ARTICLE IN PRESS

Computational Statistics and Data Analysis ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

journal homepage: www.elsevier.com/locate/csda

RelaxMCD: Smooth optimisation for the Minimum Covariance Determinant estimator

M. Schyns [a,*], G. Haesbroeck [b], F. Critchley [c]

[a] University of Liège, HEC-Management School, QuantOM, Bd. du Rectorat 7 (B31), 4000 Liège, Belgium

[b] University of Liège, Institute of Mathematics, Grande traverse, 4000 Liège, Belgium

[c] The Open University, Department of Mathematics and Statistics, Milton Keynes, United Kingdom

ARTICLE INFO

Article history:
Received 23 July 2008
Received in revised form 4 November 2009
Accepted 4 November 2009
Available online xxxx

ABSTRACT

The Minimum Covariance Determinant (MCD) estimator is a highly robust procedure for estimating the centre and shape of a high dimensional data set. It consists of determining a subsample of h points out of n which minimises the generalised variance. By definition, the computation of this estimator gives rise to a combinatorial optimisation problem, for which several approximate algorithms have been developed. Some of these approximations are quite powerful, but they do not take advantage of any smoothness in the objective function. Recently, in a general framework, an approach transforming any discrete and high dimensional combinatorial problem of this type into a continuous and low-dimensional one has been developed and a general algorithm to solve the transformed problem has been designed. The idea is to build on that general algorithm in order to take into account particular features of the MCD methodology. More specifically, two main goals are considered: (a) adaptation of the algorithm to the specific MCD target function and (b) comparison of this 'tuned' algorithm with the usual competitors for computing MCD. The adaptation focuses on the design of 'clever' starting points in order to systematically investigate the search domain. Accordingly, a new and surprisingly efficient procedure based on a suitably equivariant modification of the well-known k-means algorithm is constructed. The adapted algorithm, called RelaxMCD, is then compared by means of simulations with FASTMCD and the Feasible Subset Algorithm, both benchmark algorithms for computing MCD. As a by-product, it is shown that RelaxMCD is a general technique encompassing the two others, yielding insight into their overall good performance.

1. Introduction

The definition of the Minimum Covariance Determinant (MCD) estimator introduced by Rousseeuw (1985) goes as follows. In a sample of n data points, one has to select the subsample of size $h \approx \frac{n}{2}$ minimising the generalised variance (i.e. the determinant of the covariance matrix based on these points). The location and scatter MCD estimates are then given by the mean and covariance matrix of the optimal subsample. The MCD estimator is quite attractive: its definition is simple and intuitively appealing, while it has good theoretical properties (see Butler et al. (1993)). However, its computation is hard since the corresponding optimisation problem is combinatorial.

The computational complexity of the MCD estimator has given rise to an active research area (Hawkins, 1994; Rousseeuw and Van Driessen, 1999; Hawkins and Olive, 1999, 2002; Bernholt and Fisher, 2004). Indeed, an exact computation naively

* Corresponding author. Tel.: +32 4 366 31 91; fax: +32 4 366 2767.
E-mail addresses: M.Schyns@ulg.ac.be (M. Schyns), G.Haesbroeck@ulg.ac.be (G. Haesbroeck), F.Critchley@open.ac.uk (F. Critchley).

0167-9473/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.csda.2009.11.005

requiring the consideration of all the $\binom{n}{h}$ subsamples in order to select the best one would be infeasible even for relatively small data sets. Agulló (1998) greatly improved this naive exact algorithm by incorporating the branch and bound technique, but the computation time remains prohibitive for sample sizes exceeding 100 and dimensions greater than 10. Therefore, one has to resort to approximate algorithms to compute the MCD estimators. The first proposal was a basic resampling scheme where a given number of random starting subsamples of h points provided trial estimates among which the one minimising the objective function was selected. Often, each random subset is improved by additional iterative steps, called 'concentration steps' in the FASTMCD algorithm of Rousseeuw and Van Driessen (1999), or 'swaps' in the Feasible Solution Algorithm of Hawkins (1994) and Hawkins and Olive (1999). The concentration steps consist of selecting the h points with smallest Mahalanobis distances with respect to the current trial estimates and defining a new subset with these points, while Hawkins' swaps interchange the selected and unselected points achieving the biggest decrease in the objective function. Both ideas rely on a necessary condition of the MCD solution. In the first case, the determinant of the more concentrated points will be smaller than or equal to the determinant of the initial subsample. In the second case, at the optimum, the MCD objective function should not decrease when performing any pairwise swap. Therefore, as soon as no further swaps can decrease the objective value, the obtained subset can be considered as a feasible solution for the MCD problem. The drawback of FSA with respect to FASTMCD is that its computation time is much bigger, even if an improved FSA algorithm prevents the algorithm from spending too much time on performing swaps. Note that these two algorithms will be considered in the following as benchmarks for the computation of MCD. Other heuristic procedures like simulated annealing or tabou search (Todorov, 1992; Woodruff and Rocke, 1994) have also been tried for the computation of some robust estimators but they do not seem as efficient, at least for the MCD, or do not always satisfy some desirable statistical properties.

The combinatorial definition of the MCD estimator seems to prevent the design of algorithms taking advantage of smoothness properties. However, the relaxation strategy proposed by Critchley et al. (forthcoming) allows us to transform this discrete and high dimensional optimisation problem (corresponding to a search space containing $\binom{n}{h}$ potential candidates) into a continuous and low dimensional one, opening the door to the use of classical descent along gradients and other concavity properties. Section 2 briefly summarises this relaxation procedure applied to the MCD objective function, while Section 3 focuses on the construction of the starting points which can be geometrically designed instead of chosen at random as advocated by Critchley et al. (forthcoming). This refined version of the basic algorithm will be referred to as the *RelaxMCD* algorithm. The performance of RelaxMCD with respect to the above mentioned competitors for computing the MCD estimators is measured by means of simulations in Section 4, while Section 5 gives our concluding remarks.

2. Smooth MCD

Let n denote the sample size, k the dimension and consider the data set $X := (x_i^T)$. The MCD estimators correspond to the empirical mean and covariance matrix computed on a subsample of h points of X. Let $0 < m < \frac{n}{2}$ represent the number of points not determining the MCD estimator, i.e. $h = n - m$. The number h is related both to the breakdown point of the MCD procedure (which is approximately $\frac{n-h}{n}$) and to its efficiency. From now on, the value yielding the maximum breakdown point will be used as default, i.e. $h = [\frac{n+k+1}{2}]$ where $[z]$ denotes the largest integer smaller than or equal to z.

At the optimal subset, the MCD estimates can be seen as weighted mean and covariance matrix for which m of the n weights are equal to 0 while the remaining h are given by $1/h$. More generally, following the notation of Critchley et al. (forthcoming), the MCD optimisation problem may be defined as follows.

Denoting by $\mathbb{P}^n$ the set of all probability n-vectors, let P represent $\mathrm{diag}(p)$ for any $p \in \mathbb{P}^n$. The weighted mean and covariance matrix characterised by the weight vector p can be written as

$$\bar{x}(p) = X^T p \quad \text{and} \quad \hat{\Sigma}(p) = X^T(P - pp^T)X = M(p) - \bar{x}(p)\bar{x}(p)^T,$$

where $M(p) = X^T P X$. The set over which $\hat{\Sigma}^{-1}(p)$ is properly defined will be denoted by $\mathbb{P}(\hat{\Sigma}^{-1})$. The MCD objective function is then

$$t(p) = \log\det(\hat{\Sigma}(p)), \tag{1}$$

where the logarithm is taken in order to achieve concavity as shown in Proposition 1 (the proofs are kept for the Appendix).

Proposition 1. *The function $t(.)$ given in (1) is concave on $\mathbb{P}(\hat{\Sigma}^{-1})$. More precisely, $t(.)$ is either constant or strictly concave on all line segments $[p, p^*]$ in $\mathbb{P}(\hat{\Sigma}^{-1})$. Moreover, constancy arises iff $\bar{x}(p) = \bar{x}(p^*)$ and $\hat{\Sigma}(p) = \hat{\Sigma}(p^*)$.*

The MCD estimates are given by $(\bar{x}(\hat{p}), \hat{\Sigma}(\hat{p}))$, where $\hat{p}$ solves the minimisation problem under constraints:

$$\hat{p} = \operatorname*{argmin}_{p \in \mathbb{P}^n_{-m}} t(p),$$

where $\mathbb{P}^n_{-m}$ comprises all $p \in \mathbb{R}^n$ satisfying

$$0 \leqslant p_i \leqslant \frac{1}{n-m}, \quad \text{(i.e. bound constraints)}, \tag{2}$$

$$p_1 + \cdots + p_n = 1, \quad \text{(i.e. a linear constraint)}, \tag{3}$$

Please cite this article in press as: Schyns, M., et al., RelaxMCD: Smooth optimisation for the Minimum Covariance Determinant estimator. Computational Statistics and Data Analysis (2009), doi:10.1016/j.csda.2009.11.005

with $t(p)$ given by (1). The concavity of $t(p)$ ensures that the optimisation process ends up at a vertex (as required in the MCD problem) and not at an interior point. An algorithm constructed to solve this problem when $t(p)$ is relatively smooth and concave is outlined in Critchley et al. (forthcoming). Basically, starting at an initial point p_0, the algorithm follows the opposite direction of the centred gradient (in order to satisfy (3)) until reaching a boundary (since, due to the concavity, getting as far away as possible from the current position maximally decreases the function), where the value of at least one coordinate of the probability vector is fixed (according to (2)). It continues like this until reaching a vertex. In the following, such a vertex will often be referred to as an h-subset (this subset containing the observations corresponding to a positive weight in the vertex).

Proposition 2 derives the centred gradient corresponding to the MCD target function. Since one has to stay in $\mathbb{P}^n$ at each iteration of the descent, gradients need to be projected, as Proposition 2 further details.

Proposition 2. *For the MCD objective function defined in (1), one gets* $\forall\, p \in \mathbb{P}(\widehat{\Sigma}^{-1})$*:*

$$t^c(p) = (I_n - J_n)(D(p)), \tag{4}$$

where $J_n := n^{-1}1_n 1_n^t$ *and* $D(p)^t = (D_{11}(p) \ldots D_{kk}(p))$ *with*

$$D_{ii}(p) = (x_i - \bar{x}(p))^t\, \hat{\Sigma}^{-1}(p)(x_i - \bar{x}(p)). \tag{5}$$

Critchley et al. (forthcoming) note that a terminal vertex p^* is not always a candidate local minimum of the target function, in the sense that every feasible direction from it is uphill. They provide the following necessary and sufficient condition for such a vertex to have this property:

$$p^* \text{ is a candidate local minimum of } t(p) \text{ iff } \min_{i:p_i^*=0} t_i^c(p^*) \geq \max_{i:p_i^*=1/h} t_i^c(p^*), \tag{6}$$

i.e., all 'excluded' observations ($p_i^* = 0$) have bigger centred gradient coordinates than 'included' observations ($p_i^* = 1/h$).

If (6) does not hold, swaps are applied in order to get to a candidate local minimum. Different strategies for these local improvements are enumerated in Critchley et al. (forthcoming). The simplest strategy, called 1-swaps, is to select and swap the two observations that correspond to respectively the minimal and maximal values of the centred gradient in expression (6). This would lead to the largest local decrease (in a subspace of dimension 2) of the objective function. This scheme can be generalised by swapping, say, pairs of observations if that still leads to a local decrease of the target function. One could think of swapping the largest possible number of observations, yielding so-called l_{max}-swaps, or one could choose the dimension (i.e. the number of observations to swap) in order to get the biggest decrease of the objective function, yielding so-called $l_{deepest}$-swaps. Critchley et al. (forthcoming) do not thoroughly compare these different swapping strategies. Some numerical comparisons will, then, be reported in Section 4 but it is already worth noting that Proposition 2 combined with the general definition of the l_{max}-swap strategy gives a first insight into the FASTMCD methodology. Indeed, as already explained in the Introduction, each C-step keeps the most concentrated observations, when concentration is measured by means of robust distances based on the current mean and scatter matrix. Since the centred gradient of the MCD target function is defined in terms of these robust distances too, the ordering of these being preserved by the applied projection, the l_{max}-swap technique is doing the same job as C-steps. Moreover, Proposition 2 ensures that the C-steps of FASTMCD stop if and only if the algorithm has reached a candidate local minimum. One could say that the MCD algorithm of Critchley et al. (forthcoming) encompasses FASTMCD.

3. Selection of initial points

3.1. Speed, robustness and depth

The performance of any algorithm for computing the MCD can be measured on three criteria: robustness, low value and speed. The MCD estimator being a highly robust procedure, the primary focus in the design of algorithms for computing it should be to achieve robustness. Then, a good compromise between good minimisation and speed should be aimed at. Conditioning on robustness, the compromise between the two other criteria can be represented by any point in Fig. 1 where the horizontal axis describes speed and the vertical one the performance of the minimisation. Any researcher working on the computation of the MCD estimators wishes his algorithm to lay on the corner *Perfection* where an exact answer is obtained instantly. However, even if exact methods are available in the literature to find the global optimum of nonlinear problems under constraints (see e.g. Horst and Tuy (1995) or Pardalos and Rosen (1987)), applying these would require too much computation time. Any compromise, as heuristics like FASTMCD, FSA or RelaxMCD suggest, drives the algorithm somewhere inside the square (see for example, the positions of FASTMCD and FSA according to general belief).

It is important to note here that, even if reaching the global optimum instead of a local one is preferable, it is *not* required for many purposes, a 'good enough' solution being good enough to fully achieve statistical objectives. Now, if the minimisation performance of an algorithm may be improved while keeping a competitive computation time, this might be worth it. In this Section, it is shown that such an improvement may be obtained when the starting points are carefully selected.

Please cite this article in press as: Schyns, M., et al., RelaxMCD: Smooth optimisation for the Minimum Covariance Determinant estimator. Computational Statistics and Data Analysis (2009), doi:10.1016/j.csda.2009.11.005

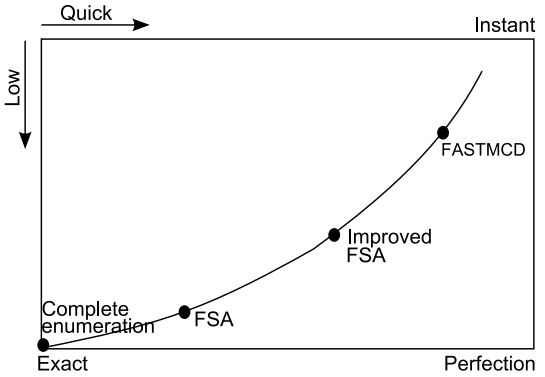

Fig. 1. Compromise between the criteria low and fast.

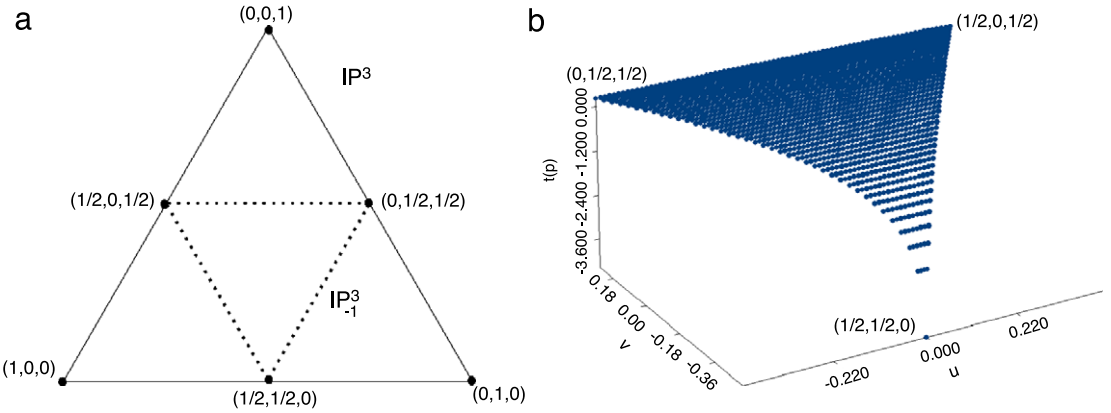

Fig. 2. (a) Two-dimensional representation of the sets $\mathbb{P}^3$ and $\mathbb{P}^3_{-1}$; (b) representation of $t(p)$ on $\mathbb{P}^3_{-1}$.

3.2. Random starting points

The three heuristics of interest start from initial points and go down, from there, in a deterministic way to a candidate local minimum. As commonly acknowledged, the quality of the attained solution depends on the corresponding initial point. Such a statement can be nicely illustrated, at least for the smooth algorithm, on panel (b) of Fig. 2 which represents the target function (1) on the set $\mathbb{P}^3_{-1}$. The horizontal plane represents the values of the probability vector p (the three dimensions may be reduced to two by taking into account the linear constraint (3), as panel (a) illustrates) while the vertical axis gives the corresponding value for the target function. This plot is based on the data set $\{-1, -0.8, 1\}$, and the MCD is computed using the best 2-subset. These 2-subsets are associated to the vertices of the triangle $\mathbb{P}^3_{-1}$ and the corresponding probability vectors are indicated on the plot of $t(p)$. The global minimum of $t(p)$ is achieved when the probability mass is evenly distributed on the two closest observations, as expected. Now, one sees that the MCD objective function is nicely shaped but can have several local minima located in different 'valleys'. One can tell that, as soon as one starts descending the valley, there is no turning back. The vertex at which the iterative procedure ends will be completely determined by where the descent started.

Therefore, in order to ensure that the algorithms will have the opportunity to fall down into good valleys, many random starting points are typically selected. As one increases the number of random starts, the probability to reach the global optimum (or at least a good local one) increases too. Unfortunately, so does the total computation time. The idea here is to replace some of these random starts by strategically selected points. Hopefully, a small number of these 'not-random' starting points could perform as well as (or even better than) a much larger number of random starts as far as the minimisation process is concerned. The optimal strategy would then be to combine these designed points together with some random starts depending on the available computation time.

Some discussion concerning the construction of the initial points is already given in Rousseeuw and Van Driessen (1999), with emphasis on the robustness criteria of the algorithm. Claiming that the priority is to get clean solutions, these authors argue that 'clean' starts should be privileged. In order to increase the chance of getting such a clean start, FASTMCD works initially with random elemental subsets (consisting of $k+1$ observations). A proper initial point is then obtained by selecting the h observations most concentrated with respect to the mean and covariance matrix based on the $(k+1)$-subset. In FSA, subsets of size h are chosen randomly, resulting in a lower overall robustness of the procedure since it is easy to show that the probability to get clean subsets is much bigger when these subsets contain $k+1$ observations rather than h.

Please cite this article in press as: Schyns, M., et al., RelaxMCD: Smooth optimisation for the Minimum Covariance Determinant estimator. Computational Statistics and Data Analysis (2009), doi:10.1016/j.csda.2009.11.005

Following Rousseeuw and Van Driessen's approach, the first strategy implemented in RelaxMCD is to start the descent from N vertices for which $k+1$ coordinates chosen at random are set at $\frac{1}{k+1}$ while the others are equal to 0 (each such vertex corresponding to an elemental subset). The same inflation step as in FASTMCD is then applied to obtain a vertex lying in $\mathbb{P}^n_{-m}$. This random drawing can be slightly modified in order to ensure that each observation is included in at least one elemental subset. For that, the set of indices $I = \{1, \ldots, n\}$ is randomly reordered, yielding the set $\tilde{I}$. A first starting vertex is built by selecting the observations corresponding to the first $k+1$ indices of $\tilde{I}$, the second vertex to the next $(k+1)$'s, …and so on. If needed, the last elements of $\tilde{I}$ are combined with the first elements to build a last $(k+1)$-vertex. This simple scheme allows one to construct $N = [(n+k)/(k+1)]$ starting points where each observation is represented at least once. When more starting points are required, the same procedure can be applied to another random ordering of the index set I.

This elemental subset strategy is motivated by the need to construct a robust algorithm for computing a robust estimator. However, it does not rely on the specific characteristics of the MCD target function as other strategies developed in the next section do.

3.3. Starting points chosen by design

More advanced schemes may be considered in order to construct starting points.

First, one can depart from a somewhat neutral position which does not favour any of the cases: $p_\circ = \frac{1}{n}1_n$. It is an internal point in $\mathbb{P}^n_{-m}$ but, contrary to FASTMCD and FSA, starting points are not required to be vertices in RelaxMCD. The main drawback of this neutral choice is that it gives the same weight to good as well as to contaminated observations while, as mentioned in Rousseeuw and Van Driessen (1999), it is more difficult to end up clean when the descent starts from a contaminated point.

A second strategy is based on the intuitive idea that if one could get to the top of a (concave) mountain, one would be in a nice position to get an overall sight of the search space and to find the best descending path to the minimum. Now, finding the maximum of a concave function is a much simpler problem (even under constraints) than finding the minimum. Moreover, for the MCD target function, one can simply iterate swaps from a random vertex up to the maximum since it is very unlikely that the maximum occurs at an interior point, as Proposition 3 formally shows.

Proposition 3. *Let $t(.)$ be the objective function of the MCD estimator. The maximum of $t(p)$ over $\mathbb{P}^n_{-m}$ will be at a relative interior point with probability* 0.

Let v_{max} denote the maximum vertex. It is worth mentioning that while the true minimum of the MCD target function should be achieved at a clean h-subset, the converse probably holds for the maximum. As soon as some observations are contaminated, v_{max} is likely to attribute a positive weight to at least one of them. Also, even if one has a nice global view from the maximum, only local gradient information is used in order to initiate the descent from the top. Nevertheless, v_{max} can be selected as a starting point in its own right. However, it seemed that this maximal vertex could also be used as a kind of mirror to construct other, and hopefully promising, starting vertices, as *opposite* to the maximum as possible. Indeed, one may expect that the vertex leading to the minimum will be quite different from v_{max}. Therefore, knowing v_{max}, a promising vertex could be constructed by interchanging the role of each observation: if one observation is included in the maximal h-subset (i.e its corresponding coordinate in v_{max} is $1/h$), exclude it and vice versa. Of course, proceeding like that makes one jump from the space $\mathbb{P}^n_{-m}$ to the space $\mathbb{P}^n_{-h}$, which is not convenient for starting the descent. A way out of this problem is to work the other way round: apply this transformation to the maximum of $t(p)$ over $\mathbb{P}^n_{-h}$ in order to end up with a vertex of $\mathbb{P}^n_{-m}$. Such a transformed vertex will be denoted as v_{opp} to emphasise the fact that it is opposed to the maximum.

The opposition with respect to v_{max} could be less extreme than what is done in the construction of v_{opp}. For example, one can simply try to get away from the maximum by only a few steps and construct equally spaced starting points on the resulting subspace in order to fully investigate it. Schematically, this corresponds to distributing uniformly starting points on a crown placed on a head (i.e. the concave mountain).

Different strategies have been considered to precisely construct this crown of initial vertices but here is the one that has been implemented in RelaxMCD. First, one chooses m as a multiple of $k+1$ (keeping $m \leq n$) and the vertex v'_{opp} opposite to the maximum computed on $\mathbb{P}^n_{-h}$ is derived. By design, v'_{opp} gives an equal weight of $1/m$ to m observations. These observations are then split at random in groups of $k+1$ observations, each of these subgroups leading to a vertex of $\mathbb{P}^n_{-n+k+1}$. These vertices should be quite clean (since they are opposed to the maximum); they are all located at the same distance equal to $(h+k+1)$ (where distance between two vertices is defined by the number of swaps one has to perform to jump from one vertex to the other) from the maximum vertex and at exactly the same distance, equal to $(k+1)$, from each other. Moreover, they are orthogonal to each other.

3.4. Equivariant k-means

As already stressed, starting from contaminated vertices increases the risk to end up at a contaminated candidate local minimum. On the other hand, if one could start the descent from an initial vertex lying close to the global optimum, there would be more chance to reach this optimum. Moreover, Butler et al. (1993) proved that the optimum subset of the MCD optimisation problem is separated from the other observations by an ellipsoid, i.e. the h selected observations form a kind

Please cite this article in press as: Schyns, M., et al., RelaxMCD: Smooth optimisation for the Minimum Covariance Determinant estimator. Computational Statistics and Data Analysis (2009), doi:10.1016/j.csda.2009.11.005

of compact group. It could then be interesting to search for clusters of observations in order to start descending from the corresponding vertices (for which the probability mass is equally distributed on the selected coordinates) hoping that at least one of these is close to the optimal vertex.

For finding these promising starting clusters, the well known k-means algorithm may be used. In order to get c clusters, a basic version of this algorithm (see Johnson and Wichern (1992)) proceeds as follows: select c observations at random to be the seeds of the c clusters. Initially, each cluster contains exactly one observation, its seed. Then, for each observation, compute its distance with respect to the centre of each cluster and associate it to the closest one. Recompute the centre of each modified cluster and repeat the previous step until each observation remains in the same cluster.

There are three problems with this basic approach, i.e.

- the final number of observations in each cluster is not known in advance and cannot be expected to be exactly equal to h, as would be a priori natural for a starting vertex;
- the number of clusters needs to be specified beforehand, while the 'optimal' number surely depends on the data;
- the affine equivariance property satisfied by the whole optimisation process up to now does not hold if one uses, as is traditionally the case in the k-means algorithm, Euclidean distances.

Adjustments need to be found to overcome each of these three drawbacks. The first two are relatively easily dealt with, while the latter is crucial in this MCD context. Accordingly, we refer to our adjusted procedure as (affine) *equivariant* k-means.

For the first, the adjustment is easy since, as already discussed in the previous section, one may inflate or deflate the clusters according to Mahalanobis distances, leaving out clusters containing less than $k + 1$ observations since distances cannot be properly defined in that case. Note also that, when the number of observations in a given cluster, n_c say, is bigger than h, the probability vector with a mass of $1/n_c$ on the corresponding observations and 0 to the others is a valid starting point of $\mathbb{P}^n_{-m}$ from which the descent can be performed.

For the choice of the number of clusters, there is no magic number which would work for all kinds of data configuration. Basically, one hopes to distinguish between clean and contaminated data. Two clusters could be enough when the contamination is concentrated in one area. If the contamination is spread out, one may expect that a cluster would be needed for every cloud of contaminated points. However, the number and location of outliers are usually unknown before the computation of the robust estimator. Moreover, the equivariant k-means algorithm is not designed to find the optimal solution of the MCD problem but is simply used as a heuristic to try to provide at least one good starting point. Increasing the number of such starting points looks like a good strategy. Therefore, the equivariant k-means algorithm is applied for different values of c starting with $c = k$ (believing that the dimension could be interpreted as a kind of complexity measure) and considering all integer values from k down to 2. By default, each cluster computed for each c is used as a starting point. However, when k is big ($k > 15$ say), the number of starts based on the equivariant k-means algorithm is truncated to save computation time.

The last adjustment consists of ensuring affine equivariance throughout. For that, the Euclidean distances are replaced by appropriate Mahalanobis distances. By 'appropriate', we mean that a reliable (i.e. uncontaminated) covariance matrix should be used as metric (not depending on the clusters). The 'best' covariance matrix being the goal of the main optimisation process, one can only rely on approximations. Again, several alternatives are possible but, at the end of the day, the covariance matrix based on the best solution obtained so far in RelaxMCD (i.e. starting from other starting points) was selected. From one application of the equivariant k-means algorithm to the next, the metric can be updated if a better one is available. Note that it is also possible to update the covariance matrix and apply again the equivariant k-means algorithm without modifying the number of clusters. It is interesting to note that, while the k-means clustering idea is used here to derive starting points for the computation of the MCD estimator, García-Escudero and Gordaliza (2007) proceeded the other way round by using the MCD estimator for constructing heterogeneous robust clustering.

3.5. Further adaptations of the general methodology

All starting points presented so far were designed without *a priori* information, but one can also compute some *a posteriori* based on results acquired so far. Two other types of starting point are detailed below, but were not incorporated in RelaxMCD for the simulation study. In the search of a good compromise between depth, robustness and speed, including them did not seem worthwhile due to the good overall performance of the algorithm as it stands and due to the increased computation time they would require. Nevertheless, for extremely complex problems, or for applications for which depth is considered more important than speed, they could be an option.

Free swaps: At each step of Hawkins' FSA, all possible swaps of two observations (one included in the current h-subset and one excluded from it) are considered and the best one is kept. Trying all possibilities gives the guarantee that the optimal 1-swap is performed. Moreover, unlike the local 1-swaps proposed in this paper which need to start downhill and stay in the same valley, Hawkins' swaps can jump over neighbouring hills. In other words, when RelaxMCD is blocked at a candidate local minimum down a valley, FSA can still go out of the valley to pursue its descent. Of course, the weakness of Hawkins' steps is the time required to consider all possible swaps before being able to select the best one. Simulations will show that the local 1-swaps of RelaxMCD are already quite time consuming, even though the order in which they have to be performed

Please cite this article in press as: Schyns, M., et al., RelaxMCD: Smooth optimisation for the Minimum Covariance Determinant estimator. Computational Statistics and Data Analysis (2009), doi:10.1016/j.csda.2009.11.005

is determined without trials. In order to integrate the 'jumping' idea of Hawkins' swaps in RelaxMCD without increasing tremendously its computation time, one could think of adding one Hawkins' swap (called *free swap* here to emphasise the fact that they do not need to start in a downhill direction) after having reached by means of local swaps (either l_{max} or 1-swaps) a candidate local minimum. One then can restart the process from this interesting starting point.

Hard core: Leaving the random starts aside, the algorithm starts from a given number, N_d say, of designed points. Therefore, it reaches N_d solutions, among which only the best one is kept. Up to now, the other solutions are discarded even if they may contain some valuable information. Indeed, if one observation is included in all the optimal subsamples obtained, then we can be quite confident that it lies in the centre of the data set. It is then sensible to construct a hard core comprising all the observations lying in all the optimal solutions; i.e. the intersection of the optimal h-subsets. This hard core contains strictly less than h observations (except if all the solutions are identical in which case applying the core method will not bring any new result). It cannot as such correspond to a new optimal vertex. When it contains at least $k+1$ points, a new h-subset can however be easily constructed by keeping the h observations with smallest robust distances with respect to the observations in the hard core. This inflation step can either be performed in one go or recursively by adding only one observation at a time.

It is hoped that the hard core set will be big since this means that many solutions are close one to the others and hopefully close to the centre of the data cloud. However, when the number of designed initial points is large, it may happen that the intersection does not cover the minimal number of $k+1$ points, making the hard core useless.

This hard core strategy focuses on the most concentrated points in the data set, as intuitively should be the target of the MCD problem. It is efficient when the algorithm gets trapped in local minima whose optimal subsamples are quite close, but slightly different, from the overall best subset. Computing the hard core and then inflating it with the most concentrated points can shift the solution to the global one.

These two additional steps will not be used in the simulation study of the next section since the purpose there will be to focus on the performance of the different types of descent and of the designed starting points, without taking into account 'external' improvements due to additional adjustments. In practice though, we would recommend to apply these adjustments by default since they may lead to a different and better solution while not increasing significantly the computation time (at least if only one final free swap is applied).

4. Simulations

At this point of the description of RelaxMCD, many questions arise. First, is the relaxation idea of Critchley et al. (forthcoming) of interest for the MCD problem? At least two efficient algorithms for computing the MCD estimators are already widely available: FASTMCD developed by Rousseeuw and Van Driessen (1999) and implemented by default in most statistical packages (SAS, R, Matlab, …) and FSA proposed by Hawkins (1994) and improved by Hawkins and Olive (1999). There is clearly the need to compare the performance of the newly introduced methodology with these two benchmark algorithms. Then, a whole section of this paper is devoted to the construction of non-random starting points. Does that improve the overall performance of the relaxation technique or would it be as efficient to rely only on random drawings? Finally, one-swaps, l_{max}-swaps, $l_{deepest}$-swaps (and one could add many other types of swaps) can be applied. Which is the best and/or the quickest way of descending?

These questions will be addressed by means of simulations. After detailing some data configurations, the relevance of the non-random starting points will be measured and an optimal strategy for the selection of these outlined. Then, our focus will be on the comparison of the performance of RelaxMCD with respect to the above mentioned competitors. Different criteria of performance will be considered according to speed, robustness and depth.

4.1. Simulation set-up

Artificial data sets were simulated to try to get answers to the questions outlined above. Mixtures of normal distributions were especially investigated, as is traditionally the case when evaluating the performance of a robust procedure. More specifically, data were generated according to the distribution

$$(1-\delta)N(0,I)+\delta N(\mu,\sigma^2 I_k), \tag{7}$$

where δ is the percentage of contamination, μ is the shift parameter (arbitrarily put on the first axis) and σ is the dispersion of the contaminating distribution. Non-spherical mixture configurations are also addressed (see Section 4.3).

Rousseeuw and Van Driessen (1999) compared the performance of FASTMCD and FSA under model (7) with parameter values μ and σ selected such that the cloud of contaminated data lies quite far away from the main bulk of data. A more challenging, and probably more realistic, configuration would allow the contamination to be closer to the good data points, without overlapping with it. To get such a set-up, the dispersion parameter σ was set to 1 and μ to $(1+\sigma)\sqrt{\chi^2_{0.95,k}}\mathbf{e}_1$ where $\mathbf{e}_1$ is the first unit vector in $\mathbb{R}^k$. In that case, the 95% confidence ellipsoids constructed for each of the two normal distributions of model (7) share a tangent hyperplane as illustrated, in the bivariate case, in Fig. 3.

Please cite this article in press as: Schyns, M., et al., RelaxMCD: Smooth optimisation for the Minimum Covariance Determinant estimator. Computational Statistics and Data Analysis (2009), doi:10.1016/j.csda.2009.11.005

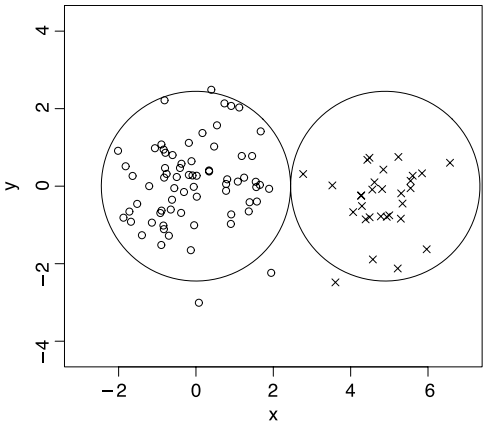

Fig. 3. Simulation set-up (7) illustrated for $k = 2$, $\delta = 0.3$ and a number of observations equal to 100. The empty circles correspond to clean data while the crosses indicate outliers.

Several sample sizes, dimensions and percentages of contamination were considered: n was taken equal to 100, 200, 500, and 1000 with dimensions k equal to 5, 10, 15, and 20 and the percentage of contamination was fixed at 10%, 20% and 30%. For each data configuration, one thousand data sets were generated. Note that, whatever the algorithm, the time required to compute the MCD estimator increases with the sample size and the dimension. When the sample contains more than 500 observations, Rousseeuw and Van Driessen (1999) apply some data partitioning to speed up the process. Such an idea could have been integrated in RelaxMCD (and FSA), but this has not been pursued here. When 1000 replications of RelaxMCD had to be done on big data sets, a compromise between speed and depth had to be settled, as will be further explained later.

4.2. Strategies for starting points

The different proposals made in Section 3 for the construction of starting points will be compared in this section. More precisely, the following starting points will be considered: the maximum vertex v_{max}, the neutral point $p_\circ = \frac{1}{n}1_n$, the vertex v_{opp} opposite to the maximum achieved on $\mathbb{P}^n_{-h}$, the opposite vertices v_{ring_j} constructed orthogonally to each others and at a given distance from the maximum vertex, the equivariant k-means vertices v_{km_j} as well as random vertices (either totally random v_{rdn_j} or orthogonally built v_{rndu_j}). One sees that some strategies lead to a single starting point, others to a larger number. The term 'family of starting points' will be used to distinguish the sets of starting points derived by means of the different strategies. Let A and B denote two such families.

Since the goal of the optimisation process is to minimise the value of the objective function, the criterion *depth* will be the one of interest to test the quality of the starting points. When comparing two families of starting points, the focus will be on the best of their elements, i.e. the one which leads RelaxMCD to the lowest value. For each simulation, the performance measure of family A with respect to family B is therefore defined by

$$\mathrm{obj}_i(A|B) = \sqrt[k]{\frac{\det \hat{\Sigma}(p_i^A)}{\det \hat{\Sigma}(p_i^B)}}, \quad 1 \le i \le \mathrm{nsim}, \tag{8}$$

where, for simulation i, p_i^A is the optimal vertex obtained when starting from a point of the set A and p_i^B the best vertex obtained when starting from an element of the set B. Note that one could express the performance measures (8) in terms of the objective function $t(p)$, but we found it easier to interpret results without the log. When the median of the measures $\mathrm{obj}_i(A|B)$ is approximately equal to 1, this means that the two sets of starting points performed similarly. When it is larger than 1, the best starting points of set A do not reach, overall, vertices as low as those reached from the best starting vertices of set B. In the sequel, boxplots based on 1000 values of (8) are represented in order to visualise the comparison between the two families.

More specifically, in order to get an overall measure of performance, the set B was taken as the union of all the sets. Therefore, for each simulation, the set B contains, by definition, the best starting point and the measure (8) is always greater than or equal to 1. Moreover, to be sure to focus only on the performance of the starting points, only one type of local improvement will be considered: l_{max}-swaps.

Fig. 4 shows the results when $n = 200$, $k = 10$ and $\delta = 30\%$ while Fig. 5 corresponds to $n = 200$, $k = 20$ and $\delta = 30\%$. Similar results were observed for all the other data configurations.

Before interpreting the results, it is important to note again that the number of elements of the set A depends on the selected strategy. The set A may be a singleton (for example, $A = \{v_{max}\}$ or $A = \{p_\circ\}$) or a much bigger set (for example, A may consist of many random starts, or of many equivariant k-means vertices). On the other hand, it is clear that the chance to get deep increases when the number of descents increases also. For fair comparisons, Figs. 4 and 5 should be divided

Please cite this article in press as: Schyns, M., et al., RelaxMCD: Smooth optimisation for the Minimum Covariance Determinant estimator. Computational Statistics and Data Analysis (2009), doi:10.1016/j.csda.2009.11.005

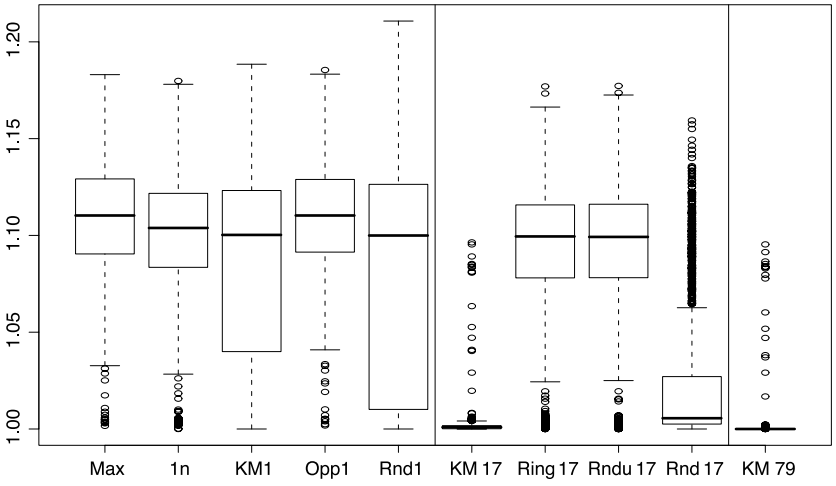

Fig. 4. Boxplots of the performance measure (8) for $n = 200$, $k = 10$ and $\delta = 30\%$; vertical lines separate groups of comparable starting strategies.

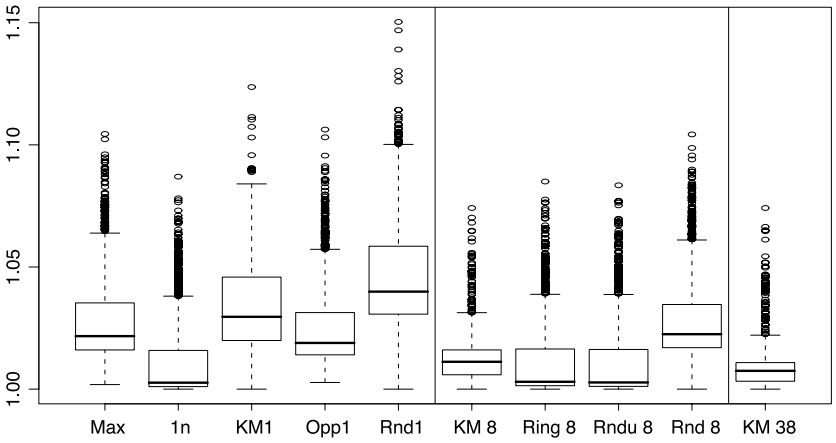

Fig. 5. Boxplots of the performance measure (8) for $n = 200$, $k = 20$ and $\delta = 30\%$; vertical lines separate groups of comparable starting strategies.

into three parts. The five first boxplots are constructed for strategies where A contains only one starting point. The next four boxplots are built for bigger sets: the first one corresponds to starting points based on the equivariant k-mean clusters, the second one to opposite vertices built orthogonally, the third one to random starting points based on a reordering of the index set, and the fourth one to random vertices. The number of elements in each of these sets has been forced to be identical. The common size of these sets is given by the maximal number of crown (or ring) points one can build for the considered configuration. The size is written next to the name of the strategy in the figures. For the equivariant k-means approach, if one cannot keep all vertices, only the first clusters are selected. However, to get an idea of the performance of the cluster methodology, an additional boxplot is represented for the set consisting of all the equivariant k-means starting points (according to the algorithm outlined in the previous section).

Both figures show that the equivariant k-means strategy is doing a good job with respect to the other approaches and it is worth noting that this fact holds for all the other considered sample sizes, dimensions and percentages of contamination. For small sample sizes, the corresponding boxplots are even concentrated on 1, meaning that, for most simulations, the best starting points belong to that family. One can also notice that the orthogonal starting points v_{ring_i} and v_{rndu_i} perform better and better as the dimension gets bigger and bigger while pure random starting points v_{rnd_i} behave well for small dimensions but become less and less efficient by comparison to the other strategies when the dimension increases. The neutral point $p_\circ$ performs surprisingly well when $k = 20$ while, as expected, starting the descent from the maximum of the function, i.e. from the assumed most contaminated starting point, does not seem to be a good strategy.

Based on these results, a preliminary conclusion would be that the best minimisation procedure is obtained when one uses all the different kinds of starting point. However, when speed is a major concern (as in Section 4.4 below), the number of starting points may be decreased and we would advise to use only the starting points constructed from the equivariant k-means clusters and $p_\circ$.

Please cite this article in press as: Schyns, M., et al., RelaxMCD: Smooth optimisation for the Minimum Covariance Determinant estimator. Computational Statistics and Data Analysis (2009), doi:10.1016/j.csda.2009.11.005

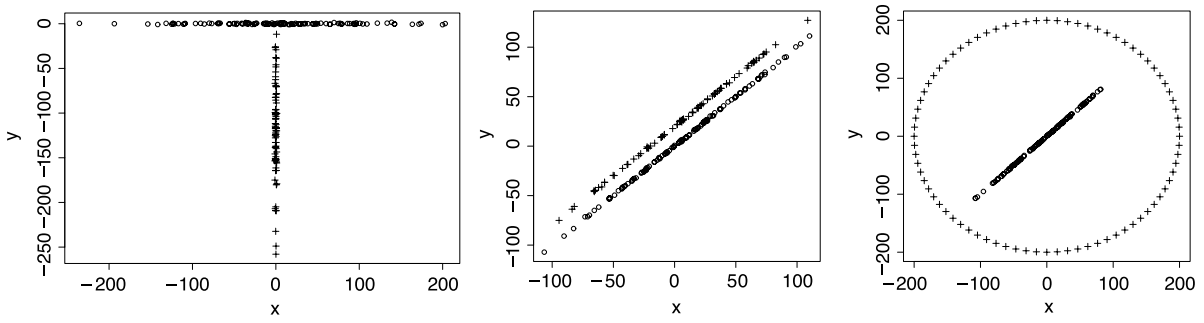

Fig. 6. Additional configurations represented in two dimensions with $n = 200$ observations: orthogonal contamination (left panel), cigar contamination (middle panel) and halo contamination (right panel).

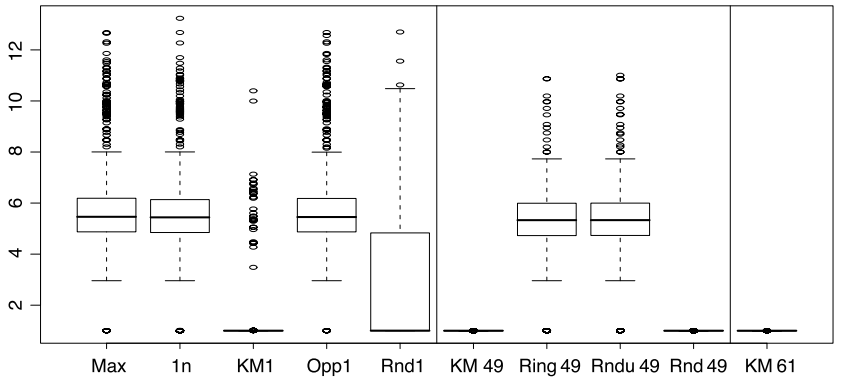

Fig. 7. Boxplots of the performance measure (8) for $n = 200$, $k = 3$ and $\delta = 40\%$ under the orthogonal contamination set-up (left panel of Fig. 6); vertical lines separate groups of comparable starting strategies.

4.3. Non-spherical data configurations

The simulations of the previous section were conducted under spherical configurations of the data and the equivariant k-means family of starting points was shown to perform well. One could however wonder whether the good performance of these starting points is not due to the particular simulation set-up considered: a mixture of two spherical distributions with identical shapes, as Fig. 3 clearly shows. Referees suggested trying other configurations which one might assume *a priori* to be less favourable to the equivariant k-means technique. The main data cloud is now distributed according to a nearly degenerate multivariate distribution, while the three following structures are used for the outliers:

- The outliers are distributed according to a nearly degenerate normal distribution having its principal axis orthogonal to the principal axis of the distribution of the clean data.
- The outliers are distributed according to a nearly degenerate normal distribution translated from the main bulk of data and scaled such that they form a kind of cigar parallel with the main cloud.
- The outliers are distributed on a halo around the main cloud of data points.

Fig. 6 illustrates these three additional data configurations in two dimensions.

Figs. 7–9 represent the boxplots of the performance measure (8) when 1000 data sets are generated according to these three models for $n = 200$, $k = 3$ and under a percentage of contamination equal to $\delta = 40\%$. The main message is clear. The equivariant k-means starting strategy continues to do well for these data configurations, performing comparably to its closest competitors (randomly chosen starts). Since the form of data configuration is, of course, generally unknown to the user in practice – while the equivariant k-means procedure has been seen to perform best for spherical mixtures – in case one wants to restrict the number of starting points, focusing on those derived from the equivariant k-means procedure is certainly a good strategy.

4.4. Comparisons of the algorithms

4.4.1. Performance measures

The three dimensions *speed*, *depth* and *robustness* will be taken into account in order to compare, here, the performance of RelaxMCD, FASTMCD and FSA on the data configuration (7). For FSA, the improved version (Hawkins and Olive (1999))

Please cite this article in press as: Schyns, M., et al., RelaxMCD: Smooth optimisation for the Minimum Covariance Determinant estimator. Computational Statistics and Data Analysis (2009), doi:10.1016/j.csda.2009.11.005

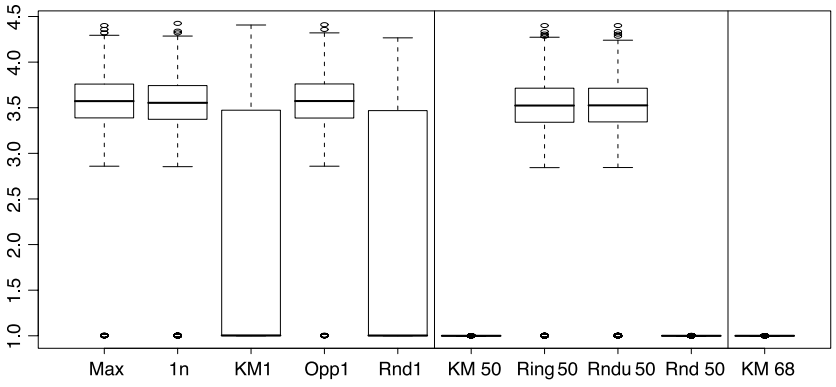

Fig. 8. Boxplots of the performance measure (8) for $n = 200$, $k = 3$ and $\delta = 40\%$ under the cigar-shaped contamination set-up (middle panel of Fig. 6); vertical lines separate groups of comparable starting strategies.

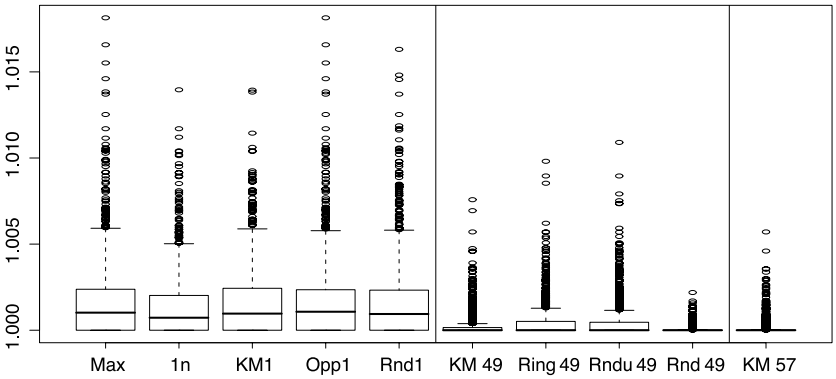

Fig. 9. Boxplots of the performance measure (8) for $n = 200$, $k = 3$ and $\delta = 40\%$ under the halo contamination set-up (right panel of Fig. 6); vertical lines separate groups of comparable starting strategies.

was chosen since the time for running the basic procedure is prohibitive for the simulation set-up considered here. For RelaxMCD, only $p_\circ$ and the starting points based on equivariant k-means are considered.

As far as the optimisation problem is concerned, speed and depth would be the main concerns (as Fig. 1 illustrates). However, when dealing with algorithms computing robust estimators, robustness should also be achieved. Let us then introduce the performance measures applied to each of these criteria.

Depth can only be assessed in a comparative way. Indeed, except for small sample sizes and dimensions, it is not possible to get the exact (or deepest) solution of the MCD problem. One can only determine, for each simulated data, whether one of the algorithms has reached lower values than the others. Therefore, a performance measure for depth will be defined for all combinations of two algorithms and will simply consist of deriving the percentages of simulations for which these two algorithms performed equivalently, as well as the percentages of those where one of them got lower than the other.

Of course, if only depth is at stake, the naive enumeration of all h-subsets would necessarily be the best way of proceeding. However, it is clearly infeasible in a reasonable time. A comparison of the speed of the procedures would allow us to better interpret the depth performance. The difficulty when comparing speed is that, while the execution time is often measured in seconds, less than one second is generally required by these algorithms to reach a solution, at least when less than 200 observations are considered. Moreover, the most important point here is that one wishes to get a result fast, without requiring that the algorithm be a fraction of second quicker than another one. Therefore, depending on the cases, two different measures will be reported: either the mean computation time over the 1000 simulations if this mean is larger than a few seconds, or the percentage of simulations for which less than one second was required.

The last performance measure should be a kind of robustness indicator since the MCD estimator should resist up to 50% of contamination in the data. It is maybe worth mentioning that the link between robustness and depth is not so clear. First of all, the MCD objective function has many local minima. These may be quite different as far as the objective values are concerned but quite similar if one considers the resulting statistical results (robustness, estimation of location and/or scatter). All these robust local minima would then be 'equivalent' and as good from a statistical point of view. In practice, one could even say that these solutions are 'interchangeable', but in a comparison of the performance of the quality of the minimisation, preference will be given to the solution achieving the smallest objective value and also to the algorithm yielding it. Conversely, when contamination is much concentrated in the data, a contaminated solution might end up as the lowest one (think of a point mass which would, with some good points, construct an h-subset of a cigar shape, meaning that

Please cite this article in press as: Schyns, M., et al., RelaxMCD: Smooth optimisation for the Minimum Covariance Determinant estimator. Computational Statistics and Data Analysis (2009), doi:10.1016/j.csda.2009.11.005

Table 1
Depth performance measures under simulation set-up (7), with different sample sizes and dimensions and with $\delta = 30\%$.

n	k	Depth					
		RelaxMCD vs. FASTMCD		RelaxMCD vs. FSA		FASTMCD vs. FSA	
		$<$ (%)	$=$ (%)	$<$ (%)	$=$ (%)	$<$ (%)	$=$ (%)
100	10	82.9	1.4	61.8	2.9	34.0	0.8
200	5	47.2	29.1	90.8	5.0	87.1	4.0
200	10	90.4	1.4	84.3	2.3	51.2	0.7
200	15	90.4	0.2	64.9	0.5	22.4	0.2
200	20	98.2	0.0	30.0	0.9	0.9	0.0
500	10	87.6	3.9	94.8	1.4	77.0	1.3
1000	10	82.3	11.8	99.2	0.1	96.0	1.7

Table 2
Speed and Robustness performance measures under simulation set-up (7), with different sample sizes and dimensions and with $\delta = 30\%$.

n	k	Speed			Robustness		
		RelaxMCD (%)	FASTMCD (%)	FSA (%)	RelaxMCD (%)	FASTMCD (%)	FSA (%)
100	10	91	63	64	85.7	88.9	71.8
200	5	82	66	66	99.4	100	99.9
200	10	52	59	35	98.2	99.8	78.9
200	15	45	45	23	65.3	30.2	25.3
200	20	40	30	18	19.6	1.7	5.2
500	10	3.67′	0.61′	2.06′	98.9	99.9	91.0
1000	10	19′	0.56′	4.95′	99.4	100	96.7

the determinant of the corresponding covariance matrix would be close to zero). In that particular case, the lowest solution would not be convenient from a statistical point of view. Since simulated data are used, it is easy to check whether an obtained solution contains outliers or not. Robustness performance will therefore be assessed by means of the percentages of simulations for which the optimal vertex is clean.

4.4.2. Results

All simulations were performed on a dual core 2.4 GHz windows system and using the stand-alone versions of the three softwares. The default values of the input parameters were chosen for FASTMCD. It could have been interesting to see how the performance measures vary according to the values attributed to some of them (particularly the number of starting points) but the Fortran code that was used did not let the user modify many of the inputs. Choosing the number of starting points was however possible for the FSA procedure. In order to simplify comparisons, it was decided to choose the same number of starting points for RelaxMCD and FSA.

While the next section will be devoted to the overall comparison between the different descent strategies of RelaxMCD, only the l_{max}-swap strategy will be considered here. Therefore, comparing FASTMCD and RelaxMCD will provide valuable information about the performance of the designed starting points with respect to random ones, the way of descending being similar in the two algorithms.

Tables 1 and 2 provide a summary of the most representative results. For some of the sample sizes and dimensions and for a percentage of contamination equal to 30%, Table 1 reports the depth performance measures. For all pairs of algorithms, the column '$<$' corresponds to the percentages of simulations for which the first algorithm reached a strictly lower minimum than the second one while the column indicated by '$=$' describes the occurrence of comparable situations (same value of the objective function, but not necessarily the same vertex). One sees that RelaxMCD goes further down with respect to the two other algorithms in most cases. When $n = 200$ and $k = 5$, its score with respect to FASTMCD is under 50% but, when considering also the simulations for which both algorithms are equivalent, the score goes up to 76%. This illustrates again the idea of 'interchangeable' solutions. When FASTMCD and RelaxMCD gets different solutions, these are often close one to the other, but with a slight advantage for RelaxMCD as far as the depth is concerned. Also, when compared to FSA, RelaxMCD gets lower performance depth scores when the ratio n/k gets small. The depth comparison of FSA and FASTMCD is more tricky: when the ratio n/k is small, the performance measure is in favour of FSA and vice-versa when the ratio n/k gets bigger.

Of course, most algorithms could improve their depth performance if the number of starting points were increased. These comments must therefore be analysed taking into account the speed of the procedures. Table 2 lists the speed performance measures as well as the robustness scores of the three algorithms.

When only speed is considered, one can certainly conclude that all algorithms are fast. The mean computation time is reported only when the sample size gets above 500 in which case RelaxMCD (and especially the starting step based on the equivariant k-means algorithm) and FSA do take more time than FASTMCD (which applies the partitioning idea as explained previously). For all other data configurations, the mean computation times lie below 2 seconds and the percentages of simulations requiring less than one second to run are given. These percentages decrease with n and k (as expected) being

Please cite this article in press as: Schyns, M., et al., RelaxMCD: Smooth optimisation for the Minimum Covariance Determinant estimator. Computational Statistics and Data Analysis (2009), doi:10.1016/j.csda.2009.11.005

quite equivalent for RelaxMCD and FASTMCD, but lower for FSA. While it is true that increasing the number of starting points of FSA would improve its depth results, it would deteriorate even more its poor speed performance.

Turning now to the second panel of Table 2, one might conclude that the algorithms do not achieve high robustness as desired. Indeed, Table 2 lists the percentages of simulations for which a clean result has been reached and the reported values can be quite small. One has, however, to remember that the contamination is not extreme as Fig. 3 illustrates. Some outliers (indicated as x) are so close to the good data that one may easily imagine a final h-subset including some of them. Obviously, that would not result in the breakdown of the estimators, as one usually expects when a contaminated h-subset is used. Nevertheless, we expect the optimum subset to be in the centre of the good data cloud for most simulations. Comparing the performance of all algorithms, one sees that RelaxMCD does a good job, followed by the two others which are quite equivalent on that criteria. When the dimension k increases, the robustness of each algorithm decreases but RelaxMCD remains robust far longer, e.g. for 200 observations in dimension 20, RelaxMCD still provides a clean solution in 20% of the cases against only 2% for FASTMCD and 5% for FSA.

The reported results were based on 30% of contamination since that is the configuration where the differences of behaviour between the three algorithms were most obvious. Note that when there is less contamination in the data, e.g. 10% or 20%, each algorithm performs well. Up to 200 observations, each of them is fast and 100% robust. As far as depth is concerned, the same conclusions as those given above hold: RelaxMCD outperforms the two other algorithms, but FASTMCD behaves nicely for small dimensions.

4.5. Comparison of descent strategies

Up to now, all the results were based on the l_{max}-swaps descent strategy. The same experiments were repeated with 1-swaps and $l_{deepest}$-swaps. According to speed, robustness and depth, $l_{deepest}$-swaps lead to the same results as those obtained with l_{max}-swaps, but with more steps in each part of the descent (these steps are quick but necessary). From the same starting points, 1-swaps usually reach deeper and more robust solutions than l_{max}-swaps but the price to pay for that is a much bigger computation time. The same depth could probably be attained with l_{max}-swaps by increasing the number of starting points. Since l_{max}-swaps provide the best compromise between all criteria, this is the descent strategy that RelaxMCD uses by default. These comparisons of descent illustrate again that FASTMCD is built efficiently since its C-steps are equivalent to the l_{max}-swaps procedure, shown to be the best descent technique among those tried out.

4.6. Other simulation set-ups and small data sets

The performance of the three algorithms was also compared on the following simulation set-up: the outliers were divided into two groups instead of being clustered in one area. The first cluster was located on the first axis and the second one on the second axis, the distance between the centres of these groups ensuring again that the corresponding 95% confidence regions do not overlap. Simulations showed that the same overall conclusions as those derived on set-up (7) can be drawn (the results are therefore not included to save space). Another set-up often considered of interest in robustness consists of concentrating the contamination on a point-mass. This configuration has not been investigated as systematically as the other two, but some preliminary results show that, when dealing with high percentages of contamination, the algorithms have a tendency to include the point-mass in the optimal h-subset, yielding a nearly singular optimal covariance matrix. This behaviour was already pointed out by Peña and Prieto (2001) and corresponds to a kind of breakdown of the MCD criterion and not to the breakdown of the algorithms designed to compute it, since the contaminated h-subset has indeed the lowest determinant among all vertices reached. Further investigation is needed to measure the risk of finding such 'cigars' when computing the MCD estimators in practice.

Comparing algorithms by means of simulations is convenient since the optimal solution and the number and location of outliers are known in advance. However, one should also illustrate the good performance on real data, the standard small data sets of Rousseeuw and Leroy (1987) being a kind of usual test for newly introduced algorithms. Applying RelaxMCD to these data sets yielded the same h-subsets as those listed in Table 1 of Rousseeuw and Van Driessen (1999), in a fraction of second as does FASTMCD.

5. Conclusions

In this paper, the general relaxation methodology outlined in Critchley et al. (forthcoming) was further specialised in order to fit, as well as possible, the MCD optimisation problem. The main focus has been on the design of clever starting points, even if some additional improvements intrinsically related to the MCD estimator (i.e. core inflation, free swaps) were also briefly introduced. Non-random starting points allow us to systematically (and, in our case, geometrically) cover the search space and seem to perform quite well with respect to starting points based on random sampling. This is especially true of the equivariant k-means approach.

The 'tuned' algorithm, called RelaxMCD, was then compared by means of simulations with the two most well-known algorithms computing the MCD estimators: FASTMCD of Rousseeuw and Van Driessen (1999) and Improved FSA of Hawkins and Olive (1999). These comparisons showed that relaxMCD achieves an efficient compromise between the different criteria considered of interest in robust computational statistics: depth, speed and robustness.

Please cite this article in press as: Schyns, M., et al., RelaxMCD: Smooth optimisation for the Minimum Covariance Determinant estimator. Computational Statistics and Data Analysis (2009), doi:10.1016/j.csda.2009.11.005

Now, one could wonder whether there is any point introducing yet another algorithm while already two highly efficient ones are widely available. However, we believe that the approach followed in the construction of RelaxMCD is worthwhile since it relies on theoretical properties of the target function like concavity and smoothness, while the others do not take advantage of these. Moreover, as mentioned above, the different steps of RelaxMCD give insights into the corresponding steps applied in FASTMCD or FSA.

Appendix

Before proving Proposition 1, some algebraic results are worth stating.

Lemma 1. *Let p and p^* be two points in $\mathbb{P}^n_{-m}$ and $0 < \lambda < 1$.*

$$\bar{x}((1-\lambda)p + \lambda p^*) = (1-\lambda)\bar{x}(p) + \lambda\bar{x}(p^*),$$
$$M((1-\lambda)p + \lambda p^*) = (1-\lambda)M(p) + \lambda M(p^*),$$
$$\hat{\Sigma}((1-\lambda)p + \lambda p^*) = (1-\lambda)\hat{\Sigma}(p) + \lambda\hat{\Sigma}(p^*) + A,$$

where $A = \lambda(1-\lambda)(\bar{x}(p) - \bar{x}(p^*))(\bar{x}(p^*) - \bar{x}(p))^T$ is a rank one matrix and where $M(p)$ denotes the second moment.

Proof of Proposition 1. By definition,

$$\begin{aligned}
t((1-\lambda)p + \lambda p^*) &= \log\det\hat{\Sigma}\left((1-\lambda)p + \lambda p^*\right)\\
&= \log\det\Big((1-\lambda)\hat{\Sigma}(p) + \lambda\hat{\Sigma}(p^*) + A\Big)\\
&= \log\{\det((1-\lambda)\hat{\Sigma}(p) + \lambda\hat{\Sigma}(p^*))(1 + \lambda(1-\lambda)(\bar{x}(p) - \bar{x}(p^*))^T\\
&\quad \times [(1-\lambda)\hat{\Sigma}(p) + \lambda\hat{\Sigma}(p^*)]^{-1}(\bar{x}(p) - \bar{x}(p^*)))\}\\
&= \log\det[(1-\lambda)\hat{\Sigma}(p) + \lambda\hat{\Sigma}(p^*)] + \log\{1 + \lambda(1-\lambda)(\bar{x}(p) - \bar{x}(p^*))^T\\
&\quad \times [(1-\lambda)\hat{\Sigma}(p) + \lambda\hat{\Sigma}(p^*)]^{-1}(\bar{x}(p) - \bar{x}(p^*))\}\\
&\geq \log\det[(1-\lambda)\hat{\Sigma}(p) + \lambda\hat{\Sigma}(p^*)] \text{ with equality iff } \bar{x}(p) = \bar{x}(p^*), \text{ as log is strictly increasing}\\
&\geq (1-\lambda)\log\det\hat{\Sigma}(p) + \lambda\log\det\hat{\Sigma}(p^*) \text{ with equality iff } \hat{\Sigma}(p) = \hat{\Sigma}(p^*),\\
&\quad \text{as } \log\det(.) \text{ is strictly concave on the set of positive definite matrices}\\
&= (1-\lambda)t(p) + \lambda t(p^*).
\end{aligned}$$

Thus, $t(.)$ is concave on $\mathbb{P}(\hat{\Sigma}^{-1})$. Further, $t(.)$ is linear on the line segment $[p, p^*]$ in $\mathbb{P}(\hat{\Sigma}^{-1})$ iff $\bar{x}(p) = \bar{x}(p^*)$ and $\hat{\Sigma}(p) = \hat{\Sigma}(p^*)$. Let us denote the condition '$\bar{x}(p) = \bar{x}(p^*)$ and $\hat{\Sigma}(p) = \hat{\Sigma}(p^*)$', condition C. Condition C can also be written as $\bar{x}(p) = \bar{x}(p^*)$ and $M(p) = M(p^*)$.

Equivalently, $t(.)$ is strictly concave on the line segment $[p, p^*]$ in $\mathbb{P}(\hat{\Sigma}^{-1})$ iff *not* C is valid, i.e. iff either $\bar{x}(p) \neq \bar{x}(p^*)$ or $M(p) \neq M(p^*)$. Now, C implies $\hat{\Sigma}(p) = \hat{\Sigma}(p^*)$, and then $t(p) = t(p^*)$.

So, condition C implies that $t(.)$ is constant on $[p, p^*]$, while clearly, *not* C implies that $t(.)$ is not constant as it is strictly concave. Therefore, C is equivalent to saying that $t(.)$ is constant on $[p, p^*]$ and the proof is complete. □

Before proving Proposition 2, it is worth recalling that if A is nonsingular and if b is a vector of the same order, then,

$$\det(A \pm bb^T) = (\det A)\{1 \pm b^T A^{-1} b\}. \tag{9}$$

Also, as is easily verified, if A is nonsingular and if B is of the same order, then for all small enough $\varepsilon > 0$,

$$(A + \varepsilon B)^{-1} = A^{-1} - \left[\sum_{l=1}^{\infty}(-1)^{l+1}(\varepsilon A^{-1}B)^l\right]A^{-1}. \tag{10}$$

The following lemma will be useful:

Lemma 2. *If A and B are symmetric of the same order with A positive definite then, for all small enough $\varepsilon > 0$,*

$$\log\det(A + \varepsilon B) = \log\det(A) + \varepsilon\, trace(A^{-1}B) - \frac{1}{2}\varepsilon^2 trace((A^{-1}B)^2) + O(\varepsilon^3).$$

Proof of Proposition 2. Let $p \in \mathbb{P}(\widehat{\Sigma}^{-1})$ and $d \equiv (d_i) \in \mathbb{R}^n$ with $\sum_{i=1}^n d_i = 0$. Then, for all small enough $\delta > 0$:

$$\widehat{\Sigma}(p + \delta d) = \widehat{\Sigma}(p) + \delta\sum_{i=1}^{n} d_i(x_i - \bar{x}(p))(x_i - \bar{x}(p))^T - \delta^2\bar{x}(d)\bar{x}^T(d), \tag{11}$$

Please cite this article in press as: Schyns, M., et al., RelaxMCD: Smooth optimisation for the Minimum Covariance Determinant estimator. Computational Statistics and Data Analysis (2009), doi:10.1016/j.csda.2009.11.005

where $\bar{x}(d) := \sum_{i=1}^{n} d_i x_i \equiv \sum_{i=1}^{n} d_i (x_i - \bar{x}(p))$. Setting

$$A := \widehat{\Sigma}(p), \qquad B := \sum_{i=1}^{n} d_i (x_i - \bar{x}(p))(x_i - \bar{x}(p))^T \quad \text{and} \quad b := \sum_{i=1}^{n} d_i (x_i - \bar{x}(p)),$$

we have $\widehat{\Sigma}(p + \delta d) = A + \delta B - \delta^2 b b^T$. Thus,

$$\begin{aligned} t(p + \delta d) &= \log\det(A + \delta B) + \log\{1 - \delta^2 b^T (A + \delta B)^{-1} b\}, \text{ using (9).} \\ &= \log\det(A + \delta B) + \log\{1 - \delta^2 b^T A^{-1} b + O(\delta^3)\}, \text{ using (10).} \\ &= \log\det A + \delta \operatorname{trace}(A^{-1}B) - \frac{1}{2}\delta^2 \left\{\operatorname{trace}((A^{-1}B)^2) + 2 b^T A^{-1} b\right\} + O(\delta^3), \text{ using Lemma 2} \\ &= t(p) + \delta d^T (D_{ii}(p)) - \frac{1}{2}\delta^2 d^T \left\{D^{(2)}(p) + 2D(p)\right\} d + O(\delta^3). \end{aligned}$$

But: $t(p + \delta d) = t(p) + \delta d^T t^c(p) + \frac{1}{2}\delta^2 d^T t^{cc}(p) d + O(\delta^3)$.

Identifying terms completes the proof of Proposition 2. □

Proof of Proposition 3. Let $\mathbb{P}^{\circ}_{-m}$ denote the relative interior of $\mathbb{P}^{n}_{-m}$. Then, the maximum occurs at such a relative interior point

$$\begin{aligned} &\Leftrightarrow \exists\, p \in \mathbb{P}^{\circ}_{-m} \quad \text{st } t^c(p) = 0, \\ &\Leftrightarrow \exists\, p \in \mathbb{P}^{\circ}_{-m} \quad \text{st } (I_n - J_n)\,(D_{ii}(p)) = 0, \\ &\Leftrightarrow \exists\, p \in \mathbb{P}^{\circ}_{-m} \quad \text{st } (D_{ii}(p)) \propto 1_n, \\ &\Leftrightarrow \exists\, p \in \mathbb{P}^{\circ}_{-m} \quad \text{st } \forall\, i,\, D_{ii}(p) = k, \text{ since } \sum_{i=1}^{n} p_i D_{ii}(p) = k, \\ &\Leftrightarrow \exists p \in \mathbb{P}^{\circ}_{-m} \quad \text{st } \{z_i(p) := \hat{\Sigma}^{-\frac{1}{2}}(p)(x_i - \bar{x}(p)) : i = 1, \ldots, n\}, \text{ lie on a sphere of radius } \sqrt{k} \\ &\Rightarrow \exists \text{ an affine transformation st } \{x_i\} \text{ lie on a sphere.} \quad \square \end{aligned}$$

References

Agulló, J., 1998. Computing the minimum covariance determinant estimator. Universidad de Alicante.
Bernholt, T, Fisher, P., 2004. The complexity of computing the MCD-estimator. Theoretical Computer Science 326, 383–398.
Butler, R.W., Davies, P.L., Jhun, M., 1993. Asymptotics for the minimum covariance determinant estimator. The Annals of Statistics 21, 1385–1400.
Critchley, F., Schyns, M., Haesbroeck, G., Fauconnier, C., Lu, G., Atkinson, R.A., Wang, D.Q., 2009. A relaxed approach to combinatorial problems in robustness and diagnostics. Statistics and Computing (forthcoming).
García-Escudero, L.M., Gordaliza, A., 2007. The importance of the scales in heterogeneous robust clustering. Computational Statistics and Data Analysis 51, 4403–4412.
Hawkins, D.M., 1994. The feasible solution algorithm for the minimum covariance determinant estimator in multivariate data. Computational Statistics and Data Analysis 17, 197–210.
Hawkins, D.M., Olive, D.J., 1999. Improved feasible solution algorithms for high breakdown estimators. Computational Statistics and Data Analysis 30, 1–11.
Hawkins, D.M., Olive, D.J., 2002. Inconsistency of resampling algorithms for high-breakdown regression estimators and a new algorithm. Journal of the American Statistical Association 97, 136–148.
Horst, R., Tuy, H, 1995. Global optimization. In: Deterministic Approaches, 3rd ed. Springer.
Johnson, R.A., Wichern, D.W., 1992. Applied Multivariate Statistical Analysis, 3rd ed. Prentice-Hall.
Pardalos, P.M., Rosen, J.B., 1987. Constrained Global Optimization: Algorithms and Applications. In: Lecture Notes in Computer Science, Springer-Verlag, New York.
Peña, D., Prieto, F.J., 2001. Multivariate outlier detection and robust covariance matrix estimation. Journal of the American Statistical Association 43, 286–303.
Rousseeuw, P.J., 1985. Multivariate estimation with high breakdown point. In: Grossmann, W., Pflug, G., Vincze, I., Wertz, W. (Eds.), Mathematical Statistics and Applications, vol. B. Dordrecht, Reidel, pp. 283–297.
Rousseeuw, P.J., Leroy, A.M., 1987. Robust Regression and Outlier Detection. John Wiley, New York.
Rousseeuw, P.J., Van Driessen, K., 1999. A fast algorithm for the minimum covariance determinant estimator. Technometrics 41, 212–223.
Todorov, V., 1992. Computing the minimum covariance determinant estimator (MCD) by simulated annealing. Computational Statistics and Data Analysis 14, 515–525.
Woodruff, D.L., Rocke, D.M., 1994. Computable robust estimation of multivariate location and shape in high dimension using compound estimators. Journal of the American Statistical Association 89, 888–896.

Please cite this article in press as: Schyns, M., et al., RelaxMCD: Smooth optimisation for the Minimum Covariance Determinant estimator. Computational Statistics and Data Analysis (2009), doi:10.1016/j.csda.2009.11.005