Robust Double Clustering

Alessio Farcomeni*

February 27, 2007

Abstract

We propose two algorithms for robust two-mode partitioning of a data matrix in the presence of outliers. First we extend the robust k-means procedure of Gallegos and Ritter (2005) to the case of biclustering, then we slightly relax the definition of outlier and propose a more flexible and parsimonious strategy, which anyway is inherently less robust. We investigate the breakdown properties of the algorithms, and illustrate the methods with simulations and three real examples.

Keywords: biclustering, double clustering, microarrays, robustness, outliers

1 Introduction

Let X be an n by p observed data matrix. Classical (nonhierarchical) methods for cluster analysis deal with the identification of similarities among the rows of X, by claiming the existence of $I << n$ groups with close characteristics.

There are many applications nevertheless in which one may want to cluster both rows and columns. For instance in DNA microarrays analysis the observed expression of p genes on n slides is recorded, and while clustering of the genes is of primary interest, clustering of the slides leads to identification of groups among the patients and is of interest too. Other applications

*The author is grateful to Prof. Vichi for support and careful reading of a first draft. Acknowledgements go also to Francesca Martella for advice.

include marketing (for instance, clustering of customers and goods), biology, psychology, sociology. A possible approach is to separately cluster the rows and the columns. It is known anyway that this approach does not allow to specify an overall objective function, thus lacking optimality properties; and furthermore ignores the dependence structure between rows and columns. Following an idea that dates back at least to Fisher (1969); Hartigan (1972), one may find more appropriate to perform a *simultaneous* clustering of rows and columns. This is called *biclustering* or *double clustering* in the literature, see Van Mechelen *et al.* (2004); Madeira and Oliveira (2004) for a review.

While there are now many competing strategies for performing double clustering in many different situations, to the best of our knowledge there still are no studies about the robustness properties of double clustering, and very few attempts to robustify the procedures. This is crucial, since double clustering is often applied to large data matrices in which *contamination* is very likely to occur. In DNA Microarrays for instance the measuring technique itself is well known to be likely to regularly lead to the presence of outliers. Ignoring this problem may lead to estimates, and consequently groupings, that are unduly different from the true underlying structure of the uncorrupted data.

In this paper we deal with double clustering in the presence of outliers, and focus on the class of methods known as "double k-means" (Vichi, 2000). We thus assume that the generic element of the data matrix x_{ij} is a real number.

It may be argued that outliers could be isolated by considering a higher number of row and column clusters. Recall nevertheless that (i) the number of clusters must be set in advance, and that we could have not taken into account the presence of contamination; (ii) stability of the algorithms can be seriously affected by unrecognized outliers and (iii) the output of small clusters, or even singletons, may lead to problems in interpretation and use of the clustering for future prediction, classification and/or resource allocation. In marketing research, for instance, it is often the case that clusters serve for the definition of few different marketing strategies for possibly large groups of customers. Furthermore, data driven methods for setting the number of clusters may easily break down, and the curse of dimensionality may make it extremely difficult to find the right number of clusters. It often happens that $n >> p$ or $n << p$, so at least one between I and J would be particularly hard to choose without having the genuine clusters contaminated by outliers.

Robust (single-mode) nonhierarchical clustering of a data matrix was

considered among others by Kaufman and Rousseeuw (1990) (with the k-medoids method), Cuesta-Albertos *et al.* (1997), Garcia-Escudero and Gordaliza (1999), Gallegos and Ritter (2005). The former methods, except from the last, rely on α-trimming rather than direct identification of outliers. We prefer choosing the number of outliers in this work mainly because we want to identify the outliers in addition to robustification of centroid estimation. Trimming methods (Cuesta-Albertos *et al.*, 1997; Garcia-Escudero and Gordaliza, 1999) look for the $\lceil n(1-\alpha)\rceil$ observations closer to the centroid in each cluster. It is then assumed that *all* clusters are contaminated with about the same amount of outliers. The approach of Gallegos and Ritter (2005) instead discards a pre-specified *total* number of outliers o_1, which can be distributed equally among the clusters or can belong to the same cluster if needed. It is easy to modify the proposed algorithms to accommodate trimming methods, which are intuitively seen to be more robust and less flexible.

Gallegos and Ritter (2005) define a row outlier as a row of the data matrix $x_i \in \mathcal{R}^p$ for which there is at least a contamination. The implicit idea is that an outlier is an object far from its closest centroid. We will extend a simpler form of their procedure to the case of double clustering in Section 2. In Section 3 we will slightly relax the definition of outlier, providing a second algorithm. The two algorithms can be combined, as we note in Section 3.1. In Section 4 we discuss a novel strategy to choose the number of outliers. We point out that this strategy may be useful also for single-mode robust clustering methods. In Section 5 we study the breakdown points of our newly proposed procedures. In Section 6.1 we provide some simulation and real data applications are illustrated in Section 6.2. Some final remarks are given in Section 7.

1.1 Notation

We introduce now the notation we will follow throughout the paper:

$x_{i\cdot}$: i-th row of the data matrix X

$x_{\cdot j}$: j-th column of the data matrix X

n: the number of rows of the data matrix X

p: the number of columns of the data matrix X

I: the (user-specified) number of row clusters

J: the (user-specified) number of column clusters

o_1: the (user-specified) number of row outliers

o_2: the (user-specified) number of column outliers

n_r: the number of rows estimated to belong to the r-th row cluster

p_c: the number of columns estimated to belong to the c-th column cluster

2 Robust Double k-means

We will use a very common model for double clustering (Van Mechelen *et al.*, 2004):

$$X = U\bar{x}V' + E, \tag{1}$$

the only difference being that we allow for the presence of outliers. $\bar{x}$ is an IxJ matrix of centroids, while the nxI matrix U and the pxJ matrix V are binary and the row sums are less than or equal to 1; and E is a residual error term on which we do not make any distributional assumption. This setting called object and variable packing in Vichi (2000). Our first proposal for robust double clustering is a specific model of the class defined in Vichi (2000); in which we force o_1 rows of U and o_2 rows of V to sum exactly to zero, and the others to sum exactly to one. In this way we mark o_1 rows and o_2 columns as outliers: they are not formally assigned to any of the I row clusters or J column clusters, and we do not use them for estimation of the centroid matrix.

The robust double clustering model can be fit with an alternating least squares procedure. We will use a random initialization and a multistart for the algorithms in this paper. Given current estimates for U, $\bar{x}$ and V, the general iteration is given in Algorithm 1.

A short description of its rationale follows. First, we update the row memberships. For each row we compute I weighted distances from the r-th group $d_r^2(i)$. This distance is the sum of the distances between the i-th row and the r-th row of the centroid matrix, weighted with the number of columns involved in each column group. The current column outliers are not used in the computation. We then assign each row to the closest row group. Then, we determine the row outliers by not assigning to groups the o_1 rows most distant from the closest centroid. Note that the algorithm, in

Algorithm 1 Robust Double k-means

Update the row memberships

Let $d_r^2(i,c) = \sum_{j=1}^{p}(x_{ij}v_{jc} - \bar{x}_{rc})^2$, $i = 1,\ldots,n$, $c = 1,\ldots,J$, $r = 1,\ldots,I$.

Let $d_r^2(i) = \sum_c d_r^2(i,c)p_c$.

Let $r_i = \arg\min_r d_r^2(i)$.

Set $U_{ir_i} = 1$. All the other elements of the i-th row of U are set to 0.

Determine the row outliers

Determine a permutation $k : \{1,\ldots,n\} \to \{1,\ldots,n\}$ such that

$$d^2_{r_{k(1)}}(k(1)) \leq d^2_{r_{k(2)}}(k(2)) \leq \cdots \leq d^2_{r_{k(n)}}(k(n)).$$

Set $U_{k(n-o_1+1).} = U_{k(n-o_1+2).} = \cdots = U_{k(n).} = \mathbf{0}$.

Update the column memberships

Let $d_c^2(j,r) = \sum_{i=1}^{n}(x_{ij}u_{ir} - \bar{x}_{rc})^2$, $j = 1,\ldots,p$, $c = 1,\ldots,J$, $r = 1,\ldots,I$.

Let $d_c^2(j) = \sum_r d_c^2(j,r)n_r$.

Let $c_j = \arg\min_c d_c^2(j)$.

Set $V_{jr_j} = 1$. All the other elements of the j-th row of V are set to 0.

Determine the column outliers

Determine a permutation $z : \{1,\ldots,p\} \to \{1,\ldots,p\}$ such that

$$d^2_{c_{z(1)}}(z(1)) \leq d^2_{c_{z(2)}}(z(2)) \leq \cdots \leq d^2_{c_{z(n)}}(z(n)).$$

Set $V_{z(p-o_2+1).} = V_{z(p-o_2+2).} = \cdots = V_{z(p).} = \mathbf{0}$.

Estimate the centroid matrix

$\bar{x} = (U'U)^{-1}U'XV(V'V)^{-1}$.

parallel with Gallegos and Ritter (2005) suggests to sort the minimum distances, but looking o_1 times for the maximum may be more computationally advantageous.

We do the same for the column memberships and outliers, and finally estimate the centroid matrix by least squares.

Robustness stems from the fact that outliers are not involved in the computation of the distance, and then do not contribute either to classification or estimation of the centroid matrix.

We stress that outliers are unusual observations *within* their cluster. This the main motivation behind the proposal of simultaneous clustering and outlier detection, and leads to a drastic reduction of masking effects with respect to outlier identification *before* cluster analysis.

When we update the row and column memberships, all the objects are assigned to a group; and only later some of them are marked as outlying. We can use this fact to see which is the closest block for a given outlier, and similarly to realize with respect to which cluster an observation is outlying. The most important fact is that we do not use outlying observations for centroid estimation.

Gallegos and Ritter (2005) minimize a criterion based on the Mahalanobis distance; which is usually justified in the setting of model based clustering (Bock, 1996; Fraley and Raftery, 2002). The idea stems from the minimum covariance determinant (MCD) approach (Rousseeuw, 1984; Rousseeuw and Van Driessen, 1999; Hardin and Rocke, 2004), who show that by successively ordering of observations in terms of Mahalanobis distance, and discarding of the highest distances, one achieves convergence to the empirical covariance matrix with smallest determinant for the non-discarded observations. The steps of the algorithm are usually called "concentration" steps, and always yield a decrease in the determinant of the covariance matrix of the non-discarded observations. Unlike these works we favor a least squares approach, thus avoiding the use of the sum of squares and products matrix. This is particularly computationally advantageous when partitioning the columns, if n is large; or when partitioning the rows if p is large. The biggest drawback is that we do not have the additional flexibility of allowing for ellipsoidal clusters, but only look for spherical clusters. Of course, standardization of X is always useful since we also are implicitly assuming that the groups are equally scattered, as is common for k-means and trimmed k-means. These assumptions are quite strong but are seen to yield stability and good generalization properties of the estimated centroid, provided they are not grossly

violated.

Another possibility is to consider the Mahalanobis distance for partitioning the (row or column) vectors of smaller dimensionality and a least squares approach for partitioning the other mode; and generalization of the algorithms to this case is straightforward.

3 Double Clustering with Double Labeling of Outliers

The approach to double clustering of Algorithm 1 is robust with respect to contamination (see Section 5), but may lead to discard valuable information. If in fact one single dimension for a (row or column) vector is contaminated, the entire vector may (and hopefully will) be marked as outlying. The loss of information may not be negligible when the number of vectors is small. We argue that uncorrupted dimensions of the marked vector can still be used as a valuable and reliable information for the clustering, and thus propose here a more careful algorithm that makes use of a *double* labeling: each row and column will be marked as belonging to one of the groups, and will also be marked as being outlying or not.

If an object is not contaminated, then it is assigned to a group and all its dimensions contribute to the estimate of the corresponding centroid. If few dimensions of an object are corrupt, it is marked as being an outlier, but *only for the corrupt dimensions*. The uncorrupted dimensions contribute to the estimate of the corresponding centroid.

This is also useful for the sake of identifying outliers, in that we will be able to explain which dimensions contribute to classify an object as outlying, hence enhancing interpretability.

The same algorithm would result from an approach in which we look for contaminated entries, instead of contaminated objects.

We will make use of two binary vectors, η and ϕ, respectively of length n and p. If η_i is 1, the p dimensions of row i will be used for clustering, otherwise only the dimensions corresponding to a non-zero ϕ will be used; and similarly for the column outliers.

Once again we can fit this model through an alternating least squares approach. Given current estimates for U, $\bar{x}$ and V, the general iteration is given in algorithm 2.

Algorithm 2 Robust Double k-means with double labeling

Update the row memberships

Let $d_r^2(i,c) = \sum_j \max(\eta_i, \phi_j)(x_{ij}v_{jc} - \bar{x}_{rc})^2$, $i = 1,\ldots,n$, $c = 1,\ldots,J$, $r = 1,\ldots,I$.

Let $d_r^2(i) = \sum_c \sum_j d_r^2(i,c) v_{jc} \max(\eta_i, \phi_j)$.

Let $r_i = \arg\min_r d_r^2(i)$.

Set $U_{ir_i} = 1$. All the other elements of the i-th row of U are set to 0.

Determine the row outliers

Let $\nu_i = \sum_j (x_{ij} - 1/J \sum_{c=1}^{J} \bar{x}_{r_i c})^2 (1 - \phi_j)$, $i = 1,\ldots,n$.

Determine a permutation $k : \{1,\ldots,n\} \to \{1,\ldots,n\}$ such that

$$\nu_{k(1)} \leq \nu_{k(2)} \leq \cdots \leq \nu_{k(n)}.$$

Set $\eta_{k(n-o_1+1)} = \eta_{k(n-o_1+2)} = \cdots = \eta_{k(n)} = 0$, and all the other elements to 1.

Update the column memberships

Let $d_c^2(j,r) = \sum_i \max(\phi_j, \eta_i)(x_{ij}u_{ir} - \bar{x}_{rc})^2$, $j = 1,\ldots,p$, $c = 1,\ldots,J$, $r = 1,\ldots,I$.

Let $d_c^2(j) = \sum_r \sum_i d_c^2(j,r) u_{ir} \max(\phi_j, \eta_i)$.

Let $c_j = \arg\min_c d_c^2(j)$.

Set $V_{jr_j} = 1$. All the other elements of the j-th row of V are set to 0.

Determine the column outliers

Let $\xi_j = \sum_i (1 - \eta_i)(x_{ij} - 1/I \sum_{r=1}^{I} \bar{x}_{rc_j})^2$, $i = 1,\ldots,n$.

Determine a permutation $z : \{1,\ldots,p\} \to \{1,\ldots,p\}$ such that

$$\xi_{z(1)} \leq \xi_{z(2)} \leq \cdots \leq \xi_{c_z(n)}.$$

Set $\phi_{z(p-o_2+1)} = \phi_{z(p-o_2+2)} = \cdots = \phi_{z(p)} = 0$, and all the other elements to 1.

Estimate the centroid matrix

$\bar{x}_{hk} = \frac{\sum_{ij} x_{ij}u_{ih}v_{jk} \max(\eta_i,\phi_j)}{\sum u_{ih}v_{jk} \max(\eta_i,\phi_j)}$, for $h = 1,\ldots,I$ and $k = 1,\ldots,J$.

The general ideas behind algorithm 2 are the same as those behind algorithm 1. When we update the row memberships, $d_r^2(i)$ is still a weighted sum of distances, but here we discard x_{ij} from the computation not just if the j-th column is marked as outlying, but only if both the i-th row and the j-th column are marked as outlying.

There is a slight difference in the determination of the outliers step. For each row, we compute the distance between each entry and its average centroid with respect to the column groups, but only for the columns that are marked as outlying. The reason is that we estimate, in the current iteration, that outliers can only be in that columns. If a row is far from its average centroid for the columns that contain the outliers, then it contains the outliers; and otherwise it does not.

The same approach is applied to determine the column memberships and outliers. Finally, the least squares estimate of the hk-th entry of the centroid matrix is given by the average of the elements of X belonging to the h-th row group *and* k-th column group whenever they are not marked as being outliers.

Remark 3.1. *This algorithm is more parsimonious than algorithm 1, anyway, often at the price of the need of a larger number of iterations before convergence. Furthermore, as we will see below it is inherently less robust.*

Remark 3.2. *The present procedure with $J = 1$ and $0 < o_2 < p$ is a generalization of Gallegos and Ritter (2005), in the sense that we do single-mode robust partitioning marking objects as only partially contaminated.*

3.1 Combining the algorithms

By combining the algorithm in Section 2 with the algorithm of Section 3, one can combine the flexibility of the double classification of certain objects with the enhanced robustness of marking entire vectors as contaminated. If an object is entirely or mostly corrupt, then it is a genuine outlier and belongs only to the set of outliers. If an object is corrupted only in a small number of dimensions, then it can still be clustered and the uncorrupted dimensions can be used to estimate the centroid matrix.

4 Choosing the number of outliers

There are four quantities to be chosen in advance in order to apply our algorithms. First, one must choose the number of row clusters I and the number of column clusters J. Choosing the number of clusters is still an open problem in cluster analysis, and we will not deal with it in this paper. We note that in confirmatory cluster analysis and in other situations at least one of the two is forced by the application. Furthermore, as pointed out by Climer and Zhang (2006), it often happens that a range of values for choosing the number of clusters are of interest, and the desireable magnitude is problem specific. To the best of our knowledge, the only available automatic method specifically designed for double clustering (in the absence of probabilistic assumptions) is the pseudo-F of Rocci and Vichi (2004) (see also Hartigan (1978)). We point out anyway that common methods for choosing k in k-means, as we note below, are easily generalized to double clustering; and that information criteria (AIC, BIC and similar) can be exploited in the presence of probabilistic assumptions.

In this section we discuss automatic strategies for choosing the number of outliers; which is not easy in our setting since we can not rely on probabilistic assumptions on the distribution of the data. We propose here a forward search method.

Let $\bar{x}(k)$ denote the k-th biggest value of the vectorized $\bar{x}$. For each possible configuration of outliers o_1 and o_2, we suggest evaluating

$$G(o_1, o_2) = \max_k |(\bar{x}(k) - \bar{x}_0(k))| / \max(|\bar{x}(k)|, |\bar{x}_0(k)|). \tag{2}$$

where $\bar{x}_0$ is the centroid estimated with $o_1 = o_2 = 0$. We order the entries of the two matrices to avoid problems related with label switching.

The G statistic measures the maximum relative discrepancy between the inclusion of no outliers and the use of o_1 row outliers and o_2 column outliers. We list a formal and an exploratory use of the G statistic (2).

First, it can be used in a forward fashion to choose the number of outliers, with the procedure of Algorithm 3.

At each step we evaluate the difference $G - G'$, that is, the standardized maximal change in the centroid matrix obtained by adding one row or column outlier. If this identifies an additional true outlier, there will be a relatively large change in at least one of the entries of the centroid matrix, say bigger than δ. Otherwise, there will be a small (lower than δ) change

Algorithm 3 Determination of o_1 and o_2

```
Choose a stopping value δ
Choose a step size s_1 and s_2
Initialize o_1 = o_2 = 0, set G' = 0.
while max(G(o_1 + s_1, o_2), G(o_1, o_2 + s_2)) − G' > δ do
  if G(o_1 + s_1, o_2) ≥ G(o_1, o_2 + s_2) then
    Set G' = G(o_1 + 1, o_2). Increment o_1 by s_1.
  else if G(o_1, o_2 + s_2) > G(o_1 + s_1, o_2) then
    Set G' = G(o_1, o_2 + s_2). Increment o_2 by s_2.
  end if
end while
```

in the centroid matrix. The choice of δ depends on the application and on the sample size. We suggest choosing smaller values of δ for large matrices, since a single outlier may not yield a big change in the centroid matrix in the presence of many observations. Unless stated otherwise, in this paper we will use $\delta = 0.05$, $s_1 = s_2 = 1$. Larger step sizes can be used to avoid being trapped in local maxima by the presence of close outliers, and to speed up the algorithm. We recommend in general using a relatively low δ, in order to possibly overestimate the number of outlying locations. A single undetected outlier can spoil the procedure, while a discarded uncontaminated entry usually yields little information loss.

Such automatic choice of the number of outliers may lead to an algorithm that fully embeds the spirit of robust statistics. If in fact there is no contamination, o_1 and o_2 will possibly be set to zero or close, so that the robust algorithm will give results comparable to the classical algorithm. If there is contamination then o_1 and o_2 will be greater than zero, and outliers will not lead to unreliable results.

Secondly, the G statistic can also be plotted in order to evaluate the level of contamination of the data, alternatively as a function of o_1 for fixed o_2, as a function of o_2 for fixed o_1; or a 3D plot can be generated.

Instead of the proposed statistic, one can also use any of the popular statistics for evaluating the quality of clustering, such as the average Silhouette (Kaufman and Rousseeuw, 1990). Many other possibilities are discussed in Milligan and Cooper (1985); Gordon (1999). It is worth noticing that these statistics should be extended to the double clustering situation, but in many cases this is straightforward.

A further possibility for choosing the number of outliers would be to measure the agreement between different choices of o_1 and o_2, using Cohen's kappa as in Reilly *et al.* (2005) or the Rand's C_k as in Chae *et al.* (2006). When the clusterings agree even when increasing o_1 or o_2, there either are close outliers or the outliers have all been sorted out. If the main aim is clustering rather than centroid estimation, this last strategy is more appropriate and easier.

5 Robustness Properties

There are many available methods to evaluate the robustness properties of a procedure (Huber, 1981; Hampel *et al.*, 1986). We will focus here on global measures of robustness, given by *breakdown values*. Hodges (1967); Donoho and Huber (1983) define a finite-sample breakdown value as the smallest fraction of outliers that can break down the estimate in a sample. For an estimator T and data set X we denote it with $\varepsilon_n^*(T, X)$. The asymptotic breakdown value (Hampel, 1971) is the breakdown value of a procedure for an infinite number of observations. The worst case scenario is given by an infinitesimal asymptotic breakdown value.

It is straightforward to extend the results of Garcia-Escudero and Gordaliza (1999) to see that classical double k-means has infinitesimal asymptotic breakdown value. In fact a single diverging entry suffices to make one centroid diverge, so that for the double k-means the highest breakdown value is $\varepsilon_n^*(\bar{x}, X) = 1/\max(n, p)$.

Consider now our procedures, with $o_1 > 0$ and $o_2 > 0$. Consider Algorithm 1, and suppose there are corrupted entries in $o_1 + o_2$ different rows. All these can be discarded, however maliciously they are put, by marking o_1 rows and the o_2 columns to which the remaining belong. If there is only one additional corrupted entry in a row/column combination that is not marked as outlying, then at least one corrupted entry is included in the estimate of the centroid matrix, and the procedure breaks down just like classical double k-means. So for procedure of Section 2 it must be that the highest possible breakdown value is $\varepsilon_n^*(\bar{x}, X) = \frac{o_1+o_2+1}{\max(n,p)}$. Consider now Algorithm 2, and suppose there are $o_2 + 1$ corrupted entries on a single row. Then at least one corrupted observation is included in the estimate of the centroid matrix. Similarly, if there are $o_1 + 1$ corrupted entries on a given column, at least one corrupted observation is included in the estimate of the centroid matrix.

Hence, for procedure of Section 3 it must be that $\varepsilon_n^*(\bar{x}, X) = \min(\frac{o_1+1}{n}, \frac{o_2+1}{p})$.

The actual breakdown point depends on the sample data structure, refer to Garcia-Escudero and Gordaliza (1999) for further comments on this issue. We have few additional remarks: first, the results for algorithm of Section 2 are well in agreement with the breakdown values obtained in Gallegos and Ritter (2005) for robust single-mode k-means. Second, if we let $o_1 = O(n)$ and $o_2 = O(p)$ the asymptotic highest breakdown value of algorithms 1 and 2 is strictly positive. To parallel trimming methods, one could set $o_1 = \lceil n\alpha_1 \rceil$ and $o_2 = \lceil p\alpha_2 \rceil$, for α_1 and α_2 small. Third, the procedure with double labeling is inherently less robust, but it still may be very useful for a carefully chosen number of outliers.

6 Illustration of the methods

6.1 Simulations

In this section we will compare the methods in simulation, reporting for each setting the modified Rand index (Hubert and Arabie, 1985) measuring concordance between estimated and true partition, the sum of the square error (SSE) in recovering the centroid matrix, and average time (in seconds) for each iteration. We report the rand index only for the originally uncontaminated rows and columns, thus overestimating the rand-index for the classical double-k-means.

The setting is as follows: we randomly partition rows and columns into I and J groups, without any control on the size of each group. We then generate the centroid matrix according to the formula: $\bar{x}_{hk} = (k-1) * I + h$. This way, the minimum distance between blocks is 1. We then generate the data simulating independent normals centered at the opportune entry of the centroid matrix, with two levels of noise ($\sigma = 0.1$ and $\sigma = 0.5$). For each setting we perform $B = 250$ iterations.

We simulate *without* contamination (Table 1), contaminating only o_2 columns for o_1 rows (Table 2), and entirely contaminating o_1 rows and o_2 columns (Table 3). When we include outliers, we generate them by randomly sampling from a normal with unit variance and mean, for the i-th outlier, equal to -10*i*S, where S is sampled from a Rademaker distribution ($\Pr(S = 1) = \Pr(S = -1) = 0.5$). We note that in this way the outliers are well separated, but in certain simulation settings they share the same

center of certain true clusters with the only difference of a larger spread; thus making more difficult their identification and the recovering of the true partition.

In the tables we report the results in parentheses for the classical double k-means, algorithm 1 and algorithm 2.

The tables seem to arise the following comments:

- Little information loss is observed when using robust procedures in the absence of contamination, especially with large matrices. Apart from the last three cases, in which the number of outliers is grossly overestimated, also the times needed for convergence are very close.
- When we partially contaminate some objects (Table 2), the classical algorithm breaks down for small and moderate matrix size, and for a few outliers also for big matrices. The robust algorithms perform very well, with as expected algorithm 2 preferable to algorithm 1. The differences between non-robust and robust algorithms are dramatic in terms of SSE.
- This behavior is more evident when we completely contaminate some objects (Table 3), with algorithm 1 preferable to algorithm 2 as could be expected.
- Algorithm 1 is always faster to convergence than Algorithm 2
- The classical method is often much slower to convergence than the robust methods in presence of outliers, likely due to instability.

6.2 Real Data Examples

6.2.1 Macroeconomic performance of industrialized countries

We first revisit the example of Vichi (2000) about the average macroeconomic performances of the G7 most industrialized countries: France (FRA), Germany (GER), Great Britain (GBR), Italy (ITA), United States of America (USA), Japan (JAP), Canada (CAN); plus Spain (SPA). The variables measured were: Gross Domestic Product index (GDP), Inflation (INF), Budget deficit/GDP (DEF), Public debt/GDP (DEB), Long term interest rate (INT), Trade balance/GDP (TRB), unemployment rate (UNE). The measurements refer to the period 1980-1990, and are of particular interest because

n	p	σ	I	J	o_1	o_2	$m-rand$	SSE	$time$
50	10	0.1	2	2	1	1	(1, 0.84, 0.84)	(10.68, 9.73, 9.35)	(0.31, 0.34, 0.38)
50	10	0.5	2	2	1	1	(0.98, 0.83, 0.83)	(10.02, 9.63, 11.03)	(0.36, 0.41, 0.86)
200	30	0.1	4	3	1	1	(1, 0.95, 0.95)	(289, 286, 300)	(5.1, 3.6, 3.9)
200	30	0.5	4	3	1	1	(1, 0.92, 0.92)	(296,295,289)	(5.3, 4.3, 4.7)
200	30	0.1	4	3	5	1	(1, 0.84, 0.84)	(289, 316, 281)	(5.1, 3.58, 3.93)
200	30	0.5	4	3	5	1	(1, 0.92, 0.92)	(296, 298, 269)	(5.3, 4.7, 4.5)
1000	80	0.1	10	5	1	1	(1,1.00,1.00)	(20168,20593,19240)	(42,48,50)
1000	80	0.5	10	5	1	1	(1,0.99,0.99)	(20537,19931,22512)	(42,69,81)
1000	80	0.1	10	5	5	1	(1,0.98,0.96)	(20168,21567,20281)	(41,81,89)
1000	80	0.5	10	5	5	1	(1,0.97,0.97)	(20537,20282,20146)	(42,106,151)
1000	80	0.1	10	5	10	2	(1,0.94,0.92)	(20168,20991,21779)	(41,80,91)
1000	80	0.5	10	5	10	2	(1,0.94,0.91)	(20537,20300,22502)	(42,97,121)

Table 1: Simulation of uncontaminated data. In parentheses, the results respectively for classical algorithm, algorithm 1 and algorithm 2.

n	p	σ	I	J	o_1	o_2	$m-rand$	SSE	$time$
50	10	0.1	2	2	1	1	(1, 1, 1)	(10.3, 9.9, 9.8)	(0.27, 0.17, 0.20)
50	10	0.5	2	2	1	1	(0.99, 0.98, 0.98)	(10.1, 9.4, 9.9)	(0.75, 0.48, 0.69)
200	30	0.1	4	3	1	1	(1.00, 0.99, 0.99)	(316, 289, 284)	(2.3, 2.1, 2.8)
200	30	0.5	4	3	1	1	(0.91, 0.98, 0.98)	(317,283,293)	(2.5, 3.0, 3.7)
200	30	0.1	4	3	5	3	(0.78, 0.79, 1.00)	(3903, 250, 292)	(3.2, 3.2, 4.3)
200	30	0.5	4	3	5	3	(0.67, 0.79, 0.94)	(5692, 303, 232)	(3.8, 6.3, 7.8)
1000	80	0.1	10	5	1	1	(1,1,1)	(20672,20589,19240)	(54,48,50)
1000	80	0.5	10	5	1	1	(0.99,0.99,0.99)	(20642,19932,22507)	(46,69,81)
1000	80	0.1	10	5	5	1	(1,0.99,1)	(20644,21421,19876)	(50,41,45)
1000	80	0.5	10	5	5	1	(1,0.99,0.99)	(20794,21817,20096)	(62,40,80)
1000	80	0.1	10	5	10	2	(0.95,0.93,1)	(21913,22216,21012)	(106,78,105)
1000	80	0.5	10	5	10	2	(0.89,0.93,0.99)	(21401,20531,20723)	(118,79,125)

Table 2: Simulation of partially contaminated data. In parentheses, the results respectively for classical algorithm, algorithm 1 and algorithm 2.

n	p	σ	I	J	o_1	o_2	$m-rand$	SSE	$time$
50	10	0.1	2	2	1	1	(0.73, 0.99, 0.77)	(187, 12, 163)	(0.35, 0.16, 0.52)
50	10	0.5	2	2	1	1	(0.57, 0.99, 0.58)	(236, 11, 54)	(0.57, 0.22, 6.27)
200	30	0.1	4	3	1	1	(1, 1, 0.98)	(500, 270, 275)	(4.9, 1.8, 8.2)
200	30	0.5	4	3	1	1	(0.94, 1, 0.96)	(300,278,263)	(7.9, 1.9, 16.9)
200	30	0.1	4	3	5	1	(0.71, 1, 0.73)	(6424, 280, 2307)	(7.9, 3.5, 22.1)
200	30	0.5	4	3	5	1	(0.71, 0.90, 0.72)	(6502, 2060, 2926)	(10.3, 4.7, 30.5)
1000	80	0.1	10	5	1	1	(1,1,1)	(21599,21172,21466)	(41,44,150)
1000	80	0.5	10	5	1	1	(0.98,1,0.99)	(21677,21426,22859)	(42,40,109)
1000	80	0.1	10	5	5	1	(0.94,1,0.92)	(32382,21611,28364)	(77,84,96)
1000	80	0.5	10	5	5	1	(0.90,1,0.91)	(36784,20564,25927)	(100,84,169)
1000	80	0.1	10	5	10	2	(0.61,1,0.70)	(77185,30800,61653)	(151,98,134)
1000	80	0.5	10	5	10	2	(0.52,0.90,0.50)	(84750,65838,69006)	(245,117,228)

Table 3: Simulation of completely contaminated data. In parentheses, the results respectively for classical algorithm, algorithm 1 and algorithm 2.

most of the variables were considered in the parameters of the Maastricht treaty. Full data are displayed in Table 4. It is apparent that the 8 countries have slightly different performances. Clustering can be used to explore the information, and it is of great interest to be able to *identify* the outliers. In particular it is questioned whether Italy can be thought of being an outlier after standardization.

Country	GDP	INF	DEF	DEB	INT	TRB	UNE
FRA	133.40	3.00	−1.50	46.60	10.40	−2.10	8.90
GER	138.80	3.40	−1.90	43.60	6.00	5.90	6.20
GBR	125.10	6.30	−1.30	34.70	11.10	−4.00	6.80
ITA	120.20	7.60	−11.50	100.50	11.90	−0.70	11.20
SPA	92.50	7.30	−3.60	46.80	14.70	−6.50	15.90
USA	176.20	4.30	−2.50	56.20	8.70	−2.70	5.50
JAP	142.00	2.20	2.90	69.80	7.40	1.90	2.10
CAN	166.70	3.30	−4.10	71.90	10.80	1.60	8.10

Table 4: Macroeconomic measurements for G7 countries

For these data the common choice for the number of groups is $I = 3$ and $J = 2$.

With algorithm 1 and automatic choice of the number of outliers we end up setting $o_1 = 1$ and $o_2 = 0$. There is one single row outlier, which is identified by the algorithm as being Italy.

The row-partition is: {GER, JAP}, {SPA}, {FRA, GBR, USA, CAN}. while the col-partition is: {GDP, DEF,DEB,TRB},{INF,INT,UNE}. This is well in agreement with Vichi (2000), with the only exception that Italy is excluded from the partitioning.

If we use the algorithm 2, the automatic choice leads to set $o_1 = 1$ and $o_2 = 1$. Italy is once again identified as being an outlier, but only for the variable DEB. For the other variables, Italy is not considered as an outlier, and the row-partition is: {GER, JAP}, {ITA, SPA}, {FRA, GBR, USA, CAN}. Interestingly enough, this is exactly the same row partition obtained in Vichi (2000). The column partition is the same as before, and once again in agreement with Vichi (2000).

We can conclude that in this data set there is some evidence of Italy being an outlier, possibly because of an exceptional high public dept (with respect to GDP); but that this outlier does not affect the classical procedure. A completely different conclusion will be drawn about the data in next section.

6.2.2 Metallic oxide analysis data

A sampling study was designed to explore the effects of process and measurement variation on properties of lots of metallic oxide. The metal content minus 80% by weight was recorded for two Types of metallic oxide raw material, in respectively 18 and 13 lots, by two randomly chosen chemists for each sample and two samples from each lot. Data reported in Table 5 come from Bennet (1954), and were analyzed with a robust mixed model approach by Fellner (1986); Zewotir and Galpin (2007).

This is a problem of *confirmatory* cluster analysis, with $I = 2$ and $J = 1$ (sample and chemist are zero-centered random effects).

The classical k-means procedure with $k = 2$ leads to a very bad solution, in which lots 6 and 7 of Type 2 belong to one group and all the other rows belong to another group. The same result is given by partitioning around medoids (PAM). If we increase the number of groups and do classical k-means or PAM with $k = 3$, there still is a badly behaved solution: one group is made of two lots (6 and 7 of Type 2), and the rand-index for the other rows is only about 0.17 in both cases.

With algorithm 1, we end up choosing $o_1 = 3$ and $o_2 = 0$. We mark lot

17 of Type 1 and lots 6 and 7 of Type 2 as outliers. The rand-index for the remaining variables is around 0.22.

Since there are at least two entirely corrupt rows, here Algorithm 2 is not appropriate. On the other hand, we can use a combination of Algorithms 1 and 2.

By combining the algorithms we finally end up marking lot 17 of Type 1 together with 6 and 7 of Type 2 as entirely outlying (once again $o_1 = 3$ and $o_2 = 0$ for Algorithm 1). On the other hand, we set $o_1 = 1$ and $o_2 = 1$ for Algorithm 2, thereby letting lot 12 for Type 2, Sample 1, Chemist 1, have an outlying measurement. The final rand-index is 0.24.

We now compare our detected outliers with the results of Fellner (1986) and Zewotir and Galpin (2007). We generally agree with the results of Fellner (1986), with the only difference that for lot 17 of Type 2 only the last four columns are marked as outlying, and that there are no outlying measurements for lot 12. This is particularly encouraging for us, since we manage achieve a similar list of outliers without using the additional information given by a-priori knowledge of the two row-groups (Type), and use of the random effects Sample and Chemist. In this situation the two row-groups are not well separated, and the outliers could be masked. Zewotir and Galpin (2007) mark entire lot 17 of Type 1, together with 6 and 7 of Type 2 as outlying They do not end up marking any column of lot 12 of Type 2, and additionally mark lots 2,3,10,11 of Type 2; together with Chemist 1, Sample 2 for lot 4 of Type 2. Here the number of identified outliers is much higher.

We shall finally note that, being based on mixture models both Fellner (1986) and Zewotir and Galpin (2007) procedures are more flexible than our approach, and can do multiple marking: for instance both methods mark Lot 6 of Type 2 as entirely outlying, and then the second measurement for Sample 2 Chemist 2 is further marked as outlying.

6.2.3 Simultaneous clustering of genes and tissues

Our last example is about analysis of DNA microarrays. Data come from Bittner *et al.* (2000), who finally analyzed the expression levels of 3613 genes on 38 tissues biopsies; 31 cutaneous melanomas and 7 controls. Samples come from male and female patients, aged 29 to 75. Bittner *et al.* (2000) discuss the segmentation of the 31 tumor samples, obtaining two clusters of 12 and 19 samples validated through multidimensional scaling and by noting different metastatic behavior between the two groups.

Rocci and Vichi (2004) perform double clustering on this data set, finally obtaining 4 row groups (one containing differentially expressed genes) and 2 groups of samples. The group of samples is slightly different than the one in Bittner *et al.* (2000), being formed of 21 and 10 samples, the same groups obtained by Goldstein *et al.* (2002) with hierarchical clustering methods.

If we allow for as few as $o_1 = 5$, $o_2 = 5$ outliers using Algorithm 2, we get back the 12-19 clustering of Bittner *et al.* (2000). The natural choice for the number of row groups in DNA microarray analysis is $I = 3$, with the aim of separating down-regulated, up-regulated and not-differentially expressed genes. Our choice is then $I = 3$, $J = 2$, $o_1 = 5$ and $o_2 = 5$ with Algorithm 2, finally getting 6 blocks (Figure 1), with centroid matrix in Table 6. As hoped, we get two groups of differentially expressed genes, one up-regulated and one down-regulated. By looking at the centroid matrix we can in fact safely conclude that a first group of 925 genes is candidate to be up-regulated, and also contributes to the column clustering; and another group of 396 is candidate for down-regulation, and also contributes to the column clustering. Further, it seems like differential expression of genes in column group 1 is more marked (higher for up-regulated, lower for down-regulated) than the differential expression for genes in column group 2, thus explaining also the sample partitioning. Interestingly enough, the remaining bulk of 2292 genes is apparently not differentially expressed and simultaneously does not contribute to the column clustering.

Our robust approach to double clustering allows then to recover the Bittner *et al.* (2000) meaningful clustering for the samples, and furthermore to simultaneously identify groups of genes that can explain the tissue partitioning. Note that the number of genes identified by our and other methods is too large for post-screening, as it often happens when applying clustering methods instead of a multiple testing approach to DNA microarray data.

7 Conclusions

We provided two robust algorithms for double clustering, and an automatic strategy for choosing the number of outliers. The algorithms are seen to have good global robustness properties, and to be useful also in identifying outliers from different perspectives. Robust double clustering seems to yield clusters that are closer to the true structure than classical methods, as was seen both in simulation and by real applications.

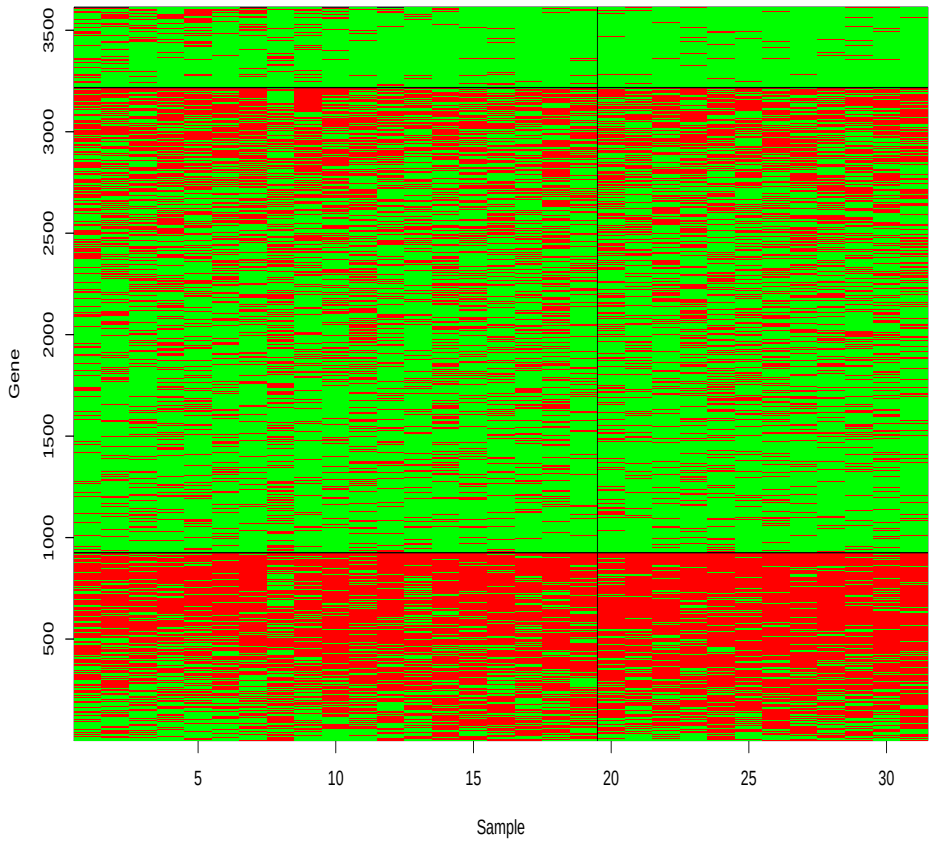

Figure 1: Clustering of Bittner *et al.* (2000) data, with $I = 3$, $J = 2$, $o_1 = o_2 = 5$, Algorithm 2.

We focused here on hard partitions, but note that generalization to fuzzy clustering or to coverings (overlapping clusters) is straightforward. In further work we will permit the location of corrupt dimensions o_2 to vary with the identified row, finally allowing for a different number and location of corrupt dimensions for each row outlier. A further robustification of the methods would be given by using a robust statistic, like the median, for estimation of the centroid matrix $\bar{x}$ (Kaufman and Rousseeuw, 1990); even if Garcia-Escudero and Gordaliza (1999) note that simply using the median instead of the mean leads to the same global robustness properties in clustering.

References

C.A. Bennet (1954). Effect of measurement error on chemical process control. *Industrial Quality Control*, **11**, 17–20.

M. Bittner, P. Meltzer, Y. Chen, Y. Jiang, E. Seftor, M. Hendrix, M. Radmacher, R. Simon, Z. Yakhini, A. Bon-Dor, N. Sampas, E. Dougherty, E. Wang, F. Maincola, C. Gooden, J. Lueders, A. Glatfelter, P. Pollock, J. Carpten, E. Gillanders, D. Leja, K. Dietrich, C. Beaudry, M. Berens, D. Alberts, and V. Sondak (2000). Molecular classification of cutaneous malignant melanoma by gene expression profiling. *Nature*, **406**, 536–540.

H.-H. Bock (1996). Probabilistic models in cluster analysis. *Computational Statistics and Data Analysis*, **23**, 5–28.

S.S. Chae, J.L. DuBien, and W.D. Warde (2006). A method of predicting the number of clusters using Rand's statistic. *Computational Statistics and Data Analysis*, **50**, 3531–3546.

S. Climer and W. Zhang (2006). Rearrangement clustering: pitfalls, remedies, and applications. *Journal of Machine Learning Research*, **7**, 919–943.

J. Cuesta-Albertos, A. Gordaliza, and C. Matràn (1997). Trimmed k-means: an attempt to robustify quantizers. *Annals of Statistics*, **25**, 553–576.

D.L. Donoho and P.J. Huber (1983). The notion of breakdown point. In: P. Bickel, K. Doksum, and J.L.Jr. Hodges, eds., *A Festschirift for Erich L. Lehmann*, 157–184. Wadsworth.

W.H. Fellner (1986). Robust estimation of variance components. *Technometrics*, **28**, 51–60.

W. Fisher (1969). *Clustering and aggregation in economics*. Johns Hopkins.

C. Fraley and A.E. Raftery (2002). Model based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, **97**, 611–631.

M.T. Gallegos and G. Ritter (2005). A robust method for cluster analysis. *Annals of Statistics*, **33**, 347–380.

L.A. Garcia-Escudero and A. Gordaliza (1999). Robustness properties of k means and trimmed k means. *Journal of the American Statistical Association*, **94**, 956–969.

D. Goldstein, D. Ghosh, and E. Conlon (2002). Statistical issues in the clustering of gene expression data. *Statistica Sinica*, **12**, 219–241.

A. Gordon (1999). *Classification*. Chapman and Hall.

F.R. Hampel (1971). A general qualitative definition of robustness. *Annals of Mathematical Statistics*, **42**, 1887–1896.

F.R. Hampel, P.J. Rousseeuw, E. Ronchetti, and W.A. Stahel (1986). *Robust Statistics: The approach based on the influence function*. Wiley.

J. Hardin and D. Rocke (2004). Outlier detection in the multiple cluster setting using the minimum covariance determinant estimator. *Computational Statistics and Data Analysis*, **44**, 625–638.

J. A. Hartigan (1978). Asymptotic distributions for clustering criteria. *The Annals of Statistics*, **6, 1**, 117–131.

J.A. Hartigan (1972). Direct clustering of a data matrix. *Journal of the American Statistical Association*, **67**, 123–129.

J.L.Jr. Hodges (1967). Efficiency in normal samples and tolerance of extreme values for some estimates of location. In: *Proc. Fifth Berkeley Symp. Math. Statist. Probab.*, vol. 1, 163–186. Univ. California Press.

P.J. Huber (1981). *Robust Statistics*. Wiley.

L. Hubert and P. Arabie (1985). Comparing partitions. *Journal of Classification*, **2**, 193–218.

L. Kaufman and P.J. Rousseeuw (1990). *Finding groups in data*. Wiley.

S.C. Madeira and A.L. Oliveira (2004). Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM transactions on computational biology and bioinformatics*, **1**, 24–45.

G.W. Milligan and M.C. Cooper (1985). An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, **50**, 159–179.

C. Reilly, C. Wang, and M. Rutherford (2005). A rapid method for the comparison of cluster analyses. *Statistica Sinica*, **15**, 19–33.

R. Rocci and M. Vichi (2004). Multimode partitioning. *In revision for: Journal of Classification*.

P.J. Rousseeuw (1984). Least median of squares regression. *Journal of the American Statistical Association*, **79**, 851–857.

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

I. Van Mechelen, H.H. Bock, and P. De Boeck (2004). Two-mode clustering methods: a structured overview. *Statistical Methods in medical research*, **13**, 363–394.

M. Vichi (2000). Double k-means clustering for simultaneous classification of objects and variables. In: S. Borra, R. Rocci, and M. Schader, eds., *Advances in classification and data analysis. Studies in classification, data analysis, and knowledge organization*, 43–52. Springer.

T. Zewotir and J.S. Galpin (2007). Residuals, leverages and outliers in mixed model. *Test*, to appear.

		Sample 1				Sample 2			
Lot	Type	Chemist 1		Chemist 2		Chemist 1		Chemist 2	
1	Type 1	4.1	4.0	4.3	4.3	4.1	4.0	4.1	4.0
2	Type 1	4.1	4.0	4.0	3.9	4.2	4.2	3.7	4.6
3	Type 1	3.5	3.5	3.4	3.6	3.4	3.3	4.0	3.5
4	Type 1	4.2	4.2	4.2	4.3	4.1	3.7	4.1	4.6
5	Type 1	3.7	3.8	3.3	3.3	3.2	3.1	3.1	3.2
6	Type 1	4.0	4.2	3.8	4.2	4.1	4.3	4.2	4.1
7	Type 1	4.0	3.8	3.8	4.0	3.6	3.8	3.9	3.8
8	Type 1	3.8	3.9	4.0	3.9	4.0	4.0	4.2	4.0
9	Type 1	4.2	4.5	4.3	4.1	3.8	3.7	3.8	3.8
10	Type 1	3.6	4.0	4.0	3.7	3.9	4.1	4.2	3.7
11	Type 1	4.6	4.6	4.0	3.4	4.4	4.5	3.9	4.1
12	Type 1	3.3	2.9	3.2	3.9	2.9	3.7	3.3	3.4
13	Type 1	4.5	4.5	4.0	4.2	3.7	4.0	4.0	3.9
14	Type 1	3.8	3.8	3.5	3.6	4.3	4.1	3.8	3.8
15	Type 1	4.2	4.1	3.8	3.8	3.8	3.8	3.9	3.9
16	Type 1	4.2	3.4	3.7	4.1	4.4	4.5	4.0	4.0
17	Type 1	3.3	3.4	3.9	4.0	2.2	2.3	2.4	2.7
18	Type 1	3.6	3.7	3.6	3.5	4.1	4.0	4.4	4.2
1	Type 2	3.4	3.4	3.6	3.5	3.7	3.5	3.1	3.4
2	Type 2	4.2	4.1	4.3	4.2	4.2	4.2	4.3	4.2
3	Type 2	3.5	3.5	4.2	4.5	3.4	3.7	3.9	4.0
4	Type 2	3.4	3.3	3.6	3.1	4.2	4.2	3.3	3.1
5	Type 2	3.2	2.8	3.1	2.7	3.0	3.0	3.2	2.7
6	Type 2	0.2	0.7	0.8	0.7	0.3	0.4	0.2	−1.0
7	Type 2	0.9	0.6	0.3	0.6	1.0	1.1	0.7	1.0
8	Type 2	3.3	3.5	3.5	3.4	3.9	3.7	3.7	3.7
9	Type 2	2.9	2.6	2.8	2.9	3.1	3.1	2.9	2.7
10	Type 2	3.8	3.8	3.9	3.8	3.4	3.6	4.0	3.8
11	Type 2	3.8	3.4	3.6	3.8	3.8	3.6	3.9	4.0
12	Type 2	3.2	2.5	3.0	3.5	4.3	3.7	3.8	3.8
13	Type 2	3.4	3.4	3.3	3.3	3.5	3.5	3.2	3.3

Table 5: Metallic Oxide Data

	Column Group 1 (19)	Column Group 2 (12)
Row Group 1 (925)	0.76	0.50
Row Group 2 (2292)	−0.08	−0.04
Row Group 3 (396)	−1.28	−0.92

Table 6: Estimated centroid for Bittner *et al.* (2000) data, with $I = 3$, $J = 2$, $o_1 = o_2 = 5$, Algorithm 2; with cardinalities in parentheses.