

Chapter 4

Outlier Detection

4.1 Introduction

Outliers are data points that are particularly inconsistent with the rest of the data (e.g. Barnett and Lewis [6]). While concise, this statement is hardly concrete: how inconsistent is an outlier, and how can we measure this inconsistency? Furthermore, how many points might we regard as outliers, and how many as “the rest of the data”? Suppose we have a tight group of 100 points, and a second group of 5 points, much further away. We might reasonably treat the latter as outliers. But if the second group had 50 points, should we be so cavalier? What if it had 500 points? At what point should we say that the first group consists of outliers, or that neither group consists of outliers?

Outliers are often defined as surprising points in relation to the rest of a data set (e.g. Ripley [105, p. 24]). This subjective definition is in contrast to areas such as feature selection, regression and segmentation, where we are attempting to find objective truths, even if we have insufficient data to succeed. This makes validation of outlier detection routines particularly troublesome: if we can’t objectively specify which points are outliers, then how can we measure how effectively our approach is finding them?

Outliers cause problems in both regression and clustering because in each case, we estimate model parameters from the data, with the goal of modelling the underlying system. Outliers are not from this underlying distribution, and so will cause us to mis-estimate the parameters. In this chapter, we focus on detecting outliers in cluster models of preference panel data. This is because we are more concerned about removing outlying consumers from the preference panels, rather than removing any products. We have very few products to start with, and these were chosen carefully by the experimenters, whereas the preference panellists were chosen at random, and there are many more of them.

Below, we discuss some sources of outliers in preference data sets, and consider what can be done about outliers in general. In Section 4.2, we discuss how various data analysis techniques deal (or fail to deal) with extreme points. We followed this in Section 4.3 with a discussion of how the accuracy of outlier detecting routines can be measured; in Section 4.4, we give a comparison of accommodation and discordancy. In Section 4.5, we turn to the specifics of detecting outliers in multivariate, structured data, and how we can combine clustering with outlier detection in preference data. We then present a novel outlier rejection approach in Section 4.6, with an experimental comparison to other methods, and include discussion of how to determine the number of outliers present in a data set. We show that this non-parametric method is more effective than several standard parametric approaches.

4.1.1 Outliers in food preference data

Gathering consumer preference data has several particular problems that are inevitable consequences of using people as instruments, including:

Taste blindness. Sensory abilities vary between individuals, according to the density of taste buds on the tongue [81]. Any randomly selected group is likely to contain some people who are poor judges of quality.

Mis-understanding the nature of the task. Although the taste tests are relatively simple, and the requirements are clearly explained, it is likely that at least some panellists will not understand what is required of them, and so will give poor responses.

Personal inconsistency. Preference tests typically take place over two days, and people's preferences are influenced by boredom, tiredness, and so on. So even an individual who has refined sensory abilities and understands the test may still give inconsistent responses.

Numerical biases. When answering scale questions, some people tend to give extreme answers (e.g. all ones or nines), while others tend to give moderate answers (e.g. all fours or fives).

Product variation. However carefully each sample is prepared and presented, there will inevitably be some variation, which could lead to different responses.

Data recording. Preference panellists are asked typically to fill in forms expressing their opinions. Depending on the design of the form, panellists may inadvertently tick the wrong box, or mark a line incorrectly. Further, when these forms are being collated, data-entry errors may occur.

Moving from the specifically food-oriented to a more general view of outliers, Barnett and Lewis [6, p.7] describe two sources of outliers: *contamination* and *extreme values*. The first of these occurs when the data set has been generated from not one but *two* distributions, one of which produces outliers. The second source is data that are drawn from a single distribution, but with some points at an extreme. These points are unlikely to occur (by definition), but they will appear occasionally. This suggests that the concept of an outlier depends partly on the model that we assume generated the data, and partly on our own expectations and requirements.

4.1.2 Actions on outliers

The utility of any action depends on what is planned next. When performing outlier detection, we must decide on the best course of action in light of how we plan to analyse the data further. Barnett and Lewis define four broad approaches to dealing with outliers [6, p.28]:

Reject. Remove outliers from the data set entirely. This has the advantage that we reduce the misleading effect of outliers, but we may be throwing away useful information. This is typically done using a *discordancy* test, described in Section 4.4.2.

Include. Keep all the data, including any outliers. This way, we guarantee that we will not lose any useful information, but we may be misled by outliers.

Accommodate. Keep all the data, but use robust methods to avoid being led astray by outliers, such as using estimates of the median or the truncated mean, rather than the global mean, of a sample.

Identify. The outliers may be “phenomena of interest” (Wu et al. [136]) in their own right, such as an exceptionally popular food, or an exceptionally active chemical. We might then wish to separate noisy outliers from interesting outliers.

In Section 4.4, we return to these options, and discuss which are most appropriate when modelling food preference data.

4.2 Influential points

It may be instructive to regard outliers as *influential points*, i.e. some subset of data that has a disproportionate effect on any model built from the data. The nature of this emphasis may vary, according to the techniques we are applying.

Least squares modelling. Many regression and clustering models are built by minimising the sum of the squares of the residual error of each data point (e.g. linear regression, Section 2.2.1 and k -means clustering, Section 3.4.4). Because the error is squared, greater influence is given to extreme points, especially when only a small sample is available.

Boosting. Boosting builds an ensemble of models using weighted data points (Section 2.5.3). In each iteration, the most erroneous points are given greater influence in subsequent models. Without extra checks, outliers can seriously distort boosted ensembles, as shown by Rätsch et al. [102].

Support vector machines. A (typically small) subset of data is used to define the decision hyperplanes or regression lines [132]. These “support vectors” are the only points to have direct influence over the model produced. The remaining data is only used indirectly, to determine which points are support vectors. In classification models, the support vectors are those points closest to the boundaries between classes. Because they are on the edge of a class, the very points that define the model are potential outliers.

Outlier detection. The most extreme points are labelled as outliers, and removed, accommodated, or otherwise dealt with.

Given univariate data, the notion of an extreme point is easy to define: it is the point with the greatest magnitude. In the multivariate case, there is no such clear definition: a point need not be extreme in any particular dimension to be an outlier. *Sub-ordering* is the name given to techniques that map multivariate data onto univariate scale, so that extreme points can be identified easily [6, p.270]. The most obvious example of a sub-ordering function is a distance measure, such as the Euclidean distance to the origin or the sample mean.

4.3 Measuring the accuracy of outlier detectors

As noted in Section 4.1, there is a degree of subjectivity in deciding whether a particular point is an outlier or not. Therefore, measuring the accuracy of an outlier detecting algorithm is problematic, as is comparing the accuracy of two such algorithms. As with most empirical data sets, preference data undoubtedly contains outliers but we have no objective means of identifying them.

One possibility to guide research is to create artificial data sets, so we can control the distribution of the data and the distribution of the outliers. But by doing so, we are shifting the subjectivity from *identifying* the outliers to *creating* them. For

example, suppose we create a set of inliers¹ by sampling from a Gaussian distribution, and then we add some outliers sampled from a uniform distribution within a similar range. Some of the “outliers” will probably appear in the middle of the Gaussian inliers, so would never be recognised as outliers. Conversely, some of the points from the Gaussian could be arbitrarily far away. The former are contamination, and the latter are extreme values. We cannot expect any algorithm to correctly classify each point according to the distribution which generated it, nor should we necessarily try to do so. There is simply insufficient information in such data sets to allow automatic identification of outliers.

An alternative method of creating an artificial data set is to draw the entire sample from a single Gaussian distribution, and then label everything with a likelihood less than some threshold as being an outlier. The outlier-detecting algorithm must then estimate this threshold, which we have arbitrarily chosen, without being given any labels on the data identifying the outliers, which is clearly a difficult test.

Given that both of these methods are far from ideal, should we use artificial data sets at all? We want to be able measure the accuracy with which outliers are detected, and we have no objective means to identify outliers within the preference data sets, so we must use artificial data sets. In subsequent sections, we will describe experiments using each of these techniques for generating data, but we should not expect any outlier detection method to achieve anything approaching 100% accuracy, for the reasons given above. More precisely, we would expect the outlier detection systems to *underestimate* the number of outliers present, as many outliers will be close to the centre(s) of the inlier distribution.

4.4 Accommodation and rejection

Four actions on outliers were listed earlier: rejection, inclusion, accommodation and identification. In the current work, identification is of little interest: the outliers are consumers on the preference panel who gave results inconsistent with the rest of the panel, so either gave faulty data or have unique tastes. In either case, we need not try to model their preferences nor to design food to satisfy them. Similarly, we don't want to simply include the outliers and distort the model, because we want models that will generalise accurately to predict (most of the) consumers' preferences. Thus there are two approaches to be considered further: we can either build a model which is “outlier-aware” and accommodates them, or identify and reject outliers. We consider these two approaches next.

¹An inlier is any data point that is not an outlier.

4.4.1 Accommodation

Conventional regression and classification approaches assume that the data are *i.i.d.* (independently and identically distributed). If this assumption holds, we can use a sample of the data to build a model, and the remainder of the data set (drawn from the same distribution) should fit the model reasonably well, as should future, as yet unseen, samples (c.f. cross validation, Section 2.5.1, p. 51).

For data sets containing outliers, it is often more useful to assume that the data are sampled from two distinct distributions, the inliers D_{in} and the outliers D_{out} . I.e. the data set is drawn from $D_{in} \cup D_{out}$. Initially, both these distributions are unknown, including their prior probabilities. We can then recast the problem as a binary classification task, where a classifier is trained to label each point as an inlier or an outlier. Unfortunately, all the records are unlabelled, because we don't know in advance which points belong to which class. As a separate and subsequent task, we may attempt to estimate D_{in} as the model of interest, and ignore both the nature of D_{out} and the data points that were generated by D_{out} .

Several suggestions have been made for relating D_{in} and D_{out} including:

Mean Slippage. This assumes that the outlier distribution is identical to the inlier distribution, but with a shifted mean. E.g. if $D_{in} = \beta X + \epsilon$, then $D_{out} = \beta X + \delta + \epsilon$, with $\delta \neq 0$ being the degree of slippage [67].

Variance Inflation. This assumes that the variance of the outliers is greater than the variance of the inliers. E.g. if $D_{in} = \beta X + \epsilon_i$, with $\epsilon_i \sim N(0, \sigma^2)$, then $D_{out} = \beta X + \epsilon_o$, with $\epsilon_o \sim N(0, K^2 \sigma^2)$, and $K > 1$. Here, K defines the amount of variance inflation [67].²

Mixed Alternative. This assumes that the data are drawn from a combination of two distributions, with a fixed probability: $x \in (1 - \pi)D_{in} + \pi D_{out}$. Here, π is the probability that a point is an outlier [23]. Of course, this requires some method to estimate π , as well as the two distributions.

Exchangeable Alternative. This assumes a family of models, each of which identifies a distinct subset of data points as being from an outlier distribution. A likelihood score can then be calculated for each alternative, and the maximum likelihood used to select the correct interpretation. This can be regarded as a Bayesian analysis of the "mean slippage" alternative described above [6, p.51].

Deterministic Alternative. In some cases, an outlier may have a clear external cause, such as measurement error, recording error, etc. In these cases, no statisti-

²One could imagine outliers caused by variance *deflation*, where $K < 1$, but we will ignore this unlikely situation.

cal testing is required, and the faulty data points can be removed without further analysis [6, p.45].

These can all be regarded as alternatives to the null hypothesis that all the data is drawn from a single distribution. “Mean Slippage” and “Variance Inflation” are special cases of “Mixed Alternative”.

Hoeting et al. [67] use the “variance-inflation” model to characterise outliers. The probability that each data point is drawn from the first (inlier) distribution is $1 - \pi$, with a probability of π of being an outlier. They then use a Bayesian approach to calculate the posterior model probability of a variety of models, each generated using a different combination of features and data points. The different sets of features and data points are generated using a Markov Chain Monte Carlo Model Composition (MC^3) approach. MC^3 performs a random walk through the model space, and is similar to simulated annealing (Section 2.4.3). Rather than choosing the best single model, Bayesian model averaging is then performed. Although the paper shows positive results on a number of small data sets, this approach is inappropriate for food design: we have too many preference records to rely on MC^3 to find the outliers³. Also, we want a single set of drivers (product features), rather than using an average model, to ease interpretation.

Figure 4.1 shows two hypothetical distributions based on the variance inflation assumption, with $K = 3$ defining the degree of inflation. Many points generated from the outlier distribution (dotted line) will lie near the mean of both distributions, and so would appear to be generated by the inlier distribution (solid line). Conversely, many points from the inlier distribution will appear arbitrarily far from the distribution mean. Given that we don’t know which distribution generated which data point, it makes little sense to describe points within these strongly overlapping regions as outliers, even if they are “contamination”.

In preference data sets, we have no a priori reason for assuming that the outliers are drawn from any particular distribution, and no way of determining what such a distribution would be. We cannot model any form of data *contamination*, and instead, we are forced to consider points as outliers only if they are *extreme* points. Accommodation methods are therefore unsuitable for our purposes, and we now turn to discordancy methods.

4.4.2 Discordancy

Rather than accommodating outliers within the model, we would rather identify and remove the outliers, before building a final model. A discordant data point is one that is statistically unlikely to have been generated from the same distribution as the rest

³Hoeting et al. describe results with 21 and 54 records; the beverage preference set P_b used here as 450 records.

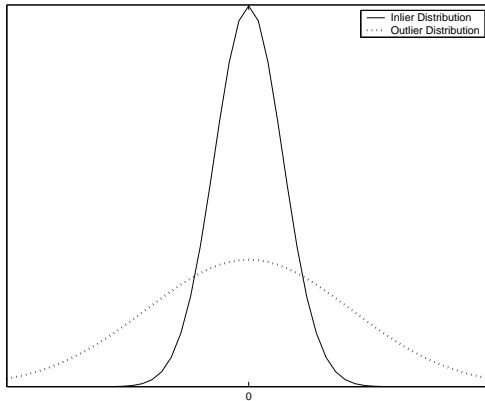


Figure 4.1: Variance inflation: inlier and outlier distributions

of the data. Numerous tests have been proposed (e.g. [6, p.94–108]), but many take the form $t = N/D$, where N measures the separation of the test point from the rest of the data, and D measures the spread of the sample. For example, to test if x_n is a discordant outlier, we might test $t = (x_n - \bar{x})/\sigma$, where \bar{x} and σ are the mean and standard deviation of the sample respectively. All of these techniques implicitly assume that the training data has no outliers⁴, while the test data may include some.

We need to know the distribution of such a test in order to calculate significance levels. Distributions commonly used in discordancy analysis are normal or gamma distributions; others have been calculated analytically or estimated empirically (e.g. using Monte Carlo methods) [6].

Discordancy tests are typically applied by firstly, identifying possible outliers, and then secondly, calculating the test statistic and its significance. This leads to problems with *masking*, where two outliers are close to each other, making each one appear to be an inlier; and *swamping*, where we avoid masking by testing for two (or more) outliers simultaneously, but in reality, only one is an outlier [23]. In either case, discordancy tests may be misled.

Many approaches to outlier detection use all of the available data to build a model, and then use that model to detect (and often to reject) outliers. For example, Tax and Duin [125] describe fitting a mixture of Gaussians to some data, and then rejecting as outliers all points whose probability of being generated by the model is less than some threshold. One drawback with this approach is that if the number of outliers in the data set is significant, then the model produced by the data may well support the outliers at least moderately well, leading to them *not* being detected at all. Extreme outliers will lead to an over-estimation of the standard deviation of the distribution. Hawkins [64]

⁴Or at least that the models are not significantly affected by any outliers present in the training data.

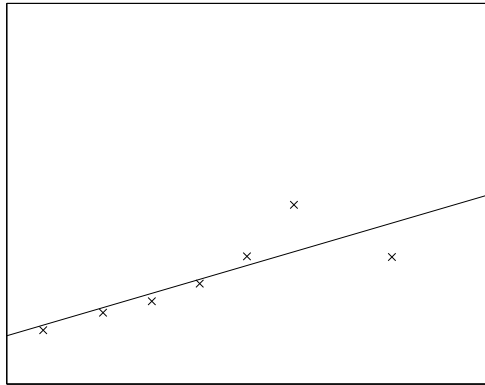


Figure 4.2: Artificial regression data without outliers. Figure modified from [15, p.210].

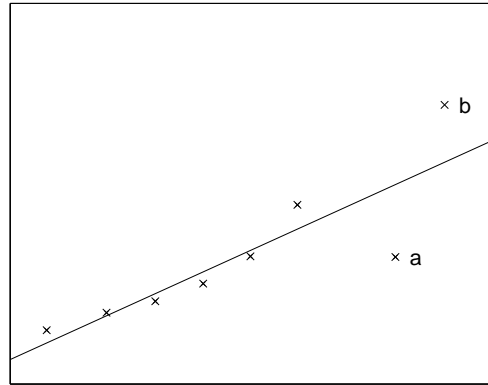


Figure 4.3: Artificial regression data with one outlier. Figure modified from [15, p.210]. Same data set 4.2, with outlier 'b' added. Point 'a' now has the largest residual error.

points out that although a normal distribution accurately approximates many other distributions across most of the values, the tails are shorter than (for example) the Student t distribution. We are interested in outliers that occur in the tails, so:

Normal approximation which is excellent for most purposes can . . . be very poor for purposes of outlier testing. [64, p.40]

Bishop [15, p.209] gives an example of how a few outliers with large errors can dominate a solution, leading to an inaccurate model. Figure 4.2 shows a linear model fitting seven data points quite well, according to least-squares optimisation. The addition of an outlying eighth point (labelled 'b') in Figure 4.3 leads to a different model. The data point fitted least well by the new model is 'a', even though this point is quite well modelled by the original line. If we treat 'a' as an outlier, remove it, and then re-estimate the model, the line will move further towards 'b', making us even less likely to regard it as an outlier. This sort of "deceptive" data set will tend to mislead many outlier detection routines.

Tax and Duin [125] propose a method based on the instability of a simple linear classifier to detect outliers, and compare this with several other techniques. The approach is to repeatedly draw bootstrap samples from the data, build a linear classifier, and calculate the predicted class for each data point in the test set. A large variation in the classes predicted by the classifiers for a point suggests that that point is an outlier, in that its class cannot be consistently predicted from the training set. The results reported in the paper show that this instability technique outperforms other techniques only if the training data set is quite small. For larger sets, the bootstrap samples show

less variance, so the linear classifiers are more similar, and the variance on the test set is reduced.

Such a classifier instability test is likely to identify as outliers points that are close to the decision boundary between two classes. It is these points that will switch from one side to the other even if the estimated boundary moves only slightly. Points that are a long way from the decision boundary will always be classified the same way, and so never be detected as outliers, however far away they are from the rest of the data. Furthermore, as the current work is focussed on clustering rather than classification, this technique is inappropriate. Robust regression is discussed comprehensively by Rousseeuw [108].

In a separate study, Tax and Duin [124] describe an outlier detection method based on Vapnik's support vector theory [132]. In the simplest case, a hypersphere is placed around the data, and reduced in radius until some points lie outside the sphere. These are labelled as outliers. The trade off between a small radius and a small number of outliers is controlled by a user-defined constant, C . The function F is minimised:

$$F = R^2 + C \sum_i \xi_i$$

where R is the radius of the sphere, and ξ_i is the distance that the i^{th} data point lies outside the sphere. ($\xi_i = 0$ for points inside the sphere.) A more sophisticated version also discussed in the paper replaces the sphere with a kernel function, allowing more complex decision boundaries to be formed by projecting the data into a higher dimensional space. Although the approach is appealing, and mathematically well-founded, Tax and Duin do not suggest a method for choosing C , a perennial problem in support vector applications. If C is given a high value, the model will predict no outliers exist; if C is given a low value, the model will predict many outliers exist.

Titterington [128] and Bebbington [9] separately published work in 1978, based on stripping convex hulls (or ellipsoids) of bivariate data. The idea is that the most outlying points in multivariate data will lie on the convex hull; these points can be "peeled" away repeatedly, to leave only the inliers. Both papers use efficient techniques to find the convex hull of a data set, but these techniques are constrained to work in two dimensions. More recent work in computational convexity efficiently solves higher dimensional problems by testing a single point for inclusion within a convex hull, which is faster than actually building the convex hull [3]. One problem with using convex hulls is that as the number of dimensions of a space increases, a higher proportion of the points sampled from that space will tend to be on the convex hull. We would need a sample growing exponentially with the dimension. In Section 4.6.4 we present the results of using the convex hull inclusion test for the three preference data sets. The

idea of repeated peeling off extreme points is related to the idea of sequential outlier rejection introduced in Section 4.6.

4.4.3 Summary: accommodation vs. discordancy

Having rejected both inclusion and identification at the start of Section 4.4, we have now discussed accommodation and discordancy. Accommodation is less than ideal, because it requires us to estimate the distribution of the outliers, and there is no reason to believe that outlying consumer preferences follow any meaningful pattern. On the other hand, discordancy tests require us to define some test statistic, and again, there is no clear solution. The choice of outlier detection method, as with other aspects of data analysis, is largely problem specific. We now turn to consider the specific problem of detecting outliers in cluster models.

4.5 Clustering and outliers

At the start of this chapter, we stated that “outliers are data points that are particularly inconsistent with the rest of the data” (p. 123). This means that if we choose to build some particular model from the data, then any outliers are only outliers *in relation to that model*, rather than in an absolute sense. For example, consider two clusters of univariate data, centred at $+2$ and -2 , with a single extra point x_o at 0 . This point is very unlikely to be from the same distribution as the rest of the data, even if it is located close to the mean of the entire sample. If we chose to model the entire data sample using a single Gaussian, then x_o would be in the centre of this model, and would not be regarded as an outlier.

Figures 4.4 and 4.5 show the same set of one-dimensional data. In the centre, is a test point x_o (ringed). Above the data are Gaussian density functions, with parameters estimated from the given sample. If we assume the data are generated from two Gaussians, then Figure 4.4 shows the test point does not fit the model⁵: it is probably an outlier. If we assume the data are generated from a single Gaussian, then Figure 4.5 shows the test point *does* fit the model: it is probably not an outlier.

We assume that the preference data sets contain clusters of similar consumers, as well as outliers. It is therefore appropriate to combine these two issues, and simultaneously cluster the data while finding outliers, in an attempt to avoid being misled.

One approach to finding outliers in clustered data is to fit a cluster model to the data, and then treat each cluster as a separate data set. Within each of these sets, we can test each suspect outlier for discordancy against the nearest cluster, and reject accordingly. However, an apparent outlier from one cluster may actually be an inlier

⁵I.e. it has a very low likelihood score according to the model

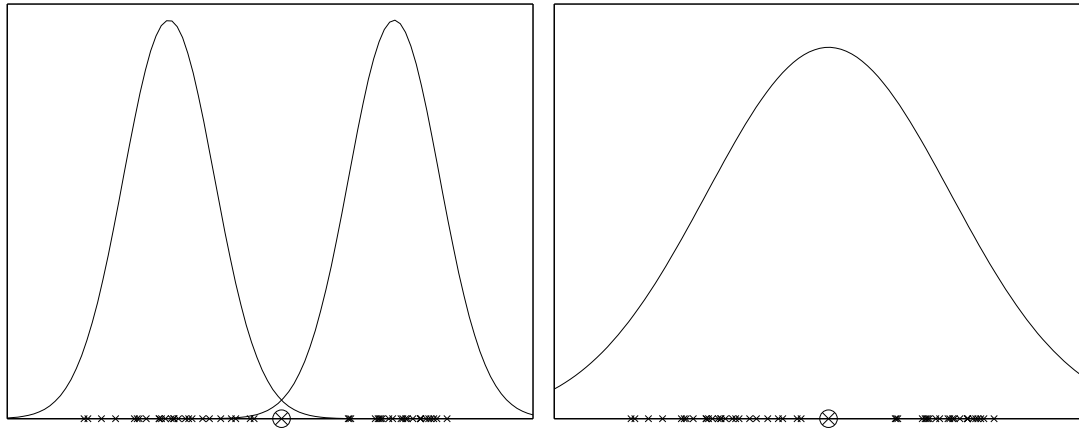


Figure 4.4: Two-cluster model. See text for details.

Figure 4.5: One-cluster model. See text for details.

from a different cluster. By changing the cluster that a single point is assigned to, we have to re-estimate the parameters of both clusters, which may lead us to change our opinion about whether the point in question is an outlier or not.

Wu et al. [136] describe using self-organising maps to cluster data and to produce a graphical representation. A domain expert was then able to use the maps to label points as being noisy outliers, “interesting” outliers, or inliers. The best prediction scores were obtained when the noisy outliers were removed, but the interesting outliers were retained. In the domain in question (glaucoma detection), interesting outliers were diseased patients, whereas noisy outliers were typically due to distraction or fatigue of the subjects. The majority of points were inliers, representing people with healthy eyes. The current work does not allow such a distinction to be made, due to the lack of knowledge about the preference panellists. Therefore all outliers must be regarded as “uninteresting” noise.

Perhaps the simplest approach to outlier detection in clustering was suggested by Weiss and Indurkha [134], who recommended searching for enough clusters to model both the main set of data and the outliers. They reasoned that outliers would be singletons or in small clusters, distant from the rest of the data, and would therefore be found as distinct clusters by any standard clustering algorithm. However, the number of clusters we search for will determine the number of outliers detected. After the model has been built, we still have to decide how small a cluster must be before we conclude that it contains only outliers. This raises more problems than it solves, particularly when the choice of clustering algorithm is likely to affect the outliers detected⁶.

⁶For example, k -means clustering tends to find equal-size clusters — see Section 3.4.6, p. 97.

4.5.1 Finding outliers in preference data

Some techniques described above require us to identify points that may be outliers before testing them. Testing each point independently of the others may fail due to masking or swamping (Section 4.4.2). Exhaustively searching through all subsets of data is combinatorially impossible⁷. There is of course a danger that if the first identification stage misses out a point, the second test stage will never test that point, and therefore never reject it as an outlier.

We listed several sources of outliers in preference data in Section 4.1.2. Due to these many sources, it is likely that a large number of the consumer preference records are outliers, perhaps as much as 20–30% [27]. Many approaches described above assume that a relatively small proportion of the data sample is contamination (e.g. Hoeting et al. [67] use a prior estimate of 2%). Any model built directly from the data will therefore be corrupted by the outliers. Given the resultant complexity, it seems improbable that all the outliers belong to one easily identifiable distribution. This suggests that identifying and rejecting outliers is more useful than accommodating them.

We can now summarise our goals and assumptions. We wish to identify and remove outliers before building a final cluster model. We must assume that a large number of outliers may be present. Further, we assume that the data set is structured, i.e. that it contains clusters.

4.5.2 Discordancy experiments

Suppose we use a mixture of Gaussians to model the data. We can calculate the maximum likelihood estimates of the parameters of the model by using the EM algorithm (Section 3.4.5). We then choose a threshold, θ , and calculate the likelihood l_i of every point in the sample with respect to the model parameters. Any point with $l_i < \theta$ is rejected as an outlier. After we have removed these outliers, we can re-estimate the model parameters, using the “clean” sample.

This is a valid approach only if the training set from which we estimate the model parameters has no outliers. If there are extreme points that are outliers, then the standard deviation of each Gaussian component will be substantially over-estimated. (This is an example of variance inflation, defined in Section 4.4.1.) So after we remove these points, and re-estimate the parameters, we would expect the standard deviation of the model to be reduced. If we use this new estimate to test further points for discordancy, it is likely that further points will be rejected. This in turn leads to an even lower estimate of the standard deviation, even more outliers, and ultimately, a model that may contain almost no data. The process converges when no remaining data points

⁷For example, the 450 records of the beverage preference set, P_b contains $2^{450} \approx 3 \times 10^{135}$ subsets.

have a likelihood threshold less than θ . We call this an “iterative discordancy test”. If the original distribution is Gaussian, then by removing points in the tails we will create a non-Gaussian distribution. Within reason, this needn’t cause problems, because Gaussian models are very robust at modelling approximately-Gaussian distributions [64].

Figure 4.6 shows data drawn from a single Gaussian distribution, with no added outliers. The discordancy test used is defined by Tax and Duin [125] to classify the test point $x_i \in \mathbf{x}$ as an outlier with respect to model M if:

$$\log(p(x_i|M)) < E[\log(p(\mathbf{x}|M))] - 3 \times \text{var}(\log(p(\mathbf{x}|M)))$$

The constant multiplier “3” corresponds to rejecting less than 1% of data in a one-dimensional distribution. As we note later (Section 4.6.6), this factor is dimension dependent.

The dots in the centre of Figure 4.6 represent inliers; each number n represents a point removed in the n^{th} iteration of this discordancy test. Clearly, the technique identifies the most outlying points as outliers, which are extreme values rather than contaminants.

The question is whether we should regard points labelled two or more as outliers. By removing the first layer, we produce a subset of data that should have no outliers in it (because we have removed them). However, if we were presented with this subset alone, with no prior knowledge, then by applying the same technique, we would still find outliers. These second-level outliers may either be “true” outliers masked by the first-level outliers, or they may be inliers.

Figure 4.7 shows the results of the same iterative discordancy test with a data sample drawn from two Gaussian components. This classifies more points as outliers, and takes more iterations to do so. The presence of the second cluster masks some outliers in both clusters, so more iterations are required before convergence.

Both these graphs show the results of single experiments; this technique is very sensitive to the exact distribution of the data, because the covariance of the data determines the width of the Gaussian components. A set of 250 experiments, each based on a different sample of 150 data points from the same two-cluster distribution as Figure 4.7, gave the mean number of outliers as 30.8, with a standard deviation of 24.0, rejecting between 1.3% and 89.3% of the points.

This iterative discordancy is closely related to the convex hull stripping methods outlined in Section 4.4.2. The sequential outlier rejection method we introduce below (Section 4.6) uses a broadly similar technique, but without requiring the analyst to choose the threshold θ .

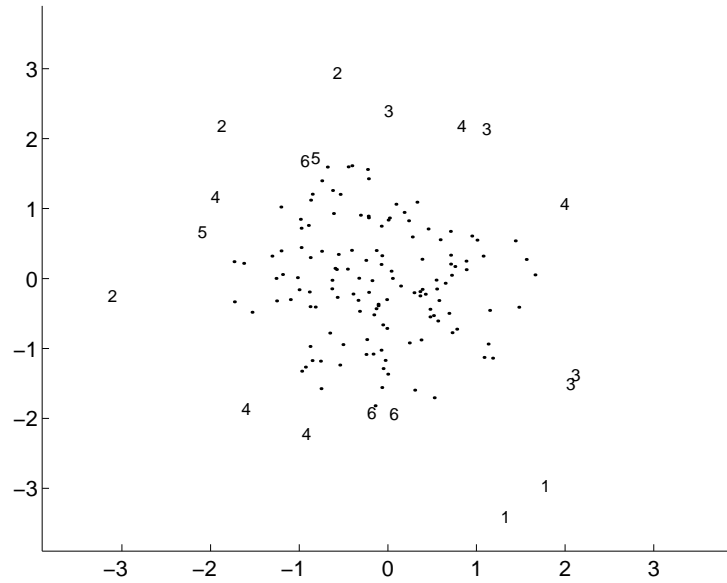


Figure 4.6: Iterative Gaussian discordancy, one cluster. Each number indicates the iteration in which that point was labelled as an outlier.

4.6 Sequential outlier rejection in clustering

We now present a new approach, namely *sequential outlier rejection* (SOR), where the data are used to build a cluster model, and exactly one data point is rejected as being a potential outlier, and removed from the data set. The model is repeatedly rebuilt using the remaining data to find further outliers. Effectively, we are iteratively using $n - 1$ points to reject the remaining point, in contrast to conventional discordancy analysis, which uses a one-pass method, where $n - t$ points are used to discover t outliers, with t initially unknown. The sequence in which the data points are removed, and the corresponding cluster error scores, provide us with extra information, that we can then use to estimate the number of outliers.

In the following analysis of the SOR algorithm, we first describe the algorithm in more detail, before moving on to consider the rejection sequences generated in more detail. In Section 4.6.3, we describe several methods of using the rejection sequences to identify outliers, giving results in Section 4.6.6 for some artificial data sets, and in Section 4.6.7 for the food preference data sets. In Section 4.6.8, we extend SOR to incorporate regression, in an attempt to produce a more useful outlier detection algorithm.

Given data set \mathbf{X} , the estimated number of clusters k and the objective function J_e , while $|\mathbf{X}| > k$, repeat:

1. Perform clustering, minimising error term, J_e over data set \mathbf{X}
2. Find ‘most outlying’ point, $x_o = \max_x J_e(x)$
3. Remove it: $\mathbf{X} \leftarrow \mathbf{X} \setminus x_o$

Figure 4.8: Sequential outlier rejection (SOR) algorithm

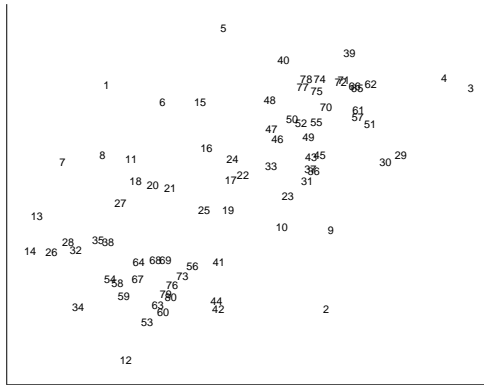


Figure 4.9: SOR sequence. Numbers indicate iteration in which points were labelled as outliers. Figure 4.10 shows the corresponding error rate.

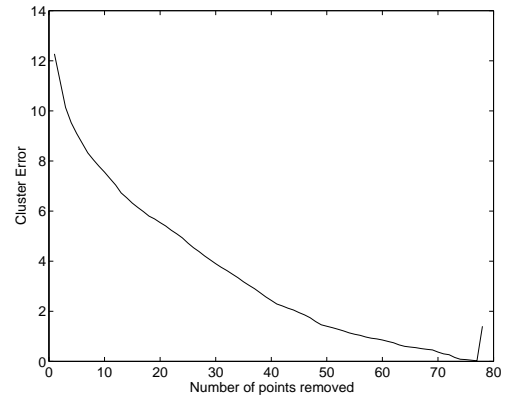


Figure 4.10: SOR error rate. Figure 4.9 shows the order in which points were removed.

that the data points round the edges have small numbers, indicating that they were removed first. The central points have much larger numbers, as they were the last to be removed by SOR.

The assumption is that the first few points rejected are outliers, while the rest are inliers. The most outlying point is the one that contributes the most to the cluster model error, J_e . For the k -means algorithm dividing data into clusters $C_1 \dots C_k$, this error is given by:

$$J_e = \sum_{i=1}^k \sum_{x \in C_i} \|x - \bar{x}_i\|^2$$

where \bar{x}_i is the mean of the i^{th} cluster. For an outlier, $x - \bar{x}_i$ will be large: when an outlier is removed, the error will drop sharply; when an inlier is removed, the error will only drop slightly. We are therefore interested in the point where the rate of change of the error slows. The sharp spike at the end of the sequence (the right-hand side of the Figure 4.10) occurs when there are almost as few data points as there are clusters,

so that removing one point will completely change the model, and the error measure becomes unstable. This tail can be ignored safely, as we can assume that all the outliers will have been removed before we reach this extreme.

4.6.2 Sequence consistency

The rejection sequences generated by SOR depend on the clustering algorithm, which in turn depends on random initialisations. We now investigate how much effect this has on the sequences. We can use Spearman's Rank correlation to compare two rejection sequences, and see how consistently the same points are rejected as outliers in each application of SOR. By repeatedly applying these algorithms to these data sets with different initialisations, we can analyse the sequence correlations in relation to the data. If approximately the same sequence is found repeatedly, we can have greater confidence that our results may be finding useful information from the data, despite the initial randomisation of the clustering algorithm parameters.

We repeat the SOR algorithm on each data set 50 times. We calculate the correlation between every pair of SOR sequences, and test these for significance. Table 4.1 shows the correlation scores for five data sets, the beverage preference data (P_b), the meat preference data (P_m), the vegetable preference data (P_v), an artificial data set consisting of four Gaussian clusters in 16 dimensions, with some added outliers⁸, and finally a second artificial set consisting of uniformly selected points from a 16-dimensional space. Both artificial sets had 350 data points, similar in scale to the preference data sets. The columns show the minimum and maximum correlations found between 50 runs of the SOR algorithm on each data set, with the final column showing the correlation score required to be significantly different from zero at the $p = 0.01$ level, according to a t -test⁹. Clearly, even the minimum correlations found are significant, demonstrating that the SOR sequences are not random, and showing that the random initialisation of the clustering algorithms is not important here. Note that by definition, the expected correlation between two random sequences is zero.

The first four rows of Table 4.1 have higher correlations than the last row. This suggests that treating the data set as a combination of outliers and clusters of inliers makes more sense for the preference data than for random uniform data. This further supports our assumptions that the preference data sets do contain outliers.

⁸We choose 16 dimensions to ease comparison with the 16-dimensional meat preference data. The standard deviation of each artificial cluster was set to 9, chosen to be similar to the standard deviation of clusters found in the meat preference data.

⁹For a correlation coefficient r with N samples, the test statistic is $t = r\sqrt{\frac{N-2}{1-r^2}}$. See Morrison [86].

Data set	Minimum correlation	Maximum correlation	Critical value for $p = 0.01$
Beverage	0.944	1.00	0.121
Meat	0.916	0.999	0.168
Vegetable	0.875	1.00	0.179
Gaussian clusters	0.907	0.999	0.138
Uniform	0.722	0.912	0.138

Table 4.1: Correlations of repeated SOR sequences

4.6.3 How many outliers?

The work above has concentrated on estimating how consistently data points are rejected. Now we discuss several approaches to estimating the number of outliers present. In Section 4.6.6, we compare these approaches experimentally.

Figure 4.11 shows how the error drops as we remove data from an artificial data set. This consisted of a single Gaussian cluster of 300 points, with 80 outliers added from a uniform distribution. Given the shape of this error curve, is there a way to estimate how many outliers exist? One approach is the “scree test” proposed by Catell [25]. This is often used to estimate the desirable number of factors to retain during factor analysis, or the number of principal components to retain during PCA. Despite the name however, it is not a formal statistical test, but rather a plot that can visually guide the analyst in estimating the required values. For example, when plotting the number of eigenvalues against the cumulative variance explained by PCA, after the first few components are used, the increase in variance explained slows. It is then up to the analyst to determine when the benefit of explaining more variance is outweighed by the cost of adding further components. It would be preferable to have a more objective, numerical approach, and we discuss several possibilities now.

Assuming that when all outliers have been removed, the rate of error drop decreases, then one approach would be to analyse the second derivative of the curve. When the rate of error drop changes, the second derivative should be at a local maximum. The simplest approximation to the second derivative is the second difference. The first difference, d^1 is calculated from the raw error sequence e thus: $d_i^1 = e_i - e_{i+1}$. The second difference, d^2 is then calculated from this: $d_i^2 = d_i^1 - d_{i+1}^1$. Figure 4.12 shows the same error curve as Figure 4.11, with the second difference (dotted line). Although the second difference is clearly higher in the area of interest (towards the left side of the graph), it is very noisy with approximately five distinct peaks. This does not give a clear indication of the number of outliers present.

A more sophisticated alternative is to use a Savitsky-Golay smoothing filter, using the algorithm presented by Press et al. [99] This was originally designed to preserve

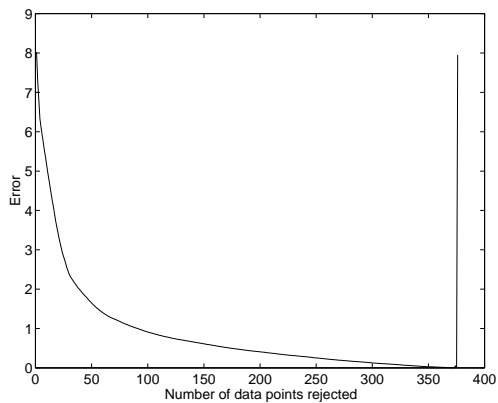


Figure 4.11: Error rate for unimodal Gaussian data with 80 added outliers

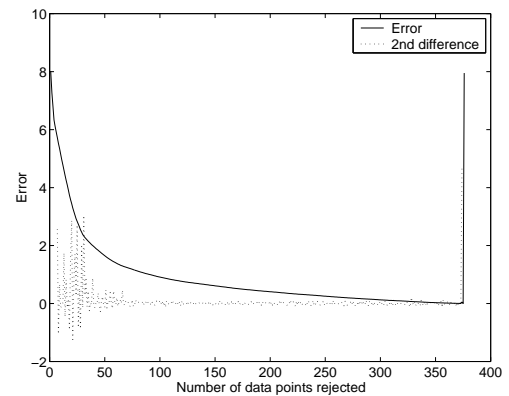


Figure 4.12: Error rate as in Figure 4.11, with second difference

higher order moments in noisy time series, and allows us to estimate smoothed derivatives of a curve when given only a set of points on the curve (as opposed to the equation of the curve, which is unknown). Figure 4.13 shows the same error curve as Figure 4.11 with the smoothed second derivative. The filter works by taking n consecutive points and building a regression model to fit them. The points are then projected onto the model, in an attempt to remove local noise, while retaining the overall structure. In this work, a filter length of 13 was used with a cubic polynomial model¹⁰. This shows a much cleaner estimate of the second derivative, with a clearer peak after around 30–40 points have been removed. Although this is less than the number of outliers created (80), some of the points sampled from the outlier distribution will actually appear close to the inliers, and so are not detected.

Another option would be to fit a polynomial (or some other parametric model) to the error curve, and then analytically calculate the equation of the second derivative, and use calculus to find the maxima. In practice, any parametric model will introduce “artificial” changes in gradient, due to the model rather than the data. For example, to produce a reasonably close fit, a high-order polynomial (e.g. 20) would be required. By using different orders of polynomial, different second derivatives with different peaks will be found, giving no reliable indication of the second derivative peak of the *data*.

Figure 4.14 shows the equivalent SOR graph for the beverage preference data. Although it is less extreme, there is still a clear change in rate, shown by a peak in the Savitsky-Golay second derivative, after approximately 45 points have been removed.

If there are no outliers in the data, then one would expect the error curve to be a straight line, at least for unstructured, uniform data. Figure 4.15 shows the SOR error curve for random uniform data (350 points 16 dimensions, sampled from range $[0, 1]$;

¹⁰I.e. Each window of 13 points was used to estimate the least-squares parameters of a cubic polynomial.

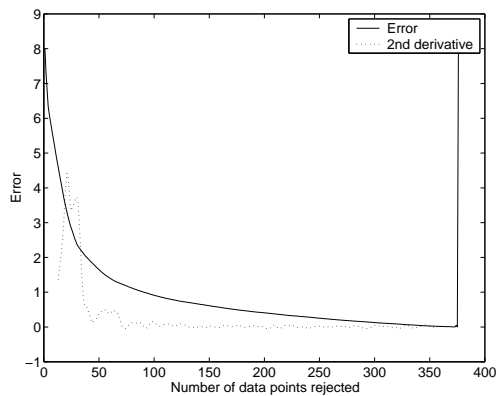


Figure 4.13: Error rate as in Figure 4.11, with smoothed second derivative

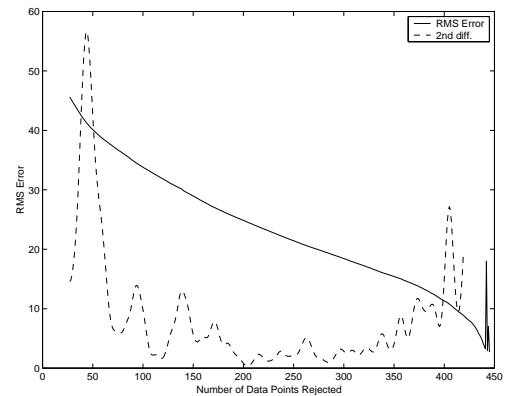


Figure 4.14: Error rate for beverage preference data, with smoothed second derivative

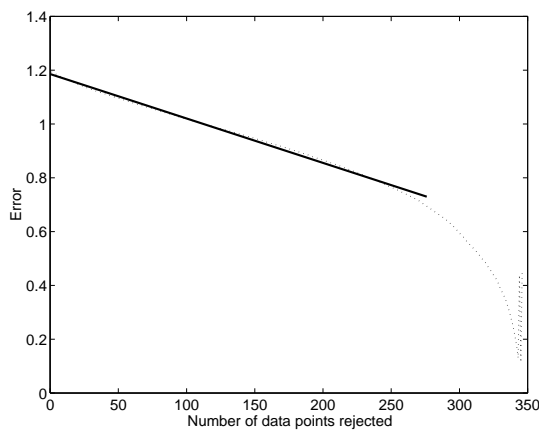


Figure 4.15: SOR error for uniform data with no outliers. Straight best-fit-line added for first 80% of curve.

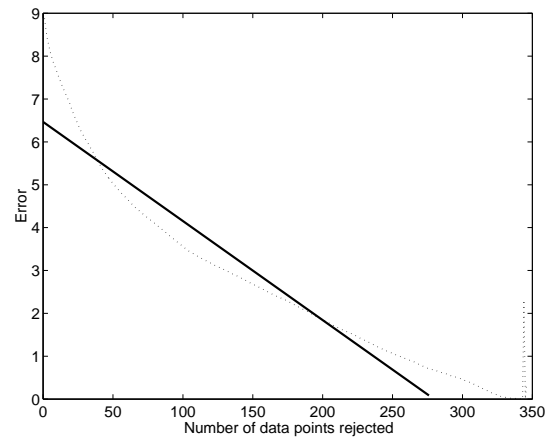


Figure 4.16: SOR error for Gaussian data with 50 outliers. Straight best-fit-line added for first 80% of curve.

SOR used k -means to search for 4 clusters). The solid line is a straight line fitted using least-squares to the first 80% of the error points. There is very little deviation between this straight line and the actual error (dotted line). Only the last 10–20% of the data show a sharp increase in the rate of error drop. By ignoring the last 20% of rejected points, we are only rejecting the hypothesis that 80% of the data are outliers — which is not an unreasonable assumption.

This suggests that if we fit a straight line to the error curve¹¹ and measure the deviation of the error curve from the line, then this may provide a simple way of estimating the number of outliers.

Figure 4.16 shows the error curve and the fitted line for some artificial structured

¹¹ignoring the last few records

data¹². The straight line first crosses the error curve after 39 points have been removed. This is less than the number of outliers created (50), but some of these points will occur within the Gaussian clusters, as noted earlier, so this heuristic seems at least worthy of further investigation.

4.6.4 Convex hull inclusion test

In Section 4.4.2, we mentioned past work using convex hulls to “strip away” outlying data points. A convex hull of a finite set of points is the smallest convex set that includes all the points. The motivation for outlier detection is that points on the hull are most extreme, and therefore likely candidates for removal.

We use the convex hull inclusion test of Bailey et al. [3] to identify which points lie on the convex hull. For each of the three preference data sets, we test each point in turn. Given n points, we test whether each point is within the remaining $n - 1$, by forming a linear programming problem. This formulation attempts to define the test point as a weighted sum of the remaining points, constrained so that the weights are non-negative and sum to unity. This programme only has a solution if the test point is within the convex hull of the remaining points.

The results in Table 4.2 clearly show the weakness of this method when applied to preference data sets. The problem is that in a high-dimensional space, at least most of the data points from a modest-sized sample will lie on the convex hull. Indeed, for two of the three data sets, every point lies on the convex hull. We gain nothing by treating these points as outliers.

Data set	Points inside convex hull
Meat preference, P_m	5
Vegetable preference, P_v	0
Beverage preference, P_b	0

Table 4.2: Convex hull inclusion test results

4.6.5 Validation methods

Having presented several alternative methods of estimating the number of outliers based on the SOR technique, we need some way of determining how effective these methods are. If we work with an artificially generated data set, we know in advance the approximate number of outliers. We can then measure the accuracy of the detection process using a simple classification accuracy measure.

¹²100 points drawn from each of four Gaussian components, in two dimensions, with a standard deviation of 2, plus 50 outliers from a uniform distribution

We take an outlier rejection sequence, with its corresponding error sequence. The linear-intercept and the Savitsky-Golay second derivative provide two methods to estimate how far down the sequence we should go before we stop labelling points as outliers. As an alternative, we use a mixture of Gaussians with a threshold on the likelihood, to be used as a discordancy test. By varying the parameters (the number of outliers in the SOR sequence, or the value of the likelihood threshold) we can calculate various classification error rates. More precisely, we can measure the true-positive rate (i.e. the proportion of correctly identified outliers), the false-positive rate (the proportion of non-outliers incorrectly identified as outliers) and so on. In principle, we could then apply some particular cost function to penalise one sort of error more than another. Without suitable knowledge, we define the error to be the (equally weighted) sum of false-positives and false-negatives in the following results.

4.6.6 Outlier detection results

We now compare the accuracy of the various techniques discussed. Two artificial data sets and the three preference data sets are used. The first two allow us to estimate the outlier classification errors, while the latter give some indication of the usefulness of these techniques when applied to real data. In the SOR experiments, the clustering algorithm used is k -means. Five techniques are compared:

Exhaustive SOR. SOR is applied to the data, removing one point at a time. At each stage, the total classification error is calculated, and the minimum error possible is taken.

Exhaustive GMM. A Gaussian mixture model is fitted to the data, and the likelihood threshold is varied to vary the number of outliers identified. The optimum value is taken.

Linear intercept. SOR is applied to the data, and a straight line is fitted to the error curve. The number of outliers is estimated corresponding to where the line first crosses the curve.

Second derivative. SOR is applied to the data, and the second derivative of the error curve is estimated using the Savitzky-Golay smoothing filter. The location of the largest peak is used to estimate the number of outliers.

n standard deviations ($n\sigma$). A Gaussian mixture model is fitted to the data, and the standard deviation σ_k of each component is calculated. All points greater than $n\sigma_k$ away from the centre of the maximum-likelihood component are identified as outliers. n is a function of the number of dimensions; in one dimension, $n = 1.96$.

Further values were estimated empirically, as described in the Appendix, Section A.2.

The first artificial data set used is drawn from a low dimensional distribution. The inliers consist of 100 points drawn from each of four two-dimensional Gaussians, which each had a standard deviation of two. The centres of the Gaussians were on the corners of a square of length 6. A further 80 outliers were added uniformly in the same region. Table 4.3 shows the number of outliers that each approach detected, and what the resultant error was. The error is defined as the number of outliers classified as inliers, plus the number of inliers classified as outliers, expressed as a percentage of the whole data set. The table shows the average of 250 experiments with this data set.

The lowest errors are associated with the two exhaustive methods, which is to be expected: heuristically choosing a threshold will never outperform choosing the best possible threshold. With the preference data sets, we don't know the outliers in advance, which is why we are developing these heuristics. Note that both the exhaustive methods underestimate the number of outliers, choosing only around half the correct number, which is 80. As stated earlier, some outliers will actually coincide with the dense regions of inlier points, so this is an acceptable estimate. The two heuristics based on the SOR curve ("linear intercept" and "second derivative") both give results only slightly worse than the best found. The $n\sigma$ results are marginally better, but still not as good as the exhaustive methods.

Technique	Estimated number of outliers	% Error (standard deviation)
Exhaustive SOR	39.55	11.89 (1.529)
Exhaustive GMM	38.73	11.93 (1.334)
Linear intercept	66.73	15.76 (0.251)
Second derivative	18.64	16.38 (0.105)
$n\sigma$	16.00	14.05 (0.505)

Table 4.3: Estimated number of outliers (low dimensional artificial data, contaminated with 80 outliers)

Table 4.4 shows the results for a higher dimensional data set. The inliers consist of 100 points drawn from each of four Gaussians, each with a standard deviation of 15, in a 16-dimensional space. The outliers were 80 points randomly selected from a uniform distribution covering the same space. The table shows the average of 250 experiments with this data set.

As with the lower dimensional experiments, the two exhaustive methods give the best results, both with errors around 1%. These results show far lower error scores than with the previous artificial data set (Table 4.3). By increasing the dimensionality, we move the cluster centres further away, leading to less overlap between clusters in our

partition, and therefore ease the identification of outliers. This highlights the difficulties of designing artificial data sets that we discussed in Section 4.3.

Technique	Estimated number of outliers	% Error (standard deviation)
Exhaustive SOR	77.94	1.11 (0.484)
Exhaustive GMM	76.82	1.30 (0.663)
Linear intercept	62.59	16.67 (0.000)
Second derivative	75.80	16.66 (0.046)
$n\sigma$	18.20	20.38 (1.306)

Table 4.4: Estimated number of outliers (high dimensional artificial data, contaminated with 80 outliers)

We can see from the results in Tables 4.3–4.4 that the “linear intercept” and “second derivative” SOR methods are not consistently better than each other. We can also see that the $n\sigma$ method tends to grossly underestimate the number of outliers present, even though both data sets were sampled from a mixture of Gaussians.

We now turn to apply SOR to the food preference data sets. This brings the advantage that our goal is to model these data sets, and as part of that, to remove outliers from them. The drawback is that we do not know in advance which points (consumers) are outliers, nor how many outliers there are. Thus, we cannot produce accuracy estimates.

4.6.7 SOR for food data

Table 4.5 compares the number of outliers classified by each technique for the three food data sets available. Because the true number and identity of the outliers is unknown, no error scores can be given. The number of clusters sought by SOR in each case corresponded to the minimum description length results given earlier (Figures 3.8–3.12, page 107), i.e. 3, 4 and 7 clusters for the meat, vegetable and beverage sets respectively. The figures are the average of 100 experiments, with the standard deviations shown in parentheses.

Clearly, using the second derivative of the SOR error curve gives an inconsistent estimate of the number of outliers, shown by the very large standard deviation scores. Of the other two methods, the linear intercept of the SOR curve tends to estimate many more outliers than the $n\sigma$ approach. The $n\sigma$ approach assumes the inlier and outliers come from a Gaussian distribution. As we found in the previous section, even when the data is drawn from a Gaussian distribution, the $n\sigma$ method still underestimated the number of outliers. We might therefore assume that more outliers exist in the preference data than the values shown in the last row of Table 4.5. For all three data sets, the estimated number of outliers is in the range of 10–20% of the data.

Technique	Meat P_m	Vegetable P_v	Beverage P_b
Linear intercept	36.0 (2.23)	29.8 (3.21)	63.5 (4.31)
Second derivative	22.0 (23.23)	21.5 (25.23)	32.3 (45.06)
$n\sigma$	22.9 (0.20)	14.6 (1.02)	42.3 (2.50)

Table 4.5: Estimated number of outliers in preference data sets. Standard deviations are shown in parentheses.

The linear intercept SOR estimation produces the most consistent results, shown by the low standard deviation scores in Tables 4.3, 4.4 and 4.5. Where the true identity of the outliers is known, this method also produces reasonable accuracy. This is further supported by the observation that $n\sigma$ seems to underestimate the number of outliers, and Table 4.5 shows that the linear intercept estimates are greater than the $n\sigma$ estimates. In comparison, the second derivative estimates are sometimes below the $n\sigma$ estimates, and have a very high standard deviation between repeated experiments.

4.6.8 SOR with weighted cluster-regression

In Section 3.9 (p. 117), we considered some difficulties in estimating the number of clusters present in the preference data sets. We argued that it was important to consider how useful the predictions were in reaching our longer term goal of understanding consumer preferences and designing food products accordingly. In the current case of outlier detection, we face a similar problem. It is impossible to objectively identify every outlier in the preference sets, making it impossible to definitively choose between the various heuristic methods that we have used so far. In Section 3.9, we attempted to aid the decision making by considering the predictive accuracy of models based on different partitions of the data. We now consider the predictive accuracy of models based on different outlier detection estimates, before returning to this question in Section 4.7.

We can use the weighted cluster-regression error score defined in Section 3.9.1, instead of the simple cluster dispersion score, as the error within the SOR algorithm. Figure 4.17 shows such an algorithm. As with the previous version of SOR, we remove the single point furthest from its corresponding cluster centre at each iteration. At each iteration, we record the cluster-regression error, whereas we had previously recorded the total cluster dispersion error.

Figure 4.18 shows how the weighted cluster-regression error score changes as we remove outliers from the meat data. We use k -means clustering with three clusters (as we predicted in Section 3.6.2), forward sequential selection (as described in Section 2.4.2), and linear regression (as described in Section 2.2.1).

At the left of the graph, we are using all the data, and so the error score corresponds to the previous results (Figure 3.23, p. 120). As we remove badly fitted points, the error

Given data set \mathbf{X} , the assumed number of clusters k and the objective function J_e , while $|\mathbf{X}| > k$, repeat:

1. Perform clustering, minimising error term, J_e , over data set \mathbf{X}
2. For each cluster c_i with $|s_i|$ members, for $i=1 \dots k$:
 - (a) Calculate mean μ_i
 - (b) Select features
 - (c) Build regression model
 - (d) Estimate mean error of i^{th} regression model, ϵ_i
3. Calculate model error, $\frac{1}{|\mathbf{X}|} \sum_{i=1}^k |s_i| \cdot \epsilon_i$
4. Find 'most outlying' point, $x_o = \max_x J_e(x)$
5. Remove it: $\mathbf{X} \leftarrow \mathbf{X} \setminus x_o$

Figure 4.17: SOR algorithm for weighted cluster-regression

tends to drop, until at the right-hand end of the graph, we have a model that has close to zero error, but is only capturing very few consumers preferences. Figures 4.19 and 4.20 show the corresponding graphs for the vegetable and beverage data sets respectively.

These graphs could be useful to an analyst to help decide where to set the balance between modelling everyone with limited accuracy, or modelling a few people very accurately. For example, the meat results of Figure 4.18 show that if we remove between about 70 and 100 people, then the weighted regression error across the three clusters remains constant, at just above 0.2. The results from the other food data sets are less clear, without a plateau, but still enable the analyst to choose a particular trade off between building an accurate model and modelling many consumers. Note that this plateau is an artefact of this particular data set, rather than a feature of the SOR algorithm used. However, highlighting these features of the data could still be useful to the analyst.

4.7 Conclusions

Outliers can occur in any data set, but preference sets are particularly susceptible due to the vagaries of human taste. Rather than attempt to model these, we wish to identify and remove the misleading responses, to avoid distorting the models that we finally build. As with many other data analysis problems, we start with a single set of data, which may or may not contain outliers. Any outliers present will influence any model we build using this data set. Using this same model to predict which points are

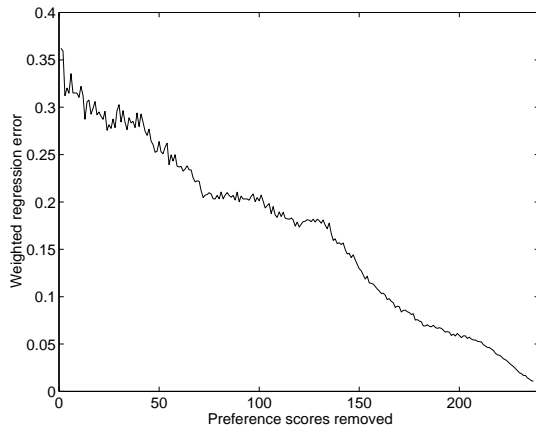


Figure 4.18: Weighted cluster regression with SOR, meat data

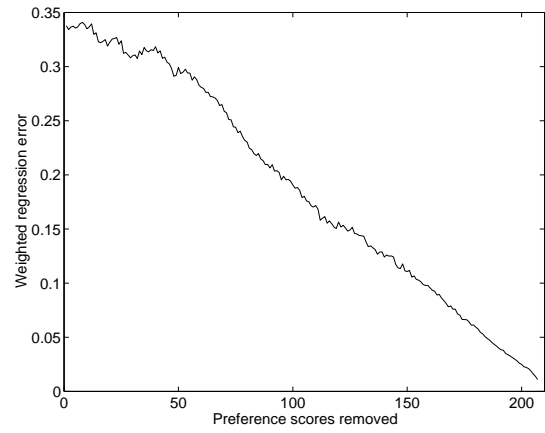


Figure 4.19: Weighted cluster regression with SOR, vegetable data

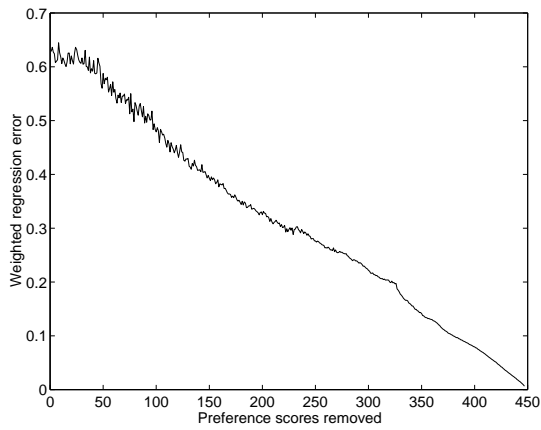


Figure 4.20: Weighted cluster regression with SOR, beverage data

outliers may be misleading. A less risky alternative is to iteratively use a model to detect a single outlier, and then to remove this point from the data set.

We have applied our novel sequential outlier rejection (SOR) algorithm to k -means clustering on several data sets, both artificial sets with known outliers and the preference sets with unknown outliers. In both cases, as each data point is removed, the cluster error drops monotonically. After a number of points have been rejected, the rate at which the cluster error drops is reduced, and remains at this lower level until very few data points are left.

The interpretation is that the points rejected at the start are outliers that were poorly modelled by the k -means clustering solution, and therefore caused a large error. As each of these points is removed, the error drops greatly. After a certain point has been reached, the points being removed are accurately modelled, and only contribute a small error. Therefore, removing these causes the cluster error to drop more slowly. SOR provides a non-parametric approach to outlier detection, which does not require the analyst to pre-specify thresholds or other such parameters. The SOR method should not be regarded as a panacea. It is not hard to conceive of a data set where sequential rejection could lead to the rejection of the wrong number of outliers, due to the original model being distorted by these outliers.

The regression error scores in Section 4.6.8 show that the presence of outliers in the preference data set makes regression models less accurate, and makes predicting preferences (and hence designing food) more difficult. We therefore suggest that iteratively removing outliers from a cluster-regression model until the regression error is “sufficiently small” is the best way to proceed, incorporating as it does the analysts’ judgement.

The flexibility of the SOR approach is such that we can replace the clustering, the regression and the feature selection algorithms used with alternatives, depending on the nature of the data we are analysing. For example, we might choose to use non-linear regression algorithms or fuzzy clustering techniques.

Given the difficulties of objectively identifying outliers, we should also ask the question, how many outliers is it useful to remove? At one extreme, we could remove no outliers at all (the “include” option mentioned in Section 4.1.2). This assumes that every preference panellist gives accurate and consistent scores; it assumes that every consumer is consistent with some real and identifiable group of consumers; it also assumes that we want to model every consumer’s tastes with equal accuracy. This is clearly unrealistic: preference panellists will make mistakes; some consumers have eccentric tastes at odd with the rest of the market; and some niche markets are so small that it is not profitable to design products to fill them.

The other extreme is to remove a very large number of outliers, as suggested by the

convex hull results (Section 4.6.4). This risks ignoring entire groups of consumers, and also risks over-simplifying the variety of preferences that exist within a market, leading to a bland, average product that satisfies no one.

If we treat a preference panel as a sample from the population, then we should consider what proportion of the population is represented. For example, the population of the UK is currently 60 million, and the population of the world is approximately 6,000 million. One preference panellist could therefore represent 300,000 UK consumers, or 3 million worldwide consumers. Each apparent outlier that we remove from the data set could be seen as ignoring a potentially vast market.

We must clearly compromise between these extremes by retaining enough data to allow us to understand the complexities of the consumer marketplace, while removing as much erroneous data as possible.