On An Optimization Problem in Robust Statistics

Biman Chakraborty
Dept. of Statistics and Applied Probability,
The National University of Singapore
6, Science Drive 2, Singapore 117546, SINGAPORE
e-mail: stabc@nus.edu.sg

Probal Chaudhuri
Theoretical Statistics and Mathematics Unit,
Indian Statistical Institute,
203, B.T. Road, Calcutta 700108, INDIA
e-mail: probal@isical.ac.in

March 18, 2005

Abstract

In this paper, we consider a large class of computational problems in robust statistics, which can be formulated as selection of optimal subsets of data based on some criterion function. To solve such problems, there are largely two classes of algorithms available in the literature. One is based on purely random search, and the other is based on deterministically guided strategies. Though these methods can achieve satisfactory results in some specific examples, none of them can be used satisfactorily for a large class of similar problems either due to their very long expected waiting time to hit the true optimum or due to their failure to come out of a local optimum when they get trapped there. Here, we propose two probabilistic search algorithms, and under some conditions on the parameters of the algorithms, we establish the convergence of our algorithms to the true optimum. We also show some results on the probability of hitting the true optimum if the algorithms are run for a finite number of iterations. Finally, we compare the performance of our algorithms to some commonly available algorithms for computing some popular robust multivariate statistics using real data sets.

Keywords: Combinatorial optimization; Iterated conditional modes; Half-space depth; Least median of squares regression; MCD estimator; Non-homogeneous Markov chains; PPS sampling; Transformation and retransformation estimates.

1 Introduction : Robust Estimation In Multivariate And Regression Analysis

Many robust estimation problems involve finding an optimal subset of the data points, which minimizes an objective function defined on a collection of subsets of the data points. For instance, the minimum covariance determinant estimates of multivariate scatter and location require getting a half sample having the minimum determinant of its covariance matrix (Rousseeuw and Leroy 1987). Computation of Tukey's half space depth (Tukey 1975) of a given point in the d-dimensional space with respect to a set of d-dimensional data points can also be formulated as the problem of finding an optimal hyperplane passing through $d-1$ of the data points

and that given point so that the number of data points on one of the two half spaces containing smaller number of data points created by that hyperplane is minimized (see e.g. Chaudhuri and Sengupta 1993).

Many well known estimates of multivariate location with high breakdown properties like coordinatewise median and spatial median are not affine equivariant. Chakraborty and Chaudhuri (1998) suggested a general procedure based on the idea of transformation and retransformation to construct affine equivariant robust estimates of location from estimates that are not affine equivariant. This idea was subsequently followed up and investigated further by Randles (2000), Hettmansperger and Randles (2002), etc. This strategy can be briefly described as follows. Construct a new coordinate system by fixing one of the data point to be the origin and the coordinate axes are given by joining the origin with other d data points, where d is the dimension of the data points. We transform all observations in the new coordinate system and compute either coordinatewise median or the spatial median in that system and then retransform back the computed median in the original coordinate system. Note that the coordinate system is based on $d+1$ observations, and the efficiency of the resulting estimate depends on the choice of those $d+1$ observations. Chakraborty and Chaudhuri (1998) and Chakraborty, Chaudhuri and Oja (1998) discussed some optimality criteria for selecting the best subset of $d+1$ observations to construct the data-driven coordinate system.

The computation of least median of squares regression estimates (Rousseeuw 1984) with d covariates and a dependent variable can also be viewed as a problem of determining two parallel hyperplanes passing through a total of $d+2$ data points in the $(d+1)$-dimensional space so that there are at least half of the data points enclosed within the region bounded by these two hypeplanes, and the distance between these two hyperplanes measured along the axis of the dependent variable is minimum.

Consider now n data points $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n \in \mathbb{R}^d$. Each of the optimization problems described above can be viewed as a special case of the general optimization problem, where one is required to determine an optimal l-subset ($l < n$) $\{\boldsymbol{X}_{i_1}, \boldsymbol{X}_{i_2}, \ldots, \boldsymbol{X}_{i_l}\}$ of the data points that minimizes a non-negative function denoted by $h(i_1, i_2, \ldots, i_l)$ $(= h(\boldsymbol{X}_{i_1}, \boldsymbol{X}_{i_2}, \ldots, \boldsymbol{X}_{i_l}))$. The function h here is deter-

mined by the specific robust estimation problem. Clearly, minimization of h by a complete enumeration of h over all possible l-subsets will be computationally prohibitive even for moderately large values of n and l. Further, in some cases (e.g., in the case of minimum covariance determinant estimation), the value of l depends on n, and it grows fast with n.

The simplest probabilistic algorithm to solve a combinatorial optimization problem is the purely random search, which chooses a subset of data points by simple random sampling, evaluates the objective function at that subset and then repeats this procedure for a large number of times. The minimum of the objective function values in these iterations is taken as the approximate global minimum for the original problem. The biggest advantage of this method is that it is very simple to implement, and if the number of iterations (say, N) goes to infinity, the probability of hitting the true optimum goes to one. In other words, it is a convergent algorithm, but in most of the problems, especially when n and l are large, the expected number of iterations required to get the true optimum is considerably high. For minimizing the criterion function in transformation retransformation procedure, no algorithm better than purely random search is available in the literature, and the same is true for algorithms for computing half-space depth for $d > 3$ and least median of squares regression with large number of covariates.

In contrast, there are some general purpose deterministic algorithms to solve many combinatorial optimization problems. For instance, coordinatewise maximal descent algorithm is well-known in combinatorial optimization, which is more popularly known as iterated conditional modes (ICM) algorithm (Besag, 1986, Kittler and Föglein, 1984) in statistics literature. A single iteration consists of l steps, where l is the size of the subset. At each step, it updates only one element of the subset fixing the rest of the elements by minimizing the objective function over that element only. Once the initial choice of the subset is fixed, it proceeds in a deterministic way. The algorithm often terminates in a local minimum next to the initial configuration after a few iterations, and the result sensitively depends on the initial configuration.

An example of problem specific guided deterministic search is the FAST-MCD algorithm (Rousseeuw and van Driessen 1999) for computing minimum covariance

determinant (MCD) estimates of multivariate location and scatter. Starting from an initial subset of $d+1$ observations for a d-variate data, it proceeds in a deterministic fashion using some structural properties of the MCD criterion. Then the same procedure is repeated for many randomly selected initial configurations. This is a fast algorithm but there is no guarantee of convergence to the global minimum, and there may not exist any initial subset of size $d+1$ from which the procedure can lead to the global minimum.

In regard to the choice between simple random search and deterministically guided search, one can draw the analogy with balancing between bias and variance in the context of estimation. In simple random search, the probability of hitting the global optimum converges to one but the expected waiting time to hit that is very high, and this is analogous to large variance being associated with small bias in a statistical estimation problem. On the other hand, guided deterministic search procedures run quite fast but they may never converge to the global optimum and may get trapped in a local optimum. This is comparable to small variance being accompanied with large bias in statistical estimation.

In this article, we will investigate two probabilistic algorithms to minimize h, which are motivated by the optimization techniques based on Markov chains with finite state spaces as used in simulated annealing (Kirkpatrick, Gelatt and Vechhi 1983). Both of these algorithms are well suited for the general optimization problem that arises in robust statistics, and consequently, it is applicable to a large number of robust estimation problems. Further, both the algorithms are guaranteed to converge to a global minimum and not to be trapped in any local minimum.

2 Probabilistic Search For The Optimum

To begin with, let $\mathcal{S}_l$ be the collection of all ordered l-tuples of the form $(i_1, \ldots, i_l)$ with $1 \leq i_r \neq i_s \leq n$, $1 \leq r \neq s \leq l$, and our goal is to determine an optimal l-tuple that minimizes

$$h(i_1, i_2, \ldots, i_l),$$

which is a nonnegative objective function of the observations $\boldsymbol{X}_{i_1}, \ldots, \boldsymbol{X}_{i_l}$. Note that the objective function h may or may not have a unique minimum. In this section, we propose two iterative algorithms, which will proceed by evaluating h on the sequence $\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_k$, where $\boldsymbol{Z}_i \in \mathcal{S}_l$ and at the same time evaluating a sequence of "best candidate minima", $\boldsymbol{M}_1, \ldots, \boldsymbol{M}_k$, such that $\boldsymbol{M}_i \in \mathcal{S}_l$ and $h(\boldsymbol{M}_1) \geq h(\boldsymbol{M}_2) \geq \cdots \geq h(\boldsymbol{M}_k) \geq \cdots$.

2.1 Algorithm 1: Search By PPS Sampling

At the k-th iteration step, let $\boldsymbol{Z}_k = (i_1^{(k)}, \ldots, i_l^{(k)}) \in \mathcal{S}_l$. Let g_k be a decreasing function defined on $[0, \infty)$ with range in $(0, 1)$, the choice of which will be specified later. The l-tuple $\boldsymbol{Z}_k$ is updated to $\boldsymbol{Z}_{k+1}$ in the following manner. First, $i_1^{(k+1)}$ is chosen from the set $\{1, \ldots, n\}$ using the standard *probability proportional to size* (PPS) sampling, where the size of the element i is taken to be $g_k\{h(i, i_2^{(k)}, \ldots, i_l^{(k)})\}$. In other words, $i_1^{(k+1)}$ is selected using a probability distribution $\{p_i\}$ with

$$p_i = \frac{g_k\{h(i, i_2^{(k)}, \ldots, i_l^{(k)})\}}{\sum_{j=1}^{n} g_k\{h(j, i_2^{(k)}, \ldots, i_l^{(k)})\}}.$$

Then, $i_2^{(k+1)}$ is chosen from the set $\{1, \ldots, n\}$ using the PPS sampling with size of the element i being $g_k\{h(i_1^{(k+1)}, i, i_3^{(k)}, \ldots, i_l^{(k)})\}$ and so on. This leads to $\boldsymbol{Z}_{k+1} = (i_1^{(k+1)}, \ldots, i_l^{(k+1)})$.

Let

$$S_k = \bigcup_{r=1}^{l} \{(i_1^{(k+1)}, \ldots, i_{r-1}^{(k+1)}, i, i_{r+1}^{(k)}, \ldots, i_l^{(k)}) | 1 \leq i \leq n\}.$$

If $h(\boldsymbol{M}_k) \leq h(\boldsymbol{z})$ for all $\boldsymbol{z} \in S_k$, then $\boldsymbol{M}_{k+1} = \boldsymbol{M}_k$, and if $h(\boldsymbol{M}_k) > h(\boldsymbol{z})$ for some $\boldsymbol{z} \in S_k$, define

$$\boldsymbol{M}_{k+1} = \arg\min_{\boldsymbol{z} \in S_k} h(\boldsymbol{z}).$$

Note that the updating of $\boldsymbol{Z}_k$ to $\boldsymbol{Z}_{k+1}$ requires altogether the evaluation of h on nl l-tuples in order to compute the sizes required in the PPS sampling and the updating of $\boldsymbol{M}_k$ does not require any additional evaluation of h. This idea is motivated by well known Gibbs sampler (Geman and Geman 1984) used in annealing techniques.

2.2 Algorithm 2: Search By Random Sampling Followed By A PPS Type Acceptance-Rejection Scheme

A computationally faster alternative to Algorithm 1 will be to use simple random sampling for the updating of each $i_r^{(k)}$ into $i_r^{(k+1)}$ instead of PPS sampling and then accepting the randomly selected element $i \in \{1, \ldots, n\}$ with probability

$$\min\left[\frac{g_k\{h(i_1^{(k+1)}, \ldots, i_{r-1}^{(k+1)}, i, i_{r+1}^{(k)}, \ldots, i_l^{(k)})\}}{g_k\{h(i_1^{(k+1)}, \ldots, i_{r-1}^{(k+1)}, i_r^{(k)}, i_{r+1}^{(k)}, \ldots, i_l^{(k)})\}}, 1\right].$$

In the case of acceptance, we set $i_r^{(k+1)} = i$, and in the case of rejection, take $i_r^{(k+1)} = i_r^{(k)}$. Further, define $\boldsymbol{m}_0 = \boldsymbol{M}_k$ and

$$\boldsymbol{m}_r = \boldsymbol{m}_{r-1}, \quad \text{if } h(i_1^{(k+1)}, \ldots, i_{r-1}^{(k+1)}, i, i_{r+1}^{(k)}, \ldots, i_l^{(k)}) \geq h(\boldsymbol{m}_{r-1}),$$

and

$$\boldsymbol{m}_r = (i_1^{(k+1)}, \ldots, i_{r-1}^{(k+1)}, i, i_{r+1}^{(k)}, \ldots, i_l^{(k)}),$$

if $h(i_1^{(k+1)}, \ldots, i_{r-1}^{(k+1)}, i, i_{r+1}^{(k)}, \ldots, i_l^{(k)}) < h(\boldsymbol{m}_{r-1})$ and define $\boldsymbol{M}_{k+1} = \boldsymbol{m}_l$. Note that it requires only l evaluations of the function h to update $(\boldsymbol{Z}_k, \boldsymbol{M}_k)$ to $(\boldsymbol{Z}_{k+1}, \boldsymbol{M}_{k+1})$. This idea is motivated by Metropolis-Hastings algorithm (Metropolis et al., 1953, Hastings, 1970).

2.3 Convergence Results

Annealing and related ideas are already discussed to some extent in computation of robust estimate in literature (Todorov 1992, Woodruff and Rocke 1993) but no convergence result or detailed study along this direction is available. Here we provide some asymptotic convergence results.

To obtain the asymptotic convergence results of the proposed algorithms, we impose the following conditions on the sequence of functions $\{g_k\}$.

Condition 1: $g_k : [0, \infty) \to (0, 1)$ is a decreasing function for any $k \geq 1$.
Condition 2: $g_k\{h(\boldsymbol{z})\} \longrightarrow 0$ as $k \longrightarrow \infty$, if $\boldsymbol{z}$ is not a minimum of h.
Condition 3: Let

$$g_k^*(\boldsymbol{z}) = \frac{g_k\{h(\boldsymbol{z})\}}{\sum_{\boldsymbol{z}' \in \mathcal{S}_l} g_k\{h(\boldsymbol{z}')\}}.$$

Then

$$\sum_k \max_{\boldsymbol{z} \in \mathcal{S}_l} |g_k^*(\boldsymbol{z}) - g_{k+1}^*(\boldsymbol{z})| < \infty. \tag{2.1}$$

Let us introduce the following notation.

$$\delta_{k,r} = \max \left\{ \left| \log \frac{g_k\{h(\boldsymbol{z})\}}{g_k(h(\boldsymbol{z}'))} \right| \ : \ \boldsymbol{z} \text{ and } \boldsymbol{z}' \text{ differ only at their } r\text{-th coordinates.} \right\}$$

and

$$\Delta_k = \max\{\delta_{k,r} \mid r = 1, \ldots, l\}.$$

We now state the main theorem on convergence of our algorithms.

Theorem 2.1 *Suppose that* $\sum_k exp(-l\Delta_k) = \infty$*, and the above conditions on the sequence of functions* $\{g_k^*\}$ *are satisfied. Then, for any initial configuration* $\boldsymbol{Z}_0$*,* $\boldsymbol{M}_k$ *converges to* $\boldsymbol{M}$ *as* $k \to \infty$*, where* $\boldsymbol{M}$ *belongs to the set of global minima of* h*.*

2.4 Implementation and the Choice of g_k

The theorem in the previous section asserts the convergence of the proposed algorithms to the global optimum. But there are several practical issues related to the implementation of the algorithms. At first, we have to decide about the sequence of functions $\{g_k\}$. If we take,

$$g_k(x) = \exp(-T_k x)$$

for some increasing sequence of positive numbers $T_k \longrightarrow \infty$, it can be easily seen that Conditions 1–3 are trivially satisfied. Such a choice of the function g_k is motivated by familiar simulated annealing and Metropolis algorithms. We still need to choose the sequence T_k, and we discuss about different choices in the following.

Let us define a rough estimate of the diameter of the range of the objective function h as

$$\Delta = \max\{|h(\boldsymbol{z}) - h(\boldsymbol{z}')| \ : \ \boldsymbol{z} \text{ and } \boldsymbol{z}' \text{ differ only at one coordinate}\}.$$

Then, the maximum oscillation of the objective function $h(\boldsymbol{z})$ during the k-th iteration satisfies $\Delta_k \leq T_k \Delta$. If the condition

$$T_k \leq \frac{1}{l\Delta} \log(k+1) \tag{2.2}$$

is satisfied for all k greater than some k_0, we have

$$\sum_{k\geq 1} \exp(-l\Delta_k) \geq \sum_{k\geq k_0} \exp(-lT_k\Delta) \geq \sum_{k\geq k_0} \frac{1}{k+1} = \infty$$

and thus the condition for convergence in Theorem 2.1 holds.

Our next concern is the speed of convergence. The following result provides some rough estimates for the speed of convergence for such a logarithmic sequence T_k.

Theorem 2.2 *If $T_k = (l\Delta)^{-1}\log(k+1)$, there exists an integer K_0 such that for $k > K_0$,*

$$P(\boldsymbol{M}_k \notin H) = O\left(k^{-(\tilde{m}-m_0)/(\tilde{m}-m_0+l\Delta)}\right), \tag{2.3}$$

where H is the set of all global minima of h, m_0 and $\tilde{m}$ denote the minimum value of h and the value next to it, respectively.

The logarithmic increase of T_k, as suggested in the above theorem, may cause extremely slow convergence. Hence faster sequences can sometimes be adopted like $T_k = Aa^k$, with $a > 1$. In particular, if computation time is limited to a number N of iterations, we have the following result on the convergence after N iterations.

Theorem 2.3 *For some suitable choices of the constants A and c, define $T_k = A(c\log N)^{k/N}$, and if we terminate after N iterations for sufficiently large N,*

$$P(\boldsymbol{M}_N \notin H) = O\left(N^{-(\tilde{m}-m_0)/(l\Delta)}\right), \tag{2.4}$$

where H is the set of all global minima of h , m_0 and $\tilde{m}$ denote the minimum value of h and the value next to it, respectively.

The optimal choices of A and c depend on the particular optimization problem (see the proof of Theorem 2.3 in the Appendix). Theorems 2.2 and 2.3 provide us with some bounds on the probabilities of hitting a true optimum in the case of logarithmic sequences and finite exponential sequences when we stop after a finite number of steps. If the difference between $\tilde{m}$ and m_0 is large, we have a faster

convergence rate, and if these values are close to each other, the probability of hitting a global minimum in k iterations becomes smaller.

Now we combine these two results in the practical implementation of our algorithms. When we run the algorithms for N iterations, we choose $T_k = C\{\log(N/2)\}^{2k/N}$ for the first $N/2$ iterations, and the constant C is selected adaptively. At the k-th iteration, we take $C = C_k = 1/h(\boldsymbol{M}_k)$. Note that $h(\boldsymbol{M}_k)$ provides an estimate of $\tilde{m}$ at the k-th iteration. For the next $N/2$ iterations, we take $T_k = C\log(k+1)$, where C is again selected adaptively with $C = C_k = 1/(l\hat{\Delta}_k)$. Here $\hat{\Delta}_k$ is an estimate of Δ in (2.2). At every iteration, we update the estimate $\hat{\Delta}_k$. In the next sections, we illustrate the performance of our algorithms with such adaptive choices of the sequence T_k using some data analytic examples.

3 Comparison With Simple Random Sampling

We consider three procedures in robust statistics, namely, Tukey's half-space depth, transformation retransformation based multivariate medians, and least median of squares regression, as examples. Computational problems in all of these methods can be formulated as combinatorial optimization problems, and we can employ our algorithms to obtain solutions to them. For all three of them, algorithms used in the current literature are essentially based on simple random sampling. In the following subsections, we briefly discuss the methods and some of the popular algorithms used for them. Then we compare the performance of our algorithms with those commonly used algorithms in terms of the CPU time consumed.

Let us recall here that in Algorithm 1, the objective function value is evaluated at nl subsets for a complete single iteration, where l is the size of the subset and n is the number of observations. Thus, for N iterations, we evaluate Nnl subsets. Similarly, for Algorithm 2, we evaluate the objective function at Nl subsets for N complete iterations. In order to have a fair comparison of our algorithms with their competitors, for a specific problem with given l and n, we select the number of iterations N for Algorithms 1 and 2 so that the total number of subsets encountered in our algorithms become comparable with the number of subsets in the search based on simple random sampling.

3.1 Tukey's Half-space Depth

Let $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ be a sample in $\mathbb{R}^d$. Consider a subset of $d-1$ indices $\alpha = \{i_1, \ldots, i_{d-1}\}$. Then we denote by $H(\alpha)$, the unique hyperplane in $\mathbb{R}^d$ containing $\boldsymbol{x}$ and $\boldsymbol{X}_i$s with $i \in \alpha$. Let $h^*(i_1, \ldots, i_{d-1})$ denote the absolute difference between the number of data points that fall in one side of $H(\alpha)$ and the number of data points that fall in its other side. Chaudhuri and Sengupta (1993) showed that $HD(\boldsymbol{x})$, the half-space depth at $\boldsymbol{x}$, can be written as

$$HD(\boldsymbol{x}) = \frac{n+d-1}{2n} - \frac{1}{2n} \max h^*(i_1, \ldots, i_{d-1}) = \min h(i_1, \ldots, i_{d-1}), \tag{3.5}$$

where $h(i_1, \ldots, i_{d-1}) = \{n + d - 1 - h^*(i_1, \ldots, i_{d-1})\}/(2n)$.

For the bivariate case, some efficient exact algorithms to compute the depth contours and the deepest point are proposed by Rousseeuw and Ruts (1996, 1998) and Ruts and Rousseeuw (1996). For dimension $d > 2$, some approximation algorithms for computing the half-space depth of a point are available in Struyf and Rousseeuw (2000), Ghosh and Chaudhuri (2005a, b), etc.

As examples, we consider four data sets. The first data set has a moderately large sample size consisting of average ratings over the course of treatment for cancer patients undergoing radiotherapy (Johnson and Wichern, 2002). There are 93 correctly measured cases with 6 variables. We refer to this data as *Radiology Data*. As a second example, we use the *Blood Glucose Data*, which was used by Reaven and Miller (1979) to examine the relationship between chemical, subclinical and overt nonketotic diabetes in 145 non-obese adult subjects. The three primary variables used in the analysis are glucose intolerance, insulin response to oral glucose and insulin resistance. In addition, fasting plasma glucose was also measured for each individual in the study. We have taken only 76 overt nonketotic diabetic patients. The third data set is related to *Pima American Indians*, which consists of 8 determinants for 500 non-diabetic Pima females aged 21 or above (Blake and Merz, 1998). Lastly, we consider the *Ionosphere Data* (Blake and Merz, 1998), which contains 351 observations on 34 measurements about the free electrons in the ionosphere obtained from radars. For each of them, we compute the half-space depth of the coordinatewise median of the data.

Table 1: Mean CPU time (in seconds) taken by Algorithms 1 and 2 to achieve a strictly smaller objective function value in computing half-space depth of the coordinatewise median than a search based on simple random sampling. The second row indicates the number of iterations N for each data set and algorithm. Percentage of times they achieved a strictly smaller objective function value compared to simple random search in N iterations is given in the parentheses in the third row.

	Algorithm 1	Algorithm 2	Simple Random Sampling
Radiology Data	0.6227	0.1372	2.5082
	(100)	(10000)	(50000)
	(98)	(100)	
Blood Glucose Data	0.0547	0.1757	1.7965
	(150)	(10000)	(60000)
	(100)	(100)	
Pima Indians Data	5.5927	2.8093	68.6097
	(50)	(25000)	(200000)
	(100)	(100)	
Ionosphere Data	21.8290	15.3802	206.1837
	(50)	(20000)	(500000)
	(100)	(100)	

From Table 1, we observe that the proposed algorithms beat simple random search almost every time except a few cases for the Radiology Data. The performances of Algorithms 1 and 2 are comparable. However, Algorithm 1 seems to work better in the smaller data sets whereas Algorithm 2 works better in larger data sets.

3.2 Transformation Retransformation Medians

For a data set $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n \in \mathbb{R}^d$, consider the transformation matrix

$$\mathbf{X}(\alpha) = [\boldsymbol{X}_{i_1} - \boldsymbol{X}_{i_0} \vdots \cdots \vdots \boldsymbol{X}_{i_d} - \boldsymbol{X}_{i_0}],$$

where α denotes a subset of $d+1$ indices $\{i_0, \ldots, i_d\}$. Chakraborty and Chaudhuri (1998) and Chakraborty, Chaudhuri and Oja (1998) have shown that for both

coordinatewise and spatial medians, the optimal subset α is obtained by minimizing

$$v(\alpha) = \frac{\text{trace}\left[\{\mathbf{X}(\alpha)\}^T \hat{\Sigma}^{-1} \mathbf{X}(\alpha)\right] / d}{\left\{\det\left[\{\mathbf{X}(\alpha)\}^T \hat{\Sigma}^{-1} \mathbf{X}(\alpha)\right]\right\}^{1/d}}, \tag{3.6}$$

over α, where $\hat{\Sigma}$ is some consistent affine equivariant estimate of the scatter matrix Σ. Hence, finding out the optimal transformation is again a combinatorial optimization problem with $h(i_0, \ldots, i_d) = v(\alpha)$, where $\alpha = \{i_0, \ldots, i_d\}$, and either purely random search or complete enumeration have been considered so far in the literature to solve this problem.

Table 2: Mean CPU time (in seconds) taken by Algorithms 1 and 2 to achieve a strictly smaller objective function value in computing the optimal transformation matrix using TR methodology than a search based on simple random sampling. The second row indicates the number of iterations N used for each data set and algorithm. Percentage of times they achieved a strictly smaller objective function value compared to simple random search in N iterations is given in the parentheses in the third row.

	Algorithm 1	Algorithm 2	Simple Random Sampling
Radiology Data	0.0033	0.0072	0.3318
	(100)	(10000)	(60000)
	(100)	(100)	
Blood Glucose Data	0.0200	0.0153	0.2053
	(150)	(10000)	(60000)
	(100)	(99)	
Pima Indians Data	0.1703	0.0137	2.2515
	(50)	(10000)	(100000)
	(100)	(100)	
Ionosphere Data	1.3827	4.6278	15.3180
	(50)	(15000)	(100000)
	(100)	(100)	

We again consider the four real data sets discussed before in Section 3.1 to find an optimal data-driven transformation matrix, which minimizes the objective function

(3.6), for each data set. In Table 2, we compare the CPU time taken by Algorithms 1 and 2 to achieve a strictly smaller objective function value than the best value obtained in a simple random search. We again observe that the proposed algorithms perform very well compared to the simple random search. However, Algorithm 1 works better for some data sets and Algorithm 2 works better in others.

3.3 Least Median of Squares Regression

Consider a linear regression model

$$y_i = \alpha + \boldsymbol{x}_i^T \boldsymbol{\beta} + \epsilon_i, \quad i = 1, \ldots, n, \tag{3.7}$$

where $\boldsymbol{x}_i$ is a p-dimensional vector of explanatory variables, y_i is the response and ϵ_i is the error term. The least median of squares (LMS) regression estimate of the parameters $(\alpha, \boldsymbol{\beta})$ is defined as

$$(\hat{\alpha}_{LMS}, \hat{\boldsymbol{\beta}}_{LMS}) = \arg\min_{\alpha, \boldsymbol{\beta}} |y_i - \alpha - \boldsymbol{x}_i^T \boldsymbol{\beta}|_{r:n},$$

where $|y_i - \alpha - \boldsymbol{x}_i^T \boldsymbol{\beta}|_{r:n}$ is the r-th order statistic of $|y_1 - \alpha - \boldsymbol{x}_1^T \boldsymbol{\beta}|, \ldots, |y_n - \alpha - \boldsymbol{x}_n^T \boldsymbol{\beta}|$ and $r = [(n + p + 1)/2]$ (see Rousseeuw 1984). Some approximation algorithms for solving this minimization problem are available in Boček and Lachout (1995), Tichavský (1991), Olson (1997). However, the most popular algorithm for computing the LMS regression estimates is the PROGRESS algorithm suggested by Rousseeuw and Leroy (1987) and later modified by Rousseeuw and Hubert (1997). It computes an approximation to the above LMS estimate of regression coefficients, which can be outlined as follows. Consider a hyperplane passing through $p+1$ data points $(y_{i_1}, \boldsymbol{x}_{i_1}), \ldots, (y_{i_{p+1}}, \boldsymbol{x}_{i_{p+1}})$ and adjust the intercept term α to get the best fitting hyperplane parallel to the above hyperplane through the data points. Take it as a candidate fit and define $h_1(i_1, \ldots, i_{p+1})$ as the $[(n+p+1)/2]$-th order statistic of the absolute residuals obtained from this candidate fit. Then PROGRESS minimizes h_1 over repeated simple random samples of such candidate fits. For $p = 1$ with complete enumeration instead of random sampling, it would lead to an exact algorithm to find LMS estimates. An alternative procedure in the case $p = 1$ is available in Edelsbrunner and Souvaine (1990).

For $p \geq 2$, the minimization of h_1 does not provide the exact LMS estimates even with a complete enumeration. In this case, consider two parallel hyperplanes $y = \tilde{\alpha}_1 + \boldsymbol{x}^T\tilde{\boldsymbol{\beta}}$ and $y = \tilde{\alpha}_2 + \boldsymbol{x}^T\tilde{\boldsymbol{\beta}}$ containing $p+2$ data points $(y_{i_1}, \boldsymbol{x}_{i_1}), \ldots, (y_{i_{p+2}}, \boldsymbol{x}_{i_{p+2}})$ among themselves. Let $h_2(i_1, \ldots, i_{p+2})$ be the $[(n+p+1)/2]$-th ordered absolute residual obtained from the fit

$$y = \frac{\tilde{\alpha}_1 + \tilde{\alpha}_2}{2} + \boldsymbol{x}^T\tilde{\boldsymbol{\beta}}.$$

Then LMS estimates are obtained by minimizing

$$h_2(i_1, \ldots, i_{p+2}) \tag{3.8}$$

over all possible subsets of $p+2$ observations. If this minimization problem could be solved by complete enumeration, one would get an exact LMS estimate. We will apply our Algorithms 1 and 2 to minimize h_2.

Table 3: Mean CPU time (in seconds) taken by Algorithms 1 and 2 to achieve a strictly smaller objective function value in computing LMS regression estimates than PROGRESS algorithm for Horse Mussels data. The second row indicates the number of iterations N used for each algorithm. Percentage of times they achieved a strictly smaller objective function value compared to PROGRESS in N iterations is given in the third row.

	Algorithm 1	Algorithm 2	PROGRESS
Mussels Data	3.7058	0.7292	7.7548
	(50)	(10000)	(50000)
	(100)	(100)	

As an example, we consider a sample of 201 horse mussels, collected in the Marlborough Sounds at the Northeast of New Zealand's South Island (Camden 1989). The response variable is mussel mass M in grams, and all quantitative predictors relate to characteristics of mussel shells: shell width W, shell height H, shell length L, each in nanometres, and shell mass S in grams. We compare the CPU time of our algorithms with PROGRESS in computing the LMS regression line of the data. In Table 3, we compare the CPU time taken by Algorithms 1 and

2 to achieve a strictly smaller objective function value in computing LMS regression estimates than PROGRESS algorithm with 50,000 subsets chosen by simple random sampling. We also report the percentage of times our algorithms attained a strictly smaller objective function value compared to PROGRESS with the above settings. It is evident that our general purpose algorithms outperform a problem specific algorithm like PROGRESS for this data set.

4 Comparison With Deterministically Guided Search

To obtain the minimum covariance determinant (MCD) estimator (Rousseeuw and Leroy, 1987) of multivariate location and scale matrix for a d-dimensional data set $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$, one needs to find a subset of l observations out of n observations whose classical covariance matrix has the lowest determinant. The MCD estimates of location and scale matrix are then the mean vector and the covariance matrix (scaled properly), respectively, of these l observations. If $l = [(n + d + 1)/2]$, MCD estimates attain the maximum possible breakdown point. So, the problem of finding the optimal subset of $[(n + d + 1)/2]$ observations can be written as the following combinatorial optimization problem:

$$\text{Minimize } h(i_1, \ldots, i_l) = \det[\text{cov}(\boldsymbol{X}_{i_1}, \ldots, \boldsymbol{X}_{i_l})]. \tag{4.9}$$

Here l, the size of the subset, grows with n, the number of observations, and an exact procedure to solve this in practice even for moderate sample size n is out of question. Rousseeuw and van Driessen (1999) proposed an algorithm called FAST-MCD to approximate the MCD estimates in higher dimensions. To describe it briefly, it starts with a subset of $d + 1$ data points, and then proceeds in a deterministic way until the objective function value stabilizes in the subsequent steps. Then it repeats the same procedure for many randomly chosen initial subsets of observations and takes the minimum of these iterations as an approximate solution to the optimization problem. While this algorithm runs quite fast, there is no guarantee of convergence to the global optimum.

To compute MCD estimates, Todorov (1992) used a simulated annealing based approach and Woodruff and Rocke (1993) suggested some heuristic search algo-

Table 4: Mean CPU time (in seconds) taken by Algorithms 1 and 2 to achieve a strictly smaller objective function value in computing MCD estimates than FAST-MCD algorithm. The second row indicates the number of iterations N used for each data set and algorithm. Percentage of times they achieved a strictly smaller objective function value compared to FAST-MCD in N iterations is given in the parentheses in the third row.

	Algorithm 1	Algorithm 2	FAST-MCD
Radiology Data	2.8157	0.1037	(4.0413)
	(100)	(10000)	(10000)
	(98)	(100)	
Blood Glucose Data	0.1770	0.1248	2.1948
	(100)	(10000)	(10000)
	(99)	(100)	
Pima Indians Data	43.6788	6.6480	71.7237
	(50)	(25000)	(10000)
	(93)	(98)	
Ionosphere Data	192.5668	180.8332	305.8227
	(50)	(20000)	(10000)
	(98)	(99)	

rithms for computing minimum volume ellipsoid estimators including simulated annealing and genetic algorithms. However, these works did not prove any results on convergence of their algorithms and they did not report any detailed study for the general combinatorial optimization problem considered here.

To illustrate the performance of our Algorithms 1 and 2 over FAST-MCD in minimizing the objective function for MCD estimates, we use the same four data sets as in Section 3.1. In Table 4, we compare the CPU time taken by Algorithms 1 and 2 to achieve a strictly smaller determinant of the covariance matrix of a subset of $[(n + d + 1)/2]$ observations (n is the sample size) than FAST-MCD algorithm with 10,000 randomly chosen initial subsets in the C-step of it (see Rousseeuw and van Driessen, 1999).

Even though the total number of possible subsets, which grows with the sample

size, in this particular optimization problem may be very large making it a very challenging problem, we observe that our algorithms performed quite well in all the data sets under consideration. We also observe that Algorithm 2 performed better than Algorithm 1. A reason behind that may be in Algorithm 2, we can have many more complete iterations in a comparable time.

Acknowledgements

The research of Biman Chakraborty was supported in part by National University of Singapore Academic Research Grant R-155-000-041-112. Parts of the work were done when Biman Chakraborty visited Indian Statistical Institute, Kolkata, India during December 2004 and Probal Chaudhuri visited Institute for Mathematical Sciences at National University of Singapore during March 2005. The hospitality and the academic support provided by the two institutes are gratefully acknowledged.

A APPENDIX : PROOFS

It is easy to see that the sequence $\{\boldsymbol{Z}_k\}$ in both Algorithms 1 and 2 form finite state-space non-homogeneous Markov chains and to show the convergence of the sequence $\boldsymbol{M}_k$ to the set of global minima, we need to prove that the sequence $\{\boldsymbol{Z}_k\}$ is *strongly ergodic*. Note that an irreducible, recurrent, homogeneous Markov chain is always ergodic and converges to its stationary distribution. However, it is not so for non-homogeneous Markov chains. The existence of a sequence of stationary distributions, which converges to the uniform distribution on the set of global minima and *weak ergodicity* provides a sufficient condition for strong ergodicity of a non-homogeneous Markov chain. In the following Lemmas, we prove the above conditions and weak ergodicity of the Markov chains associated with Algorithms 1 and 2, using Dobrushin's contraction coefficients (Dobrushin 1956a,b), to establish the strong ergodicity.

We begin by introducing some notation. Let $\boldsymbol{x} = (x_1, \ldots, x_l)$ and $\boldsymbol{y} = (y_1, \ldots, y_l)$

be two points in $\mathcal{S}_l$. Then, for every $1 \le r \le l$, define two Markov kernels on $\mathcal{S}_l$ by

$$\mathbf{P}_r^k(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} \dfrac{g_k\{h(x_1, \ldots, x_{r-1}, y_r, x_{r+1}, \ldots, x_l)\}}{\sum_{i=1}^n g_k\{h(x_1, \ldots, x_{r-1}, i, x_{r+1}, \ldots, x_l)\}}, & \text{if } x_j = y_j \text{ for every } j \neq r \\ 0, & \text{otherwise} \end{cases}$$

and

$$\mathbf{Q}_r^k(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} \dfrac{1}{n} \min\left\{\dfrac{g_k\{h(\boldsymbol{y})\}}{g_k\{h(\boldsymbol{x})\}}, 1\right\}, & \text{if } x_j = y_j \text{ for every } j \neq r \text{ and } x_r \neq y_r \\ 1 - \dfrac{1}{n} \displaystyle\sum_{i=1, i \neq x_r}^n \min\left\{\dfrac{g_k\{h(x_1, \ldots, x_{r-1}, i, x_{r+1}, \ldots, x_l)\}}{g_k\{h(x_1, \ldots, x_{r-1}, x_r, x_{r+1}, \ldots, x_l)\}}, 1\right\}, & \text{if } \boldsymbol{x} = \boldsymbol{y} \\ 0, & \text{otherwise.} \end{cases}$$

Lemma A.1 (i) *For Algorithm 1, $\{\boldsymbol{Z}_k\}_{k>0}$ is a non-homogeneous Markov chain with transition matrix*

$$\mathbf{P}_k = \prod_{r=1}^l \mathbf{P}_r^k$$

and stationary distribution g_k^*.

(ii) *For Algorithm 2, $\{\boldsymbol{Z}_k\}_{k>0}$ is a non-homogeneous Markov chain with transition matrix*

$$\mathbf{Q}_k = \prod_{r=1}^l \mathbf{Q}_r^k$$

and stationary distribution g_k^*.

Proof: (i) It is easy to observe that

$$g_k^*(\boldsymbol{x})\mathbf{P}_r^k(\boldsymbol{x}, \boldsymbol{y}) = g_k^*(\boldsymbol{y})\mathbf{P}_r^k(\boldsymbol{y}, \boldsymbol{x})$$

for every $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{S}_l$ and $r = 1, \ldots, l$. Summation of both sides over $\boldsymbol{x}$ proves that g_k^* is stationary for $\mathbf{P}_r^k$. Since the transition matrix of $\boldsymbol{Z}_k$ is given by

$$\mathbf{P}_k = \prod_{r=1}^l \mathbf{P}_r^k,$$

g_k^* is also stationary for $\mathbf{P}_k$.
(ii) In order to show that "the detailed balance condition" (Winkler 2003) holds, i.e.,

$$g_k^*(\boldsymbol{x})\mathbf{Q}_r^k(\boldsymbol{x},\boldsymbol{y}) = g_k^*(\boldsymbol{y})\mathbf{Q}_r^k(\boldsymbol{y},\boldsymbol{x}),$$

it is sufficient to consider $\boldsymbol{x},\boldsymbol{y} \in \mathcal{S}_l$ such that $x_j = y_j$ for $j \neq r$ and $x_r \neq y_r$. Then, for $h(\boldsymbol{y}) > h(\boldsymbol{x})$,

$$\begin{aligned} g_k^*(\boldsymbol{x})\mathbf{Q}_r^k(\boldsymbol{x},\boldsymbol{y}) &= \frac{g_k\{h(\boldsymbol{x})\}}{\sum_{\boldsymbol{z}'\in\mathcal{S}_l} g_k\{h(\boldsymbol{z}')\}} \times \frac{g_k\{h(\boldsymbol{y})\}}{g_k\{h(\boldsymbol{x})\}} \\ &= \frac{g_k\{h(\boldsymbol{y})\}}{\sum_{\boldsymbol{z}'\in\mathcal{S}_l} g_k\{h(\boldsymbol{z}')\}} \end{aligned}$$

and

$$g_k^*(\boldsymbol{y})\mathbf{Q}_r^k(\boldsymbol{y},\boldsymbol{x}) = \frac{g_k\{h(\boldsymbol{y})\}}{\sum_{\boldsymbol{z}'\in\mathcal{S}_l} g_k\{h(\boldsymbol{z}')\}} \times 1.$$

Similarly, it holds for $h(\boldsymbol{y}) \leq h(\boldsymbol{x})$. Thus

$$g_k^*(\boldsymbol{x})\mathbf{Q}_r^k(\boldsymbol{x},\boldsymbol{y}) = g_k^*(\boldsymbol{y})\mathbf{Q}_r^k(\boldsymbol{y},\boldsymbol{x})$$

for every $\boldsymbol{x},\boldsymbol{y} \in \mathcal{S}_l$ and $r = 1,\ldots,l$. Again, summing both sides over $\boldsymbol{x}$ proves that g_k^* is stationary for $\mathbf{Q}_r^k$. Since the transition matrix of $\boldsymbol{Z}_k$ is given by

$$\mathbf{Q}_k = \prod_{r=1}^{l} \mathbf{Q}_r^k,$$

g_k^* is also stationary for $\mathbf{Q}_k$. □

Lemma A.2 *Let m denote the minimal value of the function h, and $H = \{\boldsymbol{z} \in \mathcal{S}_l : h(\boldsymbol{z}) = m\}$, the set of global minima of h. Then,*

$$\lim_{k\to\infty} g_k^*(\boldsymbol{z}) = \begin{cases} \frac{1}{|H|}, & \text{if } \boldsymbol{z} \in H \\ 0, & \text{otherwise.} \end{cases}$$

Proof: Write $g_k^*(\boldsymbol{z})$ as

$$\begin{aligned} g_k^*(\boldsymbol{z}) &= \frac{g_k\{h(\boldsymbol{z})\}/g_k(m)}{\sum_{\boldsymbol{z}'\in\mathcal{S}_l} g_k\{h(\boldsymbol{z}')\}/g_k(m)} \\ &= \frac{g_k\{h(\boldsymbol{z})\}/g_k(m)}{|H| + \sum_{\boldsymbol{z}':h(\boldsymbol{z}')>m} g_k\{h(\boldsymbol{z}')\}/g_k(m)}. \end{aligned}$$

By Condition 2, $\sum_{\boldsymbol{z}':h(\boldsymbol{z}')>m} g_k\{h(\boldsymbol{z}')\}/g_k(m) \longrightarrow 0$ as $k \longrightarrow \infty$ and we have the desired result. □

Proof of Theorem 2.1: Let us first prove the theorem for Algorithm 1. Consider the contraction coefficient or the ergodic coefficient of the transition matrix $\mathbf{P}_k$, $c(\mathbf{P}_k) = (1/2)\max_{\boldsymbol{x},\boldsymbol{y}} \|\mathbf{P}_k(\boldsymbol{x},\cdot) - \mathbf{P}_k(\boldsymbol{y},\cdot)\|$. By Lemma 4.2.3 of Winkler (2003),

$$c(\mathbf{P}_k) = 1 - \inf_{\boldsymbol{x},\boldsymbol{y}\in\mathcal{S}_l} \sum_{\boldsymbol{z}\in\mathcal{S}_l} \mathbf{P}_k(\boldsymbol{x},\boldsymbol{z}) \wedge \mathbf{P}_k(\boldsymbol{y},\boldsymbol{z}) \leq 1 - |\mathcal{S}_l| \left\{ \inf_{\boldsymbol{x},\boldsymbol{y}\in\mathcal{S}_l} \mathbf{P}_k(\boldsymbol{x},\boldsymbol{y}) \right\}.$$

Define $m_r(\boldsymbol{x}) = \inf\{h(\boldsymbol{y}) : \ x_i = y_i, \text{ for every } i \neq r\}$ for $r = 1, \ldots, l$. Then,

$$\mathbf{P}_r^k(\boldsymbol{x},\boldsymbol{y}) = \frac{g_k\{h(\boldsymbol{x})\}/g_k\{m_r(\boldsymbol{x})\}}{\sum_{i=1}^n g_k\{h(x_1,\ldots,x_{r-1},i,x_{r+1},\ldots,x_l)\}/g_k\{m_r(\boldsymbol{x})\}} \geq \frac{\exp(-\delta_{k,r})}{n}.$$

Therefore,

$$\min_{\boldsymbol{x},\boldsymbol{y}\in\mathcal{S}_l} \mathbf{P}_k(\boldsymbol{x},\boldsymbol{y}) \geq \prod_{r=1}^{l} \frac{e^{-\delta_{k,r}}}{n} \geq \frac{e^{-l\Delta_k}}{n^l},$$

and $c(\mathbf{P}_k) \leq 1 - \exp(-l\Delta_k)$.

Now the assumption $\sum_{k>0} \exp(-l\Delta_k) = \infty$ implies that

$$\prod_{k>0} c(\mathbf{P}_k) \leq \prod_{k>0} (1 - e^{-l\Delta_k}) = 0. \tag{A.10}$$

Further, by Condition 3, $\sum_{k>0} \|g_{k+1}^* - g_k^*\| < \infty$. Then, by Theorem 4.5.1 of Winkler (2003), as $k \longrightarrow \infty$, $\nu\mathbf{P}_1\mathbf{P}_2\ldots\mathbf{P}_k$ converges to the limiting distribution that is uniform on H, and the convergence is uniform in all initial distributions ν.

In Algorithm 2, for any $\boldsymbol{x},\boldsymbol{y} \in \mathcal{S}_l$, such that $x_i = y_i$ for every $i \neq r$ and $x_r \neq y_r$, we have

$$\mathbf{Q}_r^k(\boldsymbol{x},\boldsymbol{y}) \geq \frac{1}{n}\frac{g_k\{h(\boldsymbol{y})\}}{g_k\{h(\boldsymbol{x})\}} \geq \frac{\exp(-\delta_{k,r})}{n}$$

and

$$\mathbf{Q}_r^k(\boldsymbol{x},\boldsymbol{x}) \geq \frac{1}{n} \geq \frac{\exp(-\delta_{k,r})}{n}.$$

Using similar arguments as in the case of Algorithm 1, with $\mathbf{P}_r^k$ and $\mathbf{P}_k$ replaced by $\mathbf{Q}_r^k$ and $\mathbf{Q}_k$, respectively, we have, as $k \to \infty$, $\nu\mathbf{Q}_1\mathbf{Q}_2\cdots\mathbf{Q}_k$ converges to the limiting distribution that is uniform on H, and the convergence is uniform in all initial distributions ν.

The proof is now complete by noting that for both Algorithms 1 and 2, $h(\boldsymbol{M}_k) \leq h(\boldsymbol{Z}_k)$ for any $k \geq 1$. □

Proof of Theorem 2.2: Let

$$g_\infty(\boldsymbol{z}) = \begin{cases} \frac{1}{|H|}, & \text{if } \boldsymbol{z} \in H \\ 0, & \text{otherwise} \end{cases}$$

be the uniform distribution on H, the set of global minima of h. Then, for the non-homogeneous Markov chain $\boldsymbol{Z}_k$ with kernel $\mathbf{P}_k$ and initial distribution ν, we have

$$\|\nu \mathbf{P}_1 \cdots \mathbf{P}_k - g_\infty\| \leq 2 \prod_{j=i}^{k} c(\mathbf{P}_k) + 2 \max_{j \geq i} \|g_i^* - g_\infty\| + \sum_{j=i}^{k} \|g_{j+1}^* - g_j^*\| \qquad \text{(A.11)}$$

for every $i \geq 1$. We now estimate every term on the right hand side of the above inequality.

We have noted before in the proof of Theorem 2.1 that the contraction coefficient $c(\mathbf{P}_j) \leq 1 - \exp(-T_j l\Delta) = 1 - (j+1)^{-1}$, and hence

$$\prod_{j=i}^{k} c(\mathbf{P}_j) \leq \prod_{j=i}^{k} \frac{j}{j+1} = \frac{i}{k+1}.$$

Without loss of generality, we can assume that the minimal value of m_0 of h is 0. Since the convergence is eventually monotone, the maximum in the second term of (A.11) eventually becomes $\|g_i^* - g_\infty\|$. If $\boldsymbol{z} \notin H$, then

$$|g_i^*(\boldsymbol{z}) - g_\infty(\boldsymbol{z})| = \frac{\exp\{-T_i h(\boldsymbol{z})\}}{|H| + \sum_{\boldsymbol{y} \notin H} \exp\{-T_i h(\boldsymbol{y})\}} = \frac{(i+1)^{-h(\boldsymbol{z})/(l\Delta)}}{|H| + \sum_{\boldsymbol{y} \notin H} \exp\{-T_i h(\boldsymbol{y})\}}$$

$$\leq \frac{1}{|H|}(i+1)^{-\tilde{m}/(l\Delta)}.$$

For $\boldsymbol{z} \in H$,

$$\begin{aligned} |g_i^*(\boldsymbol{z}) - g_\infty(\boldsymbol{z})| &= \left| \frac{1}{|H| + \sum_{\boldsymbol{y} \notin H} \exp\{-T_i h(\boldsymbol{y})\}} - \frac{1}{|H|} \right| \\ &\leq \frac{\sum_{\boldsymbol{y} \notin H} \exp\{-T_i h(\boldsymbol{y})\}}{|H|^2} \leq \frac{n^l - |H|}{|H|^2}(i+1)^{-\tilde{m}/(l\Delta)}. \end{aligned}$$

Thus we can write

$$\|g_i^* - g_\infty\| = O\left((i+1)^{-\tilde{m}/(l\Delta)}\right).$$

Finally, for large i, the sum

$$\sum_{j=i}^{k} |g_{j+1}^*(\boldsymbol{z}) - g_i^*(\boldsymbol{z})|$$

vanishes for $\boldsymbol{z} \notin H$, and if $\boldsymbol{z} \in H$, it is dominated by

$$|g_{k+1}^*(\boldsymbol{z}) - g_i^*(\boldsymbol{z})| \leq \|g_{k+1}^* - g_\infty\| + \|g_i^* - g_\infty\| \leq 2\|g_i^* - g_\infty\| = O\left((i+1)^{-\tilde{m}/(l\Delta)}\right).$$

Hence, a bound for the right hand side of (A.11) is given by

$$\frac{i}{k} + b_0(i+1)^{-\tilde{m}/(l\Delta)},$$

where b_0 is some constant. This becomes optimal for

$$i+1 = \left(\frac{\tilde{m}}{l\Delta} b_0\right)^{l\Delta/(\tilde{m}+l\Delta)} k^{l\Delta/(\tilde{m}+l\Delta)},$$

and we can conclude that

$$\|\nu \mathbf{P}_1 \cdots \mathbf{P}_k - g_\infty\| = O\left(k^{-\tilde{m}/(\tilde{m}+l\Delta)}\right).$$

Finally,

$$P(\boldsymbol{M}_k \notin H) \leq P(\boldsymbol{Z}_k \notin H) = O\left(k^{-\tilde{m}/(\tilde{m}+l\Delta)}\right).$$

We can have a similar result for the sequence of Markov kernels $\mathbf{Q}_k$ also as it has the same stationary distribution g_k^*. □

Proof of Theorem 2.3: Let us assume, without loss of generality, that the minimum value of the objective function h is 0, and h takes a finite set of values $0 = m_0 < m_1 < \cdots < m_r$. We then define an increasing sequence of subsets $\mathcal{F}_j = \{\boldsymbol{z} : \ h(\boldsymbol{z}) \leq m_j\} \subseteq \mathcal{S}_l$ for $j = 0, \ldots, r$. Note that, $\mathcal{F}_0$ is the set of all global minima and $\mathcal{F}_r = \mathcal{S}_l$.

Let

$$H_j = \min_{\boldsymbol{y} \in \mathcal{S}_l} \max_{\boldsymbol{x} \in \mathcal{F}_j - \mathcal{F}_{j-1}} \{h(\boldsymbol{y}) - h(\boldsymbol{x})\}^+.$$

Then H_j is essentially the height (or depth) at which a point in $\mathcal{F}_j - \mathcal{F}_{j-1}$ can be reached from a point outside. It can be shown using Theorem 4.4 of Catoni (1992) that for $\boldsymbol{z} \in \mathcal{F}_j - \mathcal{F}_{j-1}$,

$$\begin{aligned} \sup_{\boldsymbol{x} \in \mathcal{S}_l} P(\boldsymbol{Z}_k = \boldsymbol{z} | \boldsymbol{Z}_0 = \boldsymbol{x}) \;\; &\leq \;\; a_1 e^{-h(\boldsymbol{z})T_k} \\ &\quad \times \sum_{m=-\infty}^{k} e^{h(\boldsymbol{z})(T_k - T_{m+1})} a_2 e^{-H_j T_{m+1}} \prod_{i=m+2}^{k} (1 - a_2 e^{H_j T_i}), \end{aligned} \tag{A.12}$$

where $T_m = T_0$ for $m \leq 0$. The last sum stays bounded if we have

$$h(\boldsymbol{z})(T_{i+1} - T_i) \leq B e^{-H_j T_i} \quad \text{for } i > 0. \tag{A.13}$$

Proposition 6.1 of Catoni (1992) presents a more general version of the above result.

Let us take $b \geq l\Delta$, $d \leq (l\Delta)/\tilde{m}$ and $0 < \delta < \tilde{m}$, and define

$$T_k = \frac{d}{b} \left(\frac{b}{d^2 \delta} \log N \right)^{k/N}.$$

Then, for $\boldsymbol{z} \in \mathcal{F}_j - \mathcal{F}_{j-1}$, (A.13) holds. Therefore, from (A.12), we have, for $\boldsymbol{z} \in \mathcal{F}_j - \mathcal{F}_{j-1}$,

$$\sup_{\boldsymbol{x} \in \mathcal{S}_l} P(\boldsymbol{Z}_k = \boldsymbol{z} | \boldsymbol{Z}_0 = \boldsymbol{x}) \leq K e^{-h(\boldsymbol{z})T_k}. \tag{A.14}$$

Consequently, we can write

$$\max_{\boldsymbol{z} \in \mathcal{S}_l} P(h(\boldsymbol{Z}_N) \geq \delta | \boldsymbol{Z}_0 = \boldsymbol{z}) \leq b_1 \exp(-\delta T_N) \leq b_1 \exp\left(-\frac{\tilde{m}}{l\Delta} \log N \right) = b_1 N^{-\tilde{m}/(l\Delta)}$$

for some constant b_1. Therefore, for some $0 < \delta < \tilde{m}$,

$$P(\boldsymbol{M}_N \notin H) \leq P(\boldsymbol{Z}_N \notin H) = P\Big(h(Z_N) \geq \delta \Big) = O\left(N^{-\tilde{m}/(l\Delta)} \right).$$

□

REFERENCES

Besag, J. (1986) On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society, Series B*, **48**, 259–302.

Blake, C.L., and Merz, C.J. (1998) UCI Repository of machine learning databases [http://www.ics.uci.edu/∼mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science.

Boček. P., and Lachout, P. (1995) Linear programming approach to LMS–estimation. *Computational Statistics & Data Analysis*, **19** 129–134.

Catoni, O. (1992) Rough large deviations estimates for simulated annealing: Application to exponential schedules. *Annals of Probability*, **20**, 109–146.

Chakraborty, B., and Chaudhuri, P. (1998) On an adaptive transformation retransformation estimate of multivariate location. *Journal of the Royal Statistical Society, Series B*, **60**, 145–157.

Chakraborty, B., Chaudhuri, P., and Oja, H. (1998) Operating transformation retransformation on spatial median and angle test. *Statistica Sinica*, **8**, 767–784.

Chaudhuri, P., and Sengupta, D. (1993) Sign tests in multidimension : Inference based on the geometry of the data cloud. *Journal of the American Statistical Association*, **88**, 1363–1370.

Dobrushin, R.L. (1956a) Central limit theorem for non-stationary Markov chains I. *Theory of Probability and Its Applications*, **1**, 65–80.

Dobrushin, R.L. (1956b) Central limit theorem for non-stationary Markov chains II. *Theory of Probability and Its Applications*, **1**, 329–383.

Geman, S., and Geman, D. (1984) Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **6**, 721–741.

Ghosh, A. K., and Chaudhuri, P. (2005a) On data depth and distribution free discriminant analysis using separating surfaces. *Bernoulli*, **11**, 1–27.

Ghosh, A. K., and Chaudhuri, P. (2005b) On maximum depth and related classifiers. To appear in *Scandinavian Journal of Statistics*.

Hastings, W.K. (1970) Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, **57**, 97–109.

Hettmansperger, T.P., and Randles, R.H. (2002) A practical affine equivariant multivariate median. *Biometrika*, **89**, 851–860.

Johnson, R.A., and Wichern, D.W. (2002) *Applied Multivariate Statistical Analysis*, Fifth Edition, New Jersey: Prentice Hall.

Kirkpatrick, S., Gelatt, C. D., Jr., and Vechhi, M. P. (1983) Optimization by simulated annealing. *Science*, **220**, 671–680.

Kittler, J., and Föglein, J. (1984) Contextual classification of multispectral pixel data. *Image and Vision Computing*, **2**, 13–29.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., and Teller, E. (1953) Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, **21**, 1087–1092.

Olson, C.F. (1997) An approximation algorithm for least median of squares regression. *Information Processing Letters*, **63**, 237–241.

Randles, R.H. (2000) A simpler, affine invariant, multivariate, distribution-free sign test. *Journal of the American Statistical Association*, **84**, 1045–1050.

Reaven, G. M., and Miller, R. G. (1979) An attempt to define the nature of chemical diabetes using a multidimensional analysis. *Diabetologia*, **16**, 17–24.

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

Rousseeuw, P.J., and Hubert, M. (1997) Recent developments in PROGRESS. In L_1*-Statistical Procedures and Related topic*, ed. Y. Dodge, Hayward, California: Institute of Mathematical Statistics, pp. 201–214.

Rousseeuw, P.J., and Leroy, A.M. (1987) *Robust Regression and Outlier Detection*. New York: Wiley.

Rousseeuw, P.J., and Ruts, I. (1996) As 307: Bivariate location depth. *Applied Statistics*, **45**, 516–526.

Rousseeuw, P.J., and Ruts, I. (1998). Constructing the bivariate Tukey median. *Statistica Sinica*, **8**, 827–839.

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

Ruts, I., and Rousseeuw, P.J. (1996) Computing depth contours of bivariate point clouds. *Computational Statistics and Data Analysis*, **23**, 153–168.

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

Tichavský, P. (1991) Algorithm for and geometrical character of solutions in the LMS and the LTS linear regression. *Computational Statistics Quarterly C*, **6**, 139151.

Todorov, V. (1992) Computing the minimum covariance determinant estimator (MCD) by simulated annealing. *Computational Statistics & Data Analysis*, **14**, 515–525.

Tukey, J.W. (1975) Mathematics and the picturing of data. In *Proceedings of the International Congress of Mathematicians, vol. 2*, ed. R.D. James, Vancouver: Canadian Mathematical Congress, pp. 523–531.

Winkler, G. (2003) *Image Analysis, Random Fields and Dynamic Monte Carlo Methods. A Mathematical Introduction*. 2nd edition, New York: Springer.

Woodruff, D.L., and Rocke, D.M. (1993) Heuristic search algorithms for the minimum volume ellipsoid. *Journal of Computational and Graphical Statistics*, **2**, 69–95.