A Meta analysis study of outlier detection methods in classification

Edgar Acuna and Caroline Rodriguez
edgar@cs.uprm.edu, caroline@math.uprm.edu
Department of Mathematics
University of Puerto Rico at Mayaguez
Mayaguez, Puerto Rico 00680

Abstract

An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism (Hawkins, 1980). Outlier detection has many applications, such as data cleaning, Fraud detection and network intrusion. The existence of outliers can indicate individuals or groups that have behavior very different to the most of the individuals of the dataset. Frequently, outliers are removed to improve accuracy of the estimators. But sometimes the presence of an outlier has a certain meaning which explanation can be lost if the outlier is deleted.
In this work we compare detection outlier techniques based on statistical measures, clustering methods and data mining methods. In particular we compare detection of outliers using robust estimators of the center and the covariance matrix used in the Mahalanobis distance, detection of outliers using partitioning around medoids (PAM), and two data mining techniques to detect outliers: The Bay's algorithm for distance-based outliers (Bay, 2003) y the LOF a density-based local outlier algorithm (Breunig et al., 2000). A decision on doubtful outliers is taken by looking into two visualization techniques for high dimensional data: The parallel coordinate plot and the surveyplot. The comparison is carried out in 15 datasets.

1 Introduction.

According to Hawkins (1980), "An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism". Almost all the studies that consider outlier identification as their primary objective are in the field of statistics. A comprehensive treatment of outliers appears in Barnet and Lewis (1994). They provide a list of about 100 discordancy tests for detecting outliers in data following well known distributions. The choice of an appropriate discordancy test depends on:

a) the distribution,

b) the knowledge of the distribution parameters,

c) the number of expected outliers, and

d) the type of expected outliers.

These methods have two main drawbacks. First, almost all of them are for univariate data making them unsuitable for multidimensional datasets. Second, all of them are distribution-based, and most of the time the data distribution is unknown. Real-world data most of the time is multivariate with unknown distribution.

Detecting outliers, instances in a database with unusual properties, is an important data mining task. People in the data mining community got interested in outliers after Knorr and Ng (1998) proposed a non-parametric approach to outlier detection based on the distance of a instance to its nearest neighbors. Outlier detection has many application among them: Fraud detection and network intrusion, and data cleaning. Frequently, outliers are removed to improve accuracy of the estimators. However, this practice is not recommendable because sometimes outliers can have very useful information. The presence of outliers can indicate individuals or groups that have behavior very different of a normal situation.

The section 2 of this paper includes a brief discussion of treatment of outliers for univariate data. The section 3 focuses in methods for detection of multivariate outliers. Four methods of outlier detection are considered: a method based on robust estimation of the Mahalanobis distance, a method based on the PAM algorithm for clustering, a distance-based method and a density-based method. The last section of this chapter covers the effect and treatment of outliers in supervised classification.

2. Univariate Outliers

Given a data set of n observations of a variable x. Let $\bar{x}$ be the mean and let s be standard deviation of de data distribution . One observation of the data set is declared as an outlier if lies outside of the interval

$$(\bar{x} - ks, \bar{x} + ks) \qquad (1)$$

where the value of k is usually taking as 2 or 3. The justification of these values relies in the fact the assuming normal distribution one expects to have a 95% percent of the data on the interval centered in the mean with a radius equals to two standard deviation. Also one expects to have the whole data inside an interval centered at the mean and three standard deviation as radius.

From equation (1), the observation x is considered an outlier is

$$\frac{|x - \bar{x}|}{s} > k \qquad (2)$$

The problem with the above criteria is that assumes normal distribution of the data something that frequently does not occur. Furthermore, the mean and standard deviation are highly sensitive to outliers.

John Tukey (1977) introduced several methods for exploratory data analysis, one of them was the *Boxplot*. The *Boxplot* is a graphical display where the outliers appear tagged. Two types of outliers are distinguished: *mild outliers* and *extreme outliers*.

An observation x is declared an *extreme outlier* if lies outside of the interval (Q_1-3×IQR, Q_3+3×IQR). Notice that the center of the interval is $(Q_1+Q_3)/2$ and its radius is 3.5×IQR, where IQR=Q_3-Q_1 , called the *Interquartile Range*, is a robust estimator of variability which can replace s in equation (2). On the other hand $(Q_1+Q_3)/2$ is a robust estimator of the center that can be used instead of $\bar{x}$ in equation (1).

An observation x is declared *a mild outlier* is lies outside of the interval (Q_1-1.5I×QR, Q_3+1.5×IQR). The interval has a center at $(Q_1+Q_3)/2$ and its radius is 2×IQR.

The numbers 1.5 and 3 are chosen by comparison with a normal distribution. All the major statistical software include boxplots among their graphical displays.

The following figures show the outliers of the features in the three classes of the dataset *Iris* detected through their boxplot

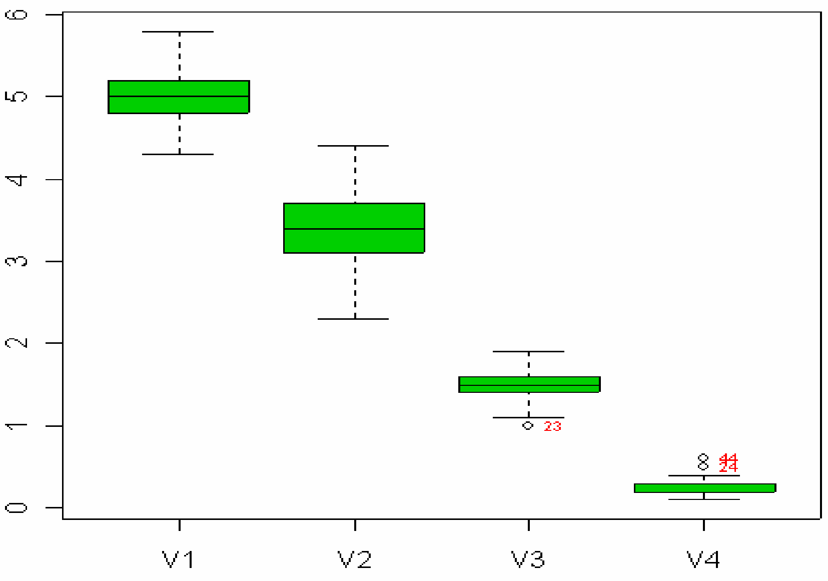

Figure 1. Outliers of the features in class 1 of the *Iris* data set

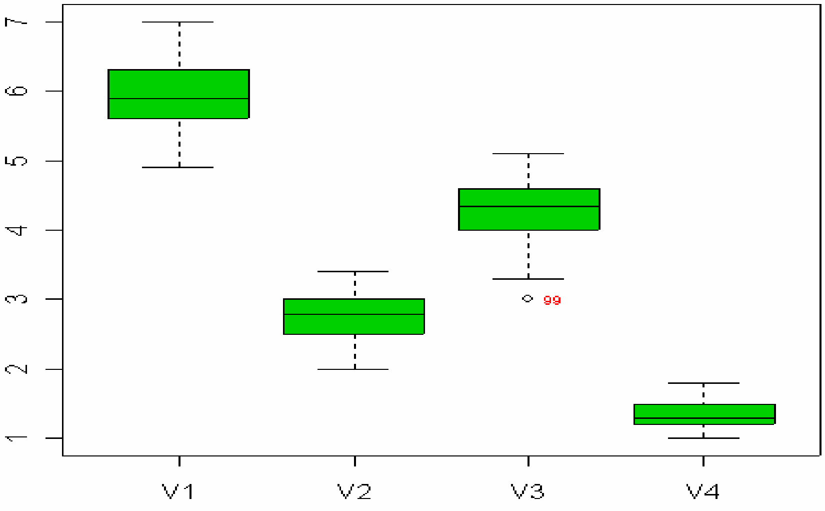

Figure 2. Outliers of the features in class 2 of the *Iris* data set

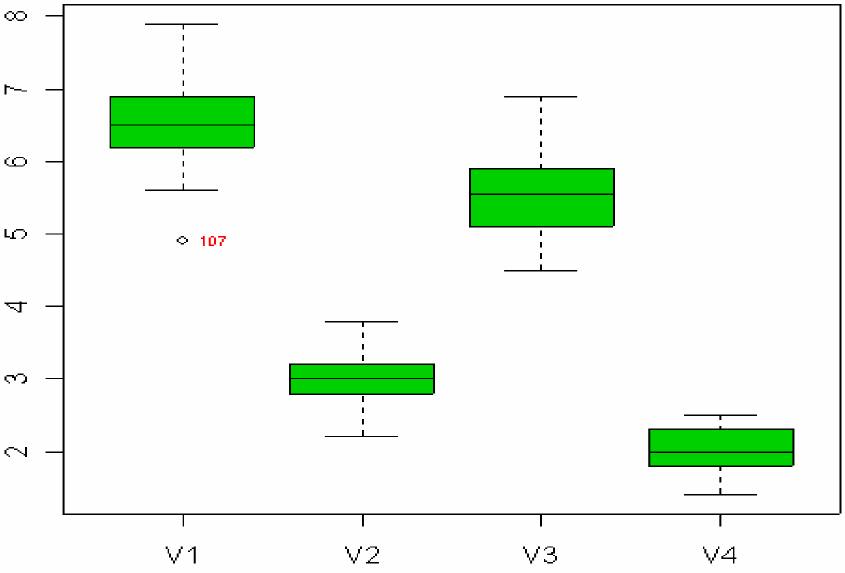

Figure 3. Outliers of the features in class 3 of the *Iris* data set.

3. Multivariate Outliers

Let us consider a data set D with p features and n instances. In a supervised classification context we must know also the classes where each of the instances belongs to. It is very common include the classes as the last column of the data matrix. The objective is to detect all the instances that seems to be unusual, these will be the multivariate outliers. One might think that multivariate outliers can be detected based on the univariate outliers on each feature but as it is shown in the figure 4 this is not true. The instance appearing in the upper right corner is a multivariate outlier but it is not an outlier in each feature. On the other hand, an instance can have values that are outliers in several features but the whole instance might be not a multivariate outlier.

There are several methods to detect multivariate outlier. The methods discussed in this paper are: statistical-based outlier detection, outlier detection by clustering, distance-based outlier detection and density-based local outlier detection. The before mentioned methods are discussed in the next sections.

3.1. Statistical based outlier detection.

Let **x** be an observation of a multivariate data set consisting of n observations and p features. Let $\bar{\mathbf{x}}$ be the centroid of the dataset, which is a p-dimensional vector with the means of each feature as components. Let **X** be the matrix of the original dataset with columns centered by their means. Then the p×p matrix **S=**1/(n-1) **X'X** represents the covariance matrix of the p features.

The multivariate version of equation (2) is

$$D^2(\mathbf{x},\bar{\mathbf{x}}) = (\mathbf{x}-\bar{\mathbf{x}})\mathbf{S}^{-1}(\mathbf{x}-\bar{\mathbf{x}}) > k \qquad (3)$$

where D^2 is called the Mahalanobis square distance from **x** to the centroid of the dataset. An observation with a large Mahalanobis distance can be considered as an outlier.

Assuming that the data follows a multivariate normal distribution then it has been shown that the distribution of the Mahalanobis distance behaves as a Chi-Square distribution for a large number of instances. Therefore the proposed cutoff point in (3) is given by k= $\chi^2_{(p,1-\alpha)}$, where χ^2 stands for the Chi-Square distribution and α is a signification level usually taken as .05.

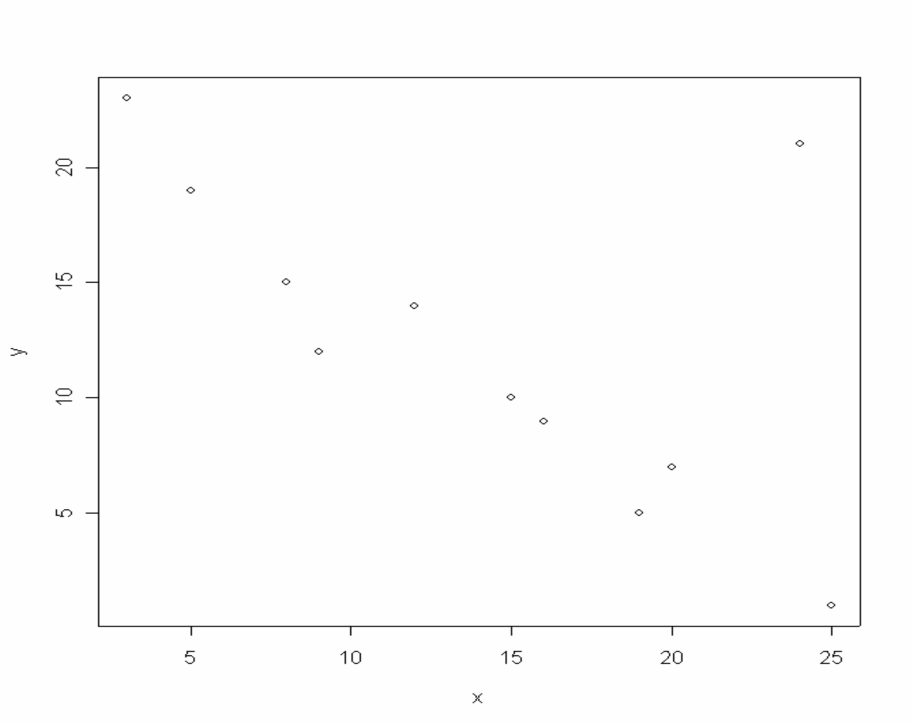

Figure 4. Example of a bidimensional outlier that is not an outlier in neither of its projections.

A basic method to detect multivariate outliers is by looking for the outliers that appear in the *boxplot* of the distribution of the Mahalanobis distance of the all instances as it is shown in the following example.

Example 1. Find out the multivariate outliers in each of the classes of the dataset Iris by building boxplots.

Notice that only two outliers (119 and 132) are detected in class 3. People in the data mining community prefers to rank the instances using an outlyngness measures rather to classify the instances in outliers and non-outliers. Rocke and Woodruff (1996) stated the Mahalanobis distance works well identifying scattered outliers. However in data with clustered outliers the Mahalanobis distance measures does not perform well detecting outliers. Data sets with multiple outliers or clusters of outliers are subject to the *masking* and swamping effects.

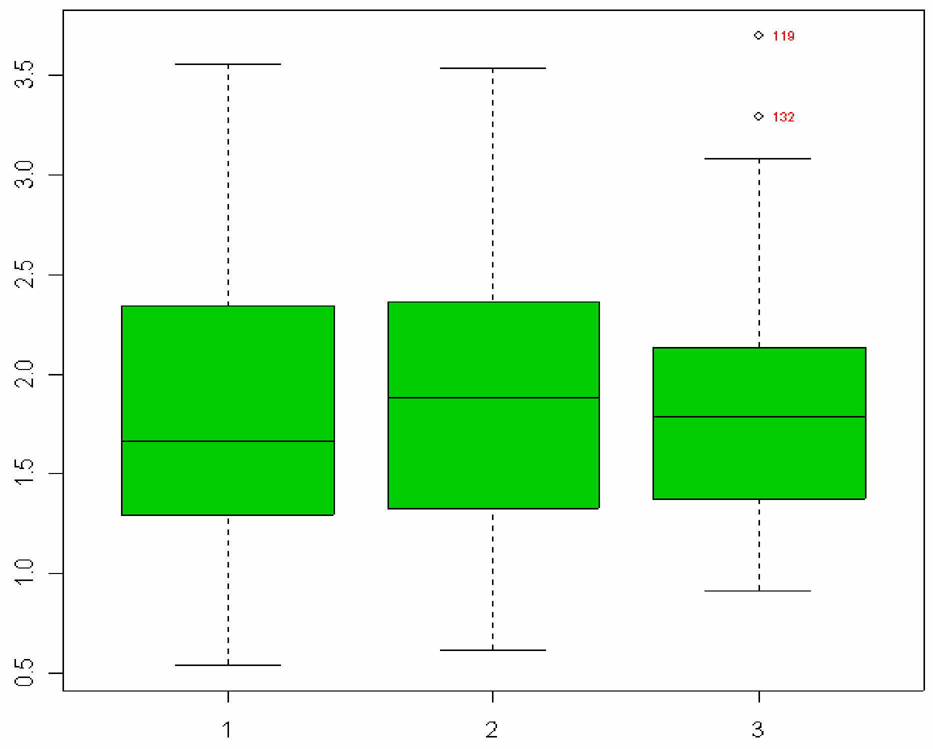

Figure 5. Detecting multivariate outliers by boxplots in the *Iris* dataset

Masking effect. It is said that an outlier masks a second one close by if the latter can be considered outlier by himself, but no any more if it is considered along with the first one. Equivalently after the deletion of one outlier other instance may emerge as an outlier.

Masking occurs when a group of outlying points skews the mean and covariance estimates toward it, and the resulting distance of the outlying point from the mean is small.

Swamping effect. It is said that an outlier swamps other instance if the latter can be considered outlier only under the presence of the first one. In other words after the deletion of one outlier other outlier may become a "good" instance. Swamping occurs when a group of outlying instances skews the mean and covariance estimates toward it and away from other "good" instances, and the resulting distance from these "good" points to the mean is large making them look like outliers.

Example 2. Consider the data set due to Hawkins, Bradu, and Kass (Rousseuw and Leroy, 1987) consisting of 75 instances and 3 features, were the first fourteen instances had been contaminated to be outliers.

Using the Mahalanobis distance only the observation 14 is detected as an outlier as is shown in Figure 6. The remaining 13 outliers appear to be masked.

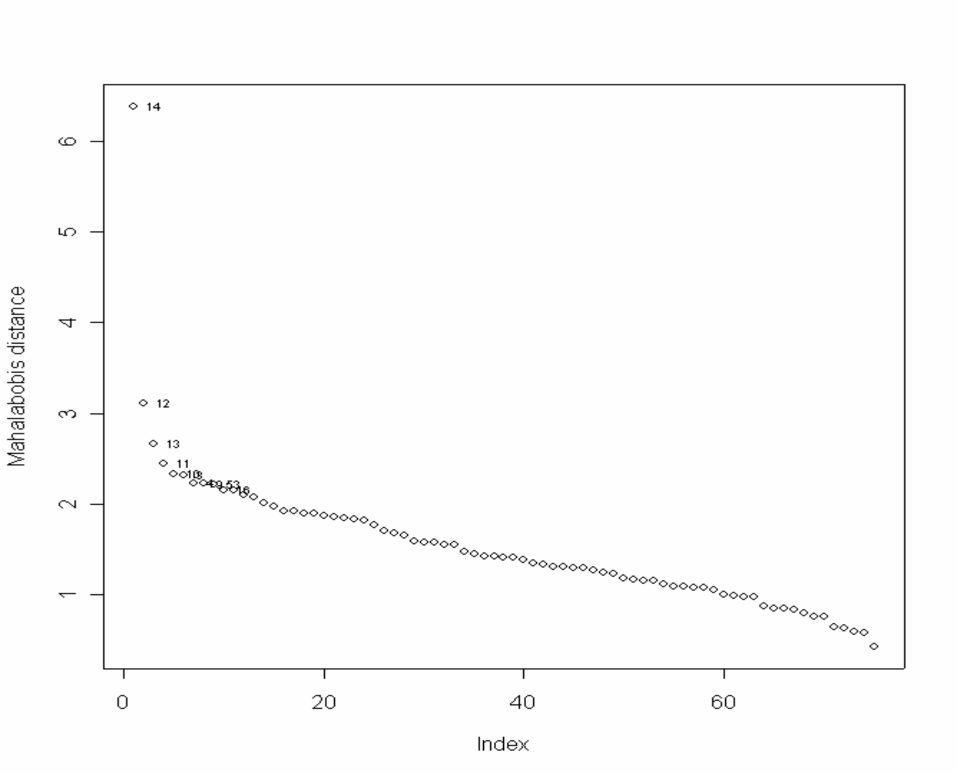

Figure 6: The Masking effect of multivariate outliers in the Hawkins data set

The masking and swamping can be solved by using robust estimates of the centroid (location) and the covariance matrix (shape), which by definition are less affected by outliers.

Outlying points are less likely to enter into the calculation of the robust statistics, so they will not be able to influence the parameters used in the Mahalanobis distance. Some robust estimators of the centroid and the covariance matrix include the minimum covariance determinant (MCD) and the minimum volume ellipsoid (MVE) both of them introduced by Rousseeuw (1985).

Minimum Volume Elipsoid (MVE) estimator is the center and the covariance of a subsample size h ($h \leq n$) that minimizes the volume of the covariance matrix associated to the subsample. Formally,

$$\text{MVE}=(\bar{\mathbf{x}}_J^*, S_J^*)\ , \qquad (4)$$

where J={set of h instances: $Vol(S_J^*) \le Vol(S_K^*)$ for all K s. t . #(K)= h}.

The value of h can be thought as the minimum number of instances which must not be outlying and usually h=[(n+p+1)/2], where [.] is the greatest integer function.

Minimun Covariance Determinant (MCD) estimator is the center and the covariance of a subsample of size h ($h \le n$) that minimizes the determinant of the covariance matrix associate to the subsample. Formally,

$$\text{MCD}=(\bar{\mathbf{x}}_J^*, S_J^*)\ , \qquad (5)$$

where J={set of h instances: $|S_J^*| \le |S_K^*|$ for all K s. t . #(K)= h}

As before, it is common to take h=[(n+p+1)/2], where [.] is the greatest integer function.

The MCD estimator underestimates the scale of the covariance matrix, so the robust distances are slightly too large, and too many instances tend to be nominated as outliers. A scale-correction has been implemented, and it seems to work well. The algorithms to compute the MVE and MCD estimators are based in combinatorial arguments (for more details see Rousseeuw and Leroy, 1987). In this paper both estimators, MVE and MCD, have been computed using function **cov.rob** available in the package **lqs** of **R**. This function uses the best algorithms available so far to computed both estimators.

Taking in account their statistical and computational efficiency, the MCD is preferred over the MVE. Replacing the classical estimators of the center and the covariance in the usual Mahalanobis distance, equation (3), by either the MVE or MCD estimator, outlying instances will not skew the estimates and can be identified as outliers by large values of the Mahalanobis distance. The most common cutoff point k is again the one based in a Ch-Square distribution, although Hardim and Rocke (2004) propose a cutoff point based on the F distribution that they claim to be a better one.

In this paper, two strategies to detect outliers using robust estimators of the Mahalanobis distances have been used: First, choose a given number of instances appearing at the top of a ranking based on their robust Mahalanobis measure. Second, choose as a multivariate outliers the instances that are tagged as outliers in the *Boxplot* of the distribution of these robust Mahalanobis distance.

Example 3. Find out the multivariate outliers in each of the classes of the *Iris* dataset by building boxplots for the distribution of the robust version of the Mahalanobis distance.
Using the **robout** function written in **R** (see appendix) and considering 10 repetitions the results appearing in the following tables have been obtained.

Table 1. Top outliers per class in the Iris dataset by frequency and the outlyingness measure using the MVE estimator

Instance	class	Frequency	Outlyingness
44	1	8	5.771107
42	1	8	5.703519
69	2	9	5.789996
119	3	8	5.246318
132	3	6	4.646023

Table 2. Top outliers in class 1 by frequency and the outlyingness measure using the MCD estimator

Instance	Class	Frequency	Outlyingness
44	1	10	6.557470
24	1	10	5.960466
69	2	10	6.224652
119	3	10	5.390844
132	3	7	4.393585

Notice that both methods detect two outliers in the first class, but the MVE method detects the instance 42 as a second outlier whereas the MCD method detects the instance 24. All the remaining outliers detected by both method are the same. Three more outliers are detected in comparison with the use of the Mahalanobis distance.

The figure 7 shows a plot of the ranking of the instances in class 3 of Iris by their robust Mahalanobis distance using the MVE estimator.

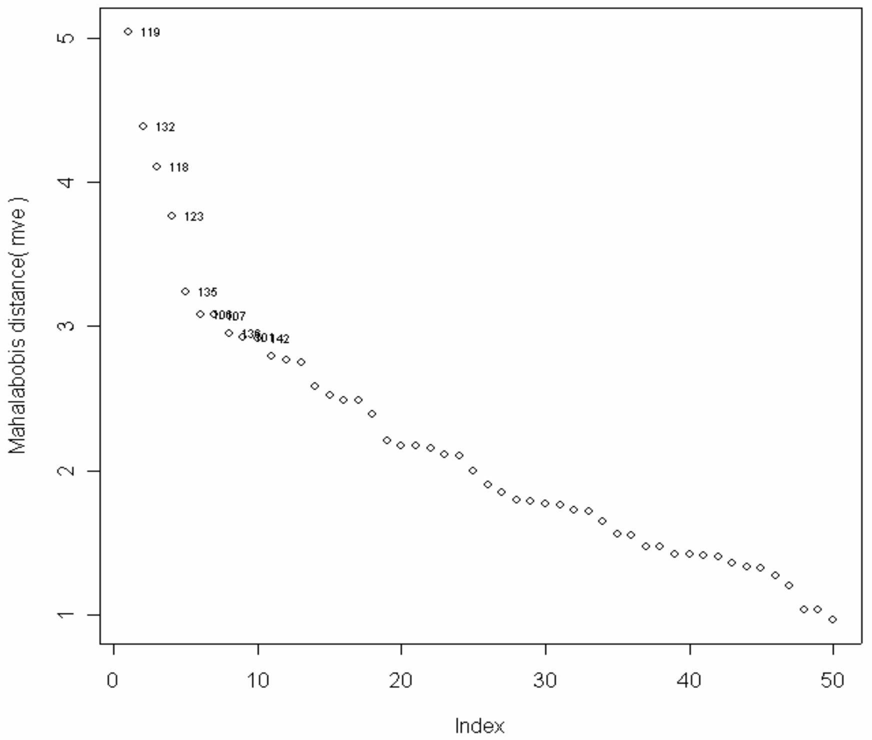

Figure 7. Plot of the instances ranked by their Mahalanobis distance using MVE estimator

The Figure 8 shows a plot of the ranking of the instances in class 3 of Iris by their robust Mahalanobis distance using the MCD estimator. According to Rocke (2002) Robust methods work well detecting scattered outliers but fail to detect clustered outliers. For this type of outlier is better to use a clustering algorithm as will be discussed in the next section.

3.2. Detection of outliers using clustering

A clustering technique can be used to detect outliers. Scattered outliers will form a cluster of size 1 and clusters of small size can be considered as clustered outliers. There are a large number of clustering techniques. In this paper only the Partitioning around Medoids (PAM) method will be considered. PAM was introduced by Kaufman and Rousseeuw (1990) uses k-clustering on medoids to identify clusters. It works efficiently on small data sets, but is extremely costly for larger ones. This led to the development of CLARA

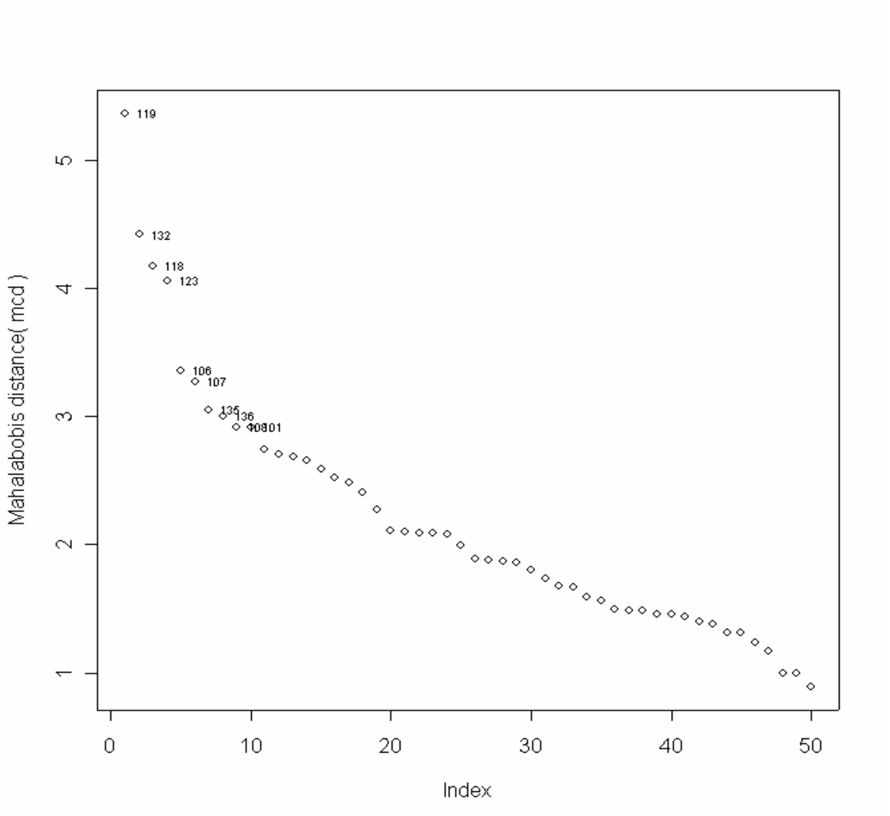

Figure 8. Plot of the instances ranked by their Mahalanobis distance using MCD estimator

(Clustering Large Applications) (Kauffman and Rousseuw, 1990) where multiple samples of the data set are generated , and then PAM it is applied to the sample. CLARA chooses the best clustering as the output, basing quality on the similarity and dissimilarity of objects in the entire set, not just the samples. A modification of CLARA that is applied to very large datasets is CLARANS (Ng and Han, 1994).

Given k, the number of partitions to construct, a partitioning method creates an initial partitioning. It then uses an iterative relocation technique that attempts to improve the partitioning by moving instances from one group to another. The general criterion of good partitioning is that instances in the same cluster are 'close' or related to each other, whereas instances of different clusters are 'far apart' or very different.

In order to find k clusters, PAM's approach is to determine a representative instance for each cluster. This representative instance called *medoid*, is meant to be the most centrally located instance within the cluster. More specifically, a *medoid* can be defined as that instance of a cluster, whose average dissimilarity to all the objects in the cluster is minimal. After finding the set of *medoids*, each object of the data set is assigned to the nearest *medoid*.

If O_j is a non selected instance and O_i is a selected medoid, we say that O_j belongs to cluster represented by O_i if $d(O_i,O_j) = min_{Oe}\, d(Oj,Oe)$ where the minimum is taken over all medoids *Oe*, and $d(O_a,O_b)$ denotes the dissimilarity or distance between instances O_a and *Ob*.

The algorithm PAM consists of two steps:

i)The BUILD-step: This step sequentially selects *k* centrally located instances, to be used as initial medoids

ii) The SWAP-step: If the objective function J = Σ $d(i,\, mv_{\mathrm{i}})$, which is the sum of the dissimilarities of all instances to their nearest medoid *mv* can be reduced by interchanging (swapping) a selected object with an unselected object, then the swap is carried out. This is continued until the objective function J can no longer be decreased.

There are *k(n-k)* possible pairs of $(O_i\,, O_h)$. For each pair computing J requires the examination of *(n-k)* non-selected instances. Thus the combined complexity is: $O(k(n-k)^2)$. Hence PAM becomes too costly for large values of *n* and *k*.

PAM is very robust to the presence of outliers and do not depend on the order in which instances are examined.

After the allocation of the instances to the k clusters, one must determine the *separation* between them. The *separation* of the cluster *C* is defined as the smallest dissimilarity between two objects; one of which belong to Cluster *C* and the other does not.

$$\mathrm{Separation}_C = \min d_{\mathrm{lh}},\ l \in C,\ h \notin C$$

If the separation of a outlier is large enough then it is considered all the instances that belong to the cluster are considered outliers. In order to detect the clustered outliers one must vary the number k of clusters until obtain clusters of small size and with a large separation from others clusters.

The algorithm PAM can be evaluated using the function **pam** available in the library **cluster** in R.

Example 4: Find out the outliers of the Iris dataset using the PAM algorithm.
Looking at the separation measures of ten clusters generated for each class, the detected outliers are shown in the table 3.

Table 3. Outliers in the Iris dataset according to the PAM algorithm

Instance	Class	Separation
42	1	0.6244998
58	2	0.6480741
61	2	0.6480741
94	2	0.6480741
99	2	0.6480741
107	3	0.9110434
118	3	0.8185353
132	3	0.8185353

Notice that in the class 3, PAM detects the instance number 107 as an outlier but it does not detect the instance 119.

3.3. Distance based outlier detection

Given a distance measure on a feature space, two different definitions of distance-based outliers are the following.

1. An instance **x** in a dataset D is an outlier with parameters p and λ if at least a fraction p of the objects are a distance greater λ from **x**. (Knorr and Ng, 1997, 1998, Knorr et al. 2000). This definition has certain difficulties such as the determination of λ and the lack of a ranking for the outliers. Thus an instance with very few neighbors within a distance λ can be regarded as stronger outlier as a an instance with more neighbors within a distance λ. Furthermore, the time complexity of the algorithm is O(kn^2), where k is the number of features and n is the number of instances. Hence it is not an adequate definition to use with datasets having a large number of instances.

2. Given the integer numbers k and n (k<n). Outliers are the top n instances with the largest distance to their k-th nearest neighbor. (Ramaswamy et al., 2000). One shortcoming of this definition is that only considers the distance to the k-th neighbor and ignores information about closer points. An alternative is to use the average distance to the k nearest neighbors is greatest. The drawback of this alternative is that it takes longer to be calculated.

In this paper a variant of one recently algorithm (Bay and Schwabacher, 2003) for distance-based outliers detection has been used.

The Bay's Algorithm.

Bay and Schwabacher (2003) proposed a simple nested loop algorithm that tries to conceal definitions 1 amd 2 . It gives linear time performance when the data is in random order and a simple pruning rule is used. The performance of the algorithm in the worst case is of quadratic order. The algorithm is described in Figure 9.

The main idea in the algorithm is that for each instance in D one keeps track of the closest neighbors found so far. When an instance's closest neighbors achieve a score lower than a cutoff then the instance is removed because it can no longer be an outlier. In this paper the score function used has been the median distance to the k neighbors. Bay used the average distance to the k neighbors, but the median is more robust than the mean. As more instances are processed the algorithm finds more extreme outliers and the cutoff increases along with pruning efficiency.

Bay and Schwabacher showed experimentally that the Bay's algorithm is linear with respect to the number of neighbors and that is almost linear with respect to the number of instances. Using 6 datasets they found a complexity of order $O(n^{\alpha})$ where α varied from 1.13 to 1.32. In this paper working with three datasets *Ionosfera*, *Vehicle* and *Diabetes* an α value near to 1.5 has been obtained.

A **bayout** function (see appendix) has been written in **R** language to perform Bay's algorithm.

Example 5. Find out the outliers of the class 3 in the *Iris* dataset using the Bay's algorithm.

Using the **bayout** function the top 20 outliers are shown in Figure 10.

Clearly the instance 107 is detected as an outlier. There is a second group that includes 119, 120,132,123 and 118.

```
Input: k: number of  nearest neighbors; n: number of outliers to return; D: dataset randomly ordered,
BS: size of blocks in which D is divided.
Let distance(x,y) return the Euclidean distance between x and y.
Let maxdist(x,Y) return the maximum distance between the instance x and the set of instances Y.
Let Closest(x,Y,k) return the k closest instances in Y to x.
Let score(x) return median distance to the k neighbors
Begin
c← 0 Set the cutoff for pruning to 0.
O←ϕ Initializate the set of outliers as the empty set.
NB←ceiling(# instances in D/BS)
    While nb<NB {
     Neighbors(b)←ϕ for all b in B(nb)
     For each d in D {
         For each b in B, b≠d{
             If |Neigbors(b)|<k or distance(b,d)<maxdist(b,Neighbors(b)){
                 Neighbors(b) ←Closest(b,Neighbors(b)Ud,k)
            If(score(Neighbors(b),b)<c{
             Remove b from B(nb)
             }}}}
   O←Top(B(nb) ∪ O, n)  ; Keep only the top n outliers
   c←min(score(o)) for all in O ; The cutoff is the score of the weakest outlier
}
end
Output: O, a set of outliers
```

Figure 9. Bay's Algorithm for finding distance-based outliers

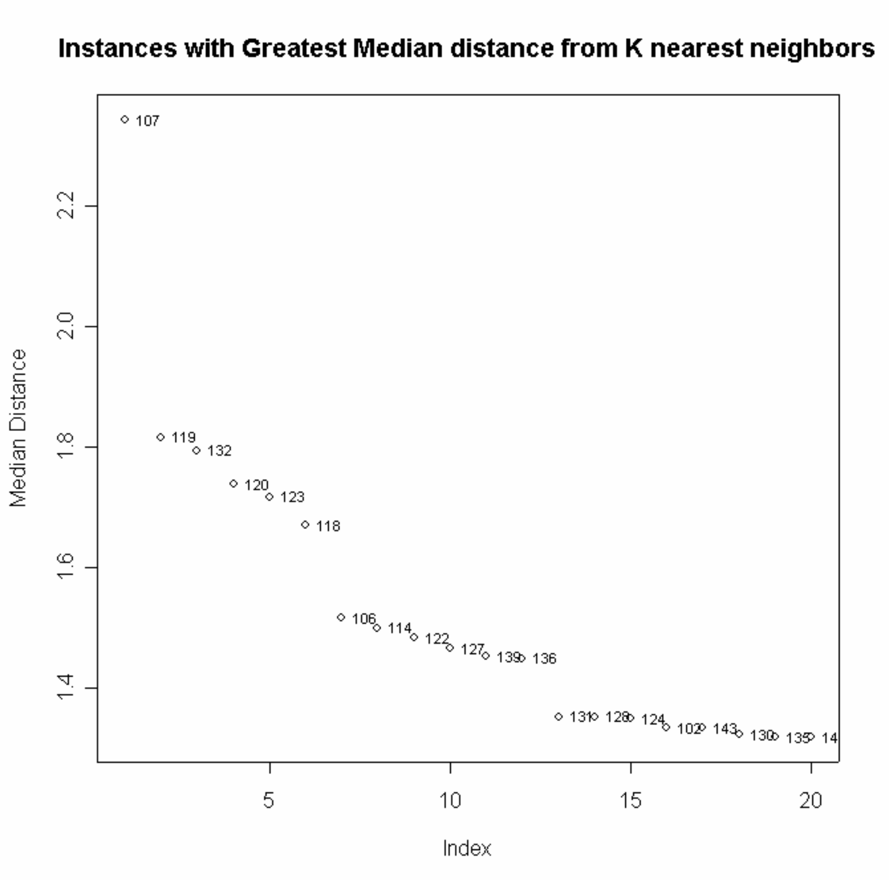

Figure 10. Instances of the class 3 in Iris dataset ranked by the Bay's algorithm outlyingness measure

3.4. Density-based local outliers.

In this type of outliers the density of the neighbors of a given instance plays a key role. Furthermore an instance is not explicitly classified as either outlier or non-outlier instead for each instance a local outlier factor (LOF) is computed which will give an indication of how strong an instance can be an outlier.

The following figure taken from Breuning et al (2000) shows the weakness of the distance-based method which identify as outlier the instance o_1, but does not consider o_2 as an outlier.

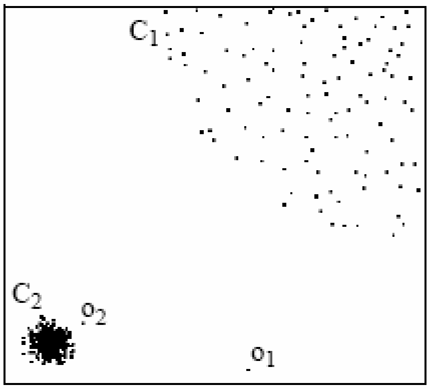

Figure 11. Example to show the weakness of the distance-based method to detect outliers

Several definitions are needed in order to formalize the algorithm

Definition 1. *k-distance of an instance x*

For any positive integer k, the k-distance of an instance x, denoted by k-distance(x), is defined as the distance d(x,y) between x an instance y ε D such that:

(i) for at least k instances y' ε D-{x} it holds that d(x,y') ≤ d(x,y)

(ii) for at most k-1 instances y' ε D-{x} it holds that d(x,y') < d(x,y).

Definition 2. *k-distance neighborhood of an instance x*

Given an instance x of a data set D its k-distance neighborhood contains every instance whose distance from x is not greater than the k-distance. That is, the set of k-nearest neighbors of x is given by

$$N_{k\text{-distance}(x)}=\{y \, \varepsilon \, D-\{x\} \text{ s.t. } d(x,y) \leq k\text{-distance}(x)\} \quad (6)$$

Definition 3. *Reachability distance of an instance x w.r.t. instance y*

Let k be a positive integer number. The reachability distance of the instance x with respect to the instance y is defined as

$$\text{reach-dist}_k(x,y)=\max\{k\text{-distance}(y),d(x,y)\} \quad (7)$$

The density-based local algorithm to detect outliers requires only one parameter, *MinPts*, which is the number of nearest neighbors used in defining the local neighborhood of the instance.

Definition 4. *Local reachability density of an instance x*

Given an instance x of a dataset D its local reachability density is defined by

$$lrd_{MinPts}(x) = 1/\{\frac{\sum_{y \in N_{MinPts}(x)} reach - dist_{MinPts}(x, y)}{| N_{MinPts}(x) |}\} \quad (8)$$

This is the inverse of the average reachability distance based on the *MinPts*-nearest neighbor de x. Finally the definition of the outlyingness measure is given below.

Definition 5. *Local outlier factor (LOF) of an instance x*

$$LOF_{MinPts}(x) = 1/\{\frac{\sum_{y \in N_{MinPts}(x)} \frac{lrd_{MinPts}(y)}{lrd_{MinPts}(x)}}{| N_{MinPts}(x) |}\} \quad (9)$$

The LOF measures the degree to which an instance x can be considered as an outlier. Breunig et al show that for instances deep inside a cluster their LOF's are close to 1 and should not be labeled as a local outlier. Since LOF is not monotonic, Breuing et al recommended to find the LOF for each instance of the datasets using MinPts-nearest neighbor, where MinPts assumes a range of values from MinPtsLB to MinPtsUB. They suggest MINPtsLB=10 and MinPtsUB=30. In this paper the default value of MinPtsUB was chosen to be 20. Having determined MInPtsLB and MinPtsUB, the LOF of each instance is computed within this range. Finally all the instances are ranked with respect to the maximum LOF value within the specified range. That is, the ranking of an instance x is based on:

$$\text{Max}\{\text{LOF}_{MinPts}(\text{x}) \text{ s.t. } MinPtsLB \le MinPts \le MinPtsUB\} \quad (10)$$

A **maxlof** function (see appendix) has been written in **R** language to perform the LOF algorithm. The LOF algorithm is shown in figure 12.

Breunig et al. states that the time complexity of the LOF depends of the dimensionality of the data. For low dimensionality data (no more than 5 features) the complexity is O(n),for medium dimensionality data (between 5 and 10 features) the complexity is O(nlogn) and for very high dimensional data the complexity is O(n^2). Using the *Ionosfera* dataset, which has 32 features, the time complexity estimated was O($n^{1.95}$).

```
Input: Dataset D, MinptsLB, MinptsUB
Maxlofevct=ϕ
For each   i in the interval [MinPtsLB, MinPtsUB]
{
1. Find the i nearest neighbors and their distance from each observation in D
2. Calculate the local reachability density for each observation in D
3. Compute the lof of each observation in D
4. Maxlofvect=max(maxlofvect, lof)
  }
end
Ouput: Maxlofvect
```

Figure 12. The LOF Algorithm

Example 6. Find out the outliers of the class 3 in the *Iris* dataset using the LOF algorithm.
Using the **maxlof** function the top 10 outliers are shown in Figure 13. Clearly the instance 107 is detected as an outlier. There is a second group that includes 119, 118,132 and 123. After that it comes instance 106.

4. Effect and treatment of outlier in supervised classification.

In the literature is frequently mentioned that the presence of outliers affects the performance of a classifiers, but there are few studies verifying such claim. This is not the case in a regression context where are a large number of studies showing the effect of outliers in regression problems
Two main aspects in supervised classification are feature selection and the misclassification error rate. In this paper an evaluation of the effect of outliers in the those aspects is considered.

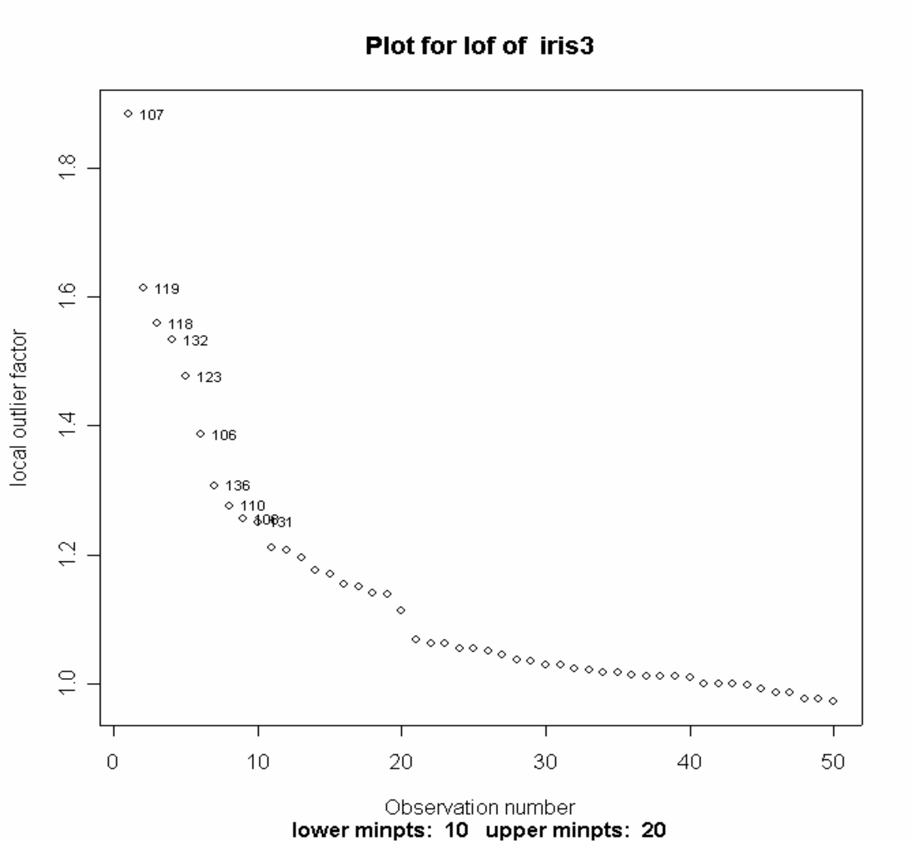

Figure 13. Instances of the class 3 in the *Iris* dataset ranked by the LOF's algorithm outlygness measure

The *Bupa* dataset , which have 345 instances, 6 features and 2 classes will be used to show the effect of outliers.

Example 7. Use the *Bupa d*ataset to show the effect of outliers in feature selection and the estimation of the misclassification error rate.

Using all the criteria described in the previous section and with the help of a parallel coordinate plot to decide about the doubtful outliers the following outliers have been detected in the *Bupa* dataset.

Outliers in class 1: (22)

168 175 182 190 205 316 317 335 345 148 183 261 311 25 167 189 312 326 343
313 20 22

Outliers in class 2 (26)

[1] 36 77 85 115 134 179 233 300 323 331 342 111 139 252 294 307 123 186 286
[20] 2 133 157 187 224 278 337

In Table 4 the misclassification error of three classifiers: LDA, knn and rpart had been computed based on: the original sample, the original sample without outliers and the original sample extracting from her a random sample of size equal to the number of outliers.

Table 4. The misclassification error rate for the LDA, knn and rpart classifiers using three different type of samples

	Original Sample	Original sample without outliers	Original Simple minus a random sub-sample
LDA	31.82	26.23	31.17
Knn(k=7)	31.55	27.65	32.26
Rpart	31.86	33.24	35.07

Notice that LDA and knn are the most affected classifiers and the least affected classifier has been the Rpart. The later makes sense since it is well known that Rpart is a classifier that is robust to outliers. Table 5 shows the feature selected using the three type of samples described before. The feature selected methods uses here are the sequential forward selection (SFS) with the three classifiers used in table 4 and the Relief

Table:5. Features selected using SFS and Relief for the three type of samples

	Original Sample	Original sample without outliers	Original sample minus a random sub-sample
SFS(lda)	5,4,3,6	5,3,4	5,4,3,6
SFS(knn)	5,3,1	5,3,1,6	5,3,4,1
SFS(rpart)	5,3,6,2	5,3,2	5,3,2
Relief	6,3,4	4,2,5,3	2,4,3

There are differences between the subset of features selected by the four methods. Finally in table 6 the misclassification error rates of the three classifiers after feature selection and for the three type of samples.

Table 6. Misclassification error rate after feature selection for the three type of samples

	Original Sample	Original sample without outliers	Original sample minus a random subsample
LDA	34.94	26.72	35.62
knn(k=7)	36.53	30.65	40.99
Rpart	37.47	32.48	39.78

Notice that the lower misclassification error are obtained for a sample where the feature selection is performed after eliminating outliers.

Another option to delete outliers is to treat them as missing values. Some people prefer the latter because avoid the loss of sample size but other people does not like it too much because it can create bias on the estimation.

References

1. Atkinson, A. (1994). Fast very robust methods for the detection of multiple outliers. Journal of the American Statistical Association, 89:1329-1339.

2. Barnett, V. and Lewis, T. (1994). *Outliers in Statistical Data.* John Wiley.

3. Bay, S.D., and Schwabacher (2003). Mining distance-based outliers in near linear time with randomization and a simple pruning rule

4. Breuning, M., Kriegel, H., Ng, R.T, and Sander. J. (2000). LOF: Identifying density-based local outliers. In Proceedings of the ACM SIGMOD International Conference on Management of Data.

5. Hadi, A. (1992). Identifying multiple outliers in multivariate data. Journal of the Royal Statistical Society B, 54:761-771.

6. Hawkins, D. (1980). *Identification of Outliers*. Chpaman and Hall. London.

7. Kaufman, L. and Rousseeuw, P.J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis.* Wiley, New York.

8. Knorr, E. and Ng, R. (1997). A unified approach for mining outliers. In Proc. KDD, pp. 219–222.

9. Knorr, E., and Ng. R. (1998). Algorithms for mining distance-based outliers in large datasets. In Proc. 24th Int. Conf. Very Large Data Bases, VLDB, pages 392–403, 24–27.

10. Knorr., E., R. Ng, and V. Tucakov. (2000). Distance-based outliers: Algorithms and applications. VLDB Journal: Very Large Data Bases, 8(3–4):237–253.

11. Ng, R.T. and Han, J. (1994). Efficient and effective clustering methods fro spatial data mining. Proc. 20th Int.Conf. on Very Large Data bases. Morgan and Kaufmann Publishers, San Francisco, 144-155.

12. Ramaswamy, S., Rastogi, R., and Shim, K. (2000). Efficient algorithms for mining outliers from large data sets. In Proceedings of the ACM SIGMOD International Conference on Management of Data.

13. Rocke, D. and Woodruff, D. (1996). Identification of outliers in multivariate data. Journal of the American Statistical Association, 91:1047-1061.

14. Rousseeuw, P (1985). Multivariate estimation with high breakdown point. Mathematical statistics and applications.

15. Rousseeuw, P. and Leroy, A. (1987). *Robust Regression and Outlier Detection*. John Wiley.

16. Rousseeuw, P. and Van Zomeren, B. (1990). Unmasking multivariate outliers and leverage points. Journal of the American Statistical Association, 85:633-639.

17. Rousseeuw, P. J. & Van Driessen, K. (1999). A Fast Algorithm for the Minimum Covariance Determinant Estimator. Technometrics, 41, 212-223.

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