2}$). We demonstrate later in this thesis that the SDE is well suited to the detection of anomalies in text. Chapter 5 shows its application to this problem and Chapter 6 gives experimental results using it to detect textual anomalies and comparing it to other well know outlier detection methods and some novel ones.

The SDE method is conceptually intuitive yet its computation can be difficult as there are infinitely many directions that data can be projected onto and we are trying to find the one direction for each observation that makes it appear to be as far away from the center of the data as possible. So, we would like to find the supremum over all possible directions (unit length vectors) $\mathbf{a} \in \mathbb{R}^p$ for the outlyingness, SD, of an observation.

$$SD(\mathbf{x}_i) = \sup_{\mathbf{a}} \frac{\mathbf{x}_i^T\mathbf{a} - \text{median}(\mathbf{Xa})}{\text{mad}(\mathbf{Xa})} \tag{2.8}$$

Where $\mathbf{x}_i^T\mathbf{a}$ is the projection of observation $\mathbf{x}_i$ in direction $\mathbf{a}$. In practice it is impossible to calculate the supremum over all possible directions, so the *maximum* is

computed over a finite set of directions instead, in an attempt to closely approximate the measure [Donoho and Gasko, 1992]. After the maximum distance for each observation (over the finite set of projection directions) is calculated, these distances, SD_i are used to calculate re-weighted covariance and location estimates. The SDE was thoroughly investigated by Maronna and Yohai [1995] who experimented with different types of re-weighting and compared it with other robust outlier procedures. Maronna and Yohai showed it to be successful for dimensions up to size 20 where the computation time required to insure a reliable estimate can become prohibitive.

The set of directions chosen is crucial to how the estimator performs and several different methods have been experimented with for how to best pick these directions (and how many directions are necessary to try for each observation). Many choices seem to give good results, Stahel [1981] picked directions by randomly choosing h observations and calculating a direction, $\mathbf{a}$, orthogonal to the hyperplane containing those h observations (this method was also used by Maronna and Yohai [1995][6] and later Hubert and Van der Veeken [2008] used a variation for skewed data[7]). Choosing directions is this manner, however, is not useful in high dimensions with limited observations. Other methods have been experimented with, but the number of directions that must be tried in high dimensions grows so rapidly it makes most methods impractical. Peña and Prieto [2001] introduced a procedure to choose a good set

[6]An implementation is available in fortran as part of the *robust* package (Insightful Robust Library) in the R statistical language `http://cran.r-project.org/web/packages/robust/index.html` this implementation is limited to cases where $n < p$.

[7]The Adjusted Outlyingness Estimator was also used in connection with robust principal component analysis [Brys et al., 2005] and is available from `http://www.agoras.ua.ac.be/` as part of the *medcouple* package, this implementation is also limited to cases where $n < p$.

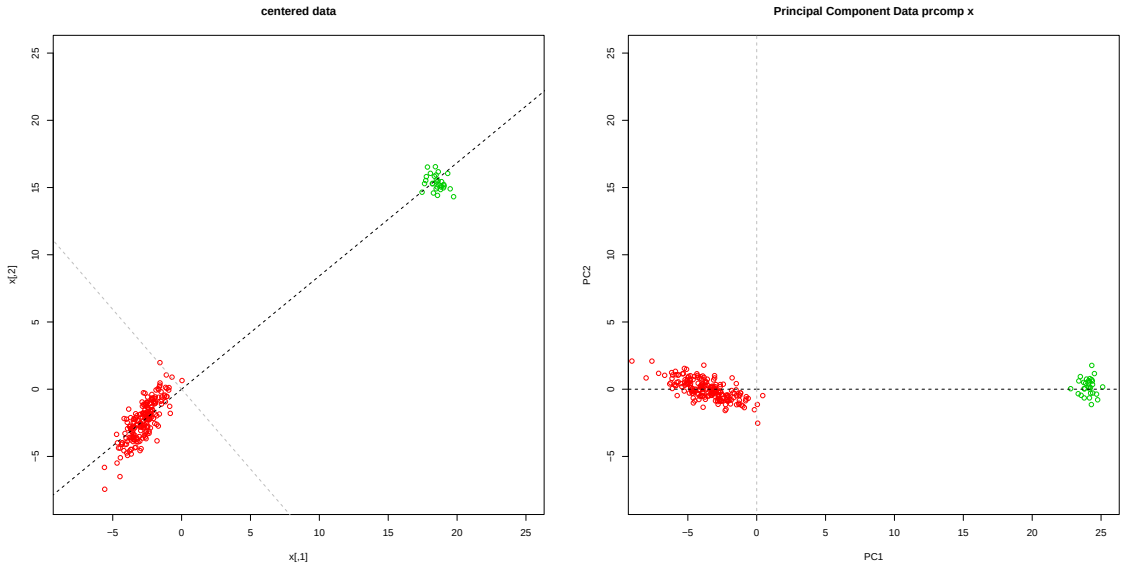
of only $2p$ directions as part of the *Kurtosis*[8] estimator, which picks directions that maximize or minimize the kurtosis. This estimator greatly reduces the computation time, as the number of directions that must be tested is small, but Filzmoser et al. [2008] indicate that there is some debate about whether this is always the best way to choose directions and as to how useful it is in high dimensions.

2.4.2 Principal Component Analysis and Singular Value Decomposition

Another approach to the high dimensionality problem is simply to reduce the dimensions of the data using *Principal Component Analysis* (PCA) (Huber [1985] explains that PCA is actually a type of Projection Pursuit). Principal component analysis is a technique used to transform data from the observed variables to a new set of variables that are uncorrelated and ordered by how much of the variance of the data they explain. PCA is often used for the task of dimensionality reduction by transforming the data in this way and then discarding all but the first k components, where we usually pick k either by determining the total variance of the data we wish to explain or by some practical constraint on size of the dimensions.

The traditional method of performing PCA is to compute the eigenvalues and eigenvectors of the covariance matrix, and to project the data onto the eigenvectors with the k largest eigenvalues (The eigenvalue decomposition of the covariance matrix gives $\Sigma = VDV^{-1}$, where the columns of V contain the eigenvectors of Σ and D is

[8]Matlab code for Kurtosis, written by the authors of the paper, is available at their website `http://halweb.uc3m.es/fjp/download3.html`, which is limited to cases where $p < 50$.



(a) Eigenvectors of $\mathbf{X}$ (b) PCA of $\mathbf{X}$

Figure 2.8: The first figure shows some data along with the eigenvectors of the data's covariance matrix. Notice that they are orthogonal and in the directions of the greatest variance. The second figure shows the data after the principal component transform. We can see that in this case we kept both components and it amounts to rotating the data so that it is in terms of the principal components.

a diagonal matrix with the eigenvalues of Σ along its diagonal.) This amounts to an orthogonal transform of the data, so that the data is in terms of the directions with greatest variance and these variables are orthogonal to each other. It should be pointed out that there is no need to compute the classical Mahalanobis distance using the inverted covariance matrix in this principal component space. Instead, because the principal components are orthogonal to each other, the distance reduces to that of Equation 2.4. Figure 2.8 shows, for some example data, the steps in PCA for a two dimensional matrix. The eigenvectors of the covariance matrix will always be orthogonal to each other and those with the largest eigenvalues correspond to the

directions in the data with the greatest variance. So, projecting the data onto the k largest eigenvectors effectively keeps only the k dimensions in the data with the largest variance. We could have kept only the first principal component for the data in Figure 2.8(a) which would have resulted in the data in Figure 2.8(b) being flattened along the horizontal axis and contain no movement in the vertical direction. It this example we would have good success detecting outliers using only the first principal component as the outliers happen to lie in this direction, but this is not always the case. Often outliers distort the variance of the data so that the principal components do not fit the data well. For this reason PCA is considered a non-robust technique.

An equivalent technique to the eigenvalue decomposition of the covariance matrix, is to take the *Singular Value Decomposition* (SVD) of the centered data matrix. Let $\tilde{\mathbf{X}}$ be the centered data matrix, $\tilde{\mathbf{X}} = \mathbf{X} - \bar{\mathbf{X}}$. The singular value decomposition of $\tilde{\mathbf{X}}$ is:

$$\tilde{\mathbf{X}} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T$$

The SVD decomposition says that $\mathbf{U}$ and $\mathbf{V}$ must be orthonormal matrices and $\mathbf{\Lambda}$ is a diagonal matrix. It turns out that the columns of $\mathbf{V}$ are the eigenvectors of $\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}$, which is helpful because this means they are the same as the eigenvectors of the covariance matrix (from Equation 2.3 we can see that the matrix $\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}$ is the covariance matrix times a scalar). $\mathbf{U}$ contains the eigenvectors of the matrix $\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T$. The matrix $\mathbf{\Lambda}$ contains its only non-zero values along its diagonal and they correspond to the square root of the eigenvalues of the covariance matrix times n minus one. The values in $\mathbf{\Lambda}$ are ordered from greatest to least, so the eigenvectors in $\mathbf{V}$ follow the same ordering and range from the vectors that account for the most variance in the

data to the least.

$$\lambda = \frac{\mathbf{\Lambda}^2}{(n-1)}, \text{ where } \lambda \text{ are the eigenvalues of the covariance matrix.}$$

The matrix $\tilde{\mathbf{X}}$ in terms of its principal components is just $\tilde{\mathbf{X}}\mathbf{V}$ (or equivalently $\mathbf{U}\mathbf{\Lambda}$) and by using only the first k eigenvectors from $\mathbf{V}$, we end up with the data in terms of the first k principal components. Let $\mathbf{Z}_{[k]}$ be the $n \times k$ matrix that is the centered data, $\tilde{\mathbf{X}}$, in terms of its first k principal components. This matrix can be computed as:

$$\mathbf{Z}_{[k]} = \tilde{\mathbf{X}}\mathbf{V}_{[k]} \tag{2.9}$$

The SVD is useful because it can be used on high dimensional data to reduce the number of dimensions to the point where we can guarantee that the covariance matrix will be invertible. Once the covariance matrix is known to be invertible, we can use any standard multivariate outlier procedure (e.g. robust Mahalanobis distance or fastMCD). It is important to note that components that are thrown away in the dimensionality reduction (when using a k in Equation 2.9 that is less than n) are effectively lost information, unless the eigenvectors have very very small eigenvalues (and thus the data has little variance in that direction). It is usually preferred to keep as many components as possible to insure that no information is wasted [Hubert et al., 2005]. It is also pointed out by Hubert et al. [2005] that the SVD transform of a matrix of any size (using all vectors in $\mathbf{V}$) results in a matrix of dimension at most n with **no loss of information**. ($\mathbf{V}$ will have at most n eigenvectors so $\tilde{\mathbf{X}}\mathbf{V}$ will be at most $n \times n$.)

This is a valuable tool for outlier detection in high dimensions, especially if a matrix has more variables than observations, $p > n$. We can simply calculate $\tilde{\mathbf{X}}\mathbf{V}$

and achieve a matrix that is of dimension at most n without loosing any information and proceed with outlier detection on this matrix. When the data is transformed in this way (without limiting the number of principal components) we say that we are representing the data in "terms of its own dimensionality".

2.4.3 Principal Component Outliers Method

In general there has been little research conducted in statistics on outlier detection in data where the number of variables is greater than the number of observations. An exception is the work of Filzmoser et al. [2008], who introduced and made available a technique they call *PCOut*[9] which makes use of many of the ideas mentioned in this chapter, including singular value decomposition, kurtosis, and robust distance in the principal component space. In later chapters, we show how to apply the PCOut method to the problem of detecting anomalies in texts and give experimental results comparing it to other methods and methods of our own.

The PCOut measure is fast to compute, works with data of any dimension, and has good success at detecting outliers. This method was shown by Filzmoser et al. to be competitive on data with a low number of dimensions (ten or less) where it was compared with many popular multivariate outlier detection methods (fastMCD, Kurtosis, and the OGK estimator of Maronna and Zamar [2002]). In higher dimensions, where tests were conducted with the number of variables ranging up to 2,000, there were no other techniques available to compare against (because none can accommodate such large dimensions). The PCOut method of outlier detection achieved 0.38%

[9] PCOut has been made available by its authors, in the R statistical language. It is found in the package *mvoutlier* at `http://cran.r-project.org/web/packages/mvoutlier/index.html`.

false negatives and 2.52% false positives on data with 2,000 dimensions and containing 10% outliers. PCOut's performance in detecting outliers in all of the artificially created test sets created by Filzmoser et al. (as well on a real test case involving microarray data) proved successful in terms of performance and robustness, making it very attractive for use when the number of dimensions in data is large.

We give a short summary of the exact procedure used in the PCout method here. The method breaks down into two separate phases one of which looks for location outliers and the other for outliers based on the scatter of the data. At the end of the procedure these two measures are combined. Let X be the data matrix with n rows, one for each observation, and p columns, one for each variable, and let x_{ij} denote the value of the i^{th} observation's j^{th} variable. Then the procedure is as follows.

PCout Algorithm:

Phase 1: Detection of Location Outliers

Step 1: Robustly sphere the data (zscore)

$$X^* = \frac{x_{ij} - \text{median}(x_{1j}, ..., x_{nj})}{\text{mad}(x_{1j}, ..., x_{nj})}$$

Step 2: Compute the matrix of principal components that account for 99% of the variance in the data. This is computed as in Equation 2.9 above, with k determined by the first k eigenvalues that sum to 99% of the total sum of all eigenvalues. We will call this matirx Z. This matrix is then robustly sphered again as in Step 1, to give a matrix Z^*.

Step 3: Compute robust kurtosis weights for each component of Z^*.

$$w_j = \left| \frac{1}{n} \sum_{i=1}^{n} \frac{\left(z^*_{ij} - \text{median}(z^*_{1j}, ..., z^*_{nj})\right)^4}{\left(\text{mad}((z^*_{1j}, ..., z^*_{nj})\right)^4} - 3 \right|, j = 1, ..., p$$

Weight each principal component by its relative weight ($z^*_{ij} \times \frac{w_j}{\sum w_j}$) and then compute the robust Mahalanobis distance for every observation. As discussed before, these are principal components (and thus all components

are orthogonal) so it is possible to compute these distances using the simplified Mahalanobis formula shown in Equation 2.4 on page 32. These distances are then transformed to make them look more like the χ^2 distribution using the scaling given in Equation 2.5, on page 34.

Step 4: Determine weights for each observation using a translated biweight. PCOut's approach to weighting observations is different from that used by most methods of outlier detection. It is a version of the translated biweight that was first used by Rocke [1996] and assigns weights as follows. Let $dloc_i$ be the scaled distances from the previous step then:

$$w_{1i} = \begin{cases} 0, & dloc_i \geq c \\ \left(1 - \left(\frac{dloc_i - M}{c - M}\right)^2\right)^2, & M < dloc_i < c \\ 1, & dloc_i \leq M \end{cases} \tag{2.10}$$

The constant M is taken to be the $33\frac{1}{3}^{\text{rd}}$ quantile of the distances (this is computed in a similar manner to the quartiles described in Section 2.2.2 pg. 25 and marks the point at which exactly one third of distances are smaller). The constant c is a cutoff set at:

$$c = \text{median}(dloc_1, ..., dloc_n) + 2.5 \times \text{mad}(dloc_1, ..., dloc_n)$$

This cutoff is very similar to the fences used by Tukey described in Section 2.2.2. Here we see that observations which have distances that are greater than the cutoff, and so are likely to be outliers, receive a weight of zero. Similarly, observations whose distances are very small, in the bottom third of the all distances, are likely to be *good* points and thus receive full weight. The distances in between this range receive a partial weighting based on their distance.

Phase 2: Detection of Scatter Outliers

Step 5: The sphered principal component matrix Z^* from Step 2 is used to produce distances by calculating the Euclidean norm for every observation.

$$dscat_i = \sqrt{\sum_{j=1}^{p} Z_{ij}^*}$$

These distances are then transformed to look more χ^2 as in Step 3.

Step 6: The distances $dscat_i$ are used to calculate weights for observations according to Equation 2.10 used in Step 4 except with different cutoffs. In

this step $c = \sqrt{\chi^2_{p,0.99}}$ (the 99^{th} quantile of the χ^2 distribution with p degrees of freedom) and $M = \sqrt{\chi^2_{p,0.25}}$. As mentioned earlier in Section 2.3.1 on page 33, these cutoffs assume that the distances, *dscat*, will approximately follow a chi-squared distribution (and they will for normal data). If the data is not normal then the scaling applied in the last step should help them appear more χ^2 like. Using quartiles of the chi-squared distribution is typical for weighting/re-weighting of observations, and closely related procedures can be found in many other outlier detection methods, including Maronna and Zamar [2002], Rousseeuw and van Driessen [1999], Maronna and Yohai [1995], and Rocke and Woodruff [1996].

Final Weights and Choosing Outliers Final weighting is calculated as a combination of the weights from the two phases. The equation for the final weights is given by:

$$w_i = \frac{(w_{1i} + s)(w_{2i} + s)}{(1 + s)^2} \tag{2.11}$$

The scaling constant s is set to 0.25 by Filzmoser et al. to insure that observations are only marked as outliers if they received low weights in both phases, rather than only one. Outliers are finally chosen as those observations whose weight w_i is less than 0.25.

2.5 Summary

This chapter presented an overview of outlier detection in statistics. We began with an introduction to univariate outlier detection, where observations are of a single variable. For instance, observations could be measurements of the length of a widget and we would like to identify any observations that seem incongruous. If we make the assumption that the length of widgets is normally distributed, we showed that we can easy construct a *robust* measure to detect these outliers. Where *robust* means a method that will give reliable estimates for all observation even when the data contains outliers. On the other hand, if we make no assumptions about the distribution of the length of widgets we show that one can inspect the data visually

to detect outliers or automatically construct "fences" based on the shape of the data which mark the boundaries of outlier and non-outliers.

The chapter proceeded with the detection of outliers in multivariate data where observations are of more than one variable. For instance, each observation could consist not only of the length of a widget, but also its width, height, and weight measurements as well. It is necessary in this case to not only look for outliers in each of these variables independently, but also to take into account the interaction of these variables. We give a full descriptions of two well known methods used for this purpose: Robust Mahalanobis Distance and the Minimum Covariance Determinant estimator.

The last part of the chapter focused on data with a large number of variables (dimensions). This type of data has often been gathered automatically and we are unwilling to make any assumptions about which variables are important identifiers of outliers. This situation arises, for example, in genomics when analyzing microarray data for gene expressions where each observation corresponds to an experiment and can consists of thousands of variables each indicating the activity of a certain gene [Sebastiani et al., 2003]. Data with a large number of variables presents unique challenges that make most multivariate outlier detection procedures unsuitable and we describe statistical methods that have been developed to overcome these challenges. Specifically we looked at projecting the data down into one dimension using the Stahel-Donoho Estimator (SDE), reducing the dimensionality of the data using Singular Value Decomposition, and Filzmoser et al.'s Principal Component Outlier detection method (PCOut). In the experimental portion of this thesis we apply the

SDE and PCOut methods to the detection of anomalies in text and compare their performance to other approaches.

Chapter 3

Related Work in Natural Language Processing

3.1 Overview

This thesis describes work on the task of detecting anomalies in textual data with no prior knowledge as to what it means to be normal or anomalous. While this is a novel task, it is closely related to and builds on several ideas and techniques from work on authorship attribution, plagiarism detection, and genre identification. These research areas have in common a collective need to measure the similarity between types of writing and a history of research involving the use of *stylometry* [Milic, 1967, 1991; Kenny, 1982], or literally the "measurement of style." The use of statistical methods with these stylistic measures has proved effective in many areas of natural language processing and in this chapter we will examine this research and how it relates to the problem of anomaly detection. We begin this chapter with a look at the

problem of authorship attribution with a focus on research that has used methods which attempt to capture stylistic qualities in text. The next section focuses on work specifically aimed at detecting stylistic inconsistencies and the following section focuses on the problem of genre identification.

3.2 Authorship Attribution

Authorship attribution refers to the automatic identification or assessment of a document's author. There is a long history of research in authorship attribution [Holmes, 1994] from many different academic disciplines including linguistics, literary forensics, literary studies and more recently computer science. In broad terms there have traditionally been two statistical approaches in authorship attribution.

Counting words The use of distributions of word frequencies to characterize an author's writing and differentiate between authors. These techniques count the distributions of words or sequences of words (n-grams) in various authors' writing. Probabilities are computed for words occurring in different authors' writing and this information is used to assign a probability to new documents as having been written by a particular author. These methods can also be used to capture topic, as similar approaches are used in classification of documents by topic [Guthrie et al., 1994; Yang, 1998; Sebastiani, 2002]. Authorship attribution techniques typically minimize this impact, by focusing on words that occur very frequently in the writing of all authors being considered.

Stylometrics The use of features that specifically attempt to capture elements of

writing that differentiate one author from another apart from the conscious choice of words. These features include counting the use of punctuation, the length of sentences, parts of speech, common misspellings, character sequences [Peng et al., 2003], and morphological features like the number of past tense words (by counting word suffixes) [Stamatatos et al., 1999]. Burrows [1992] makes the claim that computerized use of these features captures a constant aspect of an author's writing that transcends the topic of that writing. However, there is some disagreement about this, as these types of features can also successfully distinguish between genres [Kessler et al., 1997], as we describe in Section 3.4.

Among the first uses of statistical methods for identifying authorship were the pioneering works of Mendenhall [1887], who counted the average word lengths of authors and Sherman [1888] who looked at distributions of sentence lengths. These influential works studied the use of these features as indicators of an author's writing style and also explored their uses as a way of attributing authorship. These methods were evaluated qualitatively, but this research lead to many related statistical methods for authorship detection [Williams, 1975; Love, 2002].

Mosteller and Wallace [1964] carried out one of the most influential works on authorship attribution using the disputed authorship of the *Federalist Papers*. The *Federalist Papers* are a series of 85 articles written to convince early Americans to ratify the United States Constitution. Most of these articles were published during 1787 and 1788 in four New York City newspapers under the pseudonym "Publius". In the May of 1788 the full set of these articles was then published as a book using the same assumed name. The articles are known to have actually been written by

three different authors: General Alexander Hamilton, James Madison (who became the 4$^{\text{th}}$ U.S. president), and the statesman John Jay. The papers are a good test set for examining the differences in writing style between these authors because they all are essentially on the same topic. Twelve of the *Federalist Papers* have disputed authorship because both Hamilton and Madison claimed to have written them. These articles have sparked debate over who the true authors are because the styles of both authors are so similar.

Mosteller and Wallace examine the problem of determining who wrote these disputed papers using different statistical techniques and methodologies. They approach the problem by gathering various other works written by the authors and insuring that they are consistent, in terms of word usage, with the writing style these authors used in the *Federalist Papers* in the undisputed articles. In total they use 94,000 words of text written by Hamilton and 114,000 words of text written by Madison. The goal of their work is to use this text to build a statistical model for each author and then test the disputed *Federalist Papers* to see which model they are more likely to have come from.

Sentence length is dismissed by Mosteller and Wallace for use as discriminator between the two authors, because it is distributed very similarly for both. Instead they focus on *words* that discriminate well between the authors (i.e. occur more frequently in one author than another). They were inspired by similar words used with different frequencies between two authors, for instance Madison uses the word "whilst" almost exclusively while Hamilton prefers "while". These words' frequencies were not used as a *ratio* (i.e. while/whilst) as is commonly attributed to their work.

according	also	although
always	an	apt
both	by	commonly
consequently	considerable(ly)	direction
enough	innovation	kind
language	matter(s)	of
on	particularly	probability
there	this	through
to	upon	vigrous
while	whilst	work(s)

Table 3.1: The 30 Mosteller and Wallace *marker words* chosen because they discriminate well between Alexander Hamilton and James Madison.

Individual word frequencies were counted as a percentage of all words in the collection of writing (i.e. the occurrences of a particular word divided by the total number of words in the corpus). These percentages are modeled assuming independence between words, not as a ratio between particular pairs of words.

Even with the large amounts of text gathered they conclude that most words do not have a high enough frequency to give reliable results. They form an initial set of 165 words whose frequencies have the ability to discriminate between texts of the different authors. They call these words *marker words* and based on further testing on held out data they narrow this set down to the best 30. These words are shown in Figure 3.1.

Mosteller and Wallace apply a Bayesian approach[10] to modeling the distributions of these words. This involves assuming that the frequency of each word comes from a negative binomial distribution. The negative binomial distribution requires parameters so, because they are using a Bayesian approach, they estimate the distribution of

[10]Bernardo and Smith [1995] and Efron [1986] give good background on Bayesian methodologies and how they differ from traditional statistical approaches.

these parameters from the data and assume they follow a Beta distribution. Mosteller and Wallace choose to use the negative binomial and Beta distribution based on a study of the behavior of ninety very common words (different from the *marker words*). They use these distributions to predict the authorship of the undisputed works. The study by Mosteller and Wallace deals heavily with this Bayesian approach and how to apply it to a practical problem like authorship attribution. To this end, the study also compares this Bayesian approach to different traditional statistical methods for solving this problem.

This groundbreaking and comprehensive study of the *Federalist Papers* and the statistics useful for modeling word frequencies, unfortunately has had very little impact on the techniques used in the field of authorship attribution. Holmes and Forsyth [1995] give an overview of the work in Mosteller and Wallace [1964] and say with regard to their influence "they have been more admired than emulated". They popularized the use of the *Federalist Papers* as a research corpus and their *marker words*, but did very little to propagate the Bayesian approach to modeling language in authorship detection. This, according to Holmes and Forsyth [1995], is most likely due to the fact that the Bayesian approach is more difficult to understand than the traditional statistical approach and Mosteller and Wallace [1964] showed that both yielded similar results.

McColly and Weier [1983] have also experimented on the *Federalist Papers*, but they used different statistical techniques and attempted to determine the similarity between different articles. McColly and Weier use the log-likelihood ratio test between pairs of the articles in the *Federalist Papers* and then give a probability that

they were written by the same or different authors. They pair articles written by Hamilton and Madison (where authorship is known) and show that using the same word set as Mosteller and Wallace (shown in Table 3.1) it is possible to tell when two articles have been written by the same or different authors. They use a log-likelihood test to measure if the distribution of these words from each article come from the same population. If the p-value for this test is very low then then we can reject the hypothesis that they were written by the same author. For every test pairing of known authors given in McColly and Weier [1983] the correct authorship (same or different) is determined correctly by this test. In addition to these results they also come to very much the same conclusions as Mosteller and Wallace [1964] as to the authorship of the disputed articles.

The results of McColly and Weier [1983] are not comparable with those of Mosteller and Wallace because they are performing different tests. McColly and Weier [1983] are not making use of the prior distribution of words from the two authors, only comparing the distributions of the words in the two articles to see if they are similar. McColly and Weier's tests, in a way, can be seen as unsupervised as there is no training data involved. The tests are performed only with regard to the pair of articles. They are however using the *marker words* found by Mosteller and Wallace to be *good* at discriminating between Hamilton and Madison. These words were found using training data, so this technique is not actually unsupervised. The results of McColly and Weier are nonetheless impressive because, even though they use the *marker words*, they do not have prior probabilities for how likely each author is to use them.

McColly and Weier created a second type of experiment which used a word set based on non-content words that occur with high frequency. These words are not necessarily good discriminators like the words used by Mosteller and Wallace. The words used are function words that occur most often overall in the articles (they could even be used only by one author). These words were chosen to see if it is possible to correctly label pairs of articles with no prior knowledge at all. They perform the same set of tests with these words and show that the results are very poor compared to those using the *marker words*. The results of this final comparison are sometimes used to dismiss function words as bad discriminators of authorship, but this is completely misinterpreting the results. This finding tells us nothing about function words in general. It merely shows that words that were chosen to be good discriminators (like the *marker words*) will be better than words that are picked randomly just because they occur many times. (Results were never misinterpreted by McColly and Weier, but in the research of other authors, who shall remain nameless.)

A good history of the origins of many of the techniques used in authorship attribution can be found in Holmes [1994] and Love [2002] gives a nice overview of the field and its history.

3.3 Detecting Stylistic Inconsistencies

A field closely related to authorship attribution is the detection of stylistic inconsistency. Often this is to aid in the measurement of stylistic unity [Smith, 1998; McColly, 1987] in questions of multiple authorship or for the purposes of improving collaborative writing. Text written with incongruous style can be confusing and diffi-

cult to read. There has been research into automatically identifying these changes in writing style to aid multiple authors working on a single project to diagnose inconsistencies so that they may fix them. Baljko and Hirst [1999] showed that humans can intuitively detect and agree on sections of a text where the style is incongruous (because it is written by more than one author) and this has further motivated research into automatic techniques for performing this task.

Glover and Hirst [1996] were the first to directly approach the problem of detecting a change in authorship. They had subjects watch half of a video and write a summary of it and then return weeks later to watch the second half and complete the essay. They then counted different stylistic features in the essays to see if any were consistent in an author's writing between the first and second halves. They also generated all possible combinations of different authors' first and second halves and measured the same features over this data to determine the features that were good discriminators of inconsistency (or a change in author). They tested several different features:

- word length;
- sentence length;
- percentage of two and three letter words;
- distribution of parts-of-speech;
- distributions of the part-of-speech used at the beginning or end of sentences.

Results indicated that all of these features were fairly consistent with respect to a single author's style and that the percentage of two and three letter words, and the use of coordinating conjugations were particularly useful in discriminating between these writers.

Graham [2000] and Graham et al. [2005] further explore this problem using stylistic features, similar to the ones mentioned above, to identify the *points* in a document where its stylistic character changes. These can either be points indicative of a change in author, as in collaborative writing where authors have not integrated their writing styles, or points where a single author's writing style is inconsistent within a document. Graham et al. construct a corpus made from posts in a Usenet group (making the assumption every post is written by a different author and that single posts are written by the same author). They then compute many stylistic features over this corpus. They train several supervised neural network classifiers on pairs of paragraphs, either on ones that occur in the same post (thus are assumed to have the same author) or on paragraphs across different posts (the author is assumed to be different.) The classifier is tested by feeding it similar pairs of paragraphs to determine if it can tell if they occurred in the same post or a different post. They achieve an F-measure of .53 , which is an improvement on their baseline of randomly choosing the correct classification, which gives an F-measure of .23 (this is because they have an unequal number of same-author and different-author pairs). This provides some evidence that the use of stylistic features can be useful to distinguish between authors even in small segments of text (they used paragraphs which averaged 50 words in length and had a standard deviation of 41 words).

What differentiates the work of Graham et al. from most other authorship attribution research (other than their choice of classifiers) is that they show it is possible to learn what makes the style of authors' writing *different in general.* This is in contrast to most other authorship work which has typically focused on learning the particular

style of a single author, or two authors, so that work can be correctly attributed to an author automatically.

The goals in detecting stylistic inconsistencies are very similar to ours in unsupervised anomaly detection and thus many of the same techniques and ideas apply to our research. In our research, however, we are primarily interested in finding anomalies in an entire collection of data and not simply differences between one paragraph and the next, or one document and another. Another major differentiating factor is that, as with typical approaches to authorship attribution, research on detecting stylistic inconsistency uses training data to train a classifier to recognize the style of specific authors and we did not wish to be reliant on the availability of training data.

3.4 Genre Identification

The term *genre* is used to refer to a category of literary composition characterized by a particular form and style. This is often very closely related to topical content, as genres often have specific content associated with them, but the term is used to refer to the **non-topical** aspects of a piece of writing or *text type* [Biber, 1988, 1989] such as, its intended audience, communication style, or purpose. Scientific journal articles and fishing magazine articles would be examples from two different genres that happen to have individual topics associated with them, but it is possible to have different genres on the same topic, for instance a scientific journal article about a research study and a subsequent newspaper story covering that research.

A large part of the work in the study of genres has been concerned with the linguistically motivated task of defining classifications for texts that separate them

into different types based on their linguistic form. Traditional genres like *articles*, *editorials*, *reviews*, and *letters* are, according to Biber [1989], a kind of "folk typology" of texts. Biber explains that these categories are simply based on the most easily discernible external attribute, not on any internal linguistic qualities. As an example, Biber offers the fact that "two newspaper articles can range from extremely narrative and colloquial in linguistic form to extremely informational and elaborated in form", but they are still traditionally grouped into the same genre. The goal of his research has been to define new genres or *text types* that more informatively describe the internal linguistic features of texts.

Biber [1988, 1989, 1992, 1995] uses a technique called *Multidimensional Analysis* to analyze texts from a wide range traditional genres and derive a set of linguistic groupings or *text types*. He starts by defining sixty-seven features that he believes will be useful for this task based on the linguistics literature. These features are mostly composed of the percentages of different specific parts of speech (for example, the number of demonstrative pronouns, gerunds, infinitives, *wh*-relative clauses, etc.) plus a few features like the type/token ratio and the mean word length. These features are computed over all texts in his collection to construct a matrix. He then performs *Factor Analysis* on this matrix to get what he calls "textual dimensions". Factor analysis is really another name for principal component analysis (see section 2.4.2), but where typically only the components with the highest eigenvalues are kept. In Biber's case he ends up with six components or "textual dimensions" (from the original sixty-seven). These dimensions will show where the greatest variation in the data occurs.

Biber then takes each of these these six dimensions and qualitatively analyzes it to describe its bearing on different texts in the collection. For example, one of the dimensions he labeled "Narrative versus Non-narrative Concerns" and he postulates that this dimension separates texts that are concerned with narrative discourse (e.g. novels) and use a high percentage of past tense verbs and third person pronouns from text which have more descriptive or "expository" concerns (e.g. scientific papers) and avoid using past tense verbs and third person pronouns. These six dimensions are further used to cluster the original texts into eight groups, based on the number of features they have in common with these dimensions. Biber then analyses these eight clusters of documents qualitatively to produce a *typology of English texts*. These are the categories that more accurately describe the internal linguistic forms of text (*text types*) in his corpus.

A different field of research dealing with genre is that of using computers to decide the genre of unseen documents or *genre classification* [Kessler et al., 1997; Argamon et al., 1998, 2003]. This is relativity new field when compared with research on authorship attribution and has been principally motivated by a desire to improve information retrieval [Karlgren and Cutting, 1994; Karlgren, 1998; Santini, 2004]. Search engines that return results over a large collection of different types of documents and data, like the internet, could allow genre-specific searching and thus improve the quality of the results returned. Other work by Maynard et al. suggests that genre detection can also aid in information extraction [Maynard et al., 2001, 2003].

Research in this field typically makes use of at least some stylistic features to classify genres. Import work in the field was done by both Karlgren and Cutting [1994]

and Kessler et al. [1997] who have investigated different approaches to automatically classifing documents in the Brown Corpus [Francis and Kucera, 1964] according to genre. Their work makes use of discriminant analysis and logistic regression, respectively, to train on data that has been labeled with its correct genre. They make use using linguistic features (similar to Biber's features) and attempt to classify unseen data into its proper genre. Karlgren and Cutting are able to achieve a 4% error rate, when classifying documents into 2 different genres and a 27% error rate, when classifying documents into 4 genres.

Almost all work in this area involves training on a hand-crafted data set containing documents labeled with their correct genres and developing techniques to classify new documents into these genres. Unsupervised work on genre classification is rarer, but it more closely relates to our research because it does not make use of training data. Instead, this work makes use of the statistical method of clustering, where by similar objects are grouped together into subsets. Clustering techniques have typically been used in natural language processing to classify documents by topic (Manomaisupat et al. [2006] for example), but Bekkerman et al. [2006], Rauber and Müller-Kögler [2001] have employed them for the unsupervised grouping of documents by genre.

The work of Bekkerman et al. is interesting because it uses clustering on a data set with pre-defined genre markings so that results can be evaluated for accuracy. This work makes use of a multi-way distributional clustering technique to group documents into 21 marked genre categories of the British National Corpus [Burnard, 1995; Burnage and Dunlop, 1992]. They achieve 50% accuracy when classifying documents into these 21 genre categories and conclude that bag-of-word features give

4% better results for his task than features that use part of speech trigrams. This result is extremely impressive considering the number of genre categories and that no training data was used. (Especially as there is so much disagreement about what genres most accurately describe texts.)

Clustering techniques may be related, but anomaly detection is not a unsupervised classification task. Mostly this due to the fact that in anomaly detection the problem is not to group similar items together, but rather to identify ones that are most different from *all* others. Anomalous text is defined by virtue of the fact that it is simply not like some majority of the text. There is no assumption that anomalies would form a cluster by themselves or that necessarily the rest of the text would form a single cluster. If we did try to employ clustering techniques, in the hope that all 'normal' data would cluster together it would also be very difficult to determine at what point to stop the clustering so as to be left with only anomalous outliers.

3.5 Summary

This chapter presented an overview of work in computational linguistics which deals with the characterization of the style and language used in a piece of writing. Specifically we focused on research in authorship attribution, detection of stylistic inconsistency, and genre identification.

Authorship Attribution typically relies on the assumption, expressed clearly in Sinclair and Coulthard [1975] and Coulthard [2004], that every writer has their own vocabulary and preferences which they build up over time (called their *idiolect*) that influences the word choice and structure of their writing. The idea in authorship

attribution is that the individual preferences or idiolects of authors should allow us to identify them. We examined work by Mosteller and Wallace that makes use of the distribution of words in a text to distinguish between authors, as well as, work like that of Glover and Hirst which focuses on using stylistic features such as word length and sentence length to detect a change in author. These works, as well as other research, have achieved good results using these techniques and have advanced the notion that authors idiolects can at the very least be used to reliably discriminate between authors.

In the final section of the chapter we looked at identifying non-topical aspects of a text that can be used to characterize a particular text-type or genre. We reviewed the work of Biber that focused on automatically defining text-types, as well as the work of many other authors on the problem of automatically classifying documents by genre (both supervised and unsupervised).

The methods covered in this chapter for identifying authors and genres are not directly applicable to the task of anomaly detection, but nonetheless we rely heavily on many aspects of this research. In particular for the features we used to characterize text that are described in the next chapter.

Chapter 4

Characterizing Text

4.1 Overview

How best to characterize text is by no means straightforward and is influenced by the problem one is trying to solve as well as the techniques and data available. Depending on the task, or what it is about language you are trying to model, you might attempt to model the characters, words, n-grams, phonemes, sentiment, content words, or any number of other properties. An important goal of our research is the identification of pieces of text that are anomalous in very small collections (paragraphs in a document for example). This has led us to focus more on features that capture the *style* of writing, rather than on content word based features, which typically are more susceptible to the data sparsity problem of language [Guthrie et al., 2003]. The use of words in language is so varied and complex that word-based features typically require more data than stylistic features in order to see the repetition necessary to build sufficient models. We therefore concentrate on features that are based on large classes

of words and features from stylometry. Determining which stylistic measures are most useful from the literature can be difficult (see Chapter 3), as often features are used for different tasks or on different data sets. We approached this problem by implementing a wide range of features previously used in the fields of genre identification, authorship attribution, detecting stylistic inconsistency and content-analysis. We chose features that were popular in the literature and empirically determine which are best suited to the task of anomaly detection in small collections. All features described in this chapter were specifically chosen because they were used by different authors in at least three research papers. The only exceptions to this rule are the seven novel features in Section 4.4 which seemed natural extensions of the readability measures.

We have chosen to characterize a text by representing it as a vector, where the components of the vector are based on these stylistic and linguistic features. In this context, we then choose to approach anomaly detection as a type of high dimensional outlier detection (described in Section 2.4), where these features correspond to the dimensions (variables) and pieces of text correspond to observations. The vectors are computed for pieces of text and we vary the size of these pieces of text and measure the effect this has on anomaly detection. In the rest of this chapter we describe all the features we tested.

We use a total of 166 features, 31% of which are typical stylistic features, the rest being features that attempt to capture emotional tone. The features used break down as follows:

Feature Type	**Number of Features**
Simple Surface Features	19
Readability Measures	7
Obscurity of Vocabulary Features	7
Part of Speech and Syntax Features	11
Rank Features	8
Emotional Tone Features	114

4.2 Simple Surface Features

The simplest measures of style treat the text as a collection of tokens grouped into words and sentences and make use of the distributions of these features. These include things like the average length of sentences, the amount of punctuation used, and the usage of *function* words (words like prepositions, articles, and pronouns that have little bearing on content [Ellegard, 1962]). In this research we implemented the most commonly used surface features from the literature and for each piece of text computed:

1. *Average sentence length*
2. *Average word length*
3. *Average number of syllables per word*
4. *Percentage of all words that have 3 or more syllables*
5. *Percentage of all words that only have 1 syllable*
6. *Percentage of long sentences (sentences greater than 15 words)*
7. *Percentage of short sentences (sentences less than 8 words)*
8. *Percentage of sentences that are questions*
9. *Percentage of all characters that are punctuation characters*
10. *Percentage of all characters that are semicolons*
11. *Percentage of all characters that are commas*

12. *Percentage of all words that have 6 or more letters*
13. *Percentage of word types divided by the number of word tokens*
14. *Percentage of words that are subordinating conjunctions (then, until, while, since, etc.)*
15. *Percentage of words that are coordinating conjunctions (but, so, but, or, etc.)*
16. *Percentage of sentences that begin with a subordinating or coordinating conjunctions*
17. *Percentage of words that are articles*
18. *Percentage of words that are prepositions*
19. *Percentage of words that are pronouns*

4.3 Readability Measures

Readability measures [Stephens, 2006; Flesch, 1974] attempt to provide a rough indication of the reading level required for a text. These measures were first developed in the 1920's in response to teachers who wanted science text books more suited to their students' reading levels. The formulas provided a guide for how difficult a text was to read. Readability formulas are still used in education today, as well as in the military, to gauge the difficulty of training materials. All these formulas make use of a few basic features such as average sentence length, average word length, average syllables per word, words with 3 or more syllables, words with 6 or more letters. These measures are obviously lacking where true readability is concerned because they do not directly capture the obscurity of the vocabulary, whether ideas flow logically, or the complexity of the grammatical structures used, but they are nonetheless useful as an approximation of how simple a text is to read. There are many different measures

of readability and it is not clear how they correspond on different texts [Mailloux et al., 1995]. They have been used successfully in the literature to separate genres [Kessler et al., 1997; Dewdney et al., 2001] and writing style [Clough, 2000] (and possibly even authors [Luyckx and Daelemans, 2005]). We have implemented and tested all the most popular readability measures:

- **Flesch-Kincaid Reading Ease**

$$Reading\ Ease = 206.835 - 1.015\left(\frac{total\ words}{total\ sentences}\right) - 84.6\left(\frac{total\ syllables}{total\ words}\right) \quad (4.1)$$

- **Flesch-Kincaid Grade Level**

$$Grade\ Level = 11.8\left(\frac{total\ syllables}{total\ words}\right) + 0.39\left(\frac{total\ words}{total\ sentences}\right) - 15.59 \quad (4.2)$$

- **Gunning-Fog Index**

$$Fog\ Index = \left(\left(\frac{total\ words}{total\ sentences}\right) + \left(\frac{words\ with\ 3\ or\ more\ syllables}{total\ words}\right) \times 100\right) \quad (4.3)$$

- **Coleman-Liau Formula**

$$ColemanLiau = 5.89\left(\frac{total\ characters}{total\ words}\right) - 0.3\left(\frac{total\ sentences}{total\ words \times 100}\right) - 15.8 \quad (4.4)$$

- **Automated Readability Index**

$$ARI = 4.71\left(\frac{total\ characters}{total\ words}\right) + 0.5\left(\frac{total\ words}{total\ sentences}\right) - 21.43 \quad (4.5)$$

- **Lix Formula**

$$Lix = \left(\frac{total\ words}{total\ sentences}\right) + 100\left(\frac{total\ words\ with\ at\ least\ 6\ letters}{total\ words}\right) \quad (4.6)$$

- **SMOG Index**

$$SMOG = 3 + \sqrt{\frac{words\ with\ 3\ or\ more\ syllables \times 30}{total\ sentences}} \quad (4.7)$$

4.4 Obscurity of Vocabulary Usage

One distinguishing feature of writing is how ordinary or obscure the choice of vocabulary is and we created some novel features that attempt to capture this. We expect that some authors chose many words that are not very commonplace in normal writing while others may prefer to stick to more everyday, less obscure words. This captures a notion very similar to Ahmad's *weirdness* of vocabulary [Ahmad and Rogers, 2001; Ahmad and Al-Sayed, 2005], but instead of comparing the distributions of a word in a sample with a reference corpus, we look for what percentage of words in our sample occur often in our reference corpus. For every segment of text, we calculate how frequently its words appear in 10 years of newswire using the Gigaword Corpus[11] [Graff, 2003]. First we ranked all words by frequency in the Gigaword corpus, and then we make sets of words based on these frequencies. Features of this sort have never, to our knowledge, been used for similar tasks like authorship attribution, genre identification, etc., but our experimental results prove them to be extremely valuable (as can be seen in Chapter 7 when we look at feature evaluation). The sets of words we make from the Gigaword are the:

1. *Top 1000 words*
2. *Top 5000 words*
3. *Top 10,000 words*
4. *Top 50,000 words*
5. *Top 100,000 words*
6. *Top 200,000 words*

[11]For a detailed description of the Gigaword corpus see Appendix B.

7. *Top 300,000 words*

We then measure the distribution of words into these sets for any piece of text. So for any section of text we compute the percentage of words in that section that occur in each of the 7 sets of words individually. (Thus we have 7 features.)

4.5 Part of Speech and Syntax Features

We also implemented and evaluated features that make use of the part-of-speech of a word. Text is passed through the RASP[12] (Robust and Accurate Statistical Parser) system's part-of-speech tagger [Briscoe and Carroll, 2002; Briscoe et al., 2006] developed at the Universities of Sussex and Cambridge. The RASP part-of-speech tagger is trained on a subset of the British National Corpus (BNC) [Burnage and Dunlop, 1992; Burnard, 1995] and uses a hidden markov model bi-gram tagger. This tagger has good accuracy, evaluated at over 97% on texts with a vocabulary similar to training texts [Briscoe and Carroll, 2002]. The RASP team also has augmented the tagger with an unknown word model and a series of hand crafted rules for known, but rare, words and has shown that the tagger now has very little degradation in accuracy on out-of-domain texts [Briscoe et al., 2006].

All words and characters are tagged with one of 150 part-of-speech tags from the CLAWS 2 tagset [Leech et al., 1994]. We use this mark-up to compute features that capture the distribution of parts of speech.

1. *Percentage of words that are adjectives*

[12]The RASP parser has been made available by its authors and can be found at `http://www.informatics.susx.ac.uk/research/groups/nlp/rasp/`.

2. *Percentage of words that are adverbs*
3. *Percentage of words that are interrogative words (who, what, where when, etc.)*
4. *Percentage of words that are nouns*
5. *Percentage of words that are verbs*
6. *Ratio of number of adjectives to nouns*
7. *Percentage of words that are proper nouns*
8. *Percentage of words that are numbers (i.e. cardinal, ordinal, nouns such as dozen, thousands, etc.)*
9. *Diversity of POS tri-grams*

$$POS\,Trigram\,Density = \left(\frac{number\ of\ different\ POS\ trigrams}{total\ number\ of\ POS\ trigrams}\right) \times 100 \quad (4.8)$$

Texts are also run through the RASP morphological analyzer, which produces words lemmas and inflectional affixes. These are used to compute:

Percentage of passive sentences Sentences are counted as passive if they contain the following pattern:

(Form of the verb "be")(adv)*(past tense of a verb)

Percentage of words nominalizations Nominalizations are spotted by searching the suffixes produced by the RASP morphological analyzer for *tion*, *ment*, *ence*, and, *ance*.

4.6 Rank Features

Authors can often be distinguished by their preference for certain prepositions over others or their reliance on specific constructions of phrase. We capture these

preferences by keeping a ranked list sorted by frequency of several different kinds of function class words and part-of-speech bi-grams and tri-grams.

1. *Distribution of POS tri-grams list*
2. *Distribution of POS bi-gram list*
3. *Distribution of POS list*
4. *Distribution of Articles list*
5. *Distribution of Prepositions list*
6. *Distribution of Conjunctions list*
7. *Distribution of Pronouns list*
8. *Distribution of Adverbs list*

We compute all these lists and store them for every piece of text. These ranked lists are very different than the features described previously, because they are not numerical. Recall, that we are characterizing a piece of text as a vector of features, but these ranked lists do not allow for meaningful numerical values except when compared to other lists (so that their ranks can be compared). For this reason we have experimented with the use of these rank list features only when measuring the similarity or distance from one piece of text to another. When measuring the similarity between two pieces of text, for example, we can calculate eight features by comparing the eight lists from each piece of text. We compute the similarity between any two ranked lists, x and y using the Spearman Rank Correlation formula:

$$\mathcal{S}(x,y) = 1 - \frac{6\sum_{i=1}^{m} d_i^2}{m(m^2-1)} \tag{4.9}$$

where d_i is the distance between the i^{th} item in list x and that item's rank in list y and m is the number of items in each list.

4.7 General Inquirer Dictionary

The General Inquirer Dictionary (`http://www.wjh.harvard.edu/~inquirer/`) was initially developed by the Social Science Department at Harvard based on Charles Osgood's attempts to quantify the connotative meaning of words and the efforts of Dexter Dunphy [Stone et al., 1966]. It consists of mappings from words to social science content-analysis categories. These content-analysis categories attempt to capture the tone, attitude, outlook, or perspective of words and for this reason it has been used to determine the sentiment of texts [Ahmad, 2008; Tetlock, 2007]. The Inquirer dictionary we used consists of over 13,000 root words mapped into 114 categories with most words assigned to more than one category. The two largest categories are 'positive' and 'negative' which account for 1,915 and 2,291 words respectively.

The General Inquirer Dictionary's main group of categories are called *Harvard IV-4*, but it has been built up with other smaller dictionaries and by various users and contributors. This has increased the size of the dictionary and has also lead to some different branches of the dictionary being made available. A decision was made by us to use the version of the Inquirer Dictionary which is in use in the online content-analysis tool at the University of Maryland (`http://www.webuse.umd.edu:9090/`). This decision was made because this dictionary is larger (i.e. has more words) than the one currently available at Harvard and also because this dictionary does not include the extra *Lasswell* categories (which are more topical categories like 'nations' , 'transactions' , and 'wealth' and thus less concerned with the sentiment or tone of a text).

We make use of these General Inquirer Dictionary by keeping track of the per-

centage of words in a segment that fall into each of the 114 categories. So, we keep one feature for the percentage of 'positive' words, one for the percentage of 'negative' words, one for the percentage of 'hostile' words, etc. A sample of the General Inquirer Categories we use as features are shown in Table 4.7[13].

Positive	Negative	Strong
Hostile	Self-referencing	Weak
Casual slang	Think	Negate
Know	Compare	Person Relations
Need	Pleasure	Pain
Affection	Work	Active
Passive	Overstated	Understated
Agreement	Disagreement	Virtue

Table 4.1: Some of the General Inquirer *Harvard IV-4* Categories

4.8 Summary of Features

The features analyzed in this research for the detection of anomalies in text represent a large selection of the features previously used in the fields of genre identification, authorship attribution, detecting stylistic inconsistency and content-analysis. We chose to investigate a total of 166 features (listed in the previous sections of this chapter) based on several important criteria. Firstly, we concentrated only on features that do not make use of counts of content words. Content words have been used in an enormous amount of research to classify text by topic successfully, but our research is not only concerned with detecting a change in topic, but also the identification of

[13]A detailed list of all categories and their descriptions can be found at `http://www.wjh.harvard.edu/~inquirer/homecat.htm`

text that is different because the writing style, genre or tone of the text is different. It was important in our research to test whether textual anomalies can be detected without the topical clues which these words provide as one important application of this work is the identification of plagiarized passages, which will most likely be the same topic as the rest of the text. It was also important in our research to make use of features that have been widely used in other research in this field and to give an indication of how useful these features are for this task.

Chapter 5

Identifying Anomaly

5.1 Overview

In this chapter we describe our procedures for identifying anomalies. We present five different methods for anomaly detection and give detailed descriptions and implementation details of each. Two of these methods are based on procedures used in high dimensional outlier detection, two derive from related areas, and one is completely novel and devised by us specifically for anomaly detection in text. The final sections of the chapter describe the different methods we tested for normalizing the data, a overview of the distance measures that were used, and finally we give a description of the system we built for experimentation and visualization of results.

5.2 Detection Methods

We devised and tested five methods for the unsupervised detection of anomaly. They are all closely related in that they use the same set of features to characterize text and their goal is to produce a ranking of how anomalous one piece of text (segment, document, etc.) is to the rest of the text in the document (or collection of documents). They differ based on how these features are used and how segments are compared.

We have chosen to describe each method in the setting of choosing anomalous segments from a document, but note that these methods equally apply to detecting pieces of text of any size, such as picking out anomalous documents from a collection of documents. All methods measure the anomalousness of a piece of text by characterizing it as a vector (of features), $\vec{x}$, and defining a function for measuring the anomalousness of $\vec{x}$ with respect to the set of all segments in a document D.

Three of the five methods tested are variations on techniques that have been used to detect outliers in statistical data. One is a variant of average linkage clustering and one technique is to our knowledge completely novel. The methods are:

ClustDist	A distance based on average linkage clustering
SDEDist	The Stahel-Donoho Estimator distance
PCout	The weights calculated by the PCout algorthim
MeanComp	Distance from the mean of all other segments in the data
TxtCompDist	A novel method using the distance from the *textual* complement

The rest of the chapter gives a full discussion of these methods.

5.2.1 ClustDist (Average Linkage Distance)

This is a simple method for identifying anomalies that is similar to the procedure for measuring the distance between clusters in average linkage clustering [Manning and Schütze, 1999]. Intuitively, we measure the average distance from one piece of text to all other pieces of text and then average this result.

For a document D with n segments we first characterize each individual segment, i, by computing the p features over it and generating a vector of these features $\vec{v}_i$. In our experimentation we used the paragraphs in documents as well as fixed segment sizes. We define a $n \times p$ matrix $\mathbf{V}$ where each row in the matrix corresponds to a segment's feature vector.

$$\mathbf{V} = \begin{pmatrix} \vec{v}_1 \\ \vec{v}_2 \\ \vdots \\ \vec{v}_n \end{pmatrix} \quad \text{, where } \vec{v}_i \text{ is a vector of features representing segment } i$$

(Figure 5.1 shows the construction of this matrix $\mathbf{V}$).

We calculate the anomalousness of a piece of text, say x, with respect to all segments in the document, $\mathbf{V}$, by representing x as a vector of features, $\vec{x}$, and computing:

$$ClustDist(\vec{x}, \mathbf{V}) = \frac{\sum_{i=1}^{n} d\left(\vec{x}, \vec{v}_i\right)}{n} \tag{5.1}$$

Where d can be any measure of the distance between two vectors. We experimented with different methods for measuring this distance d (e.g. Euclidean, city block, cosine) and these are described in Section 5.4.

ClustDist calculates this average distance (anomalous score) for all segments in

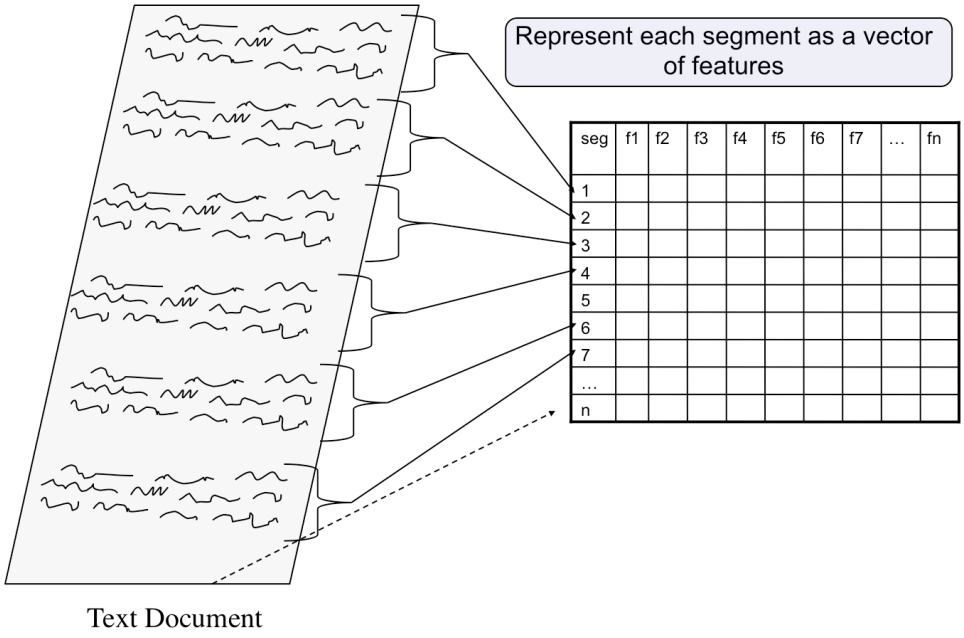


Figure 5.1: Representing Documents

a document. So, we are simply measuring how far away each segment is from every each other segment one by one and averaging these results. We hope that segments which are anomalous are on average farther away from every other segment. This can be visualized by constructing a distance matrix (see Figure 5.2) by measuring the distance between each vector and every other vector. The *anomalousness score* of any vector is the average of its distance from all other segments (a row or column in the distance matrix). This *anomalousness score* can then be used to produce a ranking. This ranking corresponds to how different (on average) a segment is to all other segments in a document.

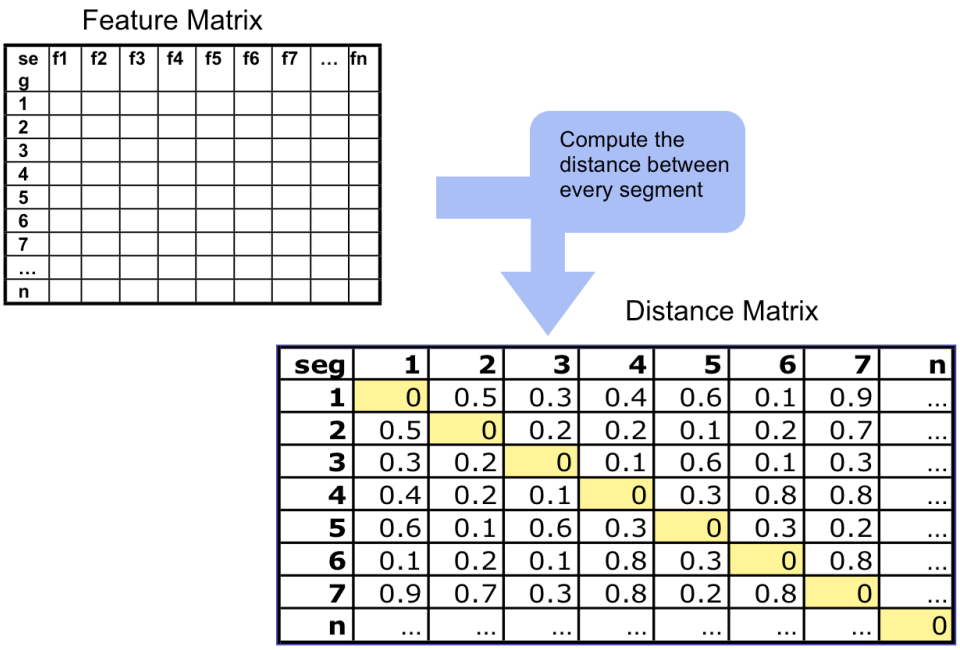


Figure 5.2: Creating a Distance Matrix

5.2.2 SDEDist (Stahel-Donoho Estimator Distance)

This measure of the anomalous of a segment is based on the Stahel-Donoho Estimator, which has been used in statistics for the identification of outliers (described in detail in Section 2.4.1). The Stahel-Donoho Estimator says that to measure an observation's outlyingness in some multivariate data, you should find a direction in space that the data can be projected onto which will give the maximum possible univariate outlyingness for that observation. Here univariate outlyingness is taken to be the robust z-score (Equation 2.1). In practice, it is impossible to project the data into all possible directions to determine which gives the largest distance, so instead we choose a finite set of directions to project the data into and take the maximum distance for observations among these projections. Let, $\mathbf{V}$, be the feature matrix for a document, D, that contains as its rows all segments in the document and as columns the p features computed over those segments. Then, for a new segment of text, x, that we have computed features over to give a column vector $\vec{x} = (f_1, f_2, ..., f_p)^T$, the SDEDist measure of anomalousness with respect to a document D's feature matrix $\mathbf{V}$ is:

$$SDEDist(\vec{x}, \mathbf{V}) = \max_{\vec{a}} \frac{\vec{x}^T\vec{a} - \text{median}(\mathbf{V}\vec{a})}{\text{mad}(\mathbf{V}\vec{a})} \tag{5.2}$$

Where a is a direction (unit length column vector) in $\mathbb{R}^p$ from our finite set of directions to test (with length one). The difficulty of using this procedure is how to choose the set of directions to test. There has been very little research in how to choose these directions when the number of features is very large, as in this case, where we have over 150 variables that we are using to characterize text. We can lessen this problem

if we first represent the data "in terms of its own dimensionality" as recommended by Hubert et al. [2005]. This is achieved by calculating the singular value decomposition of the centered feature matrix (see section 2.4.2 page 48) and taking $\mathbf{U\Lambda}$ to be our new feature matrix. This new feature matrix will be at most dimension $n \times n$ and results in no loss of information. Our SDEDist method for identifying anomalies always performs this procedure as an initial step on the feature matrix to reduce the number of dimensions and from this point on (when discussing SDEDist) we will assume that the feature matrix has undergone this transformation.

The approach we take for choosing directions is one which was used by Struyf and Rousseeuw [2000] for the task of finding the deepest location in multivariate data (which is closely related to outlier detection), but has to our knowledge not been used for outlier detection before. The procedure chooses four types of directions:

1. The p coordinate axes

2. Vectors containing an observation and the coordinate-wise median of the data. For the feature matrix $\mathbf{V}$ let the coordinate-wise median be $\bar{\mathbf{v}}$. This is just $v_i - \bar{\mathbf{v}}$ for $i = 1, ..., n$. Each of these vectors is then normalized to have length one by dividing by their Euclidean norm. (So, for a vector x, we divide by its Euclidean norm $\frac{x}{||x||} = \frac{x}{\sqrt{\sum x^2}}$)

3. Vectors containing two observations. We choose two segments at random from the data matrix call them v_a and v_b and then take as a direction the vector $v_a - v_b$ (and as before normalize it to length one). Struyf and Rousseeuw [2000] only choose 250 directions of this type, but as this process of randomly picking segments and subtracting them is extremely fast, we choose a minimum of 750

directions in this way. If the number of segments, n, is less than 50 then we forgo random sampling and just calculate all combinations as this is just $\binom{49}{2}$=1176 directions.

4. Vectors perpendicular to a subset, h, of the observations. Here we take the number of observations in h to be a third of the segments in our document, $\frac{n}{3}$, and construct the matrix by randomly selecting this many segments from $\mathbf{V}$. We then calculate the eigenvectors of its covariance matrix (see Section 2.4.2). All of these eigenvectors are then added to our set of directions (eigenvectors are already of length one, so there is no need to normalize them). As we explained in Chapter 2, these eigenvectors will point in directions of the greatest variance in the data, so the hope is that if we choose subsets and directions in this manner that one subset will contain some *good* data as well as some outliers. In this case the variance will likely be largest in the direction of the outlier and we will get an eigenvector pointing in this direction. This is exactly the kind of direction that will maximize the Stahel-Donoho distance. We repeat this process 250 times adding all eigenvectors to our set of directions on every iteration (each subset has n eigenvectors because the feature matrix has been made to be $n \times n$).

Struyf and Rousseeuw [2000] explain that the first three types of directions are mainly chosen because they are fast to compute and that the fourth type is most likely to give *good* directions that will be useful for determining how far away points are from the center of the data. In addition the 4 types of direction listed above, we add a fifth type of direction which is also fast to compute:

5. Random directions based on normal data. We add 200 directions by randomly generating 200 multivariate normal observations with mean 0 and with covariance matrix that is the identity matrix, $\mathcal{N}(0, \mathbf{I})$[14]. We treat these observations as vectors and normalize them to have a length one. We can visualize this step as randomly generating points in a spherical shape around the origin and then picking the directions that go from the origin to these points.

The SDEDist is computed for every segment in a document over this entire set of directions. We then rank segments by this degree of anomalousness.

The SDEDist was the most computationally intensive measure that we tested and required more than 4 times as much time to run as the next slowest method. This method of choosing directions for each document in our experiments generated 13,500 directions (as our test documents typically have n equal to 50) onto which the data must be projected. We implemented this method in the R statistical language, which makes use of the LAPACK [Anderson et al., 1990] linear algebra library, so matrix operations are as fast as possible, but nonetheless, Step 4 for generating directions still adds considerable computation time as we repeatedly perform SVD to get the eigenvectors of subsets.

[14]Random multivariate normal observations were generated using the *mvrnorm* function in the *MASS* package, which is part of the R statical language install.

5.2.3 PCOut (Principal Component Weighting Distance)

The PCOut method of outlier detection, developed by Filzmoser et al. [2008], is described in detail in Section 2.4.3, but to summarize, it measures the distance from observations in principal component space using dimensionality reduction (to decrease the number of components) and the kurtosis measure to weight the different components. We make use of this procedure to identify anomalies in text, by constructing a feature matrix $\mathbf{V}$ for a document D and running the PCOut method on $\mathbf{V}$. The PCOut method gives a *final weighting* for every observation (we show how to calculate this in Equation 2.11) and we use this weighting to rank our observations (which are the segments in a document). The weighting given by PCOut assigns low weights to possible outliers, so in contrast to the other methods we describe, segments are ordered from least to greatest, to be in order of anomalousness.

We would like to point out that it is possible for the PCOut method to assign the same weight to multiple segments of text, because of how it chooses its weights. For instance clear outliers should all get a weight zero. In this case, when we are ranking the segments in a document in order of anomalousness, we break ties (as in sports) by letting all segments that have the same weight be assigned the minimum rank. So, if multiple segments had rank zero they would all receive rank one (the most anomalous segment). This gives PCOut the best possible chance of performing well in our experiments (as you will see from the experimental setup in Section 6.2). Unfortunately, even with this slight advantage, we show that the PCOut method of outlier detection is never among the best performing techniques on any experiments (but it is also not the not the worst and it is extremely fast).

The PCOut method was the only method we experimented which was not implemented by us, as the source code was made available by Filzmoser et al. [2008] who introduced the PCOut measure for outlier detection in statistics. We used this code as it is freely available for the R Statistical Language and can be run on data in very high dimensions.

5.2.4 MeanComp (Distance from the vector complement)

The MeanComp method can be thought of as a "leave one out" approach, where we measure a vector $\vec{v}$'s distance to the mean of the other vectors leaving out $\vec{v}$. Firstly, just as in the other methods discussed so far, we compute all features for each segment independently and create a vector of features for each one. Next, instead of measuring a vector's difference from every other vector, as in ClustDist, we create a vector representation for that vector's complement by averaging together all of the *other* feature vectors (each feature in averaged independently for the p components of the data). Lastly, we compute each vector's difference from this *complement vector* and rank each segment by this distance.

To compute the anomalousness, $MeanComp(\vec{x}, \mathbf{V})$, of a segment of text x, we let $\mathbf{V}'$ be the matrix of feature vectors for all segments in a document except x and $\vec{\mu}'$ be the coordinate-wise mean of these vectors.

$$\mathbf{V}' = \begin{pmatrix} \vec{v}_1 \\ \vec{v}_2 \\ \vdots \\ \vec{v}_{n-1} \end{pmatrix} \quad \text{, where } n \text{ is the number of segments and } \vec{x} \notin \mathbf{V}'$$

$$\begin{aligned} \vec{\mu}' &= \mu_1, \mu_2, \ldots, \mu_p \\ &= \frac{\sum_{i=1}^{n} \mathbf{V}'_{i1}}{n}, \frac{\sum_{i=1}^{n} \mathbf{V}'_{i2}}{n}, \ldots, \frac{\sum_{i=1}^{n} \mathbf{V}'_{ip}}{n} \end{aligned} \tag{5.3}$$

We compute the anomalousness of each segment in the document as the distance from this mean (centroid) of the other segments and rank them by this score.

$$MeanComp(\vec{x}, \mathbf{V}) = d(\vec{x}, \vec{\mu}') \tag{5.4}$$

5.2.5 TxtCompDist (Distance from the textual complement)

This is a novel method for finding atypical textual data by calculating, for each segment of a document, the distance to its complement in the text (the union of the remaining segments) rather than in vector space. We construct a vector of features, $\vec{x}$, for each segment x in a document (by computing all features over the text of x) and another vector of features representing that segment's complement in the text $\vec{c_x}$ (numerical features are computed over all text except that of x to produce a single vector) and measure the distance between them.

$$TxtCompDist(\vec{x}, \mathbf{V}) = d(\vec{x}, \vec{c_x}), \quad \text{where } \vec{c_x} \text{ is the complement of } x \tag{5.5}$$

An advantage of using the complement of a segment's text in a document is that features can be computed over a much larger amount of text than is possible when treating each segment independently (especially when the number of segments, N, is large). This gives a more accurate characterization of the rest of the document than averaging the individual segments as in the vector complement approach above. This procedure makes the exact choice of segment boundaries much less important. In addition it is possible to add features that cannot be accurately computed on small pieces of text. For example, trigrams of parts of speech, adverb preference, and other ranked list features described in (Section 4.6) are sparse enough that on small pieces of text there are very few elements of these lists in common. When using the textual complement, we are more likely to have overlap in the lists so as to produce a score based on the differences in rank. The main disadvantage of this method that it requires more computation than many of the simpler methods. Unlike other methods, where features are computed over a segment exactly one time, using this method it

is necessary to recompute features for each complement in the text.

We experimented with supplementing the measure of anomaly above using our vectors of ranked lists (Section 4.6) for each segment and its complement. So, for every segment in a document we have a total of 4 vectors:

$\vec{x}$ **-** feature vector characterizing the segment

$\vec{c_x}$ **-** feature vector characterizing the complement of the segment

$\mathbf{L_x}$ **-** vector of lists for all rank features for the segment

$\mathbf{L_{c_x}}$ **-** vector of lists for all rank features for the complement of the segment

We next create a vector of Spearman scores, $\vec{\rho}_x$, by computing the Spearman rank correlation coefficient for each pair of lists in vectors $\mathbf{L_x}$ and $\mathbf{L_{c_x}}$. (All numbers in $\vec{\rho}_x$ are actually 1 minus the Spearman rank coefficient so that higher numbers mean less correlation).

$$\vec{\rho}_x = \rho_{11}, \rho_{22}, \ldots, \rho_{nn} \tag{5.6}$$

Where n is the number of lists and ρ_{11} is the Spearman rank between the first list in $\mathbf{L_x}$ and the first list in $\mathbf{L_{c_x}}$. We rank segments by a new measure of anomalousness we call $TxtCompDist_2(\vec{x})$, which we achieve by summing the values in $\vec{\rho}_x$ and the distance between the feature vector and the complement of the text vector. We compute this for all segments in a document and use it to determine which segments are most different from the rest of the document.

$$TxtCompDist_2(\vec{x}, \mathbf{V}) = d(\vec{x}, \vec{c_x}) + \sum \vec{\rho}_x \tag{5.7}$$

$TxtCompDist$ and $TxtCompDist_2$ proved more successful than any other methods in experiments and in fact clearly do better on almost every subtask than the

other methods (see Chapter 6). This is mainly due to the use of the textual complement allowing for more reliable estimates of features. Experiments show that when identifying unusual pieces of text in a document this is clearly the method that should be employed. It is relatively fast to compute and gives better results than more time consuming methods like the Stahel-Donoho Estimator.

The *Rank Features* (Section 4.6) employed by $TxtCompDist_2$, unfortunately had very little impact on the accuracy of this estimate in experiments. This indicates they may not be very useful for these types of experiments, however, their inclusion also did not (on average) lower the results. The rank features could be weighted more heavily by adapting Equation 5.7 to weight the sum of the list features, but we did not experiment with any other weightings and from our results it seems that doing so would have no positive impact. We henceforth refer to this method simply as $TxtCompDist$. For the results presented in the following Chapter, the actual estimator used was the variant of this method with the ranked lists included, i.e. $TxtCompDist_2$, but because these features had negligible impact we will use the original name. (Extended results tables in Section 6.7 show the exact difference in the performance of these approaches.)

5.3 Standardizing Variables

While the majority of the features in our feature vectors measure distributions of phenomena, and thus are percentages (% of adjectives, % of negative words, % of words that occur frequently in the Gigaword, etc.) some features are on a different scale, such as the readability formulae. Vector similarities (and differences) using

these unbalanced features will cause some features have a greater impact than others and thus could skew results toward those features. To overcome this problem and to test the impact of different scales on the performance of the methods we also experimented by performing all tests with and without standardizing the variables.

5.3.1 Zero-One Standardization of variables

We do this by scaling all variables to be between zero and one. Each value in the $n \times p$ data matrix can be standardized to produce a new matrix $\mathbf{S}$, with each value in this matrix given by:

$$s_{ij} = \frac{x_{ij} - \min(x_j)}{\max(x_j) - \min(x_j)} \tag{5.8}$$

where x_{ij} is the value of the i^{th} row and j^{th} column in the document feature matrix and $\min_j$ and $\max_j$ are the maximum and minimum values for the variable j across all observations (segments). This standardization is therefore applied for every variable in the document matrix, $\mathbf{V}$. This has the advantage of making all variables equally weighted in the decision making process, but also has the side effect of suppressing the variance of some truly vast differences or overemphasizing some very slight differences.

5.3.2 Normalizing Variables

We also experimented with normalizing all features by expressing them as their deviations from the means in units of standard deviations or z-scores. We discuss z-scores in the context of outlier detection in Section 2.2 on page 18; here they are used to standardize the scale of the variables. Let, $\mathbf{Z}$, be the data with z-scored

variables, with each value computed as:

$$z_{ij} = \frac{x_{ij} - \bar{x}_j}{\sigma_j}, \text{ for each } i = 1, ..., n \text{ and } j = 1, ..., p \tag{5.9}$$

where x_{ij} is the value of the j^{th} feature variable for the i^{th} observation (i^{th} row and j^{th} column of the $n \times p$ document feature matrix, $\mathbf{V}$), $\bar{x}_j$ is the mean of the j^{th} feature for the sample and σ_i is the standard deviation in the j^{th} feature in the sample.

All experiments have been performed using scaling, z-scores and raw feature scores so a comparison could be made between them. Overall, using z-scores or standardization on the data before outlier detection did not have a positive effect on the results (see Results, Section 6.7), but did improve experiments in the detection of authorship. Some of our methods (SDEDist and PCOut) perform robust standardization as part of their approach to the detection of anomalies anyway and so standardization of the variables beforehand is not necessary.

5.4 Distance Measures

We also experimented with several different measures for computing a vector's distance from another vector. We tested four different distance measures (listed below) and these were used with all methods and on all experiments. The results of these experiments (shown in Chapter 6). The city block distance stood out as the most successful distance measure in experiments using the textual complement (TxtCompDist), which was the best performing method of anomaly detection. The formulas we tested, for the distance between two vectors $\vec{x}$ and $\vec{y}$ of length p, are as follows.

Cosine *Dissimilarity* Measure

$$d(\vec{x}, \vec{y}) = 1 - \frac{\vec{x} \cdot \vec{y}}{||\vec{x}|| \; ||\vec{y}||} = 1 - \frac{\sum_{i=1}^{p} x_i y_i}{\sqrt{\sum_{i=1}^{p} x_i^2} \sqrt{\sum_{i=1}^{p} y_i^2}} \tag{5.10}$$

Euclidean Distance (referred to as L_2 distance)

$$d(\vec{x}, \vec{y}) = \sum_{i=1}^{p} (x_i - y_i)^2 \tag{5.11}$$

City Block Distance (also called Manhattan distance or L_1 distance)

$$d(\vec{x}, \vec{y}) = \sum_{i=1}^{p} |x_i - y_i| \tag{5.12}$$

Pearson *Dissimilarity* Coefficient

$$d(\vec{x}, \vec{y}) = 1 - \frac{1}{p} \sum_{i=1}^{p} \left(\frac{x_i - \bar{x}}{\sigma_x} \right) \left(\frac{y_i - \bar{y}}{\sigma_y} \right) \tag{5.13}$$

Here we use $\bar{x}$ to indicate the mean of a vector $\vec{x}$ and σ_x to indicate the standard deviation of $\vec{x}$.

$$\bar{x} = \frac{1}{p} \sum_{i=1}^{p} x_i \qquad \sigma = \sqrt{\frac{1}{p-1} \sum_{i=1}^{p} (x_i - \bar{x})^2}$$

5.5 Unsupervised Anomaly Detection System

We have built a system for finding anomalies in text called the *UNsupervised Anomaly Detector* (or **UNSAD**). This allowed us to test different combinations of methods, features, and segment sizes easily, and also means that all of our results are

repeatable. Another advantage is that it is now possible to test our methods on new tasks and domains very quickly.

The UNSAD system (Figure 5.3) takes documents as input and outputs a ranked list of the segments according to how anomalous they are. The system handles all tasks involved in finding textual anomalies including: segmentation (either by paragraph or as a fixed number of words), document preprocessing (cleaning,sentence splitting), running RASP (see section 4.5), computation of features, standardization of features, and running the different anomaly detection techniques with various distance measures.

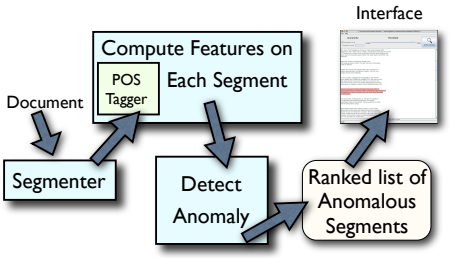


Figure 5.3: UNSAD system model

All of the major text processing work done by the system is implemented in Perl [Wall et al., 2000] and the user interface for the system was implemented using Java [Arnold and Gosling, 1998]. The Stahel-Donoho Distance method using our choice of direction was implemented by us, using the R statistical language [R Development Core Team, 2008], while the PCOut method was tested in the R language using the code provided by its authors as an R package *mvoutlier*[15].

[15]The package *mvoutlier* can be found at `http://cran.r-project.org/web/packages/mvoutlier/index.html`

The user interface (shown in Figure 5.4) for the UNSAD System allows a user to run the different unsupervised anomaly detection methods, turn on feature standardization, or change the segment size, and experiment with thresholds and then displays the results visually by highlighting segments of the document.

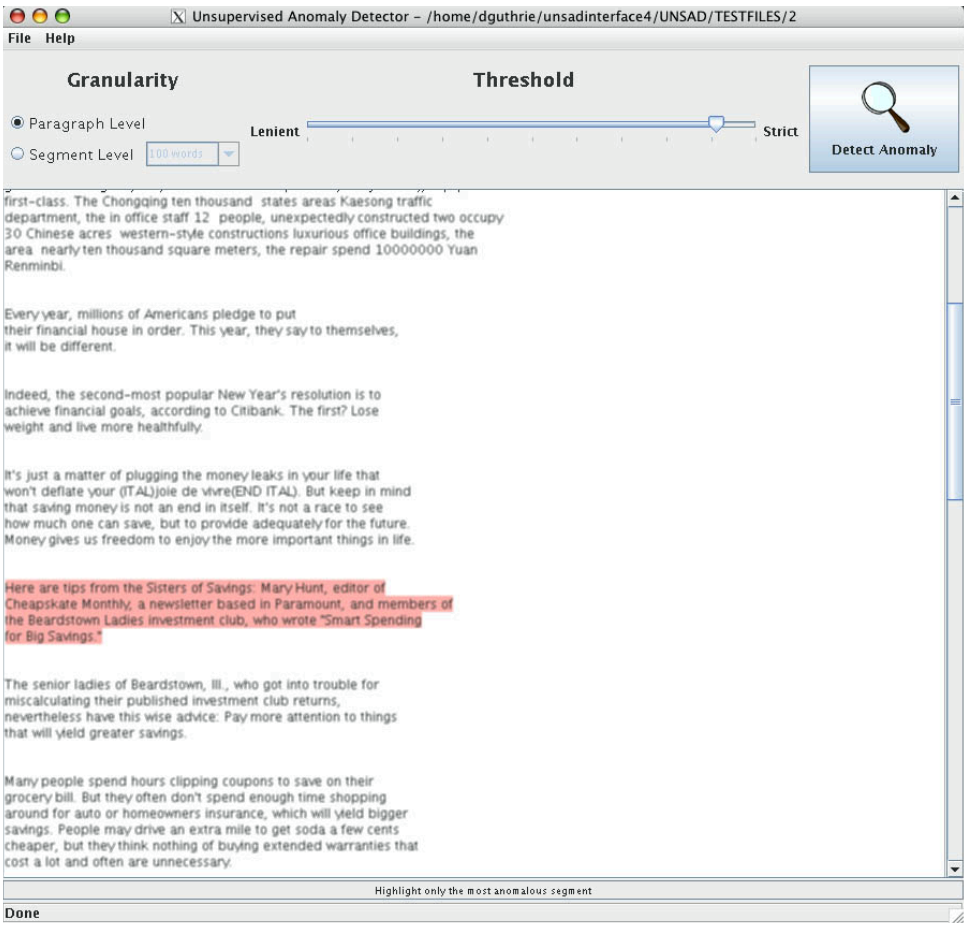


Figure 5.4: UNSAD Segment Detection Interface: the main window allows users to view a document and then press the "Detect Anomaly" button to run an anomaly detection method (specified in the preferences) on the text. Segments that the system identifies as anomalous are highlighted in red according to the threshold set.

5.6 Summary

We describe five methods for the identification of anomalies in text:

- **ClustDist** is based on average linkage clustering and measures the distance from one observation to all other observations and averages the result.
- **SDEDist** makes use of the Stahel-Donoho Estimator to find projections of the data which maximize an observations distance from the center of the observations. We give a description of this procedure and also of the method we use for choosing the projection directions.
- **PCOut** method of outlier detection measures an observations distance from the center of the observations in principal component space using kurtosis to weight these components.
- **MeanCompDist** method measures an observations distance to the mean of all other observations excluding itself.
- **TxtCompDist** method measures distance form the textual complement. It is a novel method we created specifically for anomaly detection where we measure an observations distance to a new observation obtained by recalculating all features over the entire text the observation was drawn form, excluding that observation.

We also describe two different methods for normalizing the variables as well as four different distance measures that we test along with the five methods in the next chapter.

Chapter 6

Experiments on Detection of Anomalies

6.1 Overview

In this chapter we apply the methods introduced in the previous chapter to thousands of different documents automatically constructed to contain an anomaly. We examine each methods ability to detect these anomalies and compare the results. The first section of the chapter describes the experimental setup and the assumptions we made and the following sections present results for four different anomaly detection scenarios, namely:

- Authorship anomalies
- Factual writing vs opinion writing anomalies
- Subversive article anomalies
- Machine translation anomalies

6.2 Experimental Setup

All of the unsupervised anomaly detection techniques we have developed are applicable to both small pieces of text (like a short paragraph) and to large pieces of text like documents or books. We chose to experiment on the task of finding *sections* in a document that are anomalous, but there is nothing inherent in the methods to suggest that they must be used detect anomalies within a document; they could equally well be used to detect whole documents that are anomalous with respect to a collection. The task is always about finding text that does not belong or is unusual with respect to its surroundings and it is just a matter of scope as to what those surroundings are. Many thousands of different experiments were run to detect anomalous segments in documents. We describe these with a view to investigating what types of anomaly are easiest to detect, the effect text size has on anomaly detection, the impact of standardization, and the anomaly detection technique that performs best.

Our experiments in the detection of anomalies focus primarily on detecting when the author, genre, writing style, or topic is anomalous. In these experiments we take a document that contains an anomaly and feed it to our anomaly detection program. This program returns a list of all segments ranked by how anomalous they are with respect to the whole document. If the program has performed well, then the truly anomalous segment should be at the top of the list (or very close to the top). Our assumption has been that a human wishing to detect anomaly would be pleased if they could find the truly anomalous segment in the top 3 or 5 segments marked most likely to be anomalous, rather than having to scan the whole document or collection. This may not be the case in situations where there is no reason to believe that anomalies

exist. In this case it would be necessary to label segments definitively as either outliers or non outliers and we investigate doing exactly this in Chapter 7. In this chapter we explain the experimental setup and present the results for the percentage of the time an anomalous segment can be identified in the top 1, 3, 5, 10 and 20 segments using our different methods for anomaly detection.

Test documents are artificially created by taking a "document" made up of random segments from a single source and inserting a randomly chosen segment from a different source. In this scenario, the source which makes up the majority of a document is the *normal* population while the single inserted segment is *anomalous* with respect to that population. Our anomaly detection procedures are then run over this artificially created test document with the goal of identifying the inserted segment from the different source as an anomaly. We created thousands of documents in this manner from different sources and using different sized segments, but always inserted a segment from one source into a collection of segments from a different source as illustrated in Figure 6.1.

In each of the experiments below, all test documents contain exactly one anomalous segment and exactly 50 "normal" segments. Whilst in reality it may be true that multiple segments are anomalous within a document; for the sake of simplicity of evaluation, we insert only one anomalous segment per document at a time. All methods, however, have been developed with the intention of detecting multiple anomalies in documents (if they exist) and there is nothing inherent in our procedures that precludes or hinders detecting anomalies if more than one is present in a document. In fact, in most genres of writing, the style of the writing can change greatly

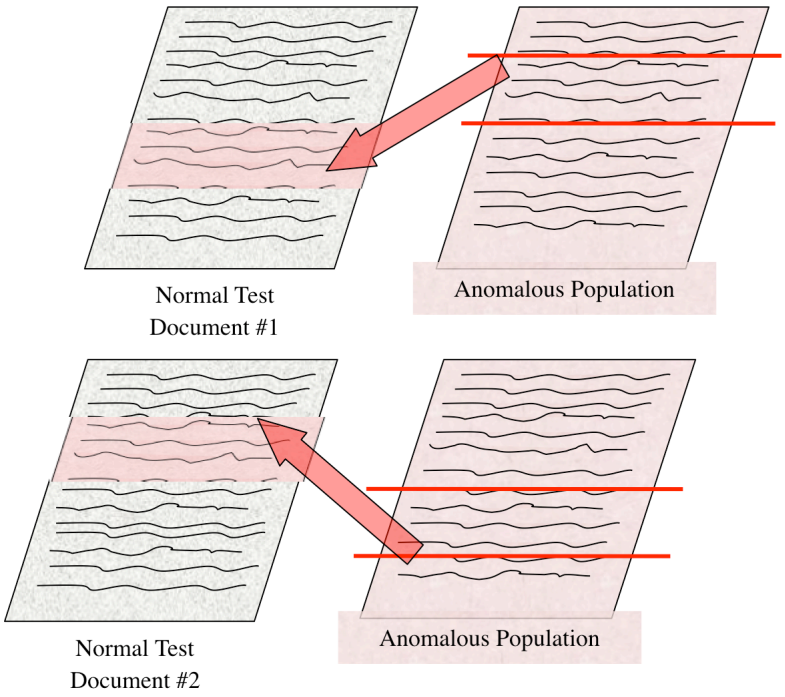


Figure 6.1: Random test documents are generated by selecting random segments from a *normal* document and inserting segments of *anomalous* text

from one paragraph (segment) to the next, so all methods for detecting anomalies in text must naturally cope with documents and collections that have fluctuation in the writing style. Documents, by virtue of the fact that not every paragraph is the same, are never perfectly homogenous and so there is a degree to which every piece of a document can be scored as anomalous with respect to the entire document.

There is an unproven assumption that what is artificially inserted into a document or collection will be the most anomalous thing within that document or collection. While this might not be true in the general case, every attempt was made to ensure the cohesiveness of the collections used in this research to minimize the chance of finding genuine, unplanned anomalies.

The work presented here looks only at fixed-length segments with pre-determined boundaries, while a real application of such a technique might be required to function with vast differences between the sizes of segments. Once again, there is nothing implicit in the method assuming fixed-length sizes, and the choice to fix certain parameters of the experiments is to better illustrate the effect of segment length on the performance of the different methods. One could simply use paragraph breaks as natural segment boundaries, or employ more sophisticated segmentation techniques if desired. For example, if these techniques were being used to detect anomalous documents in a collection of documents then no segmentation or choosing of segment sizes would be necessary as each document could naturally be thought of as simply a "large segment" and anomaly detection methods could be applied to them. If one was interested in identifying anomalies within a document, but the size of these anomalies was unknown then it might be necessary to extend the techniques presented here by iteratively looking at different segment sizes while preforming anomaly detection methods in order to identify segments and their sizes that are most anomalous for that document. Alternatively, one could apply a more computationally intensive procedure like calculating the degree of anomalousness for a sliding window of a fixed segment size that moved through the entire document. While these extensions are certainly possible, they are not the focus of this work and we assume fixed segments throughout. We choose to focus on the more general problems of what types of anomalies can be detected in text, by what methods, and the influence the size of these anomalies has on our ability to detect them.

We introduce a baseline for the following experiments that is the probability of

selecting the truly anomalous segment by chance. For instance, the probability of choosing the single anomalous segment in a document that is 51 segments long, completely by chance, when picking 3 segments is 1/51 + 1/50 + 1/49 or 6%.

We conducted thousands of experiments comparing each method using the different ways of standardization of variables across different segment sizes, genres, authors and various other testbeds. This produced an extremely large number of results (and figures comparing these results), many of which are shown in Section 6.7. For the sake of simplicity we have limited most of the tables and figures in this chapter to the best performing method. The following sections show the results using the most successful method (on average) TxtCompDist (using the textual complement) to detect anomaly in many different scenarios.

6.3 Authorship Tests

For these sets of experiments we examine whether it is possible to distinguish anomaly of authorship at the segment level. We test this by taking a document written by one author and inserting in it a segment written by a different author. We then see if this segment can be detected using our unsupervised anomaly techniques. We create our new experimental data from a collection consisting of 50 thousand words of text written by each of 8 Victorian authors:

- Bronte
- Carroll
- Doyle
- Eliot

- James
- Kipling
- Tennyson
- Wells

Test sets are created for each pair of authors by inserting a segment from one author into a document written by the other author. This creates 56 sets of experiments (one for each author inserted into every other author) and for each experiment we perform insertions one at a time into the other document (see Figure 6.1). For example we insert segments, one at a time from Bronte into Carroll and anomaly detection is performed. Likewise we insert segments one at a time from Carroll into Bronte and perform anomaly detection. Our experiment is always to test if we can detect this inserted segment.

For each of the 56 combinations of authors we insert 30 randomly chosen segments from one into the other, one at a time. We performed 56 pairs * 30 insertions each = 1,680 sets of insertion experiments. For each of these 1,680 insertion experiments we also varied the segment size to test its effect on anomaly detection. We used segment sizes of:

- **100 words**
- **500 words**
- **1000 words**

We then count what percentage of the time truly anomalous paragraphs fall within the top 1, top 3, top 5, top 10, and top 20 segments labelled by the program as anomalous. The results shown here report the average accuracy for each segment size

(over all authors and insertions). Tables and figures show the average percentage of trials in which the anomalous segment is detected in the top n documents for the TxtCompDist method (the accuracy of other methods is shown in Section 6.7).

Top n Segments	Percentage of the time found	Percentage of the time found (standardized features)	Chance
Segment size: 100 words			
1	13.54%	16.25%	1.96%
3	25.54%	31.25%	6.00%
5	32.61%	40.46%	10.21%
10	48.57%	52.04%	21.59%
20	63.79%	67.82%	49.16%
Segment size: 500 words			
1	29.01%	37.79%	1.96%
3	46.04%	50.72%	6.00%
5	49.53%	60.59%	10.21%
10	61.67%	72.40%	21.59%
20	74.30%	83.88%	49.16%
Segment size: 1000 words			
1	44.80%	48.02%	1.96%
3	54.98%	66.60%	6.00%
5	60.08%	74.07%	10.21%
10	76.26%	85.79%	21.59%
20	96.19%	97.88%	49.16%

Table 6.1: Summary of Author Results: Results presented in this table are the percentage of the time a truly anomalous segment is found in the top n segments returned, for segment sizes of 100, 500, and 1,000 words.

The average percent of time we can detect anomalous segments in the top n segments varies according to the segment size, and as expected, the average accuracy increases as the segment size increases. For 1000 word segments, the anomalous segment was found in the top 20 ranked segments about 97% of the time (85% in the top ten, 74% of the time in the top 5, 66% of the time in the top three segments, and

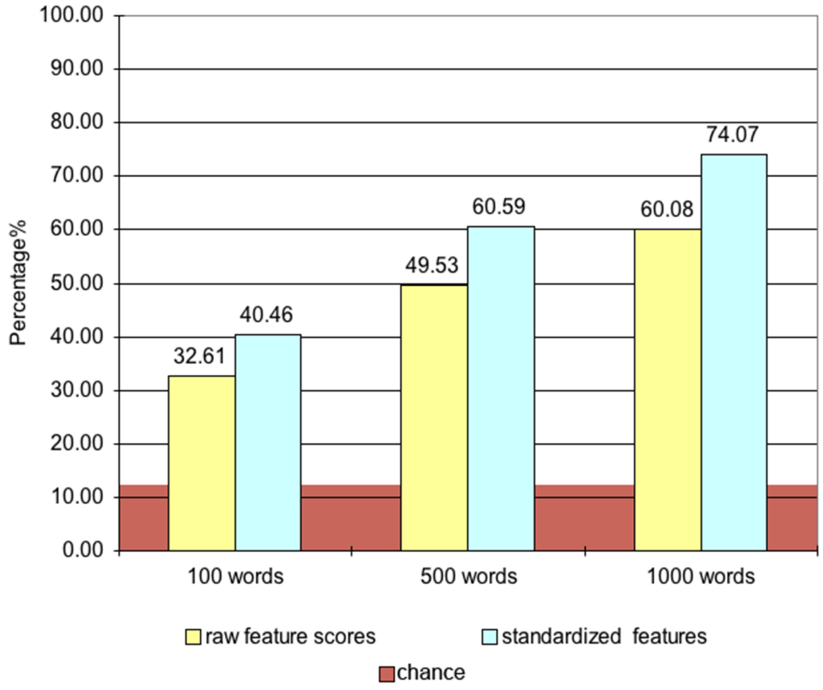


Figure 6.2: Average results for a top 5 ranking for the anomalous section (authorship).

48% returned first). For 500 word segments, average accuracy ranged from 83% down to 38% and for 100 word segments it ranged from 68% down to 16%. Even though the percentage of the time the anomalous segment is returned first is fairly low (38% for 500 word segments), this is still far better that choosing the anomalous segment by chance (1.96%). We believe that this task is particularly hard for our anomaly detection methods, as we make use of no topical information (content words) which could have greatly aided in the spotting the inserted segment. Our characterization of texts (see Chapter 4) is comprised almost entirely of stylistic features which should not differ greatly based on the topic.

6.4 Fact versus Opinion

For these sets of experiments we are testing whether opinion can be detected in a factual story. The test documents used come from similar sources (newspapers and newswire), but the style of the writing should be different as some are factual news stories and some are editorials. The opinion text is made up of editorials from 4 newspapers making up a total of 28,200 words:

- Guardian
- New Statesman
- New York Times
- Telegraph

Our factual text is a randomly chosen from the Gigaword and consists of four different 78,000 word segments one each from one of the four news wire services (see see Appendix B or a full description and examples of the corpora used):

- **Agence France Press English Service**
- **Associated Press Worldstream English Service**
- **The New York Times Newswire Service**
- **The Xinhua News Agency English Service**

Each opinion text segment is inserted into each news wire service one at a time for at least 25 insertions on each newswire. Tests are performed like the authorship tests using three different segment sizes. Results in this set of experiments were generally better than the results in the Authorship experiments. The average accuracy for 1,000 word segments in the top 10 ranking was 99% (85% in the top 3 segments.) Small segment sizes of 100 words also yielded good results and the anomaly was identified in the top 20, 78% of the time (although only 46% of the time in the top 3.)

Top n Segments	Percentage of the time found	Percentage of the time found (standardized features)	Chance
Segment size: 100 words			
1	26.50%	17.50%	1.96%
3	46.00%	36.00%	6.00%
5	49.50%	46.00%	10.21%
10	62.00%	64.00%	21.59%
20	78.50%	76.00%	49.16%
Segment size: 500 words			
1	13.50%	22.00%	1.96%
3	50.50%	59.00%	6.00%
5	80.50%	73.00%	10.21%
10	90.50%	82.50%	21.59%
20	99.00%	96.00%	49.16%
Segment size: 1000 words			
1	34.78%	53.26%	1.96%
3	85.87%	73.91%	6.00%
5	95.65%	80.43%	10.21%
10	98.91%	94.57%	21.59%
20	98.91%	98.91%	49.16%

Table 6.2: Summary of Factual Anomaly Detection: Results presented in this table are the percentage of the time a segment taken from opinion texts is found in the top n segments returned in a collection of factual articles, for segment sizes of 100, 500, and 1,000 words.

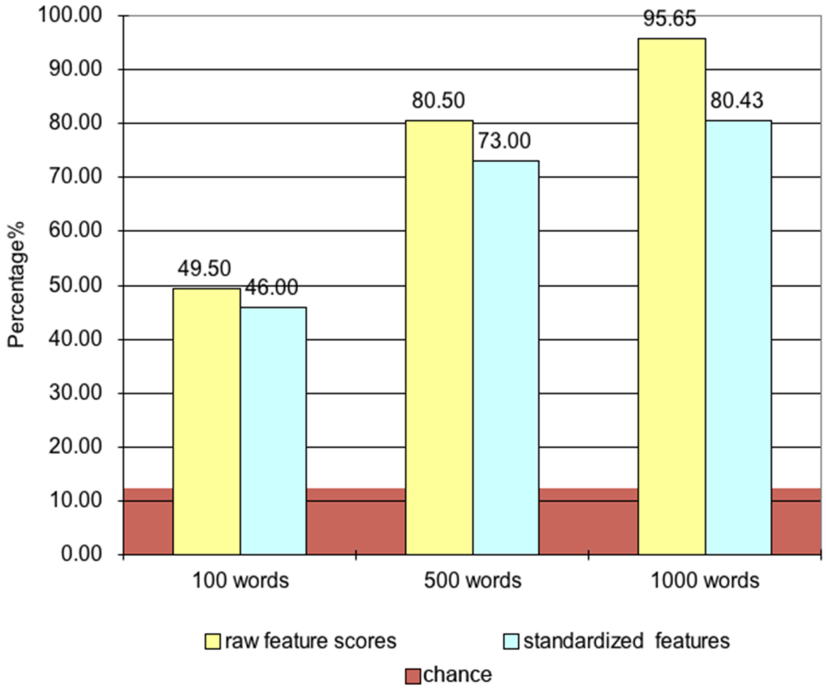


Figure 6.3: Average results for a top 5 ranking for the anomalous section (fact versus opinion).

6.5 Newswire versus the Anarchist Cookbook

In this set of experiments we test whether segments from the Anarchist Cookbook (see Appendix B) can be detected in a collection of news wire. This experiment was designed to test if we could identify very different genres using our anomaly detection techniques. The Anarchist Cookbook contains recipes for the manufacture of explosives, instructions for building telecommunications phreaking devices and other illegal activities. This writing is very procedural, as it is in the form of instructions and recipes, and also informal (e.g. "When the fuse contacts the balloon, watch out!!!"). This is very different from newswire text which is more formal, but the writing is less structured. We make use of 30,000 words from the Anarchist Cookbook as the anomalous text. Our newswire text is randomly chosen from segments of the Gigaword corpus made up of text from the four news wire services. Each Anarchist Cookbook text segment is inserted into each news wire service for at least 30 insertions on each random newswire document (of which we created 4 for each segment size). All tests are run using the three different segment sizes.

As can be see in Table 6.3, our anomaly detection technique performs much better (without standardizing the features beforehand) than on the previous experiments. Anomalies can be detected as the most anomalous segment 70% of the time in 500 word segments. These results indicate that anomalies in text (at least on this type of task) are well distinguished from the rest of the text using our methods. These methods are clearly identifying the anomalous segment as unusual when compared to other segments in the document as the segment is found in the top 10 segments 100% of the time, for 500 word segments (this results an average of $4 \times 30 = 120$ different

experiments using this segment size).

Top n Segments	Percentage of the time found	Percentage of the time found (standardized features)	Chance
Segment size: 100 words			
1	38.00%	34.00%	1.96%
3	68.00%	38.00%	6.00%
5	74.00%	46.00%	10.21%
10	88.00%	58.00%	21.59%
20	98.00%	82.00%	49.16%
Segment size: 500 words			
1	70.00%	24.00%	1.96%
3	90.00%	58.00%	6.00%
5	92.00%	76.00%	10.21%
10	100.00%	78.00%	21.59%
20	100.00%	100.00%	49.16%
Segment size: 1000 words			
1	88.78%	36.26%	1.96%
3	100.00%	58.00%	6.00%
5	100.00%	78.00%	10.21%
10	100.00%	94.00%	21.59%
20	100.00%	98.00%	49.16%

Table 6.3: Summary of Anarchist Cookbook Anomaly Detection: Results presented in this table are the percentage of the time a segment taken from the Anarchist Cookbook is found in the top n segments returned in a collection of newswire articles, for segment sizes of 100, 500, and 1,000 words.

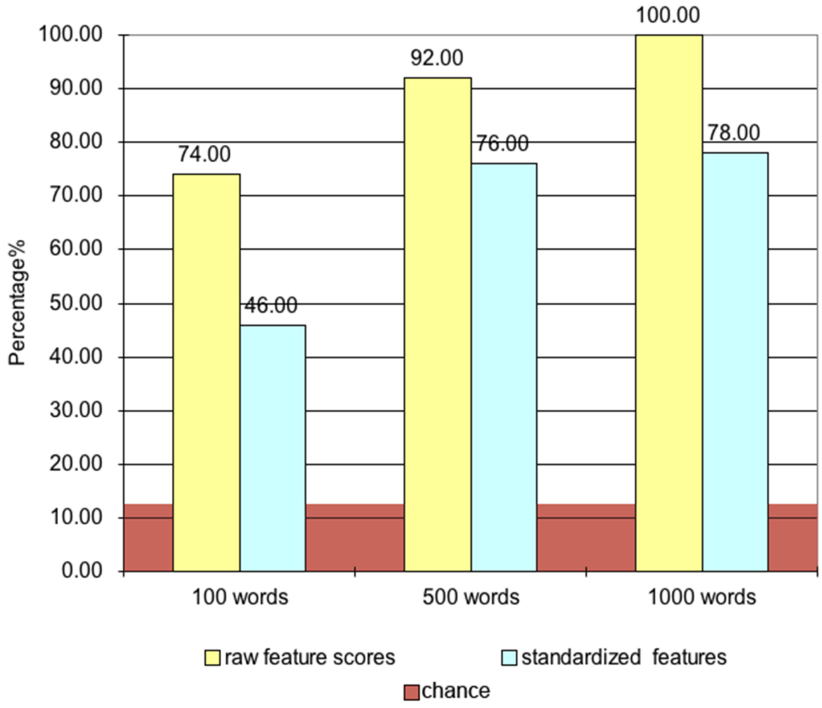


Figure 6.4: Average results for a top 5 ranking for the anomalous section (newswire versus Anarchist Cookbook).

6.6 Newswire versus Machine Translations

In this set of experiments we test whether Chinese newspaper segments that have been translated into English can be detected in a collection of English newswire. The translations of Chinese news articles are a very similar genre to the English newswire, but the translations are not perfect and so the language use is odd (see AppendixB for an example). Our methods attempt not to utilize any features about the topic of segments that would aid in anomaly detection as our features are mostly style-based. These experiments go even further to guarantee that we are clearly not detecting topical differences as our random samples of English newswire come from a huge corpus spanning many years and from different news sources, including the English news wire from the Xinhua News Agency (a Chinese news service) and so contain extremely diverse topics. Likewise, the translated Chinese news stories were chosen, by a native Chinese speaker, to be on a range of different topics. These results should therefore indicate that our anomaly detection techniques are detecting *stylistic* differences rather than more topical, author-based, or genre-based differences.

We use a corpus of 35,000 words of Chinese newspaper text that was translated into English using Google's Chinese to English translation engine. "Normal" text is randomly chosen (four times for each segment size) from the Gigaword corpus and consists of a 78,000 word segment made up of text from the four news wire services. As in the other experiments, the Chinese translations are inserted one at a time into the newswire data and anomaly detection methods were run (with three different segment sizes).

The results for this task are very good. These are the best results for anomaly

detection out of the different scenarios investigated. The results in Table 6.4 show that for 500 word segments of text we detect the anomalous segment first 83.7% of the time and that for 1000 word segments we will detect the anomaly first 96% of the time. This indicates that this task is extremely well suited to our anomaly detection technique and thus our characterization of text (features) effectively captures the differences between these types of text. While the results here are for a single anomaly detection method (TxtCompDist), in fact the accuracy of all of the best performing methods is higher on this task as well. These results are shown in Section 6.7

Top n Segments	Percentage of the time found	Percentage of the time found (standardized features)	Chance
Segment size: 100 words			
1	54.00%	36.00%	1.96%
3	60.00%	54.00%	6.00%
5	68.00%	58.00%	10.21%
10	74.00%	60.00%	21.59%
20	80.00%	76.00%	49.16%
Segment size: 500 words			
1	83.67%	59.18%	1.96%
3	87.76%	69.39%	6.00%
5	89.80%	87.76%	10.21%
10	93.88%	93.88%	21.59%
20	100.00%	100.00%	49.16%
Segment size: 1000 words			
1	96.00%	92.00%	1.96%
3	100.00%	96.00%	6.00%
5	100.00%	96.00%	10.21%
10	100.00%	100.00%	21.59%
20	100.00%	100.00%	49.16%

Table 6.4: Summary of Machine Translation Anomaly Detection: Results presented in this table are the percentage of the time a segment taken from a machine translated document is found in the top n segments returned in a collection of newswire articles, for segment sizes of 100, 500, and 1,000 words.

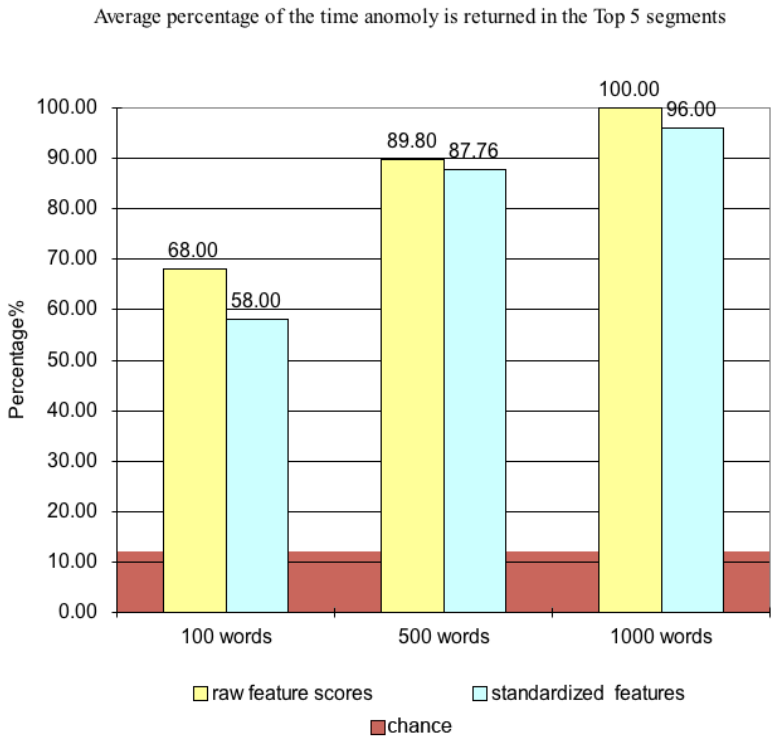


Figure 6.5: Average results for a top 5 ranking for the anomalous section (newswire versus Chinese translations).

6.7 Extended Results

In this section we show a greater selection of the results for all methods over these experiments.

	Text Segment Size: 1000 words																			
	Chinese Translation					Fact Opinion					Anarchist Cookbook					Authorship				
	top 1	top 3	top 5	top 10	top20	top 1	top 3	top 5	top 10	top20	top 1	top 3	top 5	top 10	top20	top 1	top 3	top 5	top 10	top20
ClustDist(cityblock)	32.0	36.0	44.0	44.0	64.0	21.7	30.4	39.1	69.6	69.6	16.0	28.0	34.0	51.3	65.3	12.3	16.3	20.9	35.8	71.3
ClustDist(cosine)	20.0	44.0	52.0	52.0	64.0	13.0	34.8	43.5	43.5	52.2	16.0	28.0	34.0	51.3	65.3	10.8	14.8	20.2	34.8	71.3
ClustDist(cosine)(Zscore)	20.0	44.0	52.0	52.0	64.0	13.0	34.8	43.5	43.5	52.2	6.7	10.7	12.7	14.7	19.3	10.6	14.6	20.4	35.0	71.0
ClustDist(euclid)	28.0	52.0	52.0	52.0	72.0	21.7	43.5	43.5	43.5	56.5	20.0	32.0	38.7	47.3	66.7	11.4	15.2	20.5	33.9	71.1
ClustDist(Pearson)	24.0	52.0	52.0	52.0	0.0	17.4	39.1	43.5	43.5	56.5	16.0	28.0	34.0	51.3	65.3	10.9	15.2	21.1	34.3	71.4
ClustDist(Pearson)(Zscore)	36.0	44.0	52.0	60.0	72.0	26.1	30.4	39.1	47.8	56.5	10.0	17.3	24.0	40.0	72.0	9.5	14.6	20.6	33.7	72.8
MeanComp(cityblock)	0.0	0.0	0.0	0.0	16.0	0.0	4.3	8.7	26.1	39.1	20.0	38.0	56.7	74.7	86.7	5.5	12.1	18.7	37.0	72.5
MeanComp(cityblock)(Zscore)	24.0	52.0	52.0	52.0	80.0	17.4	39.1	43.5	43.5	56.5	18.0	29.3	36.7	50.0	61.3	10.9	15.2	21.0	34.2	71.4
MeanComp(cosine)	76.0	96.0	96.0	96.0	100.0	43.5	65.2	65.2	69.6	78.3	15.3	26.7	34.0	50.7	63.3	27.5	31.6	36.1	43.5	59.4
MeanComp(cosine)(Zscore)	0.0	0.0	0.0	0.0	16.0	0.0	4.3	8.7	30.4	43.5	16.0	26.7	34.0	51.3	64.7	5.5	12.0	18.2	38.0	71.5
MeanComp(euclid)	36.0	44.0	56.0	60.0	76.0	17.4	30.4	39.1	47.8	65.2	22.0	49.3	57.3	66.7	81.3	10.2	14.2	21.1	34.7	71.7
MeanComp(Pearson)	24.0	52.0	52.0	52.0	80.0	17.4	39.1	43.5	43.5	56.5	10.0	15.3	24.7	36.0	58.7	10.9	15.2	21.1	34.3	71.4
Meth(SumDiff)	44.0	60.0	72.0	80.0	96.0	21.7	30.4	47.8	56.5	91.3	20.0	27.3	36.0	44.7	56.7	11.4	16.3	22.7	35.6	67.7
Meth(SumDiff)(Zscore)	28.0	52.0	52.0	52.0	72.0	21.7	43.5	43.5	43.5	56.5	18.0	29.3	36.7	50.0	61.3	11.4	15.2	20.5	33.9	71.1
Stahel-Donoho	72.0	96.0	96.0	96.0	100.0	39.1	78.3	87.0	91.3	91.3	90.7	99.3	100.0	100.0	100.0	41.7	54.5	62.4	73.1	85.5
PCOut weights	41.3	41.3	52.2	71.7	84.8	15.9	18.2	31.8	63.6	79.5	70.4	73.7	74.6	83.6	90.6	47.4	55.3	61.2	70.7	88.8
TxtCompDist(city)	96.0	100.0	100.0	100.0	100.0	34.8	85.9	95.7	98.9	98.9	88.0	100.0	100.0	100.0	100.0	44.8	55.0	60.1	76.3	96.2
TxtCompDist(city)(scale01)	92.0	96.0	96.0	100.0	100.0	53.3	73.9	80.4	94.6	98.9	36.0	58.0	78.0	94.0	98.0	48.0	66.6	74.1	85.8	97.9
TxtCompDist(cosine)	20.0	84.0	88.0	92.0	100.0	0.0	8.7	13.0	47.8	78.3	48.0	70.7	84.7	97.3	100.0	20.8	44.0	53.5	73.6	93.0
TxtCompDist(pearson)	20.0	84.0	88.0	92.0	100.0	0.0	8.7	13.0	47.8	82.6	48.7	71.3	84.7	97.3	100.0	20.7	44.4	53.8	73.6	93.0
TxtCompDist(euclid)	88.0	100.0	100.0	100.0	100.0	47.8	87.0	87.0	95.7	100.0	67.3	99.3	100.0	100.0	100.0	39.2	48.4	55.2	64.5	92.1
TxtCompDist-nolist(cosine)	96.0	100.0	100.0	100.0	100.0	60.9	87.0	91.3	95.7	100.0	72.0	99.3	100.0	100.0	100.0	40.6	49.5	55.0	65.8	91.6
TxtCompDist-nolist(pearson)	96.0	100.0	100.0	100.0	100.0	60.9	87.0	91.3	95.7	100.0	72.0	99.3	100.0	100.0	100.0	40.5	49.4	55.2	65.6	91.3
TxtCompDist-nolist(euclid)	88.0	100.0	100.0	100.0	100.0	47.8	87.0	87.0	95.7	100.0	67.3	99.3	100.0	100.0	100.0	39.2	48.4	55.2	64.5	92.1
TxtCompDist-nolist(city)	96.0	100.0	100.0	100.0	100.0	47.8	73.9	91.3	95.7	100.0	94.7	100.0	100.0	100.0	100.0	42.4	53.7	58.3	73.9	95.7

Table 6.5: 1000 word segments results. This table gives the percentage of trials in which the anomalous segments were identified in the top n segments for some of our methods across the different document collections. In the leftmost column is the method name followed by the distance measure and whether normalization was used. This chart shows that TxtCompDist (the textual complement method) performs best overall and also best in almost every subtask. Particularly good results come from using city block distance with the TxtCompDist method. In general the ClustDist and MeanComp methods perform significantly worse than the other methods regardless of the distance measure.

	Text Segment Size: 500 words																			
	Chinese Translation					Fact Opinion					Anarchist Cookbook					Authorship				
	top 1	top 3	top 5	top 10	top20	top 1	top 3	top 5	top 10	top20	top 1	top 3	top 5	top 10	top20	top 1	top 3	top 5	top 10	top20
ClustDist(cityblock)	20.4	20.4	34.7	46.9	75.5	10.0	10.0	10.0	22.0	58.0	4.0	18.0	28.7	46.7	67.3	8.6	15.3	19.2	32.4	62.6
ClustDist(cosine)	4.1	14.3	18.4	20.4	36.7	4.0	10.0	14.0	26.0	34.0	4.0	24.7	28.7	46.7	67.3	7.0	12.6	18.6	32.6	64.4
ClustDist(cosine)(Zscore)	4.1	14.3	18.4	20.4	34.7	4.0	10.0	14.0	26.0	34.0	6.7	10.0	11.3	16.0	19.3	7.0	12.5	18.6	32.6	64.3
ClustDist(euclid)	14.3	20.4	20.4	28.6	38.8	10.0	14.0	22.0	30.0	36.0	4.0	16.7	25.3	39.3	63.3	8.5	14.2	19.4	32.2	63.4
ClustDist(Pearson)	10.2	18.4	20.4	28.6	40.8	10.0	10.0	16.0	28.0	38.0	4.0	24.7	28.7	46.0	68.0	8.0	13.8	19.3	32.9	63.3
ClustDist(Pearson)(Zscore)	12.2	18.4	18.4	24.5	38.8	10.0	10.0	12.0	28.0	40.0	3.3	14.0	18.7	40.0	58.7	8.2	13.7	18.3	32.6	63.3
MeanComp(cityblock)	4.1	14.3	34.7	46.9	69.4	2.0	4.0	8.0	24.0	54.0	5.3	20.0	36.0	55.3	76.7	3.6	12.6	18.0	32.0	63.7
MeanComp(cityblock)(Zscore)	10.2	18.4	20.4	28.6	40.8	10.0	10.0	16.0	28.0	38.0	5.3	25.3	32.7	42.7	64.7	7.9	13.7	19.2	32.9	63.3
MeanComp(cosine)	65.3	83.7	91.8	95.9	100.0	38.0	58.0	74.0	80.0	88.0	4.0	20.0	28.7	45.3	63.3	21.7	31.5	35.4	41.3	54.8
MeanComp(cosine)(Zscore)	4.1	16.3	36.7	49.0	61.2	2.0	4.0	16.0	28.0	52.0	4.0	20.0	28.7	46.7	64.0	3.5	12.6	18.6	32.4	63.2
MeanComp(euclid)	10.2	18.4	18.4	26.5	40.8	10.0	10.0	10.0	24.0	34.0	6.0	19.3	34.7	54.7	75.3	7.1	13.6	18.8	32.4	62.9
MeanComp(Pearson)	10.2	18.4	20.4	28.6	40.8	10.0	10.0	16.0	28.0	38.0	4.0	12.7	18.0	37.3	56.7	8.0	13.8	19.3	32.9	63.3
Meth(SumDiff)	16.3	22.4	24.5	36.7	65.3	10.0	12.0	14.0	32.0	48.0	4.0	18.0	26.7	40.0	59.3	8.4	15.2	20.5	34.1	61.0
Meth(SumDiff)(Zscore)	14.3	20.4	20.4	28.6	38.8	10.0	14.0	22.0	30.0	36.0	5.3	25.3	32.7	42.7	64.7	8.5	14.2	19.4	32.2	63.4
Stahel-Donoho	49.0	83.7	89.8	98.0	100.0	26.0	72.0	78.0	86.0	92.0	68.0	98.0	100.0	100.0	100.0	24.0	41.2	49.4	59.7	76.3
PCOut weights	44.3	54.3	62.9	77.1	91.4	8.5	19.7	38.0	60.6	83.1	71.8	72.3	73.2	79.3	92.5	36.0	59.9	70.7	80.9	89.4
TxtCompDist(city)	83.7	87.8	89.8	93.9	100.0	13.5	50.5	80.5	90.5	99.0	70.0	90.0	92.0	100.0	100.0	29.0	46.0	49.5	61.7	74.3
TxtCompDist(city)(scale01)	59.2	69.4	87.8	93.9	100.0	22.0	59.0	73.0	82.5	96.0	24.0	58.0	76.0	78.0	100.0	37.8	50.7	60.6	72.4	83.9
TxtCompDist(cosine)	51.0	67.3	71.4	91.8	93.9	0.0	8.0	10.0	34.0	40.0	52.7	71.3	89.3	92.7	100.0	19.7	33.5	41.1	61.1	73.2
TxtCompDist(pearson)	53.1	67.3	71.4	93.9	95.9	0.0	8.0	10.0	34.0	40.0	52.7	71.3	90.0	92.7	100.0	19.7	33.7	43.0	61.7	73.3
TxtCompDist(euclid)	73.5	77.6	89.8	91.8	98.0	44.0	50.0	66.0	80.0	94.0	20.0	96.0	99.3	100.0	100.0	23.9	38.9	45.0	54.4	68.5
TxtCompDist-nolist(cosine)	73.5	73.5	89.8	93.9	100.0	44.0	44.0	66.0	84.0	94.0	16.7	96.0	100.0	100.0	100.0	28.0	40.7	46.2	54.9	68.7
TxtCompDist-nolist(pearson)	73.5	77.6	89.8	93.9	100.0	44.0	44.0	66.0	84.0	94.0	17.3	96.0	100.0	100.0	100.0	28.0	40.6	46.2	54.8	68.6
TxtCompDist-nolist(euclid)	73.5	77.6	89.8	91.8	98.0	44.0	50.0	66.0	80.0	94.0	20.0	96.0	99.3	100.0	100.0	23.9	38.9	45.0	54.4	68.5
TxtCompDist-nolist(city)	85.7	87.8	89.8	93.9	100.0	40.0	56.0	66.0	76.0	100.0	40.0	100.0	100.0	100.0	100.0	28.3	44.7	48.3	60.6	73.3

Table 6.6: 500 word segments results. This table gives the percentage of trials in which the anomalous segments were identified in the top n segments for some of our methods across the different document collections. In the leftmost column is the method name followed by the distance measure and whether normalization was used. This chart shows that TxtCompDist (the textual complement method) performs best overall and also best in almost every subtask (no matter what distance measure is used with it). The ClustDist and MeanComp methods perform significantly worse than the other methods regardless of the distance measure.

	Text Segment Size: 100 words																			
	Chinese Translation					Fact Opinion					Anarchist Cookbook					Authorship				
	top 1	top 3	top 5	top 10	top20	top 1	top 3	top 5	top 10	top20	top 1	top 3	top 5	top 10	top20	top 1	top 3	top 5	top 10	top20
ClustDist(cityblock)	16.0	28.0	32.0	44.0	62.0	12.0	14.0	36.0	50.0	66.0	18.3	23.3	32.0	39.3	59.3	7.5	13.8	19.9	36.4	60.8
ClustDist(cosine)	4.0	12.0	12.0	26.0	40.0	8.0	10.0	12.0	22.0	38.0	18.3	23.3	32.0	39.3	59.3	5.4	10.1	16.3	34.9	62.7
ClustDist(cosine)(Zscore)	4.0	12.0	12.0	26.0	38.0	8.0	10.0	12.0	22.0	38.0	6.0	7.3	8.0	10.7	12.7	5.5	9.9	16.1	34.8	62.7
ClustDist(euclid)	10.0	16.0	22.0	30.0	40.0	12.0	14.0	22.0	28.0	40.0	11.7	14.0	25.3	29.3	51.3	7.8	11.3	15.2	33.0	63.2
ClustDist(Pearson)	4.0	12.0	18.0	28.0	46.0	8.0	16.0	16.0	32.0	42.0	18.0	23.3	31.3	39.3	59.3	7.0	11.3	15.8	32.0	64.0
ClustDist(Pearson)(Zscore)	10.0	16.0	16.0	18.0	38.0	8.0	16.0	16.0	22.0	34.0	6.6	8.0	14.7	21.3	40.0	7.3	10.8	15.5	33.0	64.9
MeanComp(cityblock)	2.0	10.0	10.0	24.0	78.0	0.0	14.0	18.0	42.0	60.0	13.1	18.7	27.3	38.7	60.7	4.2	12.6	18.3	36.4	62.9
MeanComp(cityblock)(Zscore)	4.0	12.0	16.0	28.0	46.0	8.0	16.0	16.0	28.0	40.0	14.0	20.0	24.0	36.7	54.7	7.0	11.3	15.7	32.1	64.2
MeanComp(cosine)	50.0	54.0	58.0	72.0	90.0	58.0	60.0	60.0	74.0	90.0	18.3	23.3	32.0	41.3	59.3	12.4	20.7	30.9	41.1	57.2
MeanComp(cosine)(Zscore)	2.0	8.0	12.0	26.0	76.0	0.0	8.0	20.0	48.0	64.0	18.3	23.3	32.0	42.0	60.0	4.3	12.1	18.6	37.1	63.1
MeanComp(euclid)	8.0	14.0	18.0	24.0	40.0	8.0	14.0	16.0	24.0	34.0	16.3	25.3	29.3	43.3	62.7	6.7	9.9	16.1	33.2	64.6
MeanComp(Pearson)	4.0	12.0	18.0	28.0	46.0	8.0	16.0	16.0	32.0	42.0	5.7	7.3	14.7	22.7	38.0	7.0	11.3	15.8	32.0	64.0
Meth(SumDiff)	12.0	22.0	22.0	36.0	54.0	12.0	24.0	26.0	38.0	52.0	11.4	14.0	22.7	31.3	47.3	8.2	11.8	17.0	33.2	64.4
Meth(SumDiff)(Zscore)	10.0	16.0	22.0	30.0	40.0	12.0	14.0	22.0	28.0	40.0	14.0	20.0	24.0	36.7	54.7	7.8	11.3	15.2	33.0	63.2
Stahel-Donoho	46.0	60.0	68.0	78.0	84.0	40.0	58.0	70.0	84.0	94.0	83.3	96.7	98.7	100.0	100.0	11.8	27.4	36.8	50.5	71.2
PCOut weights	14.1	40.8	54.9	71.8	94.4	15.5	33.8	52.1	64.8	88.7	70.0	72.3	75.6	80.8	92.5	22.7	44.0	54.4	68.5	83.6
TxtCompDist(city)	54.0	60.0	68.0	74.0	80.0	26.5	46.0	49.5	62.0	78.5	38.0	68.0	74.0	88.0	98.0	13.5	25.5	32.6	48.6	63.8
TxtCompDist(city)(scale01)	36.0	54.0	58.0	60.0	76.0	17.5	36.0	46.0	64.0	76.0	34.0	38.0	46.0	58.0	82.0	16.3	31.3	40.5	52.0	67.8
TxtCompDist(cosine)	12.0	24.0	48.0	82.0	88.0	2.0	4.0	10.0	40.0	52.0	5.3	43.3	54.7	70.7	89.3	2.3	26.8	33.2	48.5	64.4
TxtCompDist(pearson)	12.0	24.0	48.0	82.0	88.0	2.0	4.0	10.0	40.0	52.0	5.3	44.0	54.7	70.7	90.7	2.5	26.9	33.7	48.6	64.2
TxtCompDist(euclid)	38.0	60.0	64.0	74.0	90.0	48.0	68.0	72.0	78.0	88.0	12.0	40.0	63.3	82.7	96.0	14.1	20.9	29.6	42.4	60.4
TxtCompDist-nolist(cosine)	32.0	60.0	64.0	82.0	90.0	46.0	66.0	72.0	84.0	90.0	23.3	54.0	66.7	90.0	98.0	14.1	22.4	31.5	45.9	61.9
TxtCompDist-nolist(pearson)	34.0	60.0	64.0	82.0	90.0	46.0	66.0	72.0	82.0	90.0	23.3	54.0	66.0	90.0	98.7	14.0	22.4	31.7	45.9	62.0
TxtCompDist-nolist(euclid)	38.0	60.0	64.0	74.0	90.0	48.0	68.0	72.0	78.0	88.0	12.0	40.0	63.3	82.7	96.0	14.1	20.9	29.6	42.4	60.4
TxtCompDist-nolist(city)	54.0	70.0	72.0	80.0	88.0	52.0	58.0	66.0	80.0	90.0	35.3	86.7	94.7	98.7	100.0	14.6	25.7	31.8	48.6	63.8

Table 6.7: 100 word segments results. This table gives the percentage of trials in which the anomalous segments were identified in the top n segments for some of our methods across the different document collections. In the leftmost column is the method name followed by the distance measure and whether normalization was used. In this chart we can see that the Stahel-Donoho Estimator was overall the best performer for 100 word segments by a very slim margin. TxtCompDist (the textual complement method with city block distance) has very close scores and still performs best on some tasks.

6.8 Conclusions from Anomaly Detection Experiments

Results for detecting anomalous segments in documents are promising. We achieve good results on all tasks and for large segments we can reliably detect anomalies as the most anomalous segment with accuracy in the high 90's for some tasks.

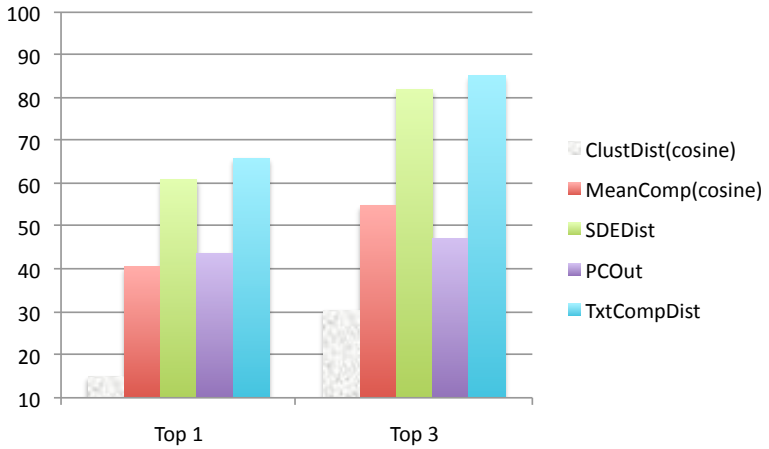


Figure 6.6: This figure shows the average time segments are found in the top 1 or 3 segments across all test sets for a segments size of 1000 words. The TxtCompDist method (Distance form the textual complement) performs best, but is closely followed by the Stahel-Donoho based method.

The results show that identifying an inserted anomaly as the most anomalous segment (Top 1) can be a difficult task. This is not surprising, given that in these experiments there is only a 2% probability of choosing this segment by chance, but we do far better than this, averaging 32% of the time for 100 word segments and 68% for one 1,000 word segments across all experiments. We do extremely well in the case of inserting Chinese newswire translated with Google into English Newswire where we can identify anomalies as the top segment returned 96% of the time for

large segments. In practical terms this means that if you can only look at the top segment returned by the system, you can be 96% certain you will have found the anomaly in a document (if it has one).

Our techniques perform best when there is a large difference in the *text type* (genre) or style as in the Anarchist Cookbook and Chinese Translation Experiments. The task with the best overall results for all methods was detecting when a machine translated news story was inserted into a collection of newswire, the worst was the task of detecting different Victorian authors. Also, it should be mentioned that in all experiments and with all detection methods results always improve as we increased the length of our segments.

On the whole our experiments show that the standardizing the scores on a scale from 0 to 1 does indeed produce better results for some types of anomaly detection, but not for all tasks we performed. The cases where it performed better than the standard raw scores were cases where the genre distinction was small (as in the authorship tests). Many of the readability formulas, for instance, distinguish these genre differences quite well but their effects on anomaly detection are greatly reduced when we standardize these scores.

The results from the testing of different procedures for detecting anomalies indicate that the TxtCompDist (Distance from the textual complement) method performs best, but the the SDEDist method also works very well. The other three methods do considerably worse. Figure 6.6 and Table 6.8 show the average over all experiments for these different methods using a segment size of 1000 words. While the SDEDist estimator is close to the accuracy of the TxtCompDist approach, it also is much slower

and required more than four times as long as other methods to compute.

	Top 1	**Top 3**
ClustDist	14.95	30.40
MeanComp	40.57	54.88
SDEDist	60.87	82.02
PCOut	43.76	47.13
TxtCompDist	**65.89**	**85.21**

Table 6.8: Results for the best performing method on 1000 words of text. This is the average over all experiments and test sets.

These results are promising, but this experimental setup makes the assumption that there is an anomaly to be identified in the document because we always return at least one segment. One could use this method to isolate segments or documents that might possibly be anomalous and our experiments show that if there were any truly anomalous segments then this strategy would be of great help. However, even picking only the single segment we judge to be most anomalous could create a lot candidate segments to be reviewed if this technique is to be applied to many different collections of data. In the next Chapter we look at solving this problem by examining the recall and precision figures for this type of anomaly detection and experimenting with whether it is possible to say with certainty that a segment is anomalous.

Chapter 7

Refinements: Thresholds and Feature Selection

7.1 Overview

Results presented in the previous chapter are promising and show that we can consistently mark anomalous segments in documents at a level much higher than chance. Specifically we showed that we could reliably identify many types of anomalies in documents if we could return five to ten segments from a document. The assumption is that other means could be used to review those five or ten segments, but you could be relatively sure all possible anomalies had been identified. This would be very useful to a human who knows that there is likely an anomaly in a document or collection because it significantly reduces their search space. It also has the advantage of high recall, that is, it insures that true anomalies will not be missed.

This methodology of returning many candidate anomalous segments has the ad-

vantage of good recall, but if you have no prior knowledge about whether a corpus or document contains an anomaly then this process may lead to far too much data being marked as "anomalous", with no indication of the degree to which we believe the data to be anomalous. It might be more useful in many situations to have an idea of the likelihood that a segment is anomalous so that only segments with a high probability of being anomalous will be marked. In this chapter we look at tailoring our unsupervised anomaly detection procedure to identify anomalies precisely. We examine choosing a threshold for anomaly scores above which the probability of true anomalies is very high by examining recall and precision, as we have defined them for anomaly detection. In the final section in this chapter we examine the usefulness of features across different dimensions of anomaly and on different sizes of text.

7.2 Defining Recall and Precision

In the previous experiments we computed a score for a segment's distance from its complement in a document. This score was then used to rank the segments by their degree of anomaly (with respect to that document). In this section we examine this score and whether it is possible to use it to pick a global threshold above which one can reliably assume a segment to be anomalous. A segment's difference from the rest of the document is computed as the city block distance of the vector representing that segment with the vector representing the rest of the document. In these experiments we set a threshold on this number and only mark segments as anomalous if they are above that threshold. Our hope was that a threshold could be set above which only actual anomalies would be identified.

Another way to think about making this unsupervised anomaly detection more precise is that we would like to pick a threshold that will have 100% precision while maximizing recall. We define recall at a given threshold as the total number of anomalous segments correctly identified across all experiments divided by the total number of anomalies.

We define precision at a threshold as the total number of segments correctly identified as anomalous in experiments using that threshold divided by all segments in all documents that have a score above that threshold.

$$\text{Recall} = \frac{\text{Anomalous segments correctly identified}}{\text{Total \# of Anomalous segments}} \tag{7.1}$$

$$\text{Precision} = \frac{\text{Anomalous segments correctly identified}}{\text{Segments marked as anomalous}} \tag{7.2}$$

We also make use of F-measure and use it as is most common in Information Retrieval with Recall and Precision weighted equally.

$$\text{F} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Recall} + \text{Precision}} \tag{7.3}$$

7.3 Varying the Threshold

Figures 7.1 and 7.2 illustrate what happens to recall and precision as we decrease our threshold on the distance score computed for each 1,000 word segment. For example, in the case of the Fact vs Opinion detection experiments (Figure 7.1) we can see that if we set the threshold to give 100% recall we achieve only 52% precision, meaning we identify all anomalous segments but half of the segments we return are not anomalous. We however are interested at the point where precision is at its maximum

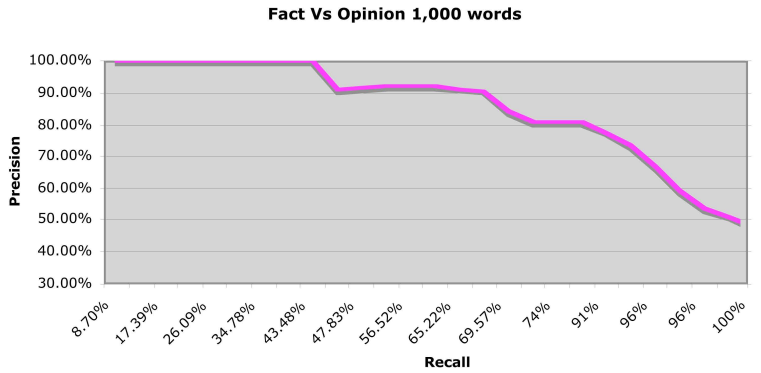


Figure 7.1: Precision versus Recall for Fact versus Opinion Experiments

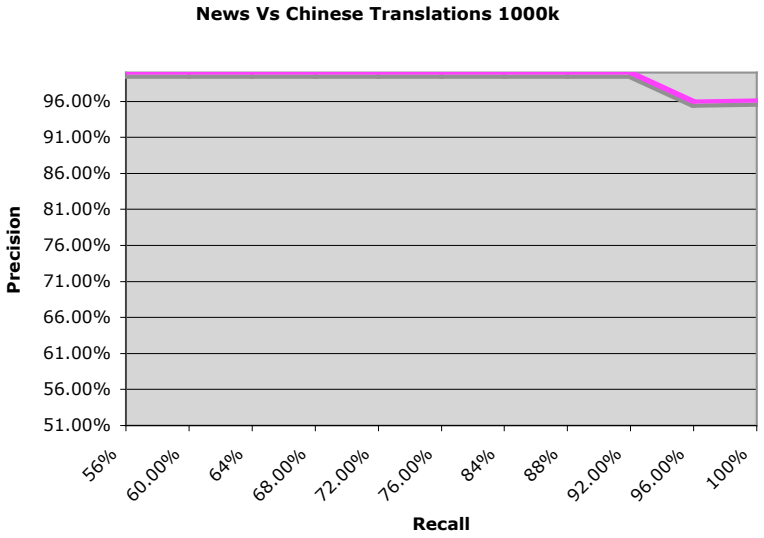


Figure 7.2: Precision versus Recall for Chinese Translation Experiments

and recall is as high as possible. These figures show how tailoring the threshold to a given experiment could increase precision and still maintain high recall. In the case of newswire with Chinese translations inserted, the chart shows that the optimum threshold where we achieve 100% precision will detect 93% of anomalies of this type. Obviously in a completely unsupervised scenario it wouldn't be possible to pick a

threshold that truly maximizes precision for new data and these Figures simply show the optimum possible for this data and this anomaly detection procedure.

7.4 Choosing Thresholds

A portion of the data used in the anomaly detection experiments was "held out" and used to choose thresholds automatically for the identification of anomaly. We automatically picked thresholds for each flavor of anomaly detection and the three different segment sizes and also experimented with picking a global threshold to use as an experiment for all segment sizes and types of anomaly detection.

Ten percent of the data was set aside to use as testing and the remaining 90% of data was used as training data to pick our thresholds. The full training data (including different genre experiments and segment sizes) consists of results for 11,880 segments, half of which are anomalous. The testing data consists of 1,320 segments, half of which are also anomalous.

We use the training data to select a threshold that gives the highest precision (usually 100%) while keeping recall as high as possible. We then fix this threshold and apply it to the segments in the testing data to see how precise and accurately we can identify anomaly. Figure 7.3 shows the results of these experiments. The figures in the "all" column show the highest threshold (most precision) for a group of experiments was picked form the training data and used across the set of test data. This gives an idea of how many anomalies we can expect to detect when we focus on making precise judgements (high precision) at the expense of recall. This might be desirable in a fully automatic system where a program is making decisions based

on whether something is labeled anomalous and we might want to err on the side of caution. For example, if an automatic system was determining whether the addition of text to an article on a website like wikipedia was anomalous or not, we would probably rather not block this addition unless we were sure it was not in keeping with the rest of the data.

Recall/Precision (Threshold)	Segment Size	Chinese Translations	Fact vs Opinion	Anarchists Cookbook	all
	100	52%/100% (369)	46%/100% (369)	36%/100% (371)	44%/100% (371)
	500	83%/100% (279)	38%/100% (280)	66%/100% (280)	62%/100% (280)
	1000	96%/100% (252)	43%/100% (252)	88%/100% (252)	76%/100% (252)
	all	46.3%/100% (370)	22.2%/100% (370)	30%/100% (370)	33%/100% (370)

Figure 7.3: Results on test data from learning the optimum threshold to maximize precision. The values in each cell show Recall/Precision and then the actual numerical value of the threshold used in parentheses. Note that 100% precision was achieved in all cases.

As we can see from Figure 7.3, the thresholds chosen to maximize precision on the training set give very good results for precision on the test set. In fact the automatic thresholds were high enough that the system only ever picked out truly anomalous segments. This means that on all of the unseen test documents the method scored 100% precision, but that not all anomalies were identified so their recall scores are lower. The table also shows results for using the highest threshold in each column and row for all experiments at the end of that column or row. The results are good for large segments as we can achieve 76% recall with 100% precision for 1,000 word

segments, but the recall drops to 44% for 100 word segments.

The threshold numbers (shown on the chart in parentheses) or the exact threshold value we learned from the training data. They indicate that there is very little difference in the optimal thresholds across different test collections, but the threshold values do change considerably between segment sizes. This indicates that to achieve high precision for short segments the threshold must be raised, as the measure is less accurate. Likewise, when the segment size is large, the threshold can be lowered while still achieving 100% precision.

These results are promising and show that for a given segment size similar thresholds apply and that it is possible to achieve high precision and with a good recall even for moderately sized segments.

7.5 Feature Selection for Unsupervised Anomaly Detection

Our approach for identifying anomaly at the segment level with no training data has proven fairly successful, but we may be interested in what features are best at distinguishing these segments across different experiments. In this section we examine the features that contribute most to this type of unsupervised anomaly detection as well as the features that contribute least. We are identifying the features that discriminated well in the experiments (actually the average over many experiments). The set of features that make something more or less anomalous are closely tied to the anomaly detection technique used and we calculated the impact that each feature

had in the context of the anomaly detection experiments in Chapter 6 using our best performing method. This tells us the value of these features on actual tasks which gives us an indication of the features that best separate different types of documents as well as insight into how the feature set could be improved for certain tasks (or all tasks).

In our experiments we used 166 features to characterize a segment of text. Here we measured the difference in the range of values each feature took across all segments in the entire document. Features that do not change very much (i.e. are consistent) across normal and anomalous segments will not aid in the detection, of anomaly, but also will not hinder it. Features that do change wildly can either greatly improve unsupervised anomaly detection if they do so in line with anomaly, or they can harm it if they do so sporadically.

Features were ranked based on their ability to separate anomalous segments from normal segments in a collection of experiments based on their contribution to differentiating anomalies. This is done, for every feature, by computing the average difference this feature has between an anomalous segment and its complement and subtracting the average difference normal segments had from their complements for the same feature. Let $\mathbf{a}$ and $\mathbf{o}$ be vectors that are of length, p, the number of variables. The vector $\mathbf{a}$ will contain the average difference from all anomalous segments' features to their complements in a set of experiments. Likewise, $\mathbf{o}$ will be the difference from normal segments' feature scores to their complements' scores (we chose 'o' here for **o**rdinary instead of an 'n' for normal because n will be used as in previous chapters

to denote the number of observations). To compute **a**, for a single test set, we have:

$$a_j = \frac{1}{n_a} \sum_{i=1}^{n_a} \left| x_{ij} - c_j^x \right|, \text{ for } j = 1, ..., p \text{ and } x \in Anomalies \tag{7.4}$$

where

x_{ij}	is the j^{th} feature of the i^{th} anomaly to be tested for this test set
c_j^x	is the j^{th} feature of the textual complement of **x**
n_a	is the number of anomalies to tested
Anomalies	is the set of all anomalies to be tested

An equivalent formula is given for **o** measuring the difference from normal segments' features to their complements' features.

$$o_j = \frac{1}{n_o} \sum_{i=1}^{n_o} \left| x_{ij} - c_j^x \right|, \text{ for } j = 1, ..., p \text{ and } x \in Normal \tag{7.5}$$

where

x_{ij}	is the j^{th} feature of the i^{th} normal segment
c_j^x	is the j^{th} feature of the textual complement of **x** (which will contain an anomaly)
n_a	is the number of normal segments
$Normal$	is the set of all normal segments

We compute a feature, j's contribution to the detection of anomalies in this test set as $a_j - o_j$. This contribution was averaged across all test sets to give the average contribution for a large group of experiments. This measure will be large positive number for features that are good at discriminating between anomalies and normal segments. If the measure is a large negative number for a feature, then the feature often hinders anomaly detection and is a bad discriminator. When this contribution measure is close to zero for a feature then it does not affect anomaly detection results overall. A feature's contribution could be close to zero if either this feature tends

not to vary much across normal and anomalous segments or if it tends to vary so randomly that it is *good* and *bad* about the same amount of time.

All Experiments				
Rank	100 word segment size	500 word segment size	1000 word segment size	All segment sizes
1	fogindex	fogindex	fogindex	fogindex
2	passivesen	passivesen	passivesen	passivesen
3	fleschease	fleschease	fleschease	fleschease
4	perlongsen	perlongsen	perlongsen	perlongsen
5	lix	lix	lix	lix
6	avgsenlength	avgsenlength	avgsenlength	avgsenlength
7	ari	perc1syll	pershortsen	ari
8	fleschgrade	pershortsen	perc1syll	perc1syll
9	gig300k	per6orMoreLetter	per6orMoreLetter	pershortsen
10	gig200k	ari	ari	per6orMoreLetter
11	gig100k	gig300k	gig300k	gig300k
12	gig50k	gig200k	gig200k	gig200k
13	gig5k	gig100k	gig100k	gig100k
14	gig10k	gig50k	gig50k	fleschgrade
15	nouns	fleschgrade	fleschgrade	gig50k

Figure 7.4: The most effective features for all experiments.

We examined the features that did best overall across all genres and experiments by calculating the measure above for all experiments (only keeping experiments with different segment sizes separate). Figure 7.4 shows that the top 15 most useful features for detecting anomaly in all experiments are fairly consistent across segment sizes. A key for the abbreviations in Figure 7.4 as well as in Figures 7.5, 7.6, and 7.7 can be found in Appendix C.

Figure 7.5 shows features that contribute negatively to all experiments. Removing these features from the feature vectors will improve the results of the unsupervised anomaly detection program. They vary a little more than the best features did across segments, but there is still a fair amount of consistency in a group of the emotional

All Experiments				
Rank	100 word segment size	500 word segment size	1000 word segment size	All segment sizes
1	econ	ritual	kin	econ
2	male	land	ani	kin
3	kin	kin	ritual	ritual
4	ritual	exprs	food	fetch
5	fetch	intrj	intrj	land
6	ord	ani	posTRIlist	male
7	exert	sky	sky	exprs
8	land	fetch	decr	exert
9	exprs	aquatic	exch	bldgpt
10	bldgpt	posTRIlist	you	route
11	route	exch	rise	ani
12	place	you	travel	rise
13	bodypt	think	posBIlist	aquatic
14	pleasur	feel	fail	fail
15	hostile	travel	land	pleasur

Figure 7.5: The least effective features for all experiments.

tone based features that do not aid in anomaly detection.

We can also look at the features that contribute most in each task setting. Figure 7.6 shows the best features for each task scenario averaged over all experiments and all segment sizes.

While the features that negatively impact anomaly detection performance with in each setting are shown in Figure 7.7.

These contribution scores showed us that although the majority of the features do contribute positively to anomaly detection, it is usually the top 15 that make the biggest difference. Likewise for each anomaly detection experiment there are approximately 5 to 10 of the 166 features that negatively impact performance, but not as consistently. These results are interesting as they can guide the search for important or unhelpful features in anomaly detection and also give an indication of

All segment sizes				
Rank	Authorship	Fact vs Opinion	Newswire vs Anarchists Cookbook	Newswire vs Chinese Translations
1	fogindex	fogindex	passivesen	fogindex
2	pershortsen	passivesen	fogindex	fleschease
3	fleschease	avgsenlength	gig300k	perlongsen
4	passivesen	perlongsen	gig200k	lix
5	perlongsen	fleschease	gig100k	avgsenlength
6	lix	lix	gig50k	ari
7	avgsenlength	ari	gig10k	perc1syll
8	perc3syll	nouns	gig5k	per6orMoreLetter
9	ari	fleschgrade	avgsenlength	fleschgrade
10	per6orMoreLetter	gig300k	perlongsen	pershortsen
11	fleschgrade	gig200k	per6orMoreLetter	perc3syll
12	perc1syll	gig100k	gig1k	polit
13	punct	gig50k	perc1syll	smog
14	smog	perc1syll	lix	coleman
15	pronouns	gig5k	nouns	pronouns

Figure 7.6: Most effective features across anomaly detection tasks.

All segment sizes				
Rank	Authorship	Fact vs Opinion	Newswire vs Anarchists Cookbook	Newswire vs Chinese Translations
1	posBIlist	pos	yes	yes
2	posTRIlist	travel	rise	you
3	yes	route	you	race
4	race	vehicle	travel	kin
5	exch	fall	race	intrj
6	aquatic	land	fail	female
7	decr	ritual	kin	color
8	feel	stay	intrj	decr
9	rise	dist	female	fall
10	nonadlt	goal	decr	think
11	vehicle	rise	think	posBIlist
12	fall	exprs	sv	punct
13	ritual	food	posBIlist	exch
14	artlist	aquatic	bldgpt	semi
15	exprs	strng	land	self

Figure 7.7: Least effective features across anomaly detection tasks.

the best features for distinguishing anomalies in different task settings.

7.6 Summary

The beginning of this chapter focuses on the problem of increasing the precision of our anomaly detection methods. It is often desirable in real world situations to have methods that may miss a few instances of anomalies, but when they identify an anomaly you can be certain it is truly anomalous. Using a supervised approach, we held out a portion of the data to use for learning the thresholds which returned the highest recall while maximizing precision. This method proved successful and we showed that on held out data we could achieve perfect precision while maintaing high recall on many tasks.

In the final section of this chapter we examined the impact each of our features had on the task of anomaly detection. We examined how well each feature performed across all experiments as well as within individual scenarios and at certain segment sizes. Features that performed particularly well on all experiments were the *readability measures*, *sentence length*, *percentage of passive sentences*, and the *obscurity of vocabulary* features.

Chapter 8

Conclusions and Future Work

8.1 Summary of Conclusions

Detecting an anomalous document (or segment of a document) when no training examples are available is a challenging research area with significant application potential. The techniques developed in this thesis are applicable to the detection of many types of "outliers" or anomalies that could occur in electronic text and do not require prior specification or training examples of what those anomalies might be. Methodologies and implementations were developed and extensively investigated for identifying textual anomalies of various sizes and types while examining the features used for this detection. We introduced a novel method for anomaly detection in text that performs better than even advanced multivariate outlier detection methods. Accuracy was shown to improve as we increased the size of the text, especially long documents of around 1,000 words, but worked well on small 100 word segments, still detecting anomalies with an accuracy well above chance. The features used to

characterize text and thus identify unusual text have been ranked by their usefulness in detecting anomalies of different types and in different sizes of texts. This is an exciting technique with a vast range of potential applications.

Main contributions of this research:

- Variations in text can be viewed as a type of anomaly or *outlier* and can be successfully detected using automatic unsupervised techniques without the use of content words. (Sections 1.1 and 6.8)

- Detection strategies that measure a piece of text's distance from its complement (see TxtCompDist in 5.2.5) are the best performing methods for detection of anomaly when no training data is available. (Sections 5.2.5 and 6.7)

- Accuracy for anomaly detection improves considerably as we increase the length of our segments. (Chapter 6)

- Stylistic features and distributions of the rarity of words are a good choice for characterizing text and detecting a broad range of anomalies. (Section 7.5)

- The most accurate unsupervised anomaly detection is possible when the anomalies are a different writing style or genre when compared to the 'normal' data (as opposed to different topic, tone or authorship). This confirms work which indicates that stylistic features are useful for genre detection. (Section 3.4)

- Thresholds for unsupervised anomaly detection can be reliably learned to maximize precision. (Section 7.4)

8.2 Future Work

Anomalous Sentences

One aim for the future of our unsupervised anomaly detection is to adapt the procedure to work on very small segments about the size of a sentence. We are currently working on a modified version of our unsupervised anomaly detection program that only uses features that scale down to as little as ten words so that we can attempt to recognize when a single sentence is anomalous with respect to its surrounding contexts. The collection of feature analysis techniques developed in this report makes it easy to determine which features are useful at this level and modify our approach accordingly. Furthermore we are also developing new sentence level unsupervised techniques that are more appropriate for unsupervised detection at the word and sentence level.

Multiple Anomalies

Our experiments in this thesis assume that there is only one anomalous segment in a document (or collection). We showed that it is possible to achieve good accuracy detecting that anomaly, but it would be interesting to see exactly how much that accuracy is impacted when documents contain more than one anomaly. If documents, for example, contained two anomalous segments then we are fairly certain this would have very little impact on our measure (especially for long documents that have 50 or more segments). On the other hand, documents that have almost 50% outliers may effect the accuracy our methods more drastically. We have strived to create methods that are *robust*, and thus not overly sensitive to outliers in the data, but it remains

to be seen exactly how resistant they are on real data that contains many anomalous segments.

Using Document Flow

Another possibility for the future of unsupervised anomaly detection is to tailor the procedure to **only** detect anomalies within documents. We identified anomalies within documents (Chapter 6), but the goal of our research was to develop a broad anomaly detection technique that could be used successfully both within documents and within collections of documents. We made no attempt to tailor our procedure specifically to anomalies in single documents. All of the procedures we developed can be seen as "bag of segments" techniques because segments can have any ordering and the results will be the same. If we take the ordering of segments into account this may give us information about the structure of a document that might aid in anomaly detection.

Documents have an implicit structure or *flow* in their text and it may be possible to exploit this fact to pinpoint where this flow is broken. Our anomaly detection techniques, thus far, treat each segment or document independently and uniformly measure its distance from all other text in a document or collection. If we focus on pieces of text within documents we might be able to gain something by taking into account a document's structure. For instance, it is likely that the second and third paragraphs in a document are more similar to each other in terms of content than the second and twentieth paragraphs would be. This is an assumption made in text tiling research [Hearst, 1997] and we believe that it could be a very beneficial addi-

tion to anomaly detection. This information could help you to spot anomalies more accurately because you could gain information about where the flow in a document is broken rather that what stands out most overall. Possible research could include adapting anomaly detection techniques to take into account the position of segments in documents. In this approach, we might weight the distance between segments, so that are segments that are closer to each other in a document are assumed more likely to be similar. The assumption that closer segments are more similar seem to be valid in terms of topic and context words, but it remains to be seen whether this would also hold true for an author's writing style or tone and thus for the techniques developed in this thesis.

Bibliography

Ahmad, K. (2008). Edderkoppspinn eller nettverk: News media and the use of polar words in emotive contexts. *Synaps*, 21:20–36.

Ahmad, K. and Al-Sayed, R. (2005). Community of practice and the special language 'ground'. In Clarke, S. and Coakes, E., editors, *Encyclopedia of Communities of Practice in Information and Knowledge Management*. IGP Reference, Hershey, PA.

Ahmad, K. and Rogers, M. A. (2001). Corpus linguistics and terminology extraction. In Wright, S.-E. and Budin, G., editors, *Handbook of Terminology Management*, volume 2. Amsterdam & Philadelphia: John Benjamins Publishing Company.

Anderson, E., Bai, Z., Dongarra, J., Greenbaum, A., McKenney, A., Croz, J. D., Hammarling, S., Demmel, J., Bischof, C., and Sorensen, D. (1990). Lapack: a portable linear algebra library for high-performance computers. In *Supercomputing '90: Proceedings of the 1990 ACM/IEEE Conference on Supercomputing*, pages 2–11, Washington, DC, USA. IEEE Computer Society.

Argamon, S., Koppel, M., and Avneri, G. (1998). Routing documents according to style. In *Proceedings of the First International Workshop on Innovative Internet Information Systems (IIIS–98)*, Pisa, Italy.

Argamon, S., Koppel, M., Fine, J., and Shimoni, A. R. (2003). Gender, genre, and writing style in formal written texts. *Text*, 23(3):321—346.

Arnold, K. and Gosling, J. (1998). *The Java Programming Language, Second Edition*. Addison-Wesley, Reading, MA.

Baljko, M. and Hirst, G. (1999). Subjectivity in stylistic assessment. *Text Technology*, 9(1):5–17.

Barnett, V. and Lewis, T. (1998). *Outliers in Statistical Data*. John Wiley & Sons, 3rd edition.

Bekkerman, R., Eguchi, K., and Allan, J. (2006). Unsupervised non-topical classification of documents. Technical report, Center for Intelligent Information Retrieval, University of Massachusetts, Amherst.

Bendre, S. M. and Kale, B. K. (1987). Masking effect on tests for outliers in normal samples. *Biometrika*, 74(4):891–896.

Bernardo, J. M. and Smith, A. F. M. (1995). *Bayesian Theory*. Chichester: Wiley.

Biber, D. (1988). *Variation Across Speech and Writing*. Cambridge University Press, Cambridge.

Biber, D. (1989). A typology of English texts. *Linguistics*, 27:3–43.

Biber, D. (1992). The multi-dimensional approach to linguistic analyses of genre variation: An overview of methodology and findings. *Computers and the Humanities*, 26(5):331–345.

Biber, D. (1995). *Dimensions of Register Variation: A Cross-Linguistic Comparison*. Cambridge University Press, Cambridge, UK.

Bolton, R. J. and Hand, D. J. (2002). Statistical fraud detection: A review. *Statistical Science*, 17(3):235–249.

Briscoe, T. and Carroll, J. (2002). Robust accurate statistical annotation of general text. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation(LREC–02)*, pages 1499–1504, Las Palmas, Gran Canaria.

Briscoe, T., Carroll, J., and Watson, R. (2006). The second release of the RASP System. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (ACL/COLING 06)*, Sydney, Australia.

Brys, G., Hubert, M., and Rousseeuw, P. J. (2005). A robustification of independent component analysis. *Journal of Chemometrics*, 19(5-7):364–375.

Brys, G., Hubert, M., and Struyf, A. (2004). A robust measure of skewness. *Journal of Computational and Graphical Statistics*, 13(4):996–1017.

Bull, J., Collins, C., Coughlin, E., and Sharp, D. (2001). Technical Review of Plagiarism Detection Software Report. Technical report, prepared for the Joint Information System Committee (JISC) by the Computer Assisted Assessment Centre at The University of Luton.

Burchfield, R. W. (1971). *Oxford English Dictionary*. Oxford University Press, Oxford.

Burge, P. and Shawe-Taylor, J. (1997). Detecting cellular fraud using adaptive prototypes. In *Proceedings of AAAI 97 Workshop on AI Approaches to Fraud Detection & Risk Management*, pages 9–13. AAAI Press.

Burnage, G. and Dunlop, D. (1992). Encoding the British National Corpus. In *Proceedings of the 13th International Conference on English Language Research on Computerised Corpora*.

Burnard, L. (1995). *Users Reference Guide for the British National Corpus*. Oxford University Computing Services, Oxford.

Burrows, J. F. (1992). Computers and the study of literature. In Butler, C. S., editor, *Computers and Written Texts*, pages 167–204. Blackwell, Oxford.

Chen, H. and Dumais, S. T. (2000). Bringing order to the Web: automatically categorizing search results. In *Proceedings of CHI-00, ACM International Conference on Human Factors in Computing Systems*, pages 145–152, Den Haag, NL. ACM Press, New York, US.

Clough, P. (2000). Analyzing style - readability. Technical report, University of Sheffield, http://ir.shef.ac.uk/cloughie/papers.html.

Clough, P., Gaizauskas, R., Piao, S., and Wilks, Y. (2002). METER: MEasuring TExt Reuse. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL02)*, Philadelphia.

Coulthard, R. M. (1992). Forensic discourse analysis. In Coulthard, R. M., editor, *Advances in Spoken Discourse Analysis*, pages 242–257. Routledge, London.

Coulthard, R. M. (1994). On the use of corpora in the analysis of forensic texts. *Forensic Linguistics: The International Journal of Speech, Language and the Law*, 1(1):27–43.

Coulthard, R. M. (2004). Author Identification, Idiolect, and Linguistic Uniqueness. *Applied Linguistics*, 25(4):431–447.

Croux, C. and Ruiz-Gazen, A. (2005). High breakdown estimators for principal components: the projection-pursuit approach revisited. *Journal of Multivariate Analysis*, 95(1):206–226.

Denning, D. E. (1987). An intrusion-detection model. *IEEE Trans. Softw. Eng.*, 13(2):222–232.

Dewdney, N., VanEss-Dykema, C., and MacMillan, R. (2001). The form is the substance: classification of genres in text. In *Proceedings of the workshop on Human Language Technology and Knowledge Management*, pages 1–8, Morristown, NJ, USA. Association for Computational Linguistics.

Donoho, D. L. (1982). Breakdown properties of multivariate location estimators. Ph.D. qualifying paper, Harvard University.

Donoho, D. L. and Gasko, M. (1992). Breakdown properties of location estimates based on halfspace depth and projected outlyingness. *The Annals of Statistics*, 20(4):1803–1827.

Efron, B. (1986). Why isn't everyone a bayesian. *American Statistician*, 40:1–11.

Ellegard, A. (1962). A statistical method for determining authorship: The Junius Letters: 1769-1772. In *13*, Gothenburg Studies in English. University of Gothenburg.

Fawcett, T. and Provost, F. (1997). Adaptive fraud detection. *Data Mining and Knowledge Discovery*, 1(3):291–316.

Fetterly, D., Manasse, M., and Najork, M. (2004). Spam, damn spam, and statistics: Using statistical analysis to locate spam web pages. In *International Workshop on the Web and Databases*, pages 1–6.

Fetterly, D., Manasse, M., and Najork, M. (2005). Detecting phrase-level duplication on the world wide web. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 170–177, New York, NY, USA. ACM.

Filzmoser, P. and Fritz, H. (2007). Exploring high-dimensional data with robust principal components. In Filzmoser, S. A. P. and Kharin, Y., editors, *Proceedings of the Eighth International Conference on Computer Data Analysis and Modeling*, volume 1, pages 43–50. Belarusian State University, Minsk.

Filzmoser, P., Maronna, R., and Werner, M. (2008). Outlier identification in high dimensions. *Computational Statistics and Data Analysis*, 52:1694–1711.

Flesch, R. (1974). *The Art of Readable Writing*. Harper and Row, New York.

Francis, W. N. and Kucera, H. (1964). *The Brown Corpus Manual: A Standard Corpus of Present-Day Edited American English, for use with Digital Computers*. Brown University, Providence, Rhode Island.

Friedman, J. H. and Tukey, J. W. (1974). A projection pursuit algorithm for exploratory data analysis. *IEEE Transactons on Computers*, C-23(9):881–890.

Gaizauskas, R., Foster, J., Wilks, Y., Arundel, J., Clough, P., and Piao, S. (2001). The METER corpus: a corpus for analysing journalistic text reuse. In *Proceedings of Corpus Linguistics*, pages 214–223, Lancaster, UK.

Gaizauskas, R. and Wilks, Y. (1998). Information Extraction: Beyond Document Retrieval. *Journal of Documentation*, 54(1):70–105.

Glover, A. and Hirst, G. (1996). Detecting stylistic inconsistencies in collaborative writing. In Sharples, M. and van der Geest, T., editors, *The new writing environment: Writers at work in a world of technology*, pages 147–168. London: Springer-Verlag.

Goodman, J. T. (2001). A bit of progress in language modeling. *Computer Speech and Language*, pages 403–434.

Graff, D. (2003). English Gigaword. Linguistic Data Consortium, catalog number LDC2003T05.

Graham, N. (2000). Automatic detection of authorship changes within single documents. MSc thesis, Department of Computer Science, University of Toronto.

Graham, N., Hirst, G., and Marthi, B. (2005). Segmenting documents by stylistic character. *Natural Language Engineering*, 11(3):397–415.

Grimmett, G. (2001). *Probability and Random Processes*. Oxford University Press, Oxford Oxfordshire.

Grubbs, F. (1960). Procedures for detecting outlying observations in samples. *Technometrics*, 11(1):1–21.

Guthrie, L., Basili, R., Zanzotto, F., Bontcheva, K., Cunningham, H., Guthrie, D., Cui, J., Cammisa, M., Liu, J. C.-C., Martin, C. F., Haralambiev, K., Holub, M., Macherey, K., and Jelinek, F. (2003). Semantic Analysis for Data Sparsity Compensation: Project report of the 2003 John's Hopkins summer workshop. Available on-line at http://www.clsp.jhu.edu/ws2003/groups/sparse/frptmainALL.pdf.

Guthrie, L., Walker, E., and Guthrie, J. A. (1994). Document classification by machine: theory and practice. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING-94)*, pages 1059–1063, Kyoto, JP.

Hardin, J. and Rocke, D. M. (2005). Distribution of robust distances. *Journal of Computational and Graphical Statistics*, 14(4):928–946.

Hassel, M. (2001). Internet as corpus - automatic construction of a swedish news corpus. In *Proceedings of the 13th Nordic Conference on Computational Linguistics (NODALIDA01)*, Uppsala, Sweden.

Hearst, M. A. (1997). Texttiling: segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64.

Hinkley, D. V. (1975). On power transformations to symmetry. *Biometrika*, 62(1):101–111.

Holmes, D. I. (1994). Authorship attribution. *Computers and the Humanities*, 28(2):87–10.

Holmes, D. I. and Forsyth, R. (1995). The Federalist revisited: New directions in authorship attribution. *Literary and Linguisstic Computing*, 10:111–127.

Huber, P. J. (1985). Projection pursuit. *The Annals of Statistics*, 13(2):435–475.

Hubert, M., Rousseeuw, P. J., and Branden, K. V. (2005). Robpca: a new approach to robust principal component analysis. *Technometrics*, 47:64–79.

Hubert, M. and Van der Veeken, S. (2008). Outlier detection for skewed data. *Journal of Chemometrics, in press.*

Jabbari, S., Allison, B., Guthrie, D., and Guthrie, L. (2006). Towards the Orwellian nightmare: separation of business and personal emails. In *Proceedings of 21st international Conference for Computational Linguistics and 44th Annual meeting of Association for Computational Linguistics (ACL/COLING–06)*, Sydney Australia.

Karlgren, J. (1998). Stylistic experiments for information retrieval. In Strzalkowski, T., editor, *Natural Language Information Retrieval.* Kluwer.

Karlgren, J. and Cutting, D. (1994). Recognizing text genres with simple metrics using discriminant analysis. In *Proceedings of the 15th conference on Computational linguistics (COLING94)*, pages 1071–1075, Morristown, NJ, USA. Association for Computational Linguistics.

Kenny, A. (1982). *The computation of style: An introduction to statistics for students of literature and humanities.* Pergamon Press, Oxford.

Kessler, B., Nunberg, G., and Schütze, H. (1997). Automatic detection of text genre. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL–97)*, pages 32–38.

Kilgarriff, A. (1997). Using word frequency lists to measure corpus homogeneity and similarity between corpora. In *Proceedings ACL SIGDAT workshop on very large corpora*, pages 231–245, Beijing and Hong Kong.

Kilgarriff, A. (2001). Comparing corpora. *International Journal of Corpus Linguistics*, 6(1):1–37.

Kilgarriff, A. and Rose, T. (1998). Measures for corpus similarity and homogeneity. In *Proceedings of the 3rd conference on Empirical Methods in Natural Language Processing*, pages 46–52, Granada, Spain.

Kimber, A. C. (1990). Exploratory data analysis for possibly censored data from skewed distributions. *Applied Statistics*, 39(1):21–30.

Koppel, M. and Schler, J. (2004). Authorship verification as a one-class classification problem. In *Proceedings of the twenty-first International Conference on Machine Learning (ICML 2004)*, page 62, New York, NY, USA. ACM Press.

Koppel, M., Schler, J., Argamon, S., and Messeri, E. (2006). Authorship attribution with thousands of candidate authors. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 659–660, New York, NY, USA. ACM.

Kruegel, C. and Vigna, G. (2003). Anomaly detection of web-based attacks. In *CCS '03: Proceedings of the 10th ACM conference on Computer and Communications Security*, pages 251–261, New York, NY, USA. ACM.

Lay, D. C. (2003). *Linear algebra and its applications*. Addison-Wesley, 3rd edition.

Leech, G., Garside, R., and Bryant, M. (1994). CLAWS 4: the tagging of the British National Corpus. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING 94)*, pages 622–8, Kyoto,Japan.

Lopuhaä, H. P. and Rousseeuw, P. J. (1991). Breakdown points of affine equivariant estimators of multivariate location and covariance matrices. *The Annals of Statistics*, 19(1):229–248.

Love, H. (2002). *Attributing Authorship: An Introduction*. Cambridge University Press.

Luyckx, K. and Daelemans, W. (2005). Shallow text analysis and machine learning for authorship attribution. In *Proceedings of the Fifteenth Meeting of Computational Linguistics in the Netherlands (CLIN 2004)*, pages 149–160.

Mahalanobis, P. C. (1936). On the generalized distance in statistics. *Proceedings of the National Institute of Science of India*, 2(1):49–55.

Mailloux, S. L., Johnson, M. E., Fisher, D. G., and Pettibone, T. J. (1995). How reliable is computerized assessment of readability? *Computers in Nursing*, 13(5):221–225.

Manning, C. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT press, Cambridge, MA. Supporting materials available at `http://www.sultry.arts.usyd.edu.au/fsnlp/` .

Manomaisupat, P., Vrusias, B., and Ahmad, K. (2006). Categorization of large text collections: Feature selection for unsupervised and supervised neural networks. In Corchado, E., Yin, H., Botti, V., and Fyfe, C., editors, *Proc. 7th Int. Data Engineering and Automated Learning Conf. (Lecture Notes on Computer Science - LNCS 4224)*, pages 1003–1013. Springer Berlin / Heidelberg.

Markou, M. and Sing, S. (2003). Novelty detection: a review- parts 1 and 2. *Signal Processing*, 83(12):2481–2521.

Maronna, R. A. and Yohai, V. J. (1995). The behavior of the Stahel-Donoho robust multivariate estimator. *Journal of the American Statistical Association*, 90(429):330–341.

Maronna, R. A. and Zamar, R. H. (2002). Robust estimates of location and dispersion for high-dimensional datasets. *Technometrics*, 44(4):307–317.

Maurer, H., Kappe, F., and Zaka, B. (2006). Plagiarism – A Survey. *Journal of Universal Computer Science*, 12(8):1050–1084.

Maynard, D., Tablan, V., Bontcheva, K., Cunningham, H., and Y.Wilks (2003). Multi-source entity recognition – an information extraction system for diverse text types. Research Memorandum CS–03–02, Department of Computer Science, University of Sheffield.

Maynard, D., Tablan, V., Ursu, C., Cunningham, H., and Wilks, Y. (2001). Named Entity Recognition from Diverse Text Types. In *Recent Advances in Natural Language Processing 2001 Conference*, pages 257–274, Tzigov Chark, Bulgaria.

McColly, W. and Weier, D. (1983). Literary attribution and likelihood-ratio tests: The case of the Middle English Pearl poems. *Computers and the Humanities*, 17(2):45–97.

McColly, W. B. (1987). Style and structure in the Middle English poem cleanness. *Computers and the Humanities*, 21(3):169–176.

McMenamin, G. R. (2002). *Forensic Linguistics: Advances in Forensic Stylistics*. CRC Press.

Mendenhall, T. (1887). The characteristic curves of composition. *Science*, 9:237–239.

Milic, L. T. (1967). *A quantitative approach to the style of Johnathan Swift*, volume 23 of *Studies in English literature*. The Hague: Mouton.

Milic, L. T. (1991). Progress in stylistics: Theory, statistics, computers. *Computers and the Humanities*, 25(6):393–400.

Mitchell, T. (1997). *Machine Learning*. McGraw-Hill International Editions.

Mosteller, F. and Tukey, J. W. (1977). *Data Analysis and Regression*. Addison-Wesley.

Mosteller, F. and Wallace, D. L. (1964). *Inference and Disputed Authorship: The Federalist*. Addison-Wesley Publishing Company, Inc., Reading, MA.

Oakes, M. (1998). *Statistics for Corpus Linguistics*. Edinburgh Textbooks in Empirical Linguistics, Edinburgh.

Papoulis, A. (2002). *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, New York.

Peña, D. and Prieto, F. J. (2001). Multivariate outlier detection and robust covariance matrix estimation. *Technometrics*, 43(3):286–300.

Peng, F., Schuurmans, D., Keselj, V., and Wang, S. (2003). Language independent authorship attribution using character level language models. In *Proceedings, 10th Conference of the European Chapter of the Association for Computational Linguistics*, pages 267–274, Budapest.

R Development Core Team (2008). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Rauber, A. and Müller-Kögler, A. (2001). Integrating automatic genre analysis into digital libraries. In *Proceedings of the First ACM-IEEE Joint Conf on Digital Libraries*, Roanoke, VA.

Rocke, D. M. (1996). Robustness properties of s-estimators of multivariate location and shape in high dimension. *The Annals of Statistics*, 24(3):1327–1345.

Rocke, D. M. and Woodruff, D. L. (1996). Identification of outliers in multivariate data. *Journal of the American Statistical Association*, 91(435):1047–1061.

Rousseeuw, P. J. (1984). Least median of squares regression. *Journal of the American Statistical Association*, 79(388):871–880.

Rousseeuw, P. J., Debruyne, M., Engelen, S., and Hubert, M. (2006). Robustness and outlier detection in chemometrics. *Critical Reviews in Analytical Chemistry*, 36(3&4):221–242.

Rousseeuw, P. J. and Leroy, A. M. (2003). *Robust Regression and Outlier Detection*. John Wiley & Sons.

Rousseeuw, P. J. and van Driessen, K. (1999). A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223.

Rousseeuw, P. J. and van Zomeren, B. C. (1990). Unmasking multivariate outliers and leverage points. *Journal of the American Statistical Association*, 85(411):633–639.

Sahlgren, M. and Karlgren, J. (2005). Counting lumps in word space: Density as a measure of corpus homogeneity. In *Proceedings of the 12th International Conference on String Processing and Information Retrieval (SPIRE05)*, pages 151–154, Buenos Aires, Argentine.

Santini, M. (2004). Identification of genres on the web: a multi-faceted approach. In *Proceedings of the 26th European Conference on Information Retrieval (ECIR-04)*.

Sato, S. and Sato, M. (1999). Toward automatic generation of web directories. In *Proceedings of International Symposium on Digital Libraries (ISDL99)*, pages 127–134, Tsukuba, Japan.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47.

Sebastiani, P., Gussoni, E., Kohane, I. S., and Ramoni, M. F. (2003). Statistical challenges in functional genomics. *Statistical Science*, 18(1):33–60.

Sherman, L. (1888). Some observations upon sentence-length in english prose. *University of Nebraska Studies*, 1:119–130.

Sinclair, J. and Coulthard, R. M. (1975). *Towards an Analysis of Discourse: the English Used by Teachers and Pupils*. Oxford University Press, London.

Small, C. G. (1990). A survey of multidimensional medians. *International Statistical Review / Revue Internationale de Statistique*, 58(3):263–277.

Smith, M. (1998). The authorship of Acts I and II of Pericles: A new approach using first words of speeches. *Computers and the Humanities*, 22:23–41.

Song, X., Wu, M., and Jermaine, C. (2007). Conditional anomaly detection. *IEEE Trans. on Knowl. and Data Eng.*, 19(5):631–645. Fellow-Sanjay Ranka.

Stahel, W. A. (1981). Breakdown of covariance estimators. Research Report 31, Fachgruppe für Statistik, Swiss Federal Institute of Technology (ETH), Zürich.

Stamatatos, E., Fakotakis, N., and Kokkinakis, G. (1999). Automatic authorship attribution. In *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics*, pages 158–164, Bergen.

Stephens, C. (2006). All about readability. Plain Language Network. http://www.plainlanguagenetwork.org/stephens/readability.html.

Stone, P. J., Dunphy, D. C., Smith, M. S., and Ogilvie, D. M. (1966). *The General Inquirer: A Computer Approach to Content Analysis: Studies in Psychology, Sociology, Anthropology & Political Science*. MIT press.

Struyf, A. and Rousseeuw, P. J. (2000). High-dimensional computation of the deepest location. *Computational Statistics and Data Analysis*, 34(4):415–426.

Tetlock, P. C. (2007). Giving content to investor sentiment. *Journal of Finance*, 62(3):1139–1168.

Tukey, J. W. (1977). *Exploratory Data Analysis*. Addison-Wesley.

Vandervieren, E. and Hubert, M. (2004). An adjusted boxplot for skewed distributions. In Antoch, J., editor, *Proceedings of Computational Statistics, 2004*, pages 1933–1940. Springer-Verlag, Heidelberg.

Wall, L., Christiansen, T., and Orwant, J. (2000). *Programming Perl, Third Edition*. OŔeilly & Assiciates, Sebastopol, CA.

Ward, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244.

Wayner, P. (2000). *Disappearing Cryptography- Information Hiding: Steganography and Watermarking*. Morgan Kaufmann, second edition.

Werner, M. (2003). *Identification of Multivariate Outliers in Large Data Sets*. PhD thesis, University of Colorado at Denver.

Wilks, Y. (2004). On the ownership of text. *Computers and the Humanities*, 38(2):115–127.

Willett, P. (1988). Recent trends in hierarchic document clustering: a critical review. *Information Processing and Management*, 24(5):577–597.

Williams, C. (1975). Mendenhall's studies of word-length distribution in the works of shakespeare and bacon. *Biometrika*, 62(1):207–212.

Woolls, D. and Coulthard, R. M. (1998). Tools for the trade. *Forensic Linguistics: The International Journal of Speech, Language and the Law*, 5(1):33–57.

Yang, Y. (1998). An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval*, 1:67–88.

Appendix A

Clustering Text

While clustering is not appropriate for anomaly detection in an unsupervised scenario (see Sections 1.3 and 3.4), because its usefulness depends on the assumption that anomalies are more similar to each other than to *normal* segments, it is nonetheless useful for visualizing the impact of features in controlled experiments. Clustering has been used successfully in the closely related problem of genre identification using stylistic features [Rauber and Müller-Kögler, 2001; Clough, 2000] and thus we thought it worthwhile to investigate it.

This appendix shows some of the initial experiments we conducted analyzing the usefulness and suitability of the all features described in chapter 4 by letting a clustering algorithm attempt to group segments with similar features. These small-scale experiments gave us insight into which features were the most beneficial for characterizing text at the segment level.

For each of these tests we analyzed segments of text (either paragraphs or documents) using of all the stylistic features described in 4.2 – 4.7 above as well as the

largest 148 categories from the General Inquirer dictionary.

A.1 Multiple Authors

In this experiment we take segment to mean an entire document and compute all features over all documents. We then use SPSS statistical package to perform the clustering. We used 21 texts by 11 different authors. We made use of the full texts obtained from project Gutenberg. The texts used are shown in table A.1.

Author	**Texts**
Louisa May Alcott	Eight Cousins Little Women
Jane Austin	Pride and Prejudice Emma Sense and Sensibility Persuasion
Forefathers (Jefferson et al.)	Declaration of Independence
Homer	Odyssey
Washington Irving	Crayon Papers Legend of Sleepy Hollow
Franz Kafka	Metamorphosis
Karl Marx	Communist Manifesto
Friedrich Nietzsche	Behond Good and Evil Thus Spake Zarathustra
Plato	Republic
Jonathan Swift	Gulliver's Travels Drapier's Letters Modest Proposal Tail of a Tub
Mark Twain	The Adventures of Tom Sawyer The Adventures of Huckleberry Finn

Table A.1: The authors and texts used for clustering analysis

All of the values for our features were normalized to z-score (described in sec-

tion 5.3) so they would all receive equal weighting regardless of their magnitude.

We used Ward's method of clustering Ward [1963] which starts by treating each observation as a cluster and then iteratively looks to merge the clusters that (when combined) will have the smallest squared distances from their mean. To determine the clusters to combine at every step, a distance based on this residual is calculated and the clusters that have the smallest distance are merged into one cluster. This process iterates until there in only one cluster left. The distance formula for determining whether to merge two clusters, $\mathbf{x}$ and $\mathbf{y}$, into a cluster $\mathbf{xy}$ is given by:

$$dist(\mathbf{x}, \mathbf{y}) = r(\mathbf{xy}) - (r(\mathbf{x}) + r(\mathbf{y}))$$

where r is the function to compute squared residuals of a cluster, given by:

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

where n is the number of observations in the cluster $\mathbf{x}$ and $\bar{\mathbf{x}}$ is the mean of cluster $\mathbf{x}$. This mean is computed in the standard way ($\frac{1}{n}\sum_{i=1}^{n} x$).

The results of this clustering is shown in figure A.1. This figure shows that using Wards clustering with these features is not perfect (some clusters include books by two different authors) yet nonetheless it is a promising result.

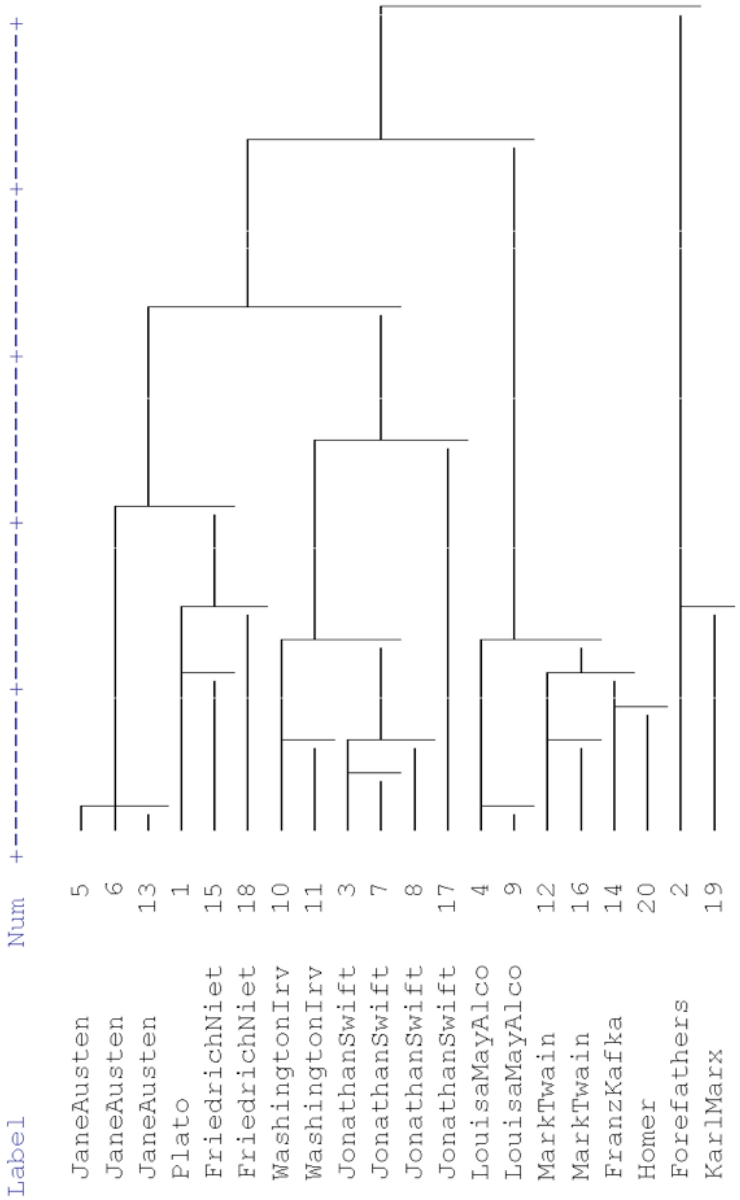


Figure A.1: Hierarchical clustering of 21 texts written by 11 authors

A.2 Different Genres

For this simple exploratory experiment we randomly chose 15 small segments from the from the Gigaword newswire corpus (ranging in size from 40 to 60 words) and inserted a random 30 word segment from the Medline corpus (not breaking across sentence boundaries). We performed the same clustering procedure as in the previous experiment. Figure A.2 shows that the result of this clustering is a clear separation between the two genres.

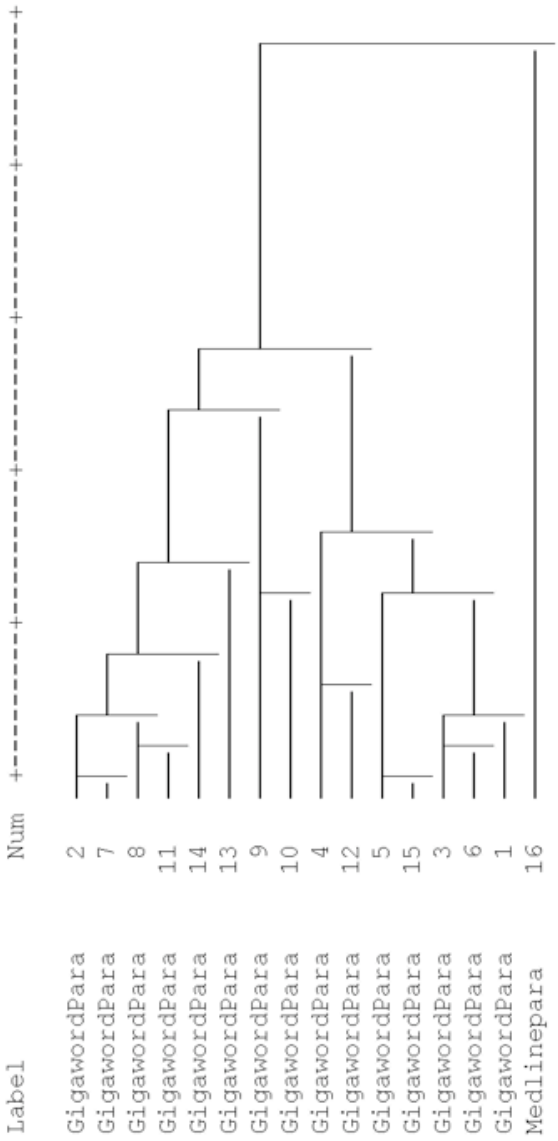


Figure A.2: Hierarchical clustering of Gigaword paragraphs with a Medline paragraph

A.3 Journal Articles

Articles from the International Journal of Corpus Linguistics were analyzed by paragraph to determine if it is possible to distinguish authors' writing at the paragraph level. We chose two articles on a similar topic from a single issue of the publication. These articles' genres are identical and both were on similar topics so this experiment gives good gauge of which features distinguish authors based primarily on their writing style. We randomly chose five paragraphs from each journal article and performed the automatic clustering as above. The results in Figure A.3 show nearly perfect grouping of the paragraphs by the same authors with the exception of one paragraph in article #2 that seems to be very different from all other articles.

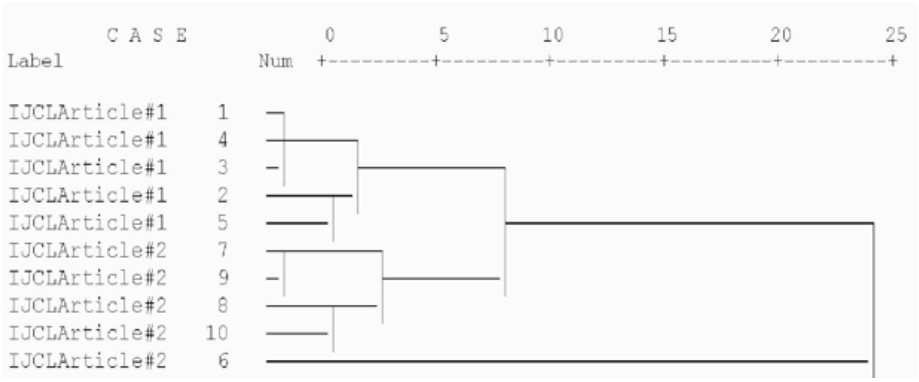


Figure A.3: Hierarchical clustering of paragraphs from articles in the International Journal of Corpus Linguistics

We also used the two journal articles described above to compute the *information gain* (see Mitchell [1997]) between the features of their observations using the WEKA machine learning toolkit. This gives us an idea of which of our features would be most useful for classifying these two journal articles to their appropriate author. Table A.2 shows the top ranked features.

Score	**Attribute**	**Examples/Description**
0.2706	Hostile	criticism, inhibit, avoid, argument
0.2245	Wordlen	Average word length
0.2218	Object	table, corpus, marker, text
0.2216	Overstated	very, every, quite, large
0.2199	Coleman-Liau	Readability Index
0.2124	DescriptiveVerbs	do, use, increase, play
0.1867	Perc3syll	Percentage of 3 or more syllable words
0.1615	Political	capitol, candidate, country
0.1612	CommonObject	mark, list, report
0.1606	Work	drive, use, done, produce
0.1562	Perc1syll	Percentage of 1 syllable words
0.1492	NegativeOutlook	complex, difficult, lack, decrease
0.1481	Academic	statistical, experiment, scientific
0.1459	NegativeH4	uncomfortable, hard, inadequate
0.1448	Fleschease	Readability measure
0.1354	SmogGraiding	Readability measure
0.1245	Communicate	thank, joke, praise

Table A.2: Feature Ranking for Clustering Experiments

A.4 Conclusion for Clustering Experiments

Clustering using z-scores and Ward's clustering method proved useful for visually testing features and showed us that grouping instances based on their feature's standard deviation from the mean has very genuine benefits. It is also clear from our analysis of these features that some contribute much more than others. This steered us to develop methods of anomaly detection that would use this information effectively.

Appendix B

Corpora

The selection of suitable corpora for the study of anomaly detection depends upon the precise definition of anomaly. For example, a document might be anomalous because it is a different genre, a different topic, or a different style of writing from the other texts in the collection. These corpora allow models of language to be defined, and techniques to be evaluated for detecting anomalous documents or segments. In this Appendix we describe all corpora used in this thesis and give examples of each.

B.1 English Gigaword

The Gigaword English Corpus is a large archive of newswire text data acquired by the Linguistic Data Consortium. The total corpus consists of over 1.7 billion words from four distinct international sources of English newswire ranging from approximately 1994-2002:

- Agence France Press English Service (afe) 1994-2002

- Associated Press Worldstream English Service (apw) 1994-2002
- The New York Times Newswire Service (nyt) 1994-2002
- The Xinhua News Agency English Service (xie) 1995-2001

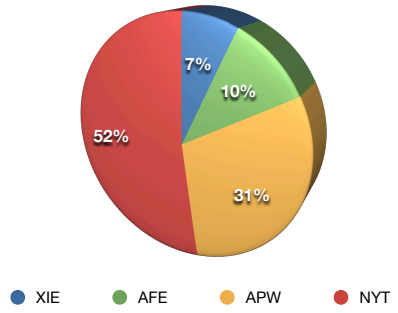


Figure B.1: Gigaword Corpus Distribution

A sample of the Gigaword corpus follows:

```
<DOC id="XIE20011101.0001" type="story" >
<HEADLINE>
Food Assistance Expedited to Ease Starvation in
Afghanistan: WFP Spokesman
</HEADLINE>
<DATELINE>
PESHAWAR (Pakistan), October 31 (Xinhua)
</DATELINE>
<TEXT>
<P>
As the freezing winter is in the offing in Afghanistan, the World Food
Programme (WFP) has picked up food assistance to hundreds of thousand
Afghans trapped in hunger and warfare in an all-out effort to avoid
famine which could trigger a humanitarian disaster.
</P>
<P>
During an interview made here on Wednesday, WFP spokesman Huggins told
Xinhua that 1,535 metric tonnes of food was leaving for afghanistan's
central highland (CHL) where a estimated population of half a million
Afghans remain in serious malnutrition for lack of daily feeding,
counting a total amount of 13,000 tonnes that have been sent in the
last 10 days from Peshawar.
</P>
```

B.2 Medline

Medline is a bibliographic database covering the fields of medicine, nursing, dentistry, veterinary medicine, the health care system, and the preclinical sciences. Medline contains bibliographic citations and author abstracts from more than 7,300 biomedical journals published in the United States and 70 other countries from 1950 to the present. In all, it comes to more than 1.2 billion words of English text. A sample abstract follows (omitting the metadata):

```
     AIM: Model of End-stage Liver Disease (MELD) score has recently
gained wide acceptance over the old Child-Pugh score in predicting
survival in patients with decompensated cirrhosis, although it is
more sophisticated.  We compared the predictive values of MELD,
Child-Pugh and creatinine-modified Child-Pugh scores in
decompensated cirrhosis.
     METHODS: A cohort of 102 patients with decompensated cirrhosis
followed-up for a median of 6 mo was studied.  Two types of modified
Child-Pugh scores estimated by adding 0-4 points to the original score
using creatinine levels as a sixth categorical variable were evaluated.
     RESULTS: The areas under the receiver operating charac-teristic
curves did not differ significantly among the four scores, but none had
excellent diagnostic accuracy (areas: 0.71-0.79).  Child-Pugh score
appeared to be the worst, while the accuracy of MELD was almost
identical with that of modified Child-Pugh in predicting short-term and
slightly better in predicting medium-term survival.
In Cox regression analysis, all four scores were significantly associated
with survival, while MELD and creatinine-modified Child-Pugh scores had
better predictive values (c-statistics: 0.73 and 0.69-0.70) than
Child-Pugh score (c-statistics: 0.65).  Adjustment for gamma-glutamate
transpeptidase levels increased the predictive values of all systems
(c-statistics: 0.77-0.81).  Analysis of the expected and observed survival
curves in patients subgroups according to their prognosis showed that all
models fit the data reasonably well with MELD probably discriminating
better the subgroups with worse prognosis.
     CONCLUSION: MELD compared to the old Child-Pugh and particularly to
creatinine-modified Child-Pugh scores does not appear to offer a clear
advantage in predicting survival in patients with decompensated cirrhosis
in daily clinical practice.
```

B.3 The Anarchist Cookbook

The Anarchist Cookbook is a set of recipes and instructions for small-scale acts of terrorism, originally written in 1969 by William Powell (often referred to by his

pseudonym, the Jolly Roger). Later editions (with additions from other authors) have expanded, encompassing more modern themes like computer hacking techniques and identity fraud. A sample recipe from the cookbook follows:

```
Making Plastic Explosives from Bleach        by The Jolly Roger
Potassium chlorate is an extremely volatile explosive compound, and has been
used in the past as the main explosive filler in grenades, land mines, and
mortar rounds by such countries as France and Germany.  Common household
bleach contains a small amount of potassium chlorate, which can be extracted
by the procedure that follows.
First off, you must obtain:
[1]        A heat source (hot plate, stove, etc.)
[2]        A hydrometer, or battery hydrometer
[3]        A large Pyrex, or enameled steel container (to weigh chemicals)
[4]        Potassium chloride (sold as a salt substitute at health and nutrition
stores)
Take one gallon of bleach, place it in the container, and begin heating it.  While
this solution heats, weigh out 63 grams of potassium chloride and add this to the
bleach being heated.  Constantly check the solution being heated with the hydrometer,
and boil until you get a reading of 1.3.  If using a battery hydrometer, boil until
you read a FULL charge.
Take the solution and allow it to cool in a refrigerator until it is between room
temperature and 0 degrees Celcius.  Filter out the crystals that have formed and
save them.  Boil this solution again and cool as before.  Filter and save the
crystals.
Take the crystals that have been saved, and mix them with distilled water in the
following proportions:  56 grams per 100 milliliters distilled water.  Heat this
solution until it boils and allow to cool.  Filter the solution and save the crystals
that form upon cooling.  This process of purification is called "fractional
crystalization".  These crystals should be relatively pure potassium chlorate.
Powder these to the consistency of face powder, and heat gently to drive off all
moisture.
Now, melt five parts Vaseline with five parts wax.  Dissolve this in white gasoline
(camp stove gasoline), and pour this liquid on 90 parts potassium chlorate (the
powdered crystals from above) into a plastic bowl.  Knead this liquid into the
potassium chlorate until intimately mixed.  Allow all gasoline to evaporate.
Finally, place this explosive into a cool, dry place.  Avoid friction, sulfur,
sulfides, and phosphorous compounds.  This explosive is best molded to the desired
shape and density of 1.3 grams in a cube and dipped in wax until water proof.  These
block type charges guarantee the highest detonation velocity.  Also, a blasting cap
of at least a 3 grade must be used.
The presence of the afore mentioned compounds (sulfur, sulfides, etc.) results in
mixtures that are or can become highly sensitive and will possibly decompose
explosively while in storage.  You should never store homemade explosives, and you
must use EXTREME caution at all times while performing the processes in this article.
```

B.4 Google Translations (Chinese to English machine translations)

Seven different Chinese newspaper articles of approximately 500 words each were chosen and run through the Google automatic translation engine to produce English texts. Web translation engines are known for their inaccuracy and ability to generate extremely odd phrases that are often very different from text written by a native speaker. The intention was to produce highly unusual texts, where meaning is approximately retained but coherence can be minimal.

A sample translation follows:

```
BBC Chinese net news:  CIA Bureau Chief Gauss told USA the senator, the card you reaches
still is attempting to avoid the American information authority, implemented the attack
to the American native place goal.  Gauss said, the card you will reach or if have the
relation other terrorist organizations sooner or later must use the biochemistry or
the nuclear weapon attack USA, this possibly only will be the time question.  But he
said, the card you reach only only are a holy war organization more widespread threat
on the one hand.  He said, in Iraq, the radical member grips the card dimension  is
using conflicts carries on the scope is more  widespread, cross national boundary terror
activity.  American Federal Bureau of Investigation bureau chief said, at present his
organization urgent matter is copes with conceals you reaches in USA'S card organizes
the member.  He said, at present in the jail and the radical church, many Muslim religion's
person is regarded as the object which the radical organization recruits.
```

Appendix C

Feature Abbreviations

Abbreviation	Description
ani	References to animals, fish, birds, and insects, including their collectivities
aquatic	References to water, including things that hold water. (e.g. beaker, steam, gulf)
ari	Automated Readability Index (see Section 4.3)
artlist	Rank list feature of the Articles
avgsenlength	Average Sentence Length
bldgpt	References to buildings, rooms, and parts of buildings
bodypt	References to parts of the body
coleman	Coleman-Liau (see Section 4.3)
color	References to color words

decr	Words that describe a decreasing change (e.g. abate, subside, cheapen)
dist	Words referring to distance and its measures
econ	Words of an economic, commercial, industrial, or business orientation
exch	Words concerned with buying, selling and trading
exert	Words concerned with bearing force or influence
exprs	Words associated with the arts, sports, and self-expression
fail	Words indicating that goals have not been achieved
fall	Words concerned with downward movement (e.g. fall, collapse, dive)
feel	Words describing particular feelings (e.g. gratitude, apathy)
female	words referring to women and social roles associated with women
fetch	Words concerned with fetching or carrying
fleschease	Flesch-Kincaid Reading Ease (see Section 4.3)
fleschgrade	Flesch-Kincaid Grade Level (see Section 4.3)
fogindex	Gunning-Fog Index (see Section 4.3)
food	Words concerning food
gig*N*k	Percentage of words occurring in top N thousand words in the Gigaword Corpus
goal	Names of end-states towards which muscular or mental striving is directed

hostile	Words indicating an attitude or concern with hostility or aggressiveness
intrj	Interjections and exclaimations
kin	Terms denoting kinship
land	Words for places occurring in nature
lix	Lix Formula (see Section 4.3)
male	Words referring to men and social roles associated with men
nonadlt	Words associated with infants through adolescents
nouns	Percentage of words that are nouns
ord	Percentage of words that are ordinal numbers
passivesen	Percentage of sentences that are passive
per6orMoreLetter	Percentage of words that have six or more letters
perc1syll	Percentage of words that are only one syllable
perc3syll	Percentage of words that have 3 or more syllables
perlongsen	Percentage of sentences greater than 15 words
pershortsen	Percentage of sentences greater than 8 words
place	References to places
pleasur	Words indicating the enjoyment of a feeling, including words indicating confidence, and interest
polit	Words having a clear political character, including political roles, collectivities, and acts
pos	Words for position
posBIlist	Rank list feature of the part-of-speech bi-grams

posTRIlist	Rank list feature of the part-of-speech tri-grams
pronouns	Percentage of words that are pronouns
punct	Percentage of all characters that are punctuation
race	Words referring to racial or ethnic characteristics
rise	Words concerned with upward movement (e.g. climb, fly, jump)
ritual	Words for non-work social rituals
route	Words concerned with the route between places
self	Pronouns referring to ones singular self
semi	Percentage of all characters that are semicolons
sky	Words for all aerial conditions, natural vapors and objects in outer space
smog	SMOG Index (see Section 4.3)
stay	Words concerned with no movement (e.g. await, lie, adhere)
strng	Words implying strength
sv	State verbs describing mental or emotional states (e.g. love, trust)
think	Words referring to the presence or absence of rational thought processes
travel	Words for all physical movement and travel from one place to another in a horizontal plane
vehicle	Words concerned with vehicle objects
yes	Words directly indicating agreement

you	Pronouns indicating another person is being addressed directly

Table C.1: Key to feature abbreviations used in Chapter 7