THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par:

Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)

Discipline ou spécialité:

Mathématiques appliquées

Présentée et soutenue par:

Salvador FLORES

le: vendredi 25 février 2011

Titre:

Problèmes d'optimisation globale en statistique robuste

Ecole doctorale:

Mathématiques Informatique Télécommunications (MITT)

Unité de recherche:

Institut de Mathématiques de Toulouse, UMR CNRS 5219

Directeur(s) de Thèse:

Anne Ruiz-Gazen, Université Toulouse 1 Capitole

Marcel Mongeau, Université Paul Sabatier

Rapporteurs

M. Fabio Schoen, Università di Firenze, Italie

M. Michael Schyns, Université de Liège, Belgique

M. Roy Welsch, Massachusetts Institute of Technology, ÉUA

Autre(s) membre(s) du jury

M. Jean-Baptiste Hiriart-Urruty, Université Paul Sabatier

Résumé

La statistique robuste est une branche de la statistique qui s'intéresse à l'analyse de données contenant une proportion significative d'observations contaminées avec des erreurs dont l'ampleur et la structure peuvent être arbitraires. Les estimateurs robustes au sens du point de rupture sont généralement définis comme le minimum global d'une certaine measure non-convexe des erreurs, leur calcul est donc un problème d'optimisation globale très coûteux. L'objectif de cette thèse est d'étudier les contributions possibles des méthodes d'optimisation globale modernes à l'étude de cette classe de problèmes.

La première partie de la thèse est consacrée au τ-estimateur pour la régression linéaire robuste, qui est défini comme étant un minimum global d'une fonction non-convexe et dérivable. Nous étudions l'impact des techniques d'agglomération et des conditions d'arrêt sur l'efficacité des algorithmes existants. Les conséquences de certains phénomènes liés au voisin le plus proche en grande dimension sur ces algorithmes agglomératifs d'optimisation globale sont aussi mises en évidence.

Dans la deuxième partie de la thèse, nous étudions des algorithmes déterministes pour le calcul de l'estimateur de moindres carrés tronqués, qui est défini à l'aide d'un programme en nombres entiers non linéaire. En raison de sa nature combinatoire, nous avons dirigé nos efforts vers l'obtention des bornes inférieures pouvant être utilisées dans un algorithme du type *branch-and-bound*. Plus précisément, nous proposons une relaxation par un programme sur le cône de deuxième ordre, qui peut être renforcée avec des coupes dont nous présentons l'expression explicite. Nous fournissons également des conditions d'optimalité globale.

Abstract

Robust statistics is a branch of statistics dealing with the analysis of data containing contaminated observations. The robustness of an estimator is measured notably by means of the *breakdown point*. High-breakdown point estimators are usually defined as global minima of a non-convex scale of the errors, hence their computation is a challenging global optimization problem. The objective of this dissertation is to investigate the potential contributions of modern global optimization methods to this class of problems.

The first part of this thesis is devoted to the τ-estimator for linear regression, which is defined as a global minimum of a nonconvex differentiable function. We investigate the impact of incorporating clustering techniques and stopping conditions in existing stochastic algorithms. The consequences of some phenomena involving the nearest neighbor in high dimension on clustering global optimization algorithms is thoroughly discussed as well.

The second part is devoted to deterministic algorithms for computing the least trimmed squares regression estimator, which is defined through a nonlinear mixed-integer program. Due to the combinatorial nature of this problem, we concentrated on obtaining lower bounds to be used in a branch-and-bound algorithm. In particular, we propose a second-order cone relaxation that can be complemented with concavity cuts that we obtain explicitly. Global optimality conditions are also provided.

Lo terrible se aprende enseguida,
y lo hermoso nos cuesta la vida...

Remerciements

Malgré l'impossibilité de tenir dans une page le vécu de toute cette période, je tâcherai d'exprimer ma gratitude à tous ceux qui m'ont accompagné pendant ces travaux de thèse de la façon la plus exhaustive possible.

Je commence par un grand merci à mes directeurs de thèse, Anne et Marcel. Leur enthousiasme et excellente disposition ont beaucoup contribué à l'accomplissement de cette thèse. Je tiens à remercier en même temps Jean-Baptiste Hiriart-Urruty, qui a suivi de près l'avancement de cette thèse, en apportant toujours de conseils et du soutien, et Felipe Alvarez, qui a guidé mes premiers pas dans le monde de la recherche et dont le soutien a été déterminant pour continuer mes études en France. Pareillement, c'est avec grand plaisir que je salue la convivialité de l'équipe Optimisation et Interactions, et plus particulièrement d'Aude Rondepierre, avec qui j'ai partagé bureau pendant une très agréable première année de thèse.

J'ai également l'honneur de remercier les professeurs Fabio Schoen, Michael Schyns et Roy Welsch pour avoir accepté de rapporter cette thèse et de s'être, par surcroît, déplaces malgré leur emploi de temps chargés pour participer à mon jury de thèse. Toujours sur le plan académique, une pensée amicale pour Matías Salibián-Barrera et Rubén Zamar, qui m'ont accueilli pour un séjour de recherche au Canada. Il a été un grand plaisir de partager avec eux et profiter de leur expérience.

Outre le développement professionnel visé en priorité par le doctorat, j'en ai aussi tiré un enrichissement personnel d'une valeur inestimable. Y ont contribué, tout d'abord, Benjamin, à qui je dois son amitié et patience envers mon français hésitant du début. Dominique, Jean-Luc et Xavier, des matheux atypiques, ainsi que Laetitia, Laurent, Tiphaine et Stefan dans le cadre du labo, et les copains chiliens Ana, Carola, Eli et Germán. J'ai aussi apprécié la cordialité des randonneurs de l'UPS et des danseurs de musiques traditionnelles, avec mention spéciale pour mon co-bureau et ami portugais Jorge. Le soutien à la distance de ma famille et de mes amis, au Chili et ailleurs, a été fondamental.

Cette thèse a pu être mené à bien grâce à une bourse Master 2 Investigación y Doctorado, Embajada de Francia en Chile - CONICYT.

Contents

Presentation of the statistical problem

Robust statistics is a branch of statistics dealing with the analysis of data that may contain large portions of contaminated observations. The origin of such contamination can be very diverse. In Figure 1, we show an example where the contamination comes from measurement errors (Rousseeuw and Leroy, 1987). For each year from 1950 to 1973, the number of outgoing international phone calls from Belgium are plotted. The bulk of the data follows a linear model; nonetheless, there are 6 observations that deviate from the majority. In fact, during the period between 1964 and 1969, there was a change on the recording system, which actually recorded the total duration, in minutes, of the international phone calls instead of the number of calls.

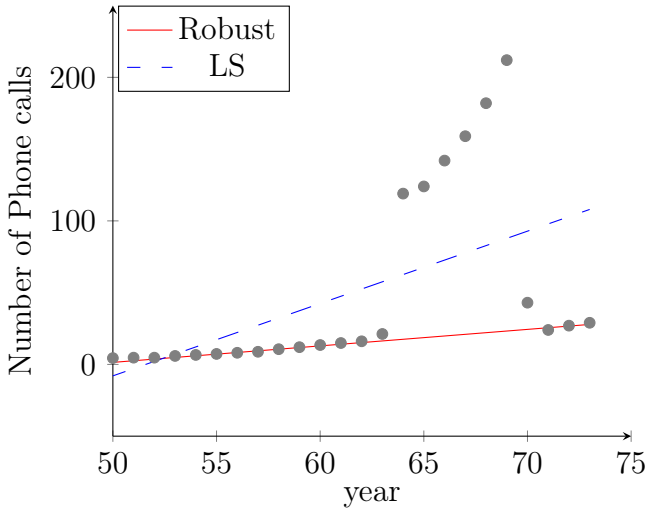

Figure 1: International phone calls from Belgium in the period 1950-1973.

A case where a subpopulation acts differently is shown in Figure 2. There are plotted, for 32 chemical compounds, a quantity called the krafft point versus a molecular descriptor called heat of formation. There is a main group that follows a regression line, but there is also, besides some few outliers at right, a second, smaller group forming what seems to be another regression line. The observations in the second group correspond to sulfonates.

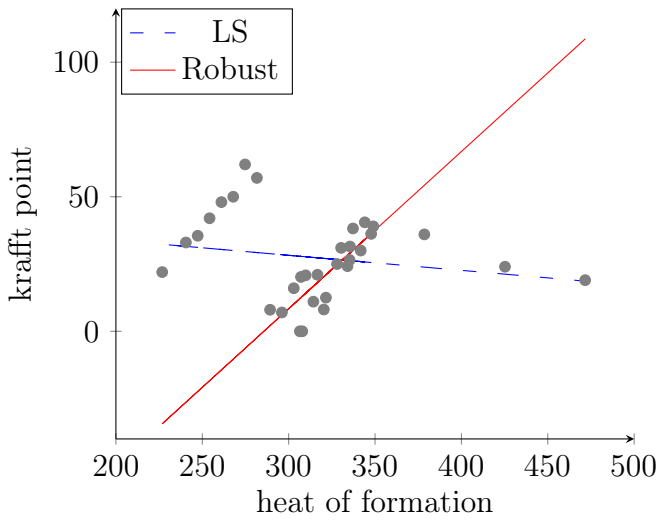

Figure 2: Krafft point for 32 chemical compounds.

Therefore, due to diversity of the possible situations, it is impossible to make assumptions on the magnitude nor on the structure of the contamination.

Besides robustness, in a sense to be specified soon, robust regression estimators satisfy statistical properties such as asymptotic normality, square-root rate of convergence and equivariance properties. Though, the use of robust estimators is not as widespread as one could expect; mainly because their computation is very time consuming and, unlike many other problems arising in statistics, the difficult problems involved in computing robust estimators are almost unknown to optimization specialists. The objective of this thesis is to explore the potential improvements that can be obtained in the computing efficiency by taking advantage of recent advances in optimization.

Robust statistics is nowadays a well-established discipline, and most classical statistical procedures have a robust counterpart. However, it is customary to consider separately the case of multivariate analysis from linear regression. In many circumstances the linear model assumption leads to work in lower dimension than in multivariate analysis.

The regression context

The linear regression context is the following:

- Observed sample $\{x_1, ..., x_n\} \subseteq \mathbb{R}^d$;
- response $y = (y_1, ..., y_n) \in \mathbb{R}^d$;
- linear model assumption

$$y_i = x_i'\beta + \delta_i,$$

with δ_i independent and identically distributed , $\mathbb{E}[\delta_i] = 0$, $Var[\delta_i] = \sigma^2$;

- The observations are rows of a matrix $X \in \mathbb{R}^{n\times d}$,

$$X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} x_1^{(1)} & x_1^{(2)} & \cdots & x_1^{(d)} \\ x_2^{(1)} & x_2^{(2)} & \cdots & x_2^{(d)} \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ x_n^{(1)} & x_n^{(2)} & \cdots & x_n^{(d)} \end{pmatrix};$$

- for $\beta \in \mathbb{R}^d$, we denote by $\mathbf{r}(\beta)$ the vector of *residuals* $\mathbf{r} = y - X\beta$, with components $r_i = y_i - x_i'\beta$.

Our objective is to obtain estimators $\hat{\beta}(X, y)$ of the parameter β satisfying additionally some *equivariance* properties:

$$\hat{\beta}(X, \lambda y + X\gamma) = \lambda\hat{\beta}(X, y) + \gamma \quad \text{for all } \lambda \in R, \gamma \in \mathbb{R}^d,$$

and for all nonsingular $d \times d$ matrix A:

$$\hat{\beta}(XA, y) = A^{-1}\hat{\beta}(X, y).$$

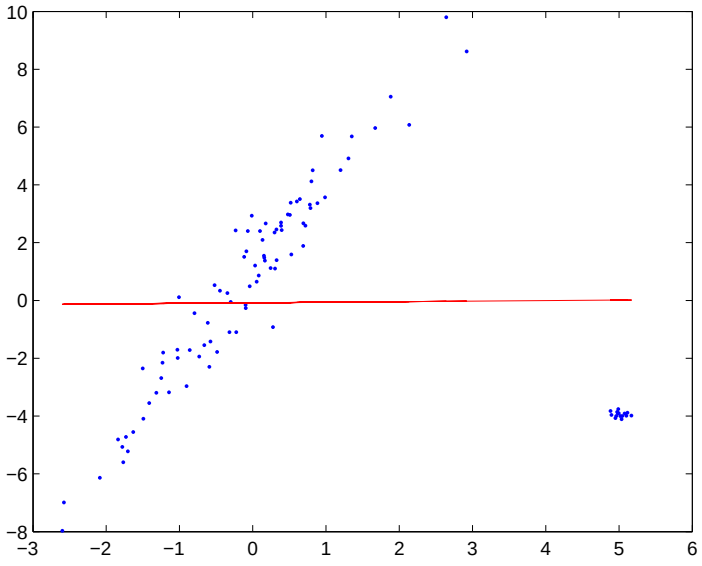

Figure 3: LS regression on a dataset with 20% of contamination.

Classical estimators satisfy these properties, but break down in presence of contamination. Let us take as an example the well-known least squares estimator, which is defined as

$$\hat{\beta}_{LS} = \underset{\hat{\beta}\in\mathbb{R}^d}{\text{Arg min}} \; S_{LS}(\mathbf{r}(\hat{\beta})),$$

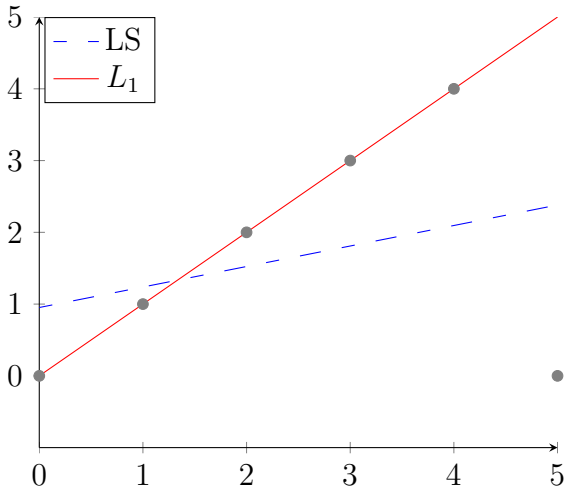

Figure 4: L_1 and LS regressions on a toy example.

where

$$S_{LS}(\mathbf{r}) = \sum_{i=1}^{n} r_i^2 = \|\mathbf{r}\|_2^2. \tag{1}$$

We can see in Figure 3 the rout of the LS estimator on a dataset with 20% of contamination. It has been observed that the quadratic scale in (1) overweights observations with large residuals. Then it seems reasonable to replace the quadratic scale by a scale with linear growth that should result in an estimator less sensitive to outlying observations. Such an estimator is the so-called (with a slight abuse of notation) L_1 estimator, defined as the minimizer of the ℓ_1 norm of residuals,

$$\hat{\beta}_{L_1} = \underset{\hat{\beta}\in\mathbb{R}^d}{\text{Arg min}} \ \|\mathbf{r}(\hat{\beta})\|_1,$$

The example of Figure 4, presented in Clarke (1983) as an application of non smooth optimization in statistics, confirms somehow this intuition; however, the ℓ_1 regression over the dataset of Figure 3 does not give better results than the least squares regression, as shown in Figure 5.

These contradictory signals raise the need for a clear definition of what «robust» means. This will be the subject of the next section.

A measure of robustness: the Breakdown Point

As the previous examples about the L_1 estimator highlight, it is necessary to formalize the idea of robustness. Hereafter we will adopt the robustness notion introduced by Donoho and Huber (1983), which is based on the concept of *breakdown point (BDP)*. Roughly speaking, the BDP of an estimator on a sample is defined as the minimum fraction of the observations that need to be replaced by arbitrary ones for

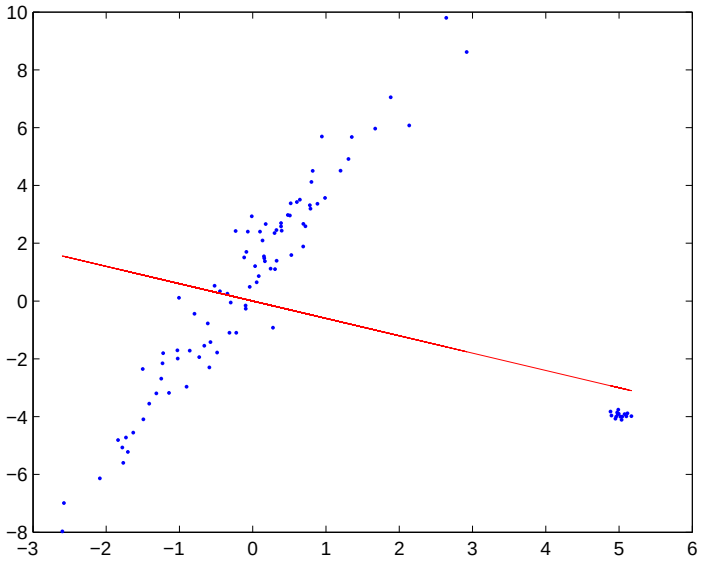

Figure 5: ℓ_1 regression on the same dataset with a 20% of contamination as in Figure 3.

the estimator to take on arbitrary values. The concept of BDP is consistent with the requirement that no hypothesis can be made on the distribution of the contamination, as it only depends on the fraction of the sample that is to be corrupted. The formal definition of the BDP is the following: consider any sample of n points (X, y) and let T be a regression estimator, $T(X, y) = \hat{\beta}$. Then consider all possible corrupted samples (X', y') that are obtained by replacing m of the original points by arbitrary values. Let us denote by $\text{bias}(m; T, X, y)$ the maximum bias that can be caused by such a contamination

$$\text{bias}(m; T, X, y) = \sup_{X', y'} \|T(X, y) - T(X', y')\|_2. \tag{2}$$

The breakdown point of the estimator T at the sample (X, y) is defined as

$$\epsilon_n^*(T, X, y) = \min \left\{ \frac{m}{n} \mid \text{bias}(m; T, X, y) = \infty \right\}.$$

In order to compare different estimators, one usually considers the asymptotic behaviour of $\epsilon_n^*(T, X, y)$:

$$\epsilon^*(T, X, y) = \lim_{n \to \infty} \epsilon_n^*(T, X, y).$$

So, as $\epsilon_n^*(\hat{\beta}_{LS}, X, y) = \epsilon_n^*(\hat{\beta}_{L_1}, X, y) = 1/n$ for any sample (X, y), one says that both the LS and the L_1 estimators have a breakdown point of 0%.

Note that in equation (2) contamination of any type and magnitude is allowed. This is the main difference with the approach of *Robust Optimization*, which searches

for an optimal solution to problems involving uncertain data on the objective function f_0 and/or on the constraints f_i, for which one can specify an uncertainty set $\mathcal{U}$:

$$\min_x \{f_0(x,\zeta) \mid f_i(x,\zeta) \leq 0,\ i = 1,...,m. \quad \forall \zeta \in \mathcal{U}\}.$$

In general, robust optimization counterparts to classical estimators will not have a higher breakdown point than the original estimator. However, in problems where the uncertainty can be confidently supposed to belong to well-defined (convex!) uncertainty sets, the robust optimization framework could be preferable, since there exists efficient polynomial-time algorithms for solving those problems. The interested reader is referred to the article by Ben-Tal and Nemirovski (2008) for further details.

The BDP of the L_1 estimator

As the example in Figure 5 suggests, $\epsilon^*(\hat{\beta}_{L_1}, X, y) = 0$ if contamination is allowed in X. However, since the L_1 estimator can be efficiently computed, there has been some interest on its BDP robustness in the case of *planned experiments*, namely, when the matrix of regressors X is deterministic and only y can be contaminated. In this case there exists some characterizations of $\epsilon^*(\hat{\beta}_{L_1}, X, y)$, which can be positive. Let m_* be the largest integer such that for any set $S \subseteq \{1,...,n\}$ of cardinality m_*,

$$\max_{\|\xi\|=1} \frac{\sum_{i \notin S} |x_i'\xi|}{\sum |x_i'\xi|} \geq \frac{1}{2},$$

then $\epsilon^*(\hat{\beta}_{L_1}, X) = (m_* + 1)/n$ (see He et al., 1990; Mizera and Müller, 2001). The combinatorial nature of this characterization makes the BDP of the L_1 estimator very difficult to compute, thus unmanageable. Giloni and Padberg (2004) gave another characterization based on the linear structure of the problem, which involves solving a Mixed-Integer Program (MIP) with $2n$ integer variables.

Nevertheless, the intuition behind the example of Figure 4 is not completely without foundation. Candes and Tao (2005) showed the following: assume that we want to recover the vector β from corrupted measurements $y = X\beta + e$, for a deterministic matrix X and an arbitrary unknown vector of errors e. Then β is the unique solution to

$$\min_{g \in \mathbb{R}^d} \|y - Xg\|_1,$$

provided that the ℓ_0 "norm", or *cardinality norm* of e, $\|e\|_0 = \text{cardinality}(\{i \mid e_i \neq 0\})$ is less than or equal to some $\eta > 0$. The number η must satisfy $\delta_\eta + \theta_{\eta,\eta} + \theta_{\eta,2\eta} < 1$, where

$$\delta_\eta = \max_{|J| \leq \eta, c \in \mathbb{R}^{|J|}} \left| \frac{\|F_J c\|^2}{\|c\|^2} - 1 \right|, \quad \theta_{\eta,\eta'} = \max_{|J| \leq \eta, |J'| \leq \eta'; c \in \mathbb{R}^{|J|}; c' \in \mathbb{R}^{|J'|}} \frac{|\langle F_J c, F_{J'} c' \rangle|}{\|c\| \|c'\|},$$

and F is any matrix F such that $FX = 0$. For an index set J, F_J denotes the sub matrix of the rows of F indexed by J. Since the result holds under the hypothesis that uncontaminated observations have null residuals, this framework is rather adapted to signal recovery. It is not surprising that in the example of Figure 4 the

majority of the observations pass exactly through a line. Going even deeper into the theoretical basis of this results we can find a result due to Fazel (2002) stating that the ℓ_1 norm is indeed the convex envelope of the cardinality norm ℓ_0 over the unit ball. In simple words, the ℓ_1 norm is the one that gives the least weight to large-residuals observations while keeping the problem convex. In the following section we will describe a class of estimators with a high breakdown point. As it could be expected, they involve nonconvex minimization problems.

High BDP estimators

Robust estimators are defined similarly to the LS and L_1 estimators, but the quadratic and linear scales of residuals are replaced by robust residual scales, which are not influenced by observations with huge residuals. We present below two such robust scales, $M-$estimates of scale and $L-$estimates of scale. The former defines robust estimators whose computation involve continuous nonconvex optimization problems, and the later gives raise to mixed-integer nonlinear programs (MINLPs).

Robust estimators based on $L-$estimates of scale

$L-$scales: definition Let $|r|_{(1)} \leq ... \leq |r|_{(n)}$ be the ordered absolute values of residuals. $L-$estimates of scale are defined as weighted ℓ_1 or ℓ_2 norms of the $|r|_{(i)}$s,

$$\sum_{i=1}^{n} a_i |r|_{(i)}, \qquad \text{or} \qquad \left(\sum_{i=1}^{n} a_i |r|_{(i)}^2 \right)^{1/2}, \tag{3}$$

where the a_is are nonnegative constants.

The least trimmed squares (LTS) estimator It is defined by minimizing an $L-$scale as in the second form in (3), with $a_i = 1$ for $1 \leq i \leq h$ and $a_i = 0$ for $h < i \leq n$, for a given h. In other words, the LTS estimator is defined as

$$\hat{\beta}_{LTS} = \underset{\hat{\beta} \in \mathbb{R}^d}{\text{Arg min}} \; S_{LTS}(\mathbf{r}(\hat{\beta})), \tag{4}$$

where

$$S_{LTS}(\mathbf{r}) = \left(\sum_{i=1}^{h} |r|_{(i)}^2 \right)^{1/2}.$$

The breakdown point of the LTS estimator depends on h, and the optimal choice is $h = \lfloor n/2 \rfloor + \lfloor (p+1)/2 \rfloor$ which leads to a BDP of nearly 50% (Rousseeuw and Leroy, 1987).

From an optimization viewpoint, problem (4) can be written as the following MINLP:

$$\min_{\substack{\hat{\beta} \in \mathbb{R}^d \\ w_i \in \{0,1\}}} \left(\sum_{i=1}^{n} w_i r_i(\hat{\beta})^2 \right)^{1/2} \tag{5}$$

subject to the constraint

$$\sum_{i=1}^{n} w_i \geq h.$$

Rousseeuw and Driessen (2006) describe a stochastic algorithm for approximating the LTS estimator. It has been implemented and incorporated into the robustbase package in R. The only known deterministic approach is that of Giloni and Padberg (2002), where the authors study the polytope defined by the constraints $w_i \in \{0,1\}, \sum w_i \geq h$ and show that the relaxed problem attains the minimum at an extreme point, which necessarily is $0-1$. Then the authors propose a heuristic algorithm.

The least median of squares (LMS) estimator The LMS estimator is defined as a minimizer of the median of the residuals:

$$\hat{\beta}_{LMS} = \underset{\hat{\beta}\in\mathbb{R}^d}{\text{Arg min}} \ \underset{i=1,\ldots,n}{\text{median}} |r|_{(i)} = \underset{\hat{\beta}\in\mathbb{R}^d}{\text{Arg min}} \ |r|_{(h)},$$

where $h = \lfloor n/2 \rfloor$. It minimizes an $L-$scale as in the first form of (3), with $a_i = 0$ except for a_h which equals 1. The breakdown point of the LMS estimator is the same as that of the LTS estimator. As for the LTS estimator, there exists a stochastic algorithm, implemented in R, for approximating the LTA estimator; Giloni and Padberg (2002) conducted a theoretical study of this estimator.

The least trimmed absolute deviations (LTA) estimator The LTA estimator is defined in the same way as the LTS estimator, but using the first form in (3), i.e it minimizes the least trimmed absolute deviations:

$$\hat{\beta}_{LTA} = \underset{\hat{\beta}\in\mathbb{R}^d}{\text{Arg min}} \ S_{LTA}(\mathbf{r}(\hat{\beta})),$$

where

$$S_{LTA}(\mathbf{r}) = \sum_{i=1}^{h} |r|_{(i)}.$$

The breakdown point of the LTA estimator is the same as that of the LTS estimator. It has the advantage with respect to LTS that by adding (or splitting) the variables it can be written as a linear MIP. Hawkins and Olive (1999) proposed to solve it exactly by simple enumeration, using the following property of the L_1 minimizer: there always exists a subset of size d for which the fit is exact, i.e the residuals are 0. Hence it suffices to enumerate all subsets of size d, instead of the more numerous subsets of size h.

Robust estimators based on $M-$estimates of scale

$M-$scales: definition The $M-$estimate of scale, $S_M(\mathbf{r})$, is defined for each $\mathbf{r}$ as the solution of

$$\frac{1}{n}\sum_{i=1}^{n} \rho\left(\frac{r_i}{S_M}\right) = b,$$

where the function $\rho : \mathbb{R} \to \mathbb{R}_+$ satisfies the following hypotheses:

$R1$. ρ is even and twice continuously differentiable.

$R2$. $\rho(0) = 0$.

$R3$. ρ is strictly increasing on $[0, c]$, for some $0 < c < +\infty$.

$R4$. ρ is constant on $[c, \infty)$.

The constant b is conveniently defined as $b = \int \rho(t) d\Phi(t)$, where Φ denotes the standard normal distribution, in order to obtain consistency for the scale at the normal error model.

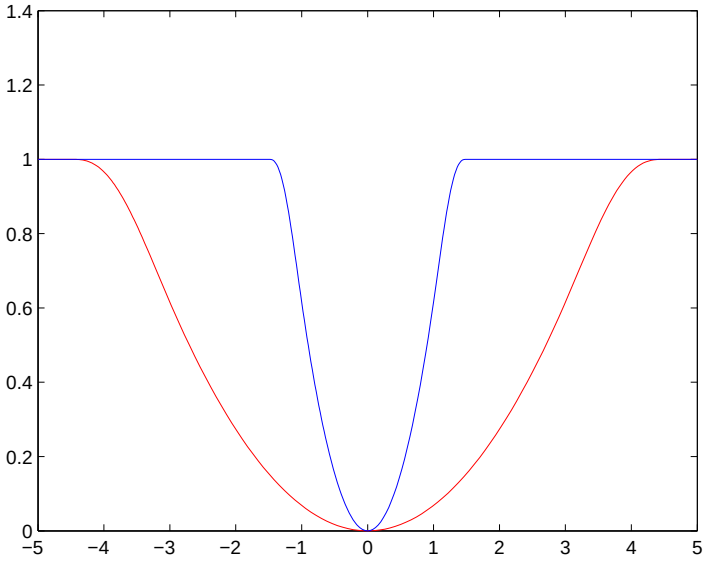

Figure 6: Two admissible ρ functions.

τ-estimator of regression The τ-estimator of regression, $\hat{\beta}_\tau$, is defined as

$$\hat{\beta}_\tau = \underset{\hat{\beta} \in \mathbb{R}^d}{\text{Arg min}} \; S_M^2(\mathbf{r}(\hat{\beta})) \frac{1}{n b_2} \sum_{i=1}^{n} \rho_2 \left(\frac{r_i(\hat{\beta})}{S_M(\mathbf{r}(\hat{\beta}))} \right), \tag{6}$$

where $S_M(\mathbf{r})$ is an $M - scale$:

$$\frac{1}{n} \sum_{i=1}^{n} \rho_1 \left(\frac{r_i}{S_M(\mathbf{r})} \right) = b_1, \tag{7}$$

ρ_1 and ρ_2 are functions satisfying $R1$-$R4$, and b_1, b_2 are real constants.

The breakdown point of τ estimators equals $\epsilon^* = b_1/\rho_1(c)$. Hence, for an adequate choice of the function ρ_1 the maximal breakdown point of 50% can be obtained. Similarly, by adjusting the function ρ_2, the asymptotic efficiency at the

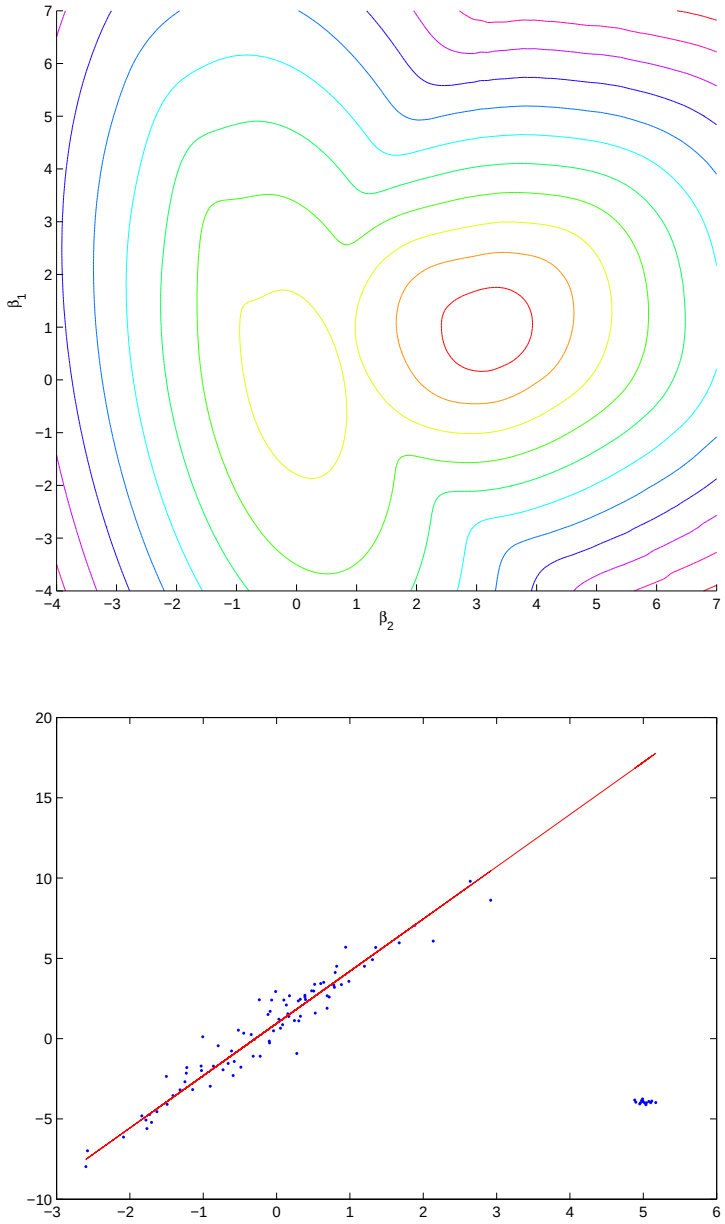

Figure 7: The τ objective function (top), and the τ-estimator fitting (bottom), for the same data as in Figure 3.

normal distribution can be made arbitrarily close to 1. A related estimator is the so-called S-estimator, defined in the same way as τ-estimators, but replacing the objective function (6) just by $\hat{\beta} \mapsto S_M^2(\mathbf{r}(\hat{\beta}))$. It has the same robustness properties of τ-estimators but a lower efficiency.

The main difficulty in computing τ-estimators comes from its lack of convexity. In fact, in the context of estimators based on residual scales, convexity and robustness are antagonist concepts. For example, for the simpler S-estimators, the first-order optimality condition reads:

$$\sum_{i=1}^{n} \rho' \left(\frac{r_i(\hat{\beta})}{S_M(\mathbf{r}(\hat{\beta}))} \right) x_i = 0$$

which is a weighted mean with weights $w_i = \rho'(r_i(\hat{\beta})/S_M(\mathbf{r}(\hat{\beta})))$. Therefore, a convex ρ would give higher weights to observations with larger residuals. Note that the weights depend on the residuals, and so they are adaptive. This is a difference with estimators based on L-scales, where the weights have a rigid structure. This is an advantage from the viewpoint of statistics, but it is a drawback from an optimization point of view. Actually, we could not find any structure to exploit, contrarily to the LTS optimization problem, which is known to be a concave minimization problem (which is a type of "structured" nonconvexity). For this reason, we will investigate stochastic algorithms for τ-estimators, that adapt themselves to a wide class of problems. For the more structured LTS we will settle for warranted optimal solutions taking advantage of the structure of the problem.

A word on the multivariate case

For completeness, we will briefly describe here the multivariate context. We have a data set $Z = \{x_1, ..., x_n\} \subseteq \mathbb{R}^d$. We suppose that the n observations are independent realizations of a certain random variable $\mathcal{X}$, which follows an elliptic distribution with induced measure absolutely continuous with respect to the Lebesgue measure, with a density h of the form

$$h(\mathbf{x}) = \det(\Sigma)^{-1/2} f((\mathbf{x} - \mu)\Sigma^{-1}(\mathbf{x} - \mu)), \tag{8}$$

where $\mu \in \mathbb{R}^d$ is called the *location*, Σ is a $d \times d$ positive definite matrix (henceforth $\Sigma \succeq 0$) called *scatter matrix*, and f is an arbitrary positive function. Two random variables with the same scatter matrix, up to some multiplicative factor, are said to have the same shape but different size. The expression in the argument of f in (8) is called *Mahalanobis distance*, it is the squared norm of the vector $x - \mu$, for the norm induced by the inner product $\langle \cdot, \cdot \rangle_{\Sigma^{-1}} = \langle \cdot, \Sigma^{-1} \cdot \rangle$. In the sequel we shall use the following notation for the Mahalanobis distance:

$$d^2(x, \mu; \Sigma) \triangleq (x - \mu)' \Sigma^{-1} (x - \mu).$$

Our objective will be to estimate the location μ and the scatter matrix Σ from the data set Z, which may be strongly contaminated. We shall search for estimators

satisfying the same equivariance properties as their regression counterparts,

$$\hat{\mu}(AZ+b) = A\hat{\mu}(Z) + b, \quad \hat{\Sigma}(AZ+b) = A\hat{\Sigma}(Z)A',$$

for any invertible $d \times d$ matrix A and any $b \in \mathbb{R}^d$.

The maximum likelihood estimators, which are also the LS estimators, are the empirical mean for location,

$$\hat{\mu}(Z) = \frac{1}{n}\sum_{i=1}^{n} x_i,$$

and the unbiased estimator

$$\hat{\Sigma}(Z) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \hat{\mu}(Z))(x_i - \hat{\mu}(Z))',$$

for the scatter matrix.

The breakdown point for multivariate location is defined in the same way as for regression, namely, the smallest fraction of observations that need to be replaced by arbitrary ones to render the bias unbounded. For the scatter matrix, a usual measure of bias is $\max(\lambda_d(\hat{\Sigma}), \lambda_1(\hat{\Sigma}))$, where $\lambda_d(\hat{\Sigma})$ and $\lambda_1(\hat{\Sigma})$ denote the largest and the smallest eigenvalue of $\hat{\Sigma}$, respectively. Robust estimators for multivariate location are defined similarly as for regression, minimizing a robust scale, but this time the residuals are replaced by the Mahalanobis distances. The τ-estimators for multivariate location and scatter are defined as minimizers of the optimization problem

$$\begin{aligned} \min_{\mu,\Sigma} \quad & \det(\Sigma)\left[\sum_{i=1}^{n}\rho_1(d^2(x,\mu;\Sigma))\right]^d \\ \text{s.t} \quad & \\ & \sum_{i=1}^{n}\rho_2(d^2(x,\mu;\Sigma)) \leq b, \\ & \mu \in \mathbb{R}^d, \Sigma \succeq 0, \text{symmetric}, \end{aligned}$$

where the functions ρ_1, ρ_2 play a role analogous to the functions in (6) and (7), and are required to satisfy the same hypotheses. The S-estimator of multivariate location and scatter is obtained through the same problem but keeping only $\det(\Sigma)$ in the objective function. An estimator analogous to the LTS regression estimator exists for multivariate robust estimation, it is called the Minimum Covariance Determinant (MCD). For a given $\lfloor n/2 \rfloor < h \leq n$, the MCD is defined as the mean and covariance matrix of a subsample of size h for which the determinant of the covariance matrix is minimum.

For S and τ-estimators there exist undocumented implementations of stochastic algorithms similar to those existing for regression. The literature for the MCD is more extensive; we can cite Rousseeuw and Driessen (1998) for two-step stochastic algorithms based on subsampling and concentration steps. Recently, Schyns et al. (2010) presented an algorithm based on a new approach consisting of relaxing the variables of the combinatorial problem for dealing finally with a smooth problem in continuous variables. Agulló (1997) proposed a branch-and-bound algorithm for the exactly computing the MCD estimator. Recently, Nguyen and Welsch (2010) proposed a method for robust covariance estimation. It is not a high breakdown point

estimator, but it is defined through a positive semidefinite program and therefore it can be efficiently computed even for large data sets.

Part I

Stochastic algorithms for approximately solving continuous global optimization problems.

Chapter I.1

Introduction

The first part of this thesis is devoted to τ-estimators for robust regression. This choice was done because the τ-estimator enjoys the best statistical properties: the robustness and the efficiency of the τ-estimator can be tuned simultaneously without trade-off. The computation of the τ-estimator of regression involves a non-convex differentiable optimization problem in continuous variables. One of its major difficulties, besides non-convexity, is that the feasible domain of optimization is the whole Euclidean space. Therefore, exhaustive exploration of the feasible domain is impossible; this rules out most deterministic algorithms for global optimization in continuous variables. At that point stochastic algorithms are one alternative, but the unboundedness of the domain appears as an issue again. However, this inconvenient has already been overcome in existing algorithms for robust regression. Most of them are based on a sampling technique called subsampling. Originally conceived for the LTS estimator, subsampling consists of drawing from a multinomial distribution a subsample of h observations, with $d+1 \leq h \leq n$, from which a candidate $\beta \in \mathbb{R}^d$ can be obtained as the LS fit to that subsample. Even if for the LTS optimization problem (4) subsampling boils down to uniformly sampling on the feasible domain $\{w \in [0,1]^n, \sum_{i=1}^n w_i = h\}$, the rationale behind subsampling, namely that by drawing enough subsamples we should have the chance to pick at least one outlier-free subsample, remains appealing for any stochastic algorithm intended to compute robust estimators.

Sampling is the cornerstone of any stochastic optimization algorithm. Once a way of sampling over the region of interest is available, a range of alternatives show up, each one more or less adapted to some specific application.

Most authors classify stochastic global optimization algorithms into (see Schoen, 1991; Törn and Žilinskas, 1989, for good surveys on the subject):

- Two-phase algorithms.
- Heuristic algorithms.
- Algorithms based on a stochastic model of the function.

The first two families of algorithms form a subclass to which we will further restrict ourselves, because they permit to take advantage of the differentiability of

the objective function to perform local searches. The reader interested in the last family of algorithms is refered to Betrò (1991) for a detailed review.

Let us briefly describe two heuristic algorithms that have shown the best performance in practice: simulated annealing (SA) and tabu search (TS). The basic idea of SA is to move through the search space by randomly choosing a neighbor of the current point that decreases the value of the objective function. However, to avoid being trapped in a local minimum, it may randomly allow a move that increases the objective function. A move from the current point to a new one is accepted with probability $\min\{1, \exp(-\Delta/T)\}$, where Δ is the potential increase on the objective function that would result from this move, and $T > 0$ is a tuning constant called temperature. Note that when $T \to 0$ the probability of accepting a move that does not improve the objective function decreases rapidly. The algorithm starts with a relatively large value of T which is then gradually reduced during the search. Eventually the algorithm freezes in a local minimum because no uphill moves can be accepted. Among all points visited by the algorithm the one with the smallest objective function is chosen as the approximate minimizer. Even if it was originally inspired from the physical process of driving a system to its minimum energy state by a "cooling" process, simulated annealing possesses a theoretical motivation. Let Ω denote the domain of interest. The Boltzmann distribution on Ω, π_T, is defined as

$$\pi_T(\beta) \propto e^{\frac{-\sigma(\beta)}{T}}, \ \beta \in \Omega \tag{I.1.1}$$

where $T > 0$ is the temperature. It can be proved that π_T converges, as $T \to 0$, to the uniform distribution over the set of global minimizers of the function σ. Since sampling from (I.1.1) would require the evaluation of σ on every point of Ω, the Metropolis algorithm can be used to obtain random samples from π_T; the Metropolis algorithm consists in simulating a sample path of a Markov process with transition kernel given by

$$K_T(x, y) = e^{\frac{[\sigma(y)-\sigma(x)]^+}{T}} R(x, y)$$

for any symmetric irreducible matrix R. Unlike SA, TS moves *deterministically* to the point that has the lowest objective value among all the neighbors. Once the search arrives at a local minimum in this way, the algorithm has to make an uphill move (the algorithm is forced to make a move even if every possible neighbor would yield an increase in objective value). In order to avoid that the algorithm immediately moves back to the local minimum in the next step, and thus become trapped, TS uses a *tabu list* that forbids moves that could yield to previously visited local minima, even if they would produce a better objective value.

On the other hand, there are two-phase algorithms. All of the algorithms in this class of algorithms proceed in three stages:

1. Sampling.
2. Local optimization.
3. Check stopping condition.

In Flores (2010), we have shown that most of the existing algorithms for robust regression fit in this class of algorithms. Furthermore, the computational study in Salibian-Barrera et al. (2008) shows that for optimization problems associated to robust regression, two-phase methods outperform heuristic algorithms. Nevertheless, stopping conditions are currently absent in existing algorithms. They also pass by clustering techniques that in some problems have helped to improve efficiency of two-phase algorithms. The impact of stopping conditions and clustering techniques for the approximation of robust regression estimators is the subject of Chapter I.2. This work is published in Flores (2010).

The main conclusion of Chapter I.2 is that stopping conditions are a valuable improvement to existing algorithms for robust regression. On the contrary, the impact of performing clustering is rather disappointing, especially in middling and high dimension. In Chapter I.3, one possible reason of the performance degradation of clustering global optimization in higher dimension is studied: the counter-intuitive behavior of the nearest neighbor in high dimension.

Chapter I.2

On the efficient computation of robust regression estimators

S. Flores.
Computational Statistics & Data Analysis 54 (12), 3044–3056.

I.2.1 Introduction

Robust regression methods have been introduced to cope with the need left by classical techniques for methods that work well in the presence of contamination in the data. Particular interest has been focused on estimators with a high breakdown point as defined by Donoho and Huber (1983). Most of these methods combine robustness with desirable statistical properties such as consistency and asymptotic normality.Nonetheless, they are defined by means of difficult global minimization problems; hence, their computation is very time consuming. We address the problem of the efficient computation of robust estimators for statistical regression based on M-scales. The principal aim of our work is to investigate to what extent the most recent developments in the field of optimization can help improve the existing computational methods.

Let us consider the classical linear regression model

$$y_i = \mathbf{x}_i'\beta + \varepsilon_i, \quad i = 1, ..., n,$$

with errors terms ε_i identically distributed with zero center and independent from the covariates $\mathbf{x}_i$. We shall denote by y the vector with components y_i, by X the $n \times d$ matrix with ith row $\mathbf{x}_i$, and by r the vector of residuals $r(\beta) = y - X\beta$ with components $r_i = y_i - \mathbf{x}_i'\beta$.

Many robust estimators of the regression coefficients $\beta \in \mathbb{R}^d$ based on n independent observations $(y_i, \mathbf{x}_i) \in \mathbb{R} \times \mathbb{R}^d$ can be defined as:

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^d}{\text{Arg min}} \quad \hat{\sigma}(r(\beta)), \tag{P}$$

where $\hat{\sigma}$ is a scale estimator. An important case of (P) is the S-estimator, defined with $\hat{\sigma}(r) = s(r)$, where $s : \mathbb{R}^n \to \mathbb{R}_+$ is an M-estimator of scale, or M-scale (Huber,

1981), defined implicitly through:

$$\frac{1}{n}\sum_{i=1}^{n}\rho\left(\frac{r_i}{s(r)}\right)=b. \tag{I.2.1}$$

The function $\rho:\mathbb{R}\to\mathbb{R}_+$ is required to satisfy the following assumptions:

1. ρ is even and twice continuously differentiable,
2. $\rho(0)=0$,
3. ρ is strictly increasing on $[0,c]$, for some $0<c<+\infty$,
4. ρ is constant on $[c,\infty)$.

Any function satisfying these assumptions will be called in the sequel a "rho function".
The constant b is conveniently defined as

$$b=\mathbb{E}_\Phi(\rho), \tag{I.2.2}$$

where Φ denotes the standard normal distribution, in order to obtain consistency for the scale at the normal error model.

The S-estimator has very good robustness properties, but it lacks efficiency. Robustness is to be understood in the sense of the *breakdown point*, which is the minimum fraction of the observations that need to be shifted for the estimator to take on arbitrary values. In fact, there is a tradeoff between robustness and efficiency, and for a breakdown point of 50%, efficiency can be as low as 28.7% (Maronna et al., 2006, pp. 131). MM-estimators were introduced to fill this gap. They are obtained by local minimization of a suitable function using an S-estimator as starting point. In this way, they can combine efficiency with a high breakdown point. See Maronna et al. (2006, Sec. 5.5) for details. However, τ-estimators have lower bias curves, and their computing effort is comparable to computing the S-estimator, which is instrumental in the computation of MM-estimators.

In this paper, we shall focus on τ-estimators, introduced in Yohai and Zamar (1988), which are defined as minimizers of the function

$$\hat{\sigma}(\beta)=\frac{s(r(\beta))^2}{nb_2}\sum_{i=1}^{n}\rho_2\left(\frac{r_i(\beta)}{s(r(\beta))}\right),\quad \beta\in\mathbb{R}^d, \tag{I.2.3}$$

where ρ_2 is a rho function and b_2 is adjusted to ρ_2 as in (I.2.2). This choice is motivated by the robustness and efficiency properties of this estimator. Indeed, τ estimators have a breakdown point $\epsilon^*=b/\rho_1(c)$; hence, for an adequate choice of the function ρ_1, the maximal breakdown point of 50% can be obtained. Similarly, by adjusting the function ρ_2, the asymptotic efficiency at the normal distribution can be made arbitrarily close to 1. However, computing τ-estimators involves solving a difficult global optimization problem. Roughly speaking, global optimization methods can be classified (Archetti and Schoen, 1984) into deterministic methods and stochastic or probabilistic methods. Deterministic methods look for a guaranteed

global optimum, while stochastic methods settle for a point that is a global optimum within an allowed margin or with a certain probability. More ambitious, deterministic methods are time consuming, and their range of applicability is limited to very specific classes of problems, or to problems that are of small size. The interested reader should refer to Agulló (2001) for a deterministic robust regression algorithm.

Most of the existing methods for computing robust estimators are stochastic and, more specifically, based on random subsampling. These methods operate by computing candidates βs based on subsamples of the observations and then starting local minimizations from each of these candidates. Therefore, they are also called multistart methods. Section I.2.2 is devoted to the local minimization aspects of computing τ-estimators. We should stress that this is the only part of our paper that is estimator-specific. The global part of our discussion is relevant to any objective function $\hat{\sigma}$, provided that a way to perform local minimizations is available. Then, in Section I.2.3, we will briefly describe clustering global optimization methods, which is a class of multistart methods that uses clustering analysis techniques (Törn and Žilinskas, 1989) in order to reduce the number of local minimizations needed to find a global minimum. Section I.2.4 discusses stopping conditions for multistart methods. Section I.2.5 shows how the existing methods for robust regression fit into the framework of clustering global optimization. In Section I.2.6, we present a few numerical tests comparing the different methods. In particular, the effectiveness of clustering techniques and stopping conditions is evaluated. We finish with our conclusions in Section I.2.7.

I.2.2 Local minimization issues

As already mentioned in the introduction, we focus on the problem of finding global minima of the function

$$\hat{\sigma}(\beta) = \frac{s(r(\beta))^2}{nb_2} \sum_{i=1}^{n} \rho_2 \left(\frac{r_i(\beta)}{s(r(\beta))} \right), \quad \beta \in \mathbb{R}^d,$$

where the *robust scale* $s(r(\beta))$ is implicitly defined by:

$$\frac{1}{n} \sum_{i=1}^{n} \rho_1(r_i(\beta)/s(r(\beta)) = b_1.$$

All the global optimization methods that we consider in this paper rely on local minimizations, and moreover, a major part of their computing time is spent in local minimizations. This is why fast and reliable local minimization algorithms are crucial. When the Hessian of the objective function is available, the most efficient local minimization algorithm is the Newton-Raphson method. Nevertheless, due to the flat parts present in the function ρ, the Hessian of the τ-objective function may contain large portions filled with zeros, and therefore, it is ill-conditioned. A good alternative for computing local minima of (I.2.3) is the *Iterated Reweighted Least Squares* (IRLS) algorithm (Salibian-Barrera et al., 2008). Although it has a slower rate of convergence, in practice it has proven to be quite efficient and stable. At

each iteration, the IRLS algorithm solves a weighted least squares problem, which is equivalent to minimizing a quadratic local approximation of the objective function. In this section, we propose to use inexact solutions of the weighted least squares problems at each iteration, and we evaluate the gain in efficiency.

The IRLS step is derived from the necessary condition for optimality:

$$g_\tau(\beta) := \frac{\partial \hat{\sigma}}{\partial \beta} = 0.$$

An expression for $g_\tau(\beta)$ has been obtained in Yohai and Zamar (1988):

$$g_\tau(\beta) = \frac{-2}{n} X' W(\beta) r(\beta).$$

Here, $W(\beta)$ denotes the diagonal matrix with entries $w_j(\beta)$, where

$$w_j(\beta) = \frac{\omega(\beta)\rho_1'(e_j(\beta)) + \rho_2'(e_j(\beta))}{e_j(\beta)}, \tag{I.2.4}$$

with the following notations:

$$e_i(\beta) = \frac{r_i(\beta)}{s(\beta)}, \quad \omega(\beta) = \frac{\sum_{i=1}^n [2\rho_2(e_i(\beta)) - \rho_2{}'(e_i(\beta))e_i(\beta)]}{\sum_{i=1}^n \rho_1'(e_i(\beta))e_i(\beta)}.$$

This leads to the matrix form of the optimality condition:

$$X'W(\beta)X\beta = X'W(\beta)y. \tag{I.2.5}$$

Disregarding the fact that the matrix W depends (nonlinearly) upon β, the system of equations (I.2.5) are the normal equations associated to the *Weighted Least Squares* problem with weights W. It is a fixed point equation, so the iterative method

$$\beta_{k+1} = (X'W(\beta_k)X)^{-1} X'W(\beta_k)y \tag{I.2.6}$$

has been proposed to solve it in Salibian-Barrera et al. (2008). These are the IRLS iterations.

For each iteration, IRLS constructs a quadratic approximation of the true objective function, where the minimum of this quadratic approximation is the subsequent iterate. The computational cost of each iteration is equivalent to the cost of computing the weights (I.2.4) and solving the system (I.2.6). Computing the weights (I.2.4) is costly because this requires the computation of an M-estimator of scale. Therefore, Salibian-Barrera et al. (2008) have proposed to replace the M-scale in (I.2.4) with an approximate value whereby the resulting iterations are known as approximated IRLS iterations. Approximated IRLS solves the $d \times d$ linear system (I.2.6) for each iteration, which corresponds to finding the minimum of a quadratic local approximation of the true objective function. However, when d is not small, the computational burden of solving a linear system for each iteration can be non-negligible. Keeping in mind that our objective is to solve (I.2.5), an approximate solution of the instrumental subproblem (I.2.6) should be enough. In fact, replacing $y = r(\beta) + X\beta$ and setting $B_k = 2X'W(\beta_k)X/n$ in (I.2.6), we obtain

$$-B_k\beta_{k+1} = -B_k\beta_k + g_\tau(\beta_k).$$

In this form, the IRLS resembles the so called *quasi-Newton methods*, whose iterations are of the form (Nocedal and Wright, 2006) $-B_k\beta_{k+1} = -B_k\beta_k + \alpha_k g_\tau(\beta_k)$, for an adequate steplength $\alpha_k > 0$.

It has been proved (Nocedal and Wright, 2006, Subsec. 5.7.1) that for quasi-Newton methods, a "good enough" direction suffices to keep convergence.

In order to evaluate the efficacy of this approach, we compared the computing time required to perform local minimizations by solving approximately (I.2.6), which will henceforth be denoted as the *Iterated Inexact Reweighted Least Squares* (IIRLS), with the computing time of the approximated IRLS iterations.

In our tests, we used the LSQR algorithm (Björck, 1996) to calculate approximated solutions to least squares problems. This algorithm can handle problems with non-square matrices, and it is stable and easily available on the Matlab environment.

In Figure I.2.1, we show the results obtained from the IIRLS. The abbreviation `IIRLS5` (respectively `IIRLS20`) stands for the algorithm performing 5 (respectively 20) iterations of the LSQR algorithm for each approximated IRLS iteration. We denoted `IARLS` as the approximated IRLS algorithm introduced in Salibian-Barrera et al. (2008) that uses approximated M-scales for computing the weights. This is subsequently used to solve (I.2.6) using a direct method.

We plot, for different values of d from 5 to 200, the average time spent by each algorithm over 500 local minimizations of the function (I.2.3) with different datasets and different starting points generated as described in Section I.2.6. In all the cases, the IIRLS algorithm found the same local minimum as IARLS, although it usually needed more iterations to converge. Nevertheless, the economy in time of performing inexact iterations compensated for the increase in the number of iterations.

We see in Figure I.2.1 that for $d = 5$, there is not a great difference among the three methods, but the difference increases with d and becomes of great importance for d in the medium and large range. Considering the fact that the quality of the results is exactly the same, it is worthwhile to use IIRLS for local minimizations.

I.2.3 Clustering methods

This section describes a technique for solving global optimization problems, such as (P), by incorporating clustering analysis techniques. The objective of clustering methods in global optimization (Törn and Žilinskas, 1989) is to identify groups of βs such that, if used as an initial point in a local minimization, every member of the same group yields the same local minimum. Thus, it would suffice to perform only one local minimization from each of these groups in order to locate all local minima, and the best of these minima would be a global minimum. A schematic explanation of clustering methods is presented in Figure I.2.2.

It consists in repeating the following steps iteratively, which we shall describe in detail in the rest of the section.

1. Sampling: sample candidates βs. Add them to the candidates sampled in previous iterations.

2. Concentration and/or selection: concentrate the sampled candidates around the minima in order to facilitate the clustering.

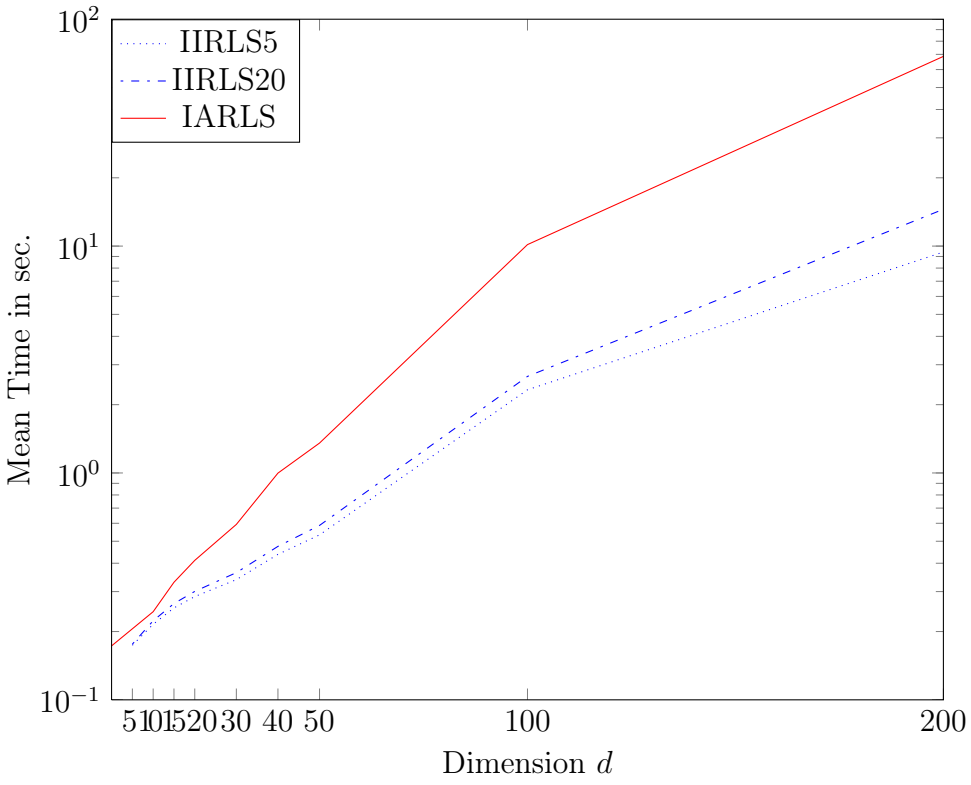

Figure I.2.1: Average times needed to find a solution to (I.2.5) with respect to d, using approximated (`IARLS`) and inexact (`IIRLS5` and `IIRLS20`) iterations.

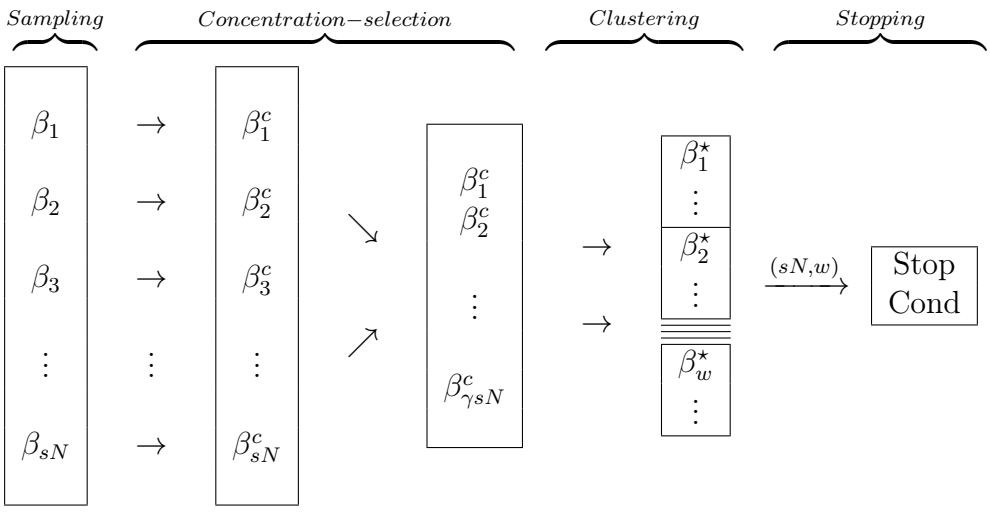

Figure I.2.2: Schematic representation of clustering methods, at the sth iteration.

3. Clustering: identify groups of candidates suspected to converge towards the same minimum.

4. Stopping condition: decide whether it is worthwhile to continue, taking into account the outcome of the local minimizations. If so, go back to (1), otherwise

stop.

Let us describe in detail the first three steps: sampling, concentration/selection and clustering. For clarity of the exposition, stopping conditions will be discussed separately in the next section.

I.2.3.1 Sampling

In absence of additional information, candidates are usually sampled uniformly in the feasible region. However, when the feasible region is unbounded, it is not completely clear how the sampling should be done.

For the particular case of robust regression, Ruppert (1992) proposed a method known as *random subsampling*: candidates $\beta_i, i = 1, ..., N$, are constructed by drawing random subsamples of size $h \geq d$ from the data and letting β_i be the least squares fit to the ith subsample. The rationale behind this method is that for a large enough N, at least one outlier-free subsample could be sampled, which should give a good candidate, hopefully in the neighborhood of a global minimum.

I.2.3.2 Concentration and/or selection

The concentration step consists of performing some iterations of a local minimization procedure, usually one, starting from each candidate. In the selection step, a prespecified fraction of the sampled candidates with lowest function value is retained. It has been proposed (Törn and Žilinskas, 1989) to do only the concentration step, only the selection step or both. For robust regression, Ruppert (1992) proposed to do selection and then concentration, while Salibian-Barrera et al. (2008) explored the use of concentration followed by selection. The term "concentration step" has already been used in Rousseeuw and Driessen (1998) to denote a particular local minimization procedure for the LTS estimator. In the sequel, we will use this term in the broader context described previously.

I.2.3.3 Clustering

Many ways to perform clustering for global optimization have been proposed (Törn and Žilinskas, 1989). Here, we review only Single Linkage, which is the simplest method (Rinnooy Kan and Timmer, 1987a).

Single linkage

At each iteration k, compute the radius $r_k = \frac{1}{\sqrt{\pi}} \left(\Gamma(1 + d/2) \frac{\xi \ln(kN)}{kN} \right)^{1/d}$, for some $\xi > 0$. Where Γ denotes the gamma function.
The single linkage algorithm consists in iterating the following three steps until all points have been assigned to a cluster

1. Choose a local minimum to be used as seed.

2. Initialize the cluster with the seed.

3. Grow cluster : given a partially constructed cluster,
   iterate :

   (a) find the unclustered point closest to the cluster.

   (b) if this point is within distance r_k from the cluster, add it to the cluster and repeat (3a). Otherwise, go back to step (1) and start the next cluster.

The theoretical convergence properties of Single Linkage have been proved to minimize a function over a bounded set S, supposing that the initial candidates have been sampled uniformly over S (and that no concentration step has been performed). The proof proceeds by estimating the probability that a local minimization is started from a point a at iteration k. This probability is bounded by the probability that there exists another candidate within distance r_k with a lower function value. This is because if the ball with center a and radius r_k contains a candidate z with lower function value, then if z is assigned to a cluster, a will be assigned to the same cluster. Moreover, if in step (1) the local minimizations are performed by first considering the candidates with lower function values, then we will not apply a local minimization to a before z is assigned to a cluster.

Therefore, the choice

$$r_k = \frac{1}{\sqrt{\pi}} \left(\Gamma\left(1 + \frac{d}{2}\right) \xi \text{Volume}(S) \frac{\ln(kN)}{kN} \right)^{1/d}, \ \xi > 0,$$

makes the probability of applying a local minimization to any candidate decrease with k, for any $0 < \eta < 1/2$, as $O(k^{1-\eta\xi})$. See Rinnooy Kan and Timmer (1987a) for the details.

In the interest of improving the effectiveness of the clustering method, we take into account the following facts that could be detrimental to the clustering method:

In Kaufman and Rousseeuw (1990, Ch.5), various techniques of agglomerative clustering are examined. The authors alert about the chaining effect of single linkage, which makes the clusters stick to each other because of the formation of chains, and argue, based on theoretical analysis and practice, that some other techniques such as *complete linkage* or *average linkage* would be better suited for most problems. We have incorporated this insight into our numerical experiments, despite the fact that the previous theoretical analysis for adjusting the radius r_k at each iteration does not carry over to the case where we consider the maximum distance to the cluster (complete linkage) or the average distance to the cluster (average linkage). In both cases the existence of a point within radius r with a lower function value does not ensure that a local minimization will not be started, unless there is a relationship between r and the (unknown) diameter of the cluster.

In Beyer et al. (1999), it had been pointed out that, for points sampled from a broad set of distributions, the distance of any of this points to its nearest neighbor becomes very close to the distance to the farthest point as the dimension increases. This essentially means that the notion of nearest neighbor loses much of its meaning in high dimension. Later on, in Aggarwal et al. (2001), the role played in this

phenomenon by the norm used to measure distances was devealed. It was shown that, in expectation as $d \to \infty$, the gap goes to 0 for the ∞-norm, tends to a constant for the Euclidean norm, and goes to ∞ for the 1-norm.

In our numerical tests in Section I.2.6 we compare the actual impact of these two factors on the performance of clustering.

I.2.4 A stopping condition

A crucial issue in global optimization algorithms is the tradeoff between solution accuracy and computing time. In practice, this dilemma is the decision about when to stop.

When using random subsampling (Maronna et al., 2006, Sec 5.7.2), the minimization (P) over the whole $\mathbb{R}^d$ is reduced to a search over a finite set of candidates generated from subsamples. For the probability of picking one outlier-free subsample from a sample with a fraction ε of contamination to be greater than $1 - \delta$, we need to sample a number N of candidates such that:

$$N \geq \frac{|log(\delta)|}{|log(1 - (1 - \varepsilon)^d)|}. \tag{I.2.7}$$

Unfortunately, this approach has some drawbacks. First, the number N in (I.2.7) grows exponentially with d. For instance, if $\delta = 0.01$ and $\varepsilon = 0.25$, we should sample 1450 candidates for $d = 20$, 25786 for $d = 30$ and more than 80 millions for $d = 50$. Furthermore, it depends on the fraction ε of contamination, which is not known in advance.

The main disadvantage of this criterion is its rigidity; it is an *a priori* criterion. Information about actually sampled candidates is completely disregarded. Thus, the algorithm will continue in the same way after 10 local minimizations as if it has found 8 local minima or only 1.

Adaptive criteria try to estimate the fraction of the search region that has been actually explored, using the observed information about the structure of each particular problem. Then, for a given level of accuracy, the running time of the algorithm will depend on the problem complexity, expressed mainly through the number of local minima that are found given a number of local minimizations.

In the sequel, we will describe an approach to this problem that uses Bayes' theorem to incorporate information gathered during the optimization process in order to decide when to stop. It was introduced in Boender and Rinnooy Kan (1987) and refined in Piccioni and Ramponi (1990). The framework is the following: a sequential sample is drawn from a multinomial distribution with an unknown number of cells and unknown cell probabilities. In our context, each cell will be associated with one minimum of problem (P) and will be filled with the subsamples whose candidates converge after a local minimization to the minimum associated with this cell. The cell probabilities will be the fraction of subsamples in the cell.

Let us consider a small illustrative example. In Figure I.2.3, we elucidate a case with $n = 20$ and $d = 4$ where the objective function in (P) has 4 local minima $\beta_1^\star, \beta_2^\star, \beta_3^\star$ and $\beta_4^\star$. Seven subsamples have been drawn, thus yielding by least squares

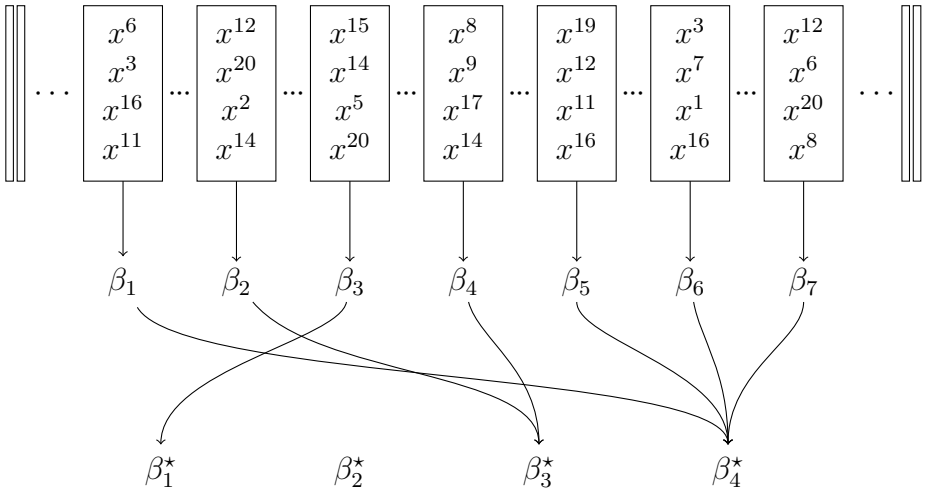

Figure I.2.3: An example of subsampling with $n = 20$ and $d = 4$. Seven subsamples have been sampled among the 4845 possible subsamples, coming from three of the four cells.

seven candidates $\beta_1, ..., \beta_7$. If local minimizations were started from these candidates, β_3 would converge towards $\beta_1^\star$. β_2 and β_4 would converge to $\beta_3^\star$. Candidates $\beta_1, \beta_5, \beta_6$ and β_7 would converge to $\beta_4^\star$, and the minimum $\beta_2^\star$ would remain undiscovered. Thus, we have observed 3 cells associated with the minima $\beta_1^\star, \beta_3^\star$ and $\beta_4^\star$, with observed frequencies of 1, 2 and 4.
In general, let $\beta_1^\star, \beta_2^\star, ..., \beta_k^\star$, be the local minima. Let ϑ_i, $i = 1, ..., k$, denote the probabilities of each cell. In the Bayesian approach, the unknowns $k, \vartheta_1, ..., \vartheta_k$ are supposed to be themselves random variables $K, \Theta_1, ..., \Theta_k$ with realizations $k, \vartheta_1, ..., \vartheta_k$ for which *a priori* distributions can be specified. Given the outcome $(m_1, ..., m_w)$ of a number of local searches, Bayes theorem is used to compute an *a posteriori* distribution. Following the approach of Piccioni and Ramponi (1990), we will suppose that different minima have different function values, $\hat{\sigma}(\beta_i^\star) \neq \hat{\sigma}(\beta_j^\star)$ for $i \neq j$. In such case, the minima can be ordered according to their function values $\hat{\sigma}(\beta_1^\star) < \hat{\sigma}(\beta_2^\star) < ... < \hat{\sigma}(\beta_k^\star)$. The same will be done with the minima found in experiments. Hence, we can compute the statistics $O_l = M_{J_l}, l = 1, ..., W_m$, where M_i denotes the observed frequency of the ith minimum ($M_1 = 1$, $M_2 = 0$, $M_3 = 2$ and $M_4 = 4$ in our example), J_l the index of the lth observed minimum (in the example $J_1 = 1$, $J_2 = 3$ and $J_3 = 4$) and W_m the number of (different) observed minima after sampling m candidates. In simple words, O_l is the frequency of the lth *observed* minimum.

A particularly interesting quantity from the stopping criteria viewpoint is $H = J_1 - 1$, which denotes the number of undiscovered local minima *with better function value than the observed minima.* Supposing *a priori* that the number of cells K follows an improper uniform discrete distribution on $[1, \infty)$, and that given $K = k$, the cell probabilities $\Theta_1, ..., \Theta_k$ are jointly uniformly distributed on the $k - 1$ unit simplex, the following conditional probability for H can be obtained, when $m \geq 2$

and $w \leq m-2$ (Piccioni and Ramponi, 1990, Corollary 2):

$$\mathbb{P}(H = h | W_m = w, O_l = m_l, {}_{l=1,\ldots,w}) = (m-w-1)\frac{(m-2)!(w+h-1)!}{(w-1)!(m+h-1)!}.$$

In particular, the probability that the global minimum has been already discovered is

$$\mathbb{P}_{m,w} := \mathbb{P}(H = 0 | W_m = w) = \frac{m-w-1}{m-1}, \qquad w \leq m-2. \tag{I.2.8}$$

Using (I.2.8) we can readily devise a stopping condition, namely, to stop when the probability of having found the global minimum reaches a prespecified threshold.

The probability $\mathbb{P}_{m,w}$ is undefined when $w = m-1$ or when $w = m$. However, in practice, this only occurs during the first two or three iterations. In this case, because we do not have evidence that the search region has been well explored, we should keep the algorithm running.

For example, let us consider the robust regression problem (P) for the stackloss dataset (Maronna et al., 2006, pp. 381). The Multistart method consist of sampling candidates βs and starting a local minimization from each of them, until the probability (I.2.8) reaches a given threshold. Let us consider the thresholds 0.3, 0.6 and 0.9. The first threshold 0.3 is reached after 4 local minimizations, which give 2 different local minima, none of which is the global one. The threshold 0.6 is reached after 9 local minimizations and 3 minima, one of which is the global minimum. Finally, the threshold 0.9 is reached after 42 local minimizations, producing 4 local minima.

Of course, since it is a random algorithm, its running is unlikely to be the same every time. What should be retained is that higher thresholds give more accurate results in the sense that the search region is more exhaustively explored, and this is done adaptively. Needless to say, a more exhaustive search will take longer than a rougher one. In general, there is not an easy way to guess how long will it take to solve problem (P) within a given accuracy, but one can always impose a time limit, and the value (I.2.8) can be given as information to the user at the end.

We would like to stress the fact that this approach works for any algorithm based on subsampling and local searches, even if the objective is to compute other estimators, such as LTS, or beyond the context of linear regression, such as the location and scatter estimation problem.

Stopping conditions based on (I.2.8) can also be used for algorithms described in Section I.2.3 that try to foresee the result of a local minimization, and to avoid it if it is likely to re-discover an existing minimum. In that case, in (I.2.8) m will be the number of sampled candidates and not the number of local searches actually carried out. The reason is that clustering methods are supposed to give the same outcome one could have obtained by starting local searches from each candidate. Nevertheless, the precision of the stopping condition will be subordinated to that of the clustering method; if it alters the outcome of the algorithm, the current *a posteriori* probability will be updated with wrong information.

I.2.5 A clustering global optimization point of view of robust regression estimation

We have already mentioned that most existing methods for computing robust regression estimators can be described as clustering methods, as described in Section I.2.3. In this section we shall describe them, and we will see how they perform each of the steps discussed in Section I.2.3.

I.2.5.1 Random resampling

The original version of random resampling was introduced by Rousseeuw in Rousseeuw (1984) for computing the least median of squares estimator, and it was further refined and adapted for S-estimators by Ruppert in Ruppert (1992). Rousseeuw's original version introduced the sampling technique used by most existing algorithms, and his algorithm consisted only of sampling and choosing the candidate with the least scale. The modification of Ruppert, illustrated in Figure I.2.4, included selection and local minimization applied, without clustering, to the best candidates. The details of each step are described below.

Given parameters N, t, do once:

- Sampling: Use Rousseeuw's random subsampling.
- Concentration-Selection: No concentration step is performed, but a selection of the t best candidates is done.
- Clustering: No clustering is performed. A local minimization is started from each candidate until convergence.
- Stopping condition: There is no stopping criterion. The number of sampled candidates is fixed in advance.

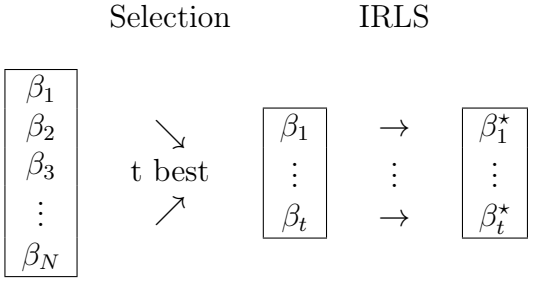

Figure I.2.4: Schematic representation of Ruppert's version of random resampling.

I.2.5.2 Fast-S, Fast-τ

The "Fast-S" algorithm was presented in Salibian-Barrera and Yohai (2006) for S-estimators, and it was modified to cope with τ-estimators in Salibian-Barrera et al.

(2008). This modification was called "Fast-τ". From the global optimization viewpoint, it extended Random Resampling by adding a concentration step. As for random resampling, we give below a schematic illustration and a detailed description of each step. Given parameters N, ς and t, iterate the following steps:

- Sampling: Use Rousseeuw's random resampling.
- Concentration: Apply ς steps of IRLS to each candidate. Select the t candidates with best objective function.
- Clustering: No clustering is performed.
- Stopping condition: There is no stopping condition; the number of sampled candidates is fixed in advance.

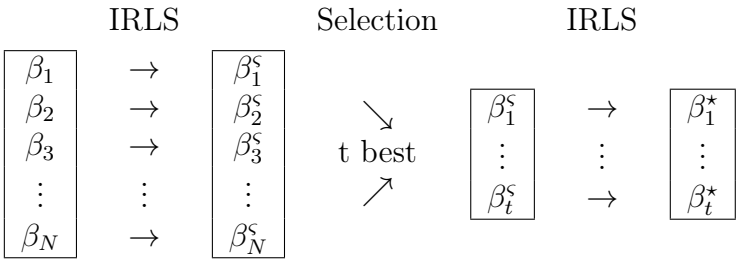

Figure I.2.5: Schematic representation of the Fast-S and Fast-τ methods.

I.2.5.3 Fast-τ with stopping condition.

As previously observed, the considered algorithms for computing robust regression estimators are all particular instances of the general clustering procedure depicted in Figure I.2.2.

The first one, random resampling, consists only of sampling and selection, Fast-τ adds a concentration step. Unlike clustering, the addition of an adequate stopping condition does not add computational work. For this reason, we introduce here a modification of Fast-τ for including stopping criteria, without doing clustering. It simply executes Fast-τ iteratively, and at the end of each iteration, it adds the (eventually new) minima found on the list of minima encountered in previous iterations, and evaluates a stopping condition. In our case, we require the probability $\mathbb{P}_{kN,w}$ defined by (I.2.8) to exceed a given threshold θ.

Our proposition is summarized in Figure I.2.6. For given parameters N, ς and t, and a probability of success θ,

Note that under this form, the modified version of Fast-τ does not exactly fit in the framework illustrated in Fig I.2.2 because the selection is performed considering only the candidates sampled in the last iteration and disregarding the ones from previous iterations.

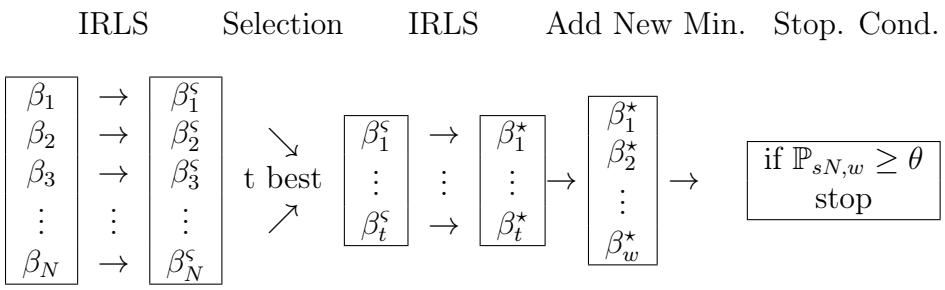

Figure I.2.6: The proposed modification of Fast-τ including a stopping condition.

I.2.6 Numerical tests

We conducted numerical experiments in order to investigate the impact of clustering techniques and stopping criteria. As in Ruppert (1992) and Salibian-Barrera et al. (2008), we consider simulated data contaminated with clustered outliers, where contamination is highly concentrated around outlying values.

- $(1-\delta)100\%$ of the points follow the regression model $y = X\beta + \varepsilon$, where the $d-1$ covariates are distributed as $N_{d-1}(0, I_{d-1})$, $x_{id} = 1$ is the intercept, $\varepsilon_i \sim N(0,1)$ and $\beta = 0$.
- $\delta 100\%$ of the points are "bad" high-leverage points: the covariates now follow a $N_{d-1}(s, 0.1^2 I_{d-1})$ distribution where $s = (100, 0, ..., 0)^t$, and the response variables $y_i \sim N(100l, 0.1^2)$. In the following we will call the parameter l the "contamination slope".

For evaluating the stopping condition, we tested only Multistart, and we compare it with Fast-τ using $N = 50$ samples. For performance evaluations, we tested all the algorithms described in Section I.2.5, except for Random Resampling (Subsection I.2.5.1). The reason for excluding Random Resampling from those tests is that we included the Fast-τ algorithm, and the numerical tests reported in Salibian-Barrera et al. (2008) already compared Fast-τ with Random Resampling, among others, and showed that it outperforms Random Resampling.

The clustering methods were implemented by the author in Matlab and are available upon request. In our test, it was always executed with the same parameters: at each iteration, $N = 100$ candidates are sampled and added to the sample, then a concentration step consisting of one IRLS iteration (cf. Section I.2.2) and a selection of the best 10% of the concentrated candidates is performed (see Figure I.2.2). Concerning the radius used for the clustering, as the minimization in (P) is done over the unbounded domain $\mathbb{R}^d$, we used the formula $r_k = \pi^{-1/2} \left(\Gamma(1 + d/2)\xi \ln(kN)/(kN)\right)^{1/d}$, for a predefined $\xi > 0$. After extensive experiments, we realized that the results are rather insensitive to the choice of the parameter ξ. We set $\xi = 40$ in our tests. For the tests involving the Fast-τ algorithm, we used the code available from the webpage of Matias Salibian-Barrera. We used the parameters $\varsigma = 2$ and $t = 5$ (see Figure I.2.5). The number N of initial candidates changed from test to test and is indicated each time.

Both Fast-τ and Single Linkage clustering were fitted with a stopping condition. In all cases, it consisted in stopping when the probability $\mathbb{P}_{m,w}$ defined in (I.2.8) reached a given threshold θ.

Concerning the performance in terms of computing time, for each algorithm we saved the required time T_A. Then, we computed the ratio $RT = T_A/T_R$, where T_R is the time required by an algorithm used as reference. We usually used Fast-τ as reference, with different parameters depending on the test. This ratio is what we call the relative computing time. In this way, the results are machine-independent; and easier to compare.

As it is not feasible to know certainly if the returned solution is the global minimum, we only give the relative τ-scale with respect to a reference algorithm, $\hat{\sigma}_A/\hat{\sigma}_R$, where $\hat{\sigma}_A$ and $\hat{\sigma}_R$ are the values of the objective (I.2.3). We call this ratio "relative τ-scale". Similarly to the results reported in Salibian-Barrera and Yohai (2006) for S-estimators, for the particular type of contamination that we are considering, often the coefficient $\hat{\beta}_1$ over the 500 samples forms two well-separated groups: one highly concentrated around the contamination slope, and another one more dispersed around 0, the slope of the data without contamination, hereafter referred to as "clean slope". In those cases, we also show the percentage of samples for which the "slope" is around 0. Note that it is not uncommon that minima with the contamination slope had better function values (I.2.3) than minima around the "clean" slope, especially for those with high proportions of outliers.

In Subsection I.2.6.1 we evaluate the impact of the stopping condition. We discuss in detail how it behaves when applied to Multistart with threshold values ranging in a wide range. Our second and third test, presented in Subsection I.2.6.2 and I.2.6.3 , evaluate the performance of the considered algorithms in two different kinds of situations. In I.2.6.2, this is done for a small problem, whose complexity varied only through the change in the dimension d. In the third test, presented in Subsection I.2.6.3, we compare the performance of the considered algorithms, the parameter used to control the complexity of the problems was the contamination slope l.

I.2.6.1 Analysis of the stopping condition

The objective of this Subsection is to scrutinize the impact of the stopping condition presented in Section I.2.4 on the algorithms in Section I.2.5. The datasets were generated as indicated at the beginning of Section I.2.6, with $n = 400$ observations in dimension $d = 15$. The contamination fraction was $\delta = 40\%$ forming two groups of 20% each with contamination slopes $l = 2$ and $l = 4$. The objective of using two groups of outliers was to increase the number of local minima, which would better illustrate the effectiveness of the stopping condition, since the complexity of the problems heavily depends on the number of local minima. The algorithms compared were Multistart (MS), which consist of sampling candidates and launching a local minimization from each of them; it is the same as the Random Resampling algorithm (cf. Section I.2.5.1) without selection. and Fast-τ with $N = 50$ samples (FT50). Multistart stopped as soon as the probability $\mathbf{P}_{m,w}$ in (I.2.8) reached a given threshold θ. In this section, we used the thresholds $0.3, 0.6, 0.9$ and 0.95.

In Table I.2.1 we show the following:

- The number of local minima, w.
- The quantity of sampled candidates, m.
- The effective value of the probability (I.2.8) at the termination of the algorithm, $\mathbf{P}_{m,w}$.
- The percentage of samples for which the estimate given by the algorithm had the clean slope, around 0, at the row "% Slope ".
- The relative running time with respect to FT50, T_{MS}/T_{FT50}.
- The relative τ-scale with respect to FT50, $\hat{\sigma}_{MS}/\hat{\sigma}_{FT50}$.

All the entries, except for the clean slope, are the average over 500 simulations.

Table I.2.1: Details of the execution of Multistart (MS) and Fast-τ with 50 samples (FT50). The datasets consisted of $n = 400$ observations in dimension $d = 15$, with a fraction of contamination of $\delta = 40\%$.

	MS				FT50
θ	0.3	0.6	0.9	0.95	
w	1.2	1.2	1.7	2.4	2.2
m	3.19	4.69	19.08	50.24	50
$\mathbf{P}_{m,w}$	0.47	0.66	0.9	0.95	0.96
% Slope	95.4	94.6	86.4	77.2	68.6
T_{MS}/T_{FT50}	0.17	0.24	0.96	2.52	1
$\hat{\sigma}_{MS}/\hat{\sigma}_{FT50}$	1.012	1.0116	1.0083	1.0045	1

By examinating the column corresponding to FT50 in Table I.2.1 we see that, in this example with few local minima, most of the time the effective value of (I.2.8) is already around 0.96; if the function used to have 5 local minima, it would be around 0.9 and it would be around 0.8 if the function had 10 local minima. The reason is that candidates are sampled in batches; thus, if each batch is of size N, the only attainable threshold values are of the form $(N - w - 1)/(N - 1)$, for positive integers w. Since in MS, candidates are sampled one by one, it stops as soon as the threshold is reached so it can be combined with low threshold values. On the contrary, algorithms that do some kind of selection need to sample in batches; for performing local searches, only from promising candidates. In the case of algorithms performing clustering, a harsh selection is needed in order to well separate clusters and prevent sticking to each other, thus the use of small batches is discouraged. As a consequence, for all of the algorithms of Section I.2.5, if the objective function does not have many local minima, only relatively high threshold values will be observed in practice.

However, MS illustrates quite well how the stopping condition works, since we see how the number of discovered local minima and the number of sampled candidates increases as the threshold increases. Even if the percentage of datasets for which the estimate had the clean slope seems to indicate the contrary, the accuracy of the result also improved. This can be seen by examining the relative τ-scale. We see that MS gives worse results than FT50 for low threshold values, but this ratio decreases for higher threshold values, showing the better performance of MS.

I.2.6.2 Many dimensions, many thresholds on a small problem

In our second test, we set the contamination fraction to $\delta = 0.2$ and the contamination slope to $l = 2$. For each $d \in \{5, 10, 15, 20\}$, we generated 500 datasets of $n = 100$ points. We compare four algorithms:

- Fast-τ with $N = 500$ candidates (FT500).
- Fast-τ with $N = 250$ candidates (FT250).
- Fast-τ with a stopping condition (SC), described in subsection I.2.5.3. The parameters are (see Figure I.2.6) $N = 100$, $\varsigma = 1$ and $t = 5$.
- Single Linkage clustering as described in Subsection I.2.3.3 (SL), with the parameters indicated at the beginning of this Section. Motivated by the observation at the end of Section I.2.4, we tested Single Linkage using the ℓ_1 and the Euclidean norm. They are denoted as SL1 and SL2, respectively.

For those algorithms incorporating a stopping condition (SC and SL), the threshold parameter θ took the values 0.95, 0.97 and 0.99. These results are shown in Table I.2.2 for the algorithms with stopping condition, and in table I.2.3 for Fast-τ with $N = 250$ and $N = 1500$ samples.

For algorithms with stopping condition, a curious situation appears, as the algorithms performing clustering always perform worse than those without clustering, but in low dimension, they find quite often a better solution than the optimal one, in the sense that they find a non-global minimum with the clean slope. As the dimension increases, however, their performance degrades both in terms of objective function and in the percentage of times that they find the clean slope. Single Linkage using the ℓ_1 norm generally had better function values than using the Euclidean norm, but the difference is negligible. In preliminary tests we also tried Complete Linkage and Average Linkage clustering, but the results were essentially identical, so we only used Single Linkage.

Overall, the Fast-τ algorithm with stopping condition using thresholds 0.95 and 0.97 achieves a good tradeoff between computing time and quality of the solution, as it gives results almost as good as FT500 or even FT1500, but within a fraction of time. By raising the threshold to 0.99, the results are similar to those of FT1500, but with a larger computing time.

Table I.2.2: Quality of the solution and computing effort for SC, SL1 and SL2. The parameters of the contamination were $l = 2$ and $\delta = 0.2$. The relative time and the relative τ-scale are with respect to FT500

		% Clean Slope			Relative τ-scale			Relative Time		
d		95%	97%	99%	95%	97%	99%	95%	97%	99%
	SC	58	58	58.2	1	1	1	0.34	0.35	0.74
5	SL1	88	88	88.2	1.019	1.019	1.02	0.15	0.16	0.49
	SL2	88.4	88.4	88.4	1.021	1.021	1.021	0.1	0.1	0.29
	SC	64.4	64.4	64.6	1	1	1	0.466	0.51	1.44
10	SL1	78.8	78.8	85.8	1.025	1.025	1.020	0.21	0.23	0.63
	SL2	78	78	85	1.024	1.024	1.021	0.12	0.13	0.33
	SC	68.8	69.2	69.8	1.003	1.002	0.999	0.48	0.64	2.45
15	SL1	32.8	33	52.4	1.104	1.103	1.069	0.23	0.24	0.65
	SL2	36.4	36.4	53.8	1.090	1.090	1.064	0.14	0.15	0.4
	SC	64.8	64.8	65.2	1.016	1.014	0.998	0.51	0.72	3.46
20	SL1	7.6	7.8	19.8	1.193	1.192	1.149	0.25	0.26	0.62
	SL2	8.4	8.4	14.8	1.206	1.206	1.192	0.18	0.18	0.40

Table I.2.3: Quality of the solution and computing effort for FT250 and FT1500. The parameters of the contamination were $l = 2$ and $\delta = 0.2$. The relative time and the relative τ-scale are with respect to FT500

d		% Clean Slope	Relative τ-scale	Relative Time
	FT250	58.2	1.000	0.57
5	FT1500	58.4	1.000	2.8
	FT250	64.8	1.000	0.62
10	FT1500	64.6	0.999	2.63
	FT250	71.4	1.001	0.64
15	FT1500	72.2	0.998	2.57
	FT250	65.4	1.007	0.65
20	FT1500	69	0.996	2.6

I.2.6.3 Varying complexity for a fixed dimension

In the next test, we examine the behavior of the algorithms for a fixed dimension, but also for various proportions of outliers and contamination slopes. The closer the contamination slope is to the clean slope, the more difficult it becomes to identify the clean one.

We tested FT250, FT500, FT1500, SL and SC in dimension $d = 10$ with different contamination slopes. The number of observations was fixed to $n = 400$, and the proportion of outliers was $\delta = 10\%, 15\%$ and 20%. For SL and SC, the stopping condition uses the threshold $\theta = 0.95$; because the preliminary test did not show significative improvements when we raised the threshold in these problems. Similarly to the previous test, the use of different variants of clustering did not significantly change the results; therefore we only tested Single Linkage clustering, with the usual Euclidean norm.

The performance was measured as in the previous test. Namely, we compare the objective function value with respect to a reference algorithm, and we keep the percentage of the samples from which convergence occurred to a minimum with a slope around 0. These two quantities were the measure of quality of the solution. We also record the time relative to the time spent by FT500, as explained in the previous section.

The quality of the solutions obtained by Fast-τ was identical to SC; thus, we do not include it in the tables. The results of these test are in Table I.2.4.

Table I.2.4: Percentage of samples where the minimum found had the clean slope, and time spent relative to FT500. For $d = 10$.

δ	10%				15%				20%			
Slope	1.1	1.4	1.7	2	1.1	1.4	1.7	2	1.1	1.4	1.7	2
	% Clean Slope											
SC	62	100	100	100	0	26.4	98	100	0	0	3.2	61.8
SL	99	100	100	100	44.2	94.6	100	100	2.8	21.4	77.4	96.4
	Relative Time											
SC	0.34	0.33	0.34	0.35	0.37	0.38	0.40	0.39	0.39	0.41	0.44	0.39
SL	0.18	0.21	0.20	0.20	0.22	0.17	0.20	0.21	0.24	0.21	0.19	0.14
FT250	0.57	0.59	0.59	0.60	0.60	0.59	0.59	0.61	0.61	0.62	0.61	0.57
FT1500	2.71	2.70	2.69	2.68	2.58	2.69	2.62	2.57	2.57	2.56	2.60	2.78
	Relative τ-scale											
SL	1.01	1	1	1	1.07	1.04	1	1	1.01	1.04	1.07	1.01

As in the previous test, the algorithm with clustering has a very good running time, about 20% of the time spent by Fast-τ, and in some cases, it gives a worse solution to the minimization problem (P) than algorithms without clustering. Thus, it does not compute the τ-estimator, but the result it gives has the clean slope. We do not have a clear explanation for that, and we think this phenomenon bears closer examination; in another article. We see once again that the stopping condition permits to find the global minimum with a high probability in a proper computing

time.

I.2.7 Conclusions and future work

We have investigated the effectiveness of the usage of clustering techniques and stopping conditions for global optimization in the particular case of robust regression. Our viewpoint is that of an user of robust regression who wants to compute robust estimators without having to adjust parameters that depend upon the details of the chosen algorithm.

The integration of a stopping condition is completely justified both by the quality of the results and by the performance in terms of computing time. Additionally, it is very simple to implement in new and existing software. It should be incorporated in algorithms not only for computing τ-estimators, but also in any algorithm based on subsampling and concentrations steps.

A threshold between 0.95 and 0.97 achieves a good compromise between efficiency and quality of the results. A higher threshold value ensures a very good solution; at an extra cost in terms of computing time. For routine utilization, a threshold of 0.96 or 0.97 should give good results at a competitive computing time, which will adapt by itself to the complexity of the problem. Additionally, it possesses a very appealing interpretation in terms of probability of finding the global optimum, independently of the particular problem under consideration.

In their present states, the existing clustering techniques for global optimization do not seem to fit the needs of robust estimation problems. However, as shown in Section I.2.5, they are a natural extension of the existing methods, and their computing times and behavior in some difficult problems suggest that they deserve further investigation.

Chapter I.3

High dimensional issues in clustering (global optimization)

I.3.1 Introduction

While evaluating the use of clustering techniques to improve existing global optimizations methods particularly suited for robust regression problems (Flores, 2010), we were very disappointed by the poor performance of some clustering techniques in dimension as moderate as 10 or 15. Trying to understand the breakdown of (single linkage) clustering, we learned that the same problem had been highlighted by computer science researchers working on high dimensional indexing (Beyer et al., 1999; Aggarwal et al., 2001), and by people working with other clustering algorithms (Schoen, 1999). In fact, the common point between clustering and indexing is the *nearest neighbor* subproblem they rely on. Therefore, the problem is unlikely to be restricted to single linkage clustering as, quoting from Locatelli and Schoen (1999, pp. 380),

« *All of the papers dealing with clustering methods base this decision on some sort of nearest neighbor statistics - that is, the decision of starting a local search is in some way connected with the fact that no point in a properly defined subset of the sample is too near to the current one*».

The phenomenon described in Beyer et al. (1999); Aggarwal et al. (2001) is: under rather mild hypothesis, the distance to the nearest point approaches the distance to the farthest point as the dimension increases. This essentially means that the notion of nearest neighbor loses much of its meaning in middling and high dimension.

The objective of this chapter is to go into the consequences of the blurring of the notion of nearest neighbor for clustering algorithms in high dimension, and to propose amendments permitting to uphold the efficacy of clustering algorithms as the dimension increases.

I.3.1.1 Clustering global optimization

We focus on the continuous global optimization problem,

$$\min_{\beta \in \Omega} \sigma(\beta), \tag{I.3.1}$$

where $\Omega \subseteq \mathbb{R}^d$ is a compact set, and $\sigma : \Omega \to \mathbb{R}$ is a continuous function. We will assume that a local minimization procedure is available.

Algorithm 1 Multilevel Single Linkage (MLSL)

Choose $N > 0$, $\xi > 0$ and $\gamma \in (0, 1]$. Set $k := 0$, $\mathcal{X} = \emptyset$ and iterate:

1. Let $k := k + 1$;
2. let

$$r = r_{k,N,\xi} = \pi^{-1/2} \left(\Gamma \left(1 + \frac{d}{2} \right) \text{vol}(\Omega) \xi \frac{log(kN)}{kN} \right)^{1/d}; \tag{I.3.2}$$

3. generate a uniform random sample of size N in Ω, add it to $\mathcal{X}$;
4. sort $\mathcal{X}$ by increasing function value, and select the γkN best of them, call it $\mathcal{X}_k$;
5. launch a local search algorithm from each point in $\mathcal{X}_k$ except if
   - the distance of the point from the boundary is less than a fixed threshold.
   - the point is "too near" from a critical point

   Or,
   - if there exists another point with better function value at a distance which is less than or equal to the radius r_k;
6. Check if some stopping condition is satisfied, otherwise repeat from step 1.

Rinnooy Kan and Timmer (1987a,b) introduced a stochastic global optimization method for solving (I.3.1) using clustering techniques. It is an improvement of the multistart (MS) method, which proceeds by sampling points uniformly on Ω, followed by a local search from each of the sampled points. The MS method has the evident disadvantage of potentially wasting time by launching several local searches yielding the same minimum. The method proposed in the cited articles, called multi-level single linkage (MLSL, Algorithm 1), aimed at overcoming this drawback by avoiding the local search if it is likely to find a minimum already known. MLSL enjoys the following asymptotic properties as the number of iterations goes to infinity:

P1. The best observed value converges to the global optimum with probability 1.

P2. The probability of starting a local search decreases to 0.

P3. The algorithm performs a finite number of local searches with probability 1, even if it runs forever.

Despite its strong theoretical properties, the necessity of revising at each iteration the decisions about local searches and some other technical details left some room for improving MLSL. This was done by Locatelli and Schoen (1996), where the authors propose a family of algorithms for global optimization satisfying properties $P1$, $P2$ and $P3$ without the computational overhead of MLSL. In a later article (Locatelli and Schoen, 1999), a particular member of this family, called simple linkage (SL, Algorithm 2) was analyzed in detail and it was shown to make almost the same decisions as MLSL.

Algorithm 2 Simple linkage (SL)

Choose $\xi > 0$ and $\varepsilon > 0$. Set $k = 1$, generate β_1 uniformly and iterate:

1. Let $k = k + 1$;
2. let r_k be defined as in (I.3.2), with N=1;
3. generate a single uniform random point β_k;
4. launch a local search from β_k except if

$$\exists j < k,\ \|\beta_j - \beta_k\|_2 \leq r_k :\ \sigma(\beta_j) \leq \sigma(\beta_k) + \varepsilon; \qquad \text{(I.3.3)}$$

5. check stopping condition, if it is not met, repeat from step 1.

In Locatelli and Schoen (1999) it was also proved that SL enjoys property P2 if and only if $\lim_{k\to\infty} k^{1/d} r_k = \infty$, and that if $\xi > 2^d/2d$ then P3 holds as well.

A key difference between MLSL and SL is that MLSL never starts local searches from points within a prescribed distance from the boundary, while SL allows local searches to be started from any point in the feasible domain. This difference is particularly important in high dimension. For instance, given a threshold distance ν, the probability of sampling from a uniform distribution within distance ν from the boundary of the unit cube is $1 - (1 - 2\nu)^d$. Moreover, if local searches from points at a distance from the boundary less than a fixed threshold ν are inhibited in SL, then property P3 holds if $\xi > 1$. Even if the practical interest of property P3 is not clear, this is a first evidence that if radius $r_{k,\xi}$ is used, the parameter ξ should vary with the dimension d.

Shortly after, Schoen (1999) pointed out some problems affecting clustering global optimization algorithms in high dimensions, and proposed some solutions. At that time there was already some evidence indicating that the use of formula (I.3.2) forced algorithms to behave much like best start (BS), a quite inefficient random sampling method that samples a point and launches a local search only if the new point has the overall best function value.

The radius in equation (I.3.2) has been derived in Boender et al. (1982); Rinnooy Kan and Timmer (1987a) by approximating the distribution of the nearest

neighbor statistic within a set of uniformly distributed points, and has been used afterward in many clustering algorithms. In fact, r_k was chosen in such a way that

$$\frac{\pi^{d/2}}{\Gamma(1+d/2)} r_k^d = \text{vol}(B_{r_k}) = \text{vol}(\Omega)\xi\frac{log(k)}{k},$$

where B_r denotes a ball of radius r for the ℓ_2 norm. The asymptotic development for the Γ function, as $d \to \infty$ for fixed k, shows that (Schoen, 1999)

$$r_k \sim \left(\xi\frac{log(k)}{k}\right)^{1/d} (d/2)^{1/2+2^{-d}}.$$

That number grows thus as $\sqrt{d}$ for large d, as the diameter of the ℓ_2 unit ball does, forcing the radius (I.3.2) to be too big at the first iterations. For these reasons, Schoen (1999) proposed to use the ℓ_∞ norm instead, for which the diameter of the unit ball is constant in any dimension. The resulting Algorithm, called ∞-Simple Linkage (∞-SL), follows the lines of Algorithm 2, using the ℓ_∞ norm in (I.3.3), and replacing radius $r_{k,\xi}$ by

$$s_k = s_{k,\xi} = (\xi \log(k)/k)^{1/d}. \tag{I.3.4}$$

Algorithm ∞-SL satisfies property P3 for any $\xi > 1$.

Nevertheless, Beyer et al. (1999) pointed out that the difference between the distance from any point to its nearest neighbor and the distance from the same point to the farthest point in the sample, hereafter simply called *the gap*, shrinks to 0 as the dimension grows. Later on, in Aggarwal et al. (2001), the role played in this phenomenon by the norm used to measure distances was revealed. It was shown, for a wide range of distributions, that in expectation as $d \to \infty$, the gap goes to 0 for the ℓ_∞ norm, tends to a constant for the ℓ_2 norm, and goes to ∞ for the ℓ_1 norm (see the left part of Figures I.3.4, I.3.3 and I.3.2).

In the next sections, using the results of Beyer et al. (1999) and Aggarwal et al. (2001) on the behavior of the nearest neighbor in high dimensional spaces, we show that the problem of choosing the radius is indeed very delicate. We propose an alternative method intended to be independent of the dimension, the feasible domain and the sampling distribution.

I.3.2 Nearest neighbor behavior in high dimension

In this section we shall show how the poor performance of clustering algorithms in high dimension, particularly their excessive resemblance to MS or BS, is related to the above-mentioned "gap phenomenon".

Let us focus on condition (I.3.3) at step 3. of Algorithm 2 (the following also holds for MLSL). Denote by D^k_{min} and D^k_{max} respectively the minimum and maximum distance from the last sampled point β_k to the other points β_j, $j = 1, ..., k-1$ in the sample. Three scenarios are possible at this iteration:

i) $r_k < D^k_{min}$. A local search will be started, independently of the function values of the rest of the sample. At this iteration the algorithm will make the same decision as Multistart, so we call this iteration a 'Multistart iteration'.

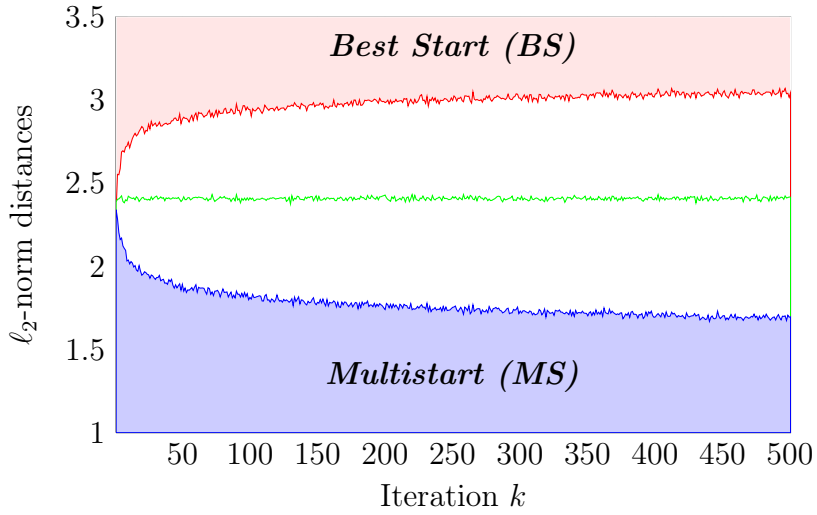

Figure I.3.1: For each iteration k, the upper curve plots the maximum distance from the last sampled point to the points sampled in previous iterations. Similarly, the lowest curve indicates the minimum distance; between them we show the median of the distances.

ii) $D^k_{min} \leq r_k < D^k_{max}$. A local search will be performed depending on the function values of the fraction of the sample that is within a distance r_k.

iii) $D^k_{max} \leq r_k$. A local search will be started only if β_k has the best overall function value. Thus, the algorithm will make the same decision as that of best start, and we shall talk of a 'best start iteration'.

We see that for small r, we are always in case (i), and the algorithm behaves like multistart. On the contrary, for large r, the function value of all the points in the sample are taken into account before performing a local search, hence the algorithm behaves like best start. It is precisely for r_k in the range between D^k_{min} and D^k_{max} that more complex algorithms can outperform basic methods like multistart and best start.

In Figure I.3.1, we illustrate the situation with $\Omega = [0,1]^{35}$. During 500 iterations, we show the maximum (top), minimum (bottom) and the median (between) of the distances from the points sampled at previous iterations to the last sampled point. For a given curve r_k, at the ks for which the curve lies on the upper region, the algorithm will behave like BS, and at those k for which the curve lies on the lower region, the algorithm will make the same decision as MS. It results clear that if the space between the top and the bottom curves wipes out and the choice of the radius does not follows accordingly, the algorithm will behave either as MS or BS.

Going into the details, Aggarwal et al. (2001, Lemma 2) show, for a large set of distributions, that for a sample of size N,

$$C(p) \leq \lim_{d\to\infty} \mathbb{E}\left(\frac{D_{max} - D_{min}}{d^{1/p-1/2}}\right) \leq (N-1)C(p), \tag{I.3.5}$$

where $C(p)$ is a constant depending on p, the distances being measured according to the p-norm, and D_{max} (resp. D_{min}) denotes the maximum (resp. minimum) distance from any point to the other points in the sample. Note that D^k_{min} and D^k_{max} depend also on the dimension and the norm. A similar result holds for "fractional norms", i.e for the maps $(x_1, ..., x_d) \mapsto (x_1^p + \cdots + x_d^p)^{1/p}$ with $p \in (0, 1)$.

The result (I.3.5) announces that:

- for $p > 2$, the gap tends to 0 as the dimension d increases, promoting subite switching from best start regime to a multistart regime;
- for $p = 2$, the gap tends, for a fixed number of sampled points, to an unknown constant;
- for $1 \leq p < 2$, the gap increases to ∞ as the dimension raises.

As in (I.3.5) the limit as $d \to \infty$ is taken for a constant sample size, and in SL-type optimization algorithms the current sample at iteration k consists of k points, we simulate the stage-wise sampling of algorithm 2 in dimensions $d = 5, 10, 15, 20, 25, ..., 105$. For $k = 2, ..., 500$, we sample uniformly a point β_k in $[0, 1]^d$, then we compute the distance for the p-norm, $\|\beta_k - \beta_j\|_p$ for $j = 1, ..., k-1$ and save the α-quantile q^k_α such that

$$\alpha = \mathbb{P}(\|\beta_k - \beta_j\|_p \leq q^k_\alpha)$$

for $\alpha = 0, 0.25, 0.5, 0.75$ and 1. In particular, the 1-quantile q^k_1 coincides with D^k_{max}, and the 0-quantile q^k_0 coincides with D^k_{min}. For fixed d and α, most of the plots of the distances against the iteration number k look much as Figure I.3.1. So a first observation is that the influence of changing number of points k is not significant. Much more interesting are the plots of the gap $q_1 - q_0$ and the inter-quartile difference $q_{.75} - q_{.25}$ for varying d and fixed k, shown in Figures I.3.4, I.3.3 and I.3.2 for the ℓ_∞, ℓ_2 and ℓ_1 norm, respectively.

As predicted by the theoretical results, for $p = \infty$ (Figure I.3.2) the gap between the curves depicting the closest point and the farthest point to β_k vanishes as the dimension increases. The augmentation of the number of points does not seem to be significant enough to change this trend. For $p = 2$ (Figure I.3.3), the variability due to the change in dimension is not significant. The situation is better when using the ℓ_1 norm, as shown in Figure I.3.4. As predicted, the gap increases with dimension. We can also observe that even in the lowest considered dimension, $d = 5$, the gap is of the same magnitude as for the ℓ_2 or ℓ_∞ norms. We conclude from these observation that the ℓ_1 norm is the best alternative for distinguishing points in any dimension. For $p < 1$, even if the maps $(x_1, ..., x_d) \mapsto (x_1^p + \cdots + x_d^p)^{1/p}$ are not norms, theoretical results say that the gap should increase faster than with $p = 1$, and this is actually the case. Nevertheless, besides the embarrassment at interpreting the outcome of functions that are not norms, they present the inconvenient that in low dimension the gap is quite close to 0.

In Figure I.3.5 we show the effect of this phenomenon on a ∞-SL-type algorithm. For the first 150 iterations k, in dimension 20, 35, 50 and 75, we plot $s_{k,\xi}$ as defined

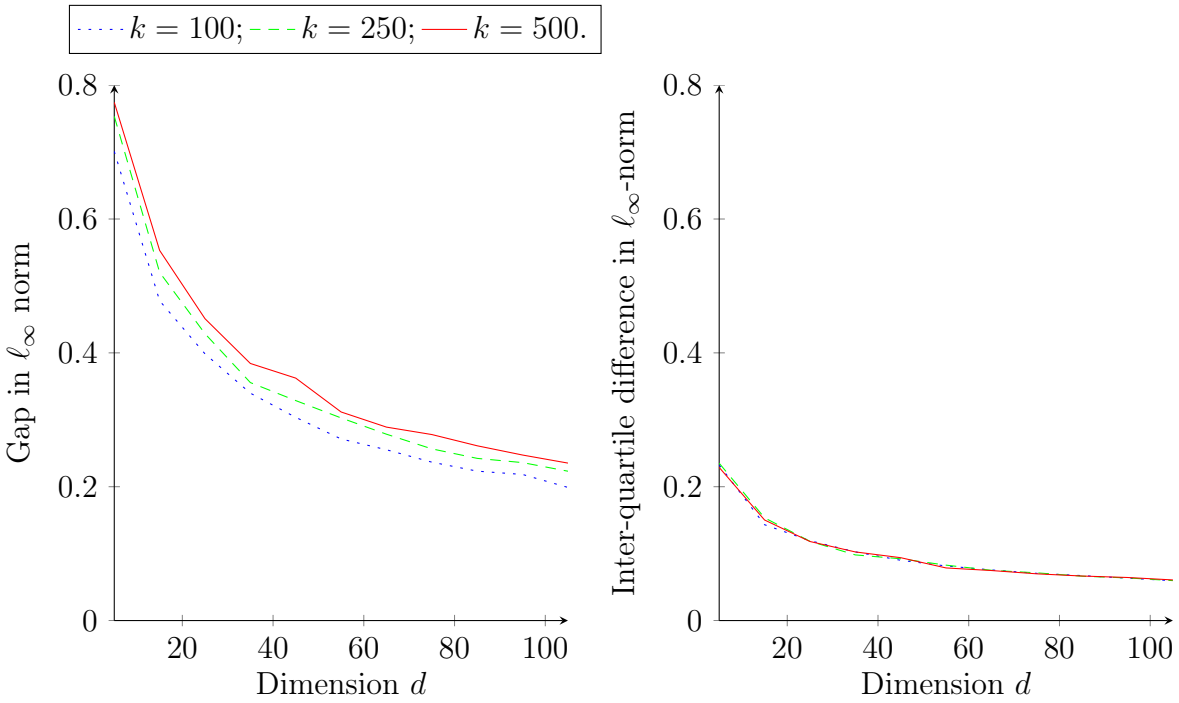

Figure I.3.2: The gap $q_1 - q_0$ (left) and the inter-quartile difference $q_{.75} - q_{.25}$ (right) for the ℓ_∞ norm.

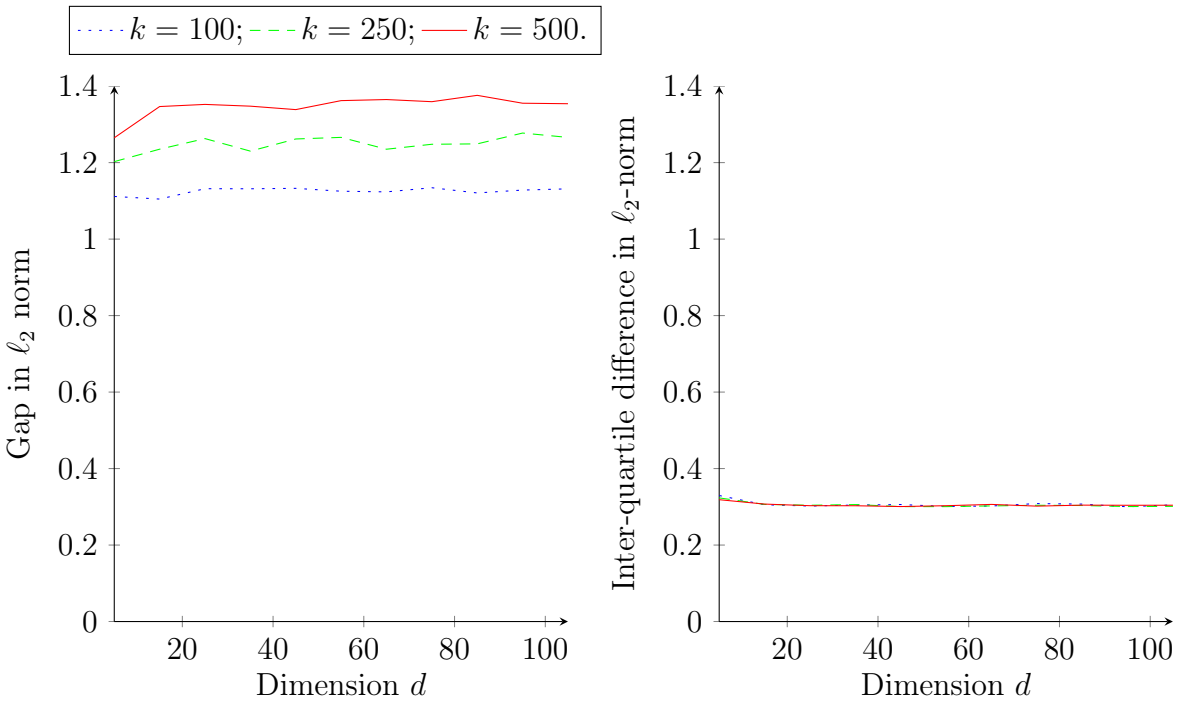

Figure I.3.3: The gap $q_1 - q_0$ (left) and the inter-quartile difference $q_{.75} - q_{.25}$ (right) for the ℓ_2 norm.

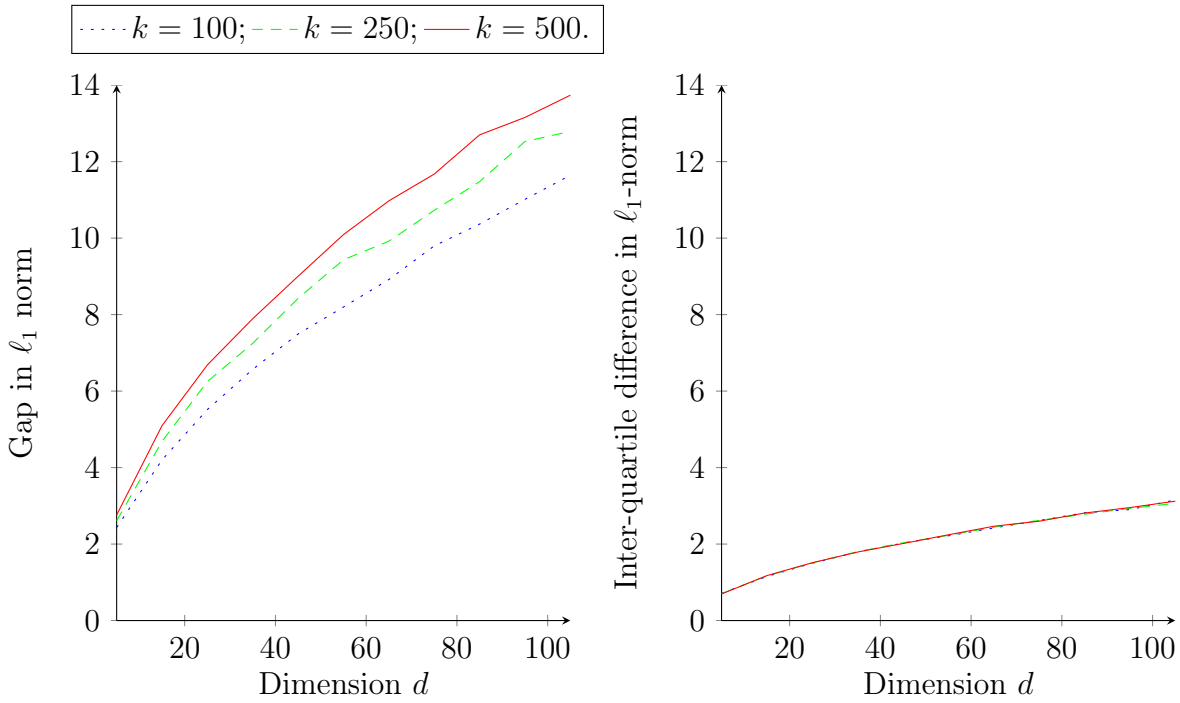

Figure I.3.4: The gap $q_1 - q_0$ (left) and the inter-quartile difference $q_{.75} - q_{.25}$ (right) for the ℓ_1 norm.

in (I.3.4) with $\xi = 2$ (dotted line), and the ℓ_∞ distances from β_k to its nearest point in the sample, the farthest point in the sample and the median of the ℓ_∞ distances from β_k to the previously sampled points, as in Figure I.3.1. We see how, as the dimension increases, both the curve s_k and the curve of the median go up, but the narrowing of the gap makes s_k to get closer and closer to the curve of the farthest distances, indicating that the algorithm behaves like BS with respect to deciding to start (or not) a local search.

However, the numerical results reported in Schoen (1999) seemed rather good. This is probably because of another modification they proposed: make the radius depend on the sampled point. Again, this modification was introduced through the parameter ξ, which is no longer fixed in advance as in Algorithm 2, but is computed once β_k has been sampled, and depends both on d and β_k: $\xi = \xi_d(\beta_k)$.

I.3.3 Replacing radius by quantiles

The direct application of the nearest neighbor distribution to estimate the clustering radius does not give good results during the first iterations. Moreover, finding a useful functional form for $r = r(k, d, \xi, \beta_k, p)$ for more general domains than the unit cube seems to be out of reach, even for uniformly distributed points.

A dual viewpoint is that of quantiles. At the beginning of iteration k, $k-1$ points $\beta_1, ..., \beta_{k-1}$ have already been sampled. Then, we sample β_k and every point $\beta_i,\ i = 1, .., k-1$ is within distance $r_0 = \min_i \|\beta_k - \beta_i\|_p$ and $r_1 = \max_i \|\beta_k - \beta_i\|_p$

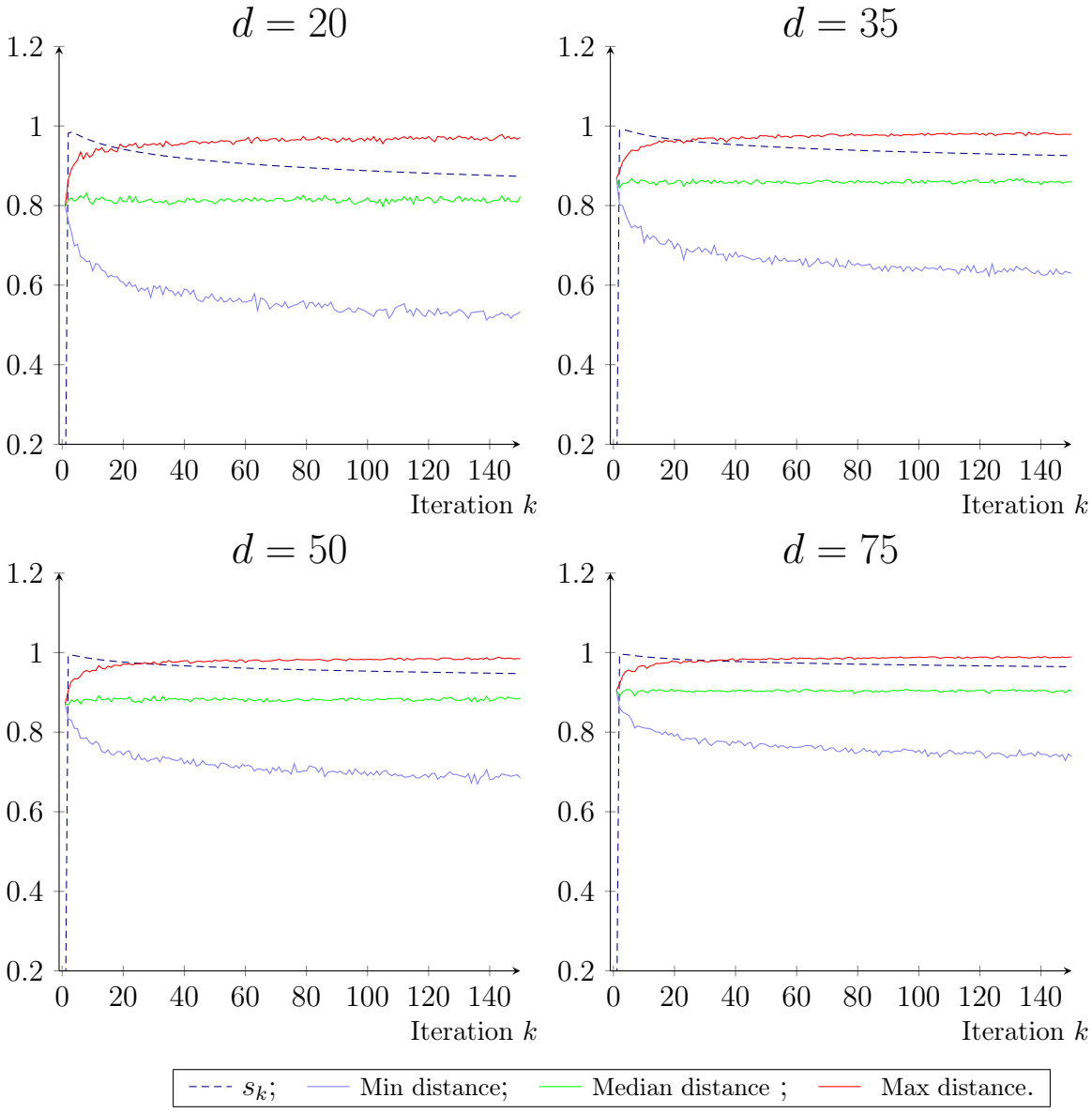

Figure I.3.5: Behavior of s_k in dimensions 20, 35, 50 and 75. Excepting s_k, all the plots have the same meaning as in Figure I.3.1

from β_k. Hence, by taking a radius $r \geq r_1$ we are considering all points, in other words, we are considering the 1-quantile. On the other extreme, by taking $r \leq r_0$ the local search is launched without considering any function value, or equivalently, taking the 0-quantile for the threshold radius.

In the same way, for intermediate radius $r_0 < r < r_1$ we are considering the α_r-quantile, with $\alpha_r = \mathbb{P}(\|\beta_k - \cdot\|_p \leq r \mid \beta_1, ..., \beta_{k-1})$. Conversely, if we consider the α-quantile, $0 < \alpha < 1$, we can associate to this quantile a radius r_α obtained as the distance from β_k to the farthest point within this quantile. Note that for $\alpha = 0$ or $\alpha = 1$ we cannot associate a unique radius, but only the intervals $[0, r_0]$ and $[r_1, \infty]$ respectively.

Algorithm 3 Quantile Global Optimization (QGO)

Choose a norm p, a sampling distribution F and a sequence α_k.
Set k=1, randomly generate $\beta_1 \sim F$ and iterate:

1. Let $k = k + 1$;
2. randomly generate $\beta_k \sim F$ and compute the ℓ_p-distance to the previously sampled points;
3. let r_k be the α_k-quantile of the distances computed in 2;
4. launch a local search from β_k except if

$$\exists j < k, \ \|\beta_k - \beta_j\|_p \leq r_k \text{ and } \quad \sigma(\beta_j) \leq \sigma(\beta_k); \tag{I.3.6}$$

5. check stopping condition, if it is not met, repeat from step 1.

Based on the later observation, we propose and evaluate Algorithm 3, called quantile global optimization (QGO). It is formulated for an arbitrary norm p and sampling distribution F, and depends on the sequence $\{\alpha_k\}$ of quantile orders.

Some key aspects should be taken into account for choosing the sequence α_k:

- At the first iteration, the probability of finding a new minima is 1, hence we perform an MS iteration.
- During the fist iterations, the probability of finding new minima is largest, so we would like to be 'close to MS'.
- As the iterations go, the probability of rediscovering known minima raises, then we would like to start leaving from MS and getting closer to BS.

These guidelines suggest taking increasing sequences α_k such that $\alpha_0 = 0$ and $\alpha_k \to 1$ as $k \to \infty$. For such a quantile order sequence, the following holds:

Theorem 1 *Consider the QGO algorithm described hereabove, with* $1 \leq p \leq \infty$ *and a sampling distribution* F *whose support contains the whole feasible domain* Ω. *Then the following properties hold, as* $k \to \infty$*:*

1. *The best observed function value converges to the minimum of the function* σ *over* Ω *with probability* 1.

2. *If the sequence α_k converges to 1, the probability of starting a local search converges to 0.*

Proof.
The first affirmation follows from the fact that since $\Omega \subseteq \text{support}(F)$, then the random sampling tends to cover the whole domain. Let us denote as $\bar{\sigma}_k$ the best observed value at iteration k. In order to prove *2*, let us rewrite step 3 of Algorithm 3 as: launch a local search if

$$\forall j < k,\ \sigma(\beta_j) > \sigma(\beta_k)\ or\ \|\beta_j - \beta_k\|_p > r_k,$$

then, if we denote by E_k^A the event «Algorithm A starts a local search at iteration k », we have

$$\begin{aligned}\mathbb{P}(E_k^{QGO}) &= \mathbb{P}\left(\left[\bigcap_{j<k}\{\sigma(\beta_j) > \sigma(\beta_k)\}\right]\bigcup\left[\bigcap_{j<k}\|\beta_j - \beta_k\|_p > r_k\right]\right) \qquad \text{(I.3.7)}\\ &\leq \mathbb{P}(\sigma(\beta_j) > \sigma(\beta_k),\ \forall j < k) + \mathbb{P}(\|\beta_j - \beta_k\|_p > r_k, \forall j < k)\\ &= \mathbb{P}(\sigma(\beta_j) > \sigma(\beta_k),\ \forall j < k) + (1 - \alpha_k).\end{aligned}$$

Next we rewrite $\{\sigma(\beta_j) > \sigma(\beta_k),\ \forall j < k\}$ as $\{\bar{\sigma}_k > \sigma(\beta_k)\}$ to conclude that $\mathbb{P}(\sigma(\beta_j) > \sigma(\beta_k),\ \forall j < k) = \mathbb{P}(\{\bar{\sigma}_k > \sigma(\beta_k)\}) \to 0$, because $\bar{\sigma}_k \to \inf_\Omega \sigma$. Since $1 - \alpha_k \to 0$ as well, the second assertion follows. □

In order to illustrate how the QGO algorithm works, in Figure I.3.6 we display an hyperbolic tangent-shaped sequence $\alpha_k \to 1$, and the corresponding radii obtained using quantiles in the same situation as that of Figure I.3.1. We can observe that at the beginning the algorithm behaves like MS, and tends asymptotically to behave like BS, and this, independently of the dimension, the norm used to measure distances, or the domain Ω.

A notable such sequence is $\alpha_k = 1 - p_k$, where p_k denotes the probability of finding a new minimum. This proposal follows closely the previous reasoning, since at the first iteration the probability of finding a new minimum is 1, because no local minima are known, so we have $\alpha_k = 0$, which means performing a multistart-type iteration. Then, as the iterations go, $p_k \to 0$, and $\alpha_k \to 1$. In practice the probabilities p_k are unknown. One possibility is to estimate them with the information gathered from previous iterations, as in Boender and Rinnooy Kan (1987) or Piccioni and Ramponi (1990). In Piccioni and Ramponi (1990), the probability of having found a global optimum at iteration k, provided that in the previous iterations the algorithm has found w_k local minima, is:

$$\alpha_k = 1 - p_k = \frac{k - w_k - 1}{k - 1},$$

then $\alpha_k \to 1$ as $k \to \infty$, and therefore the probability of launching a local search decreases to 0.

However, there is still a lot of freedom for choosing the sequence α_k. The main difficulty for imposing conditions on the sequence α_k comes from the fact that in QGO we are not just proposing to replace distance thresholds by quantiles, but

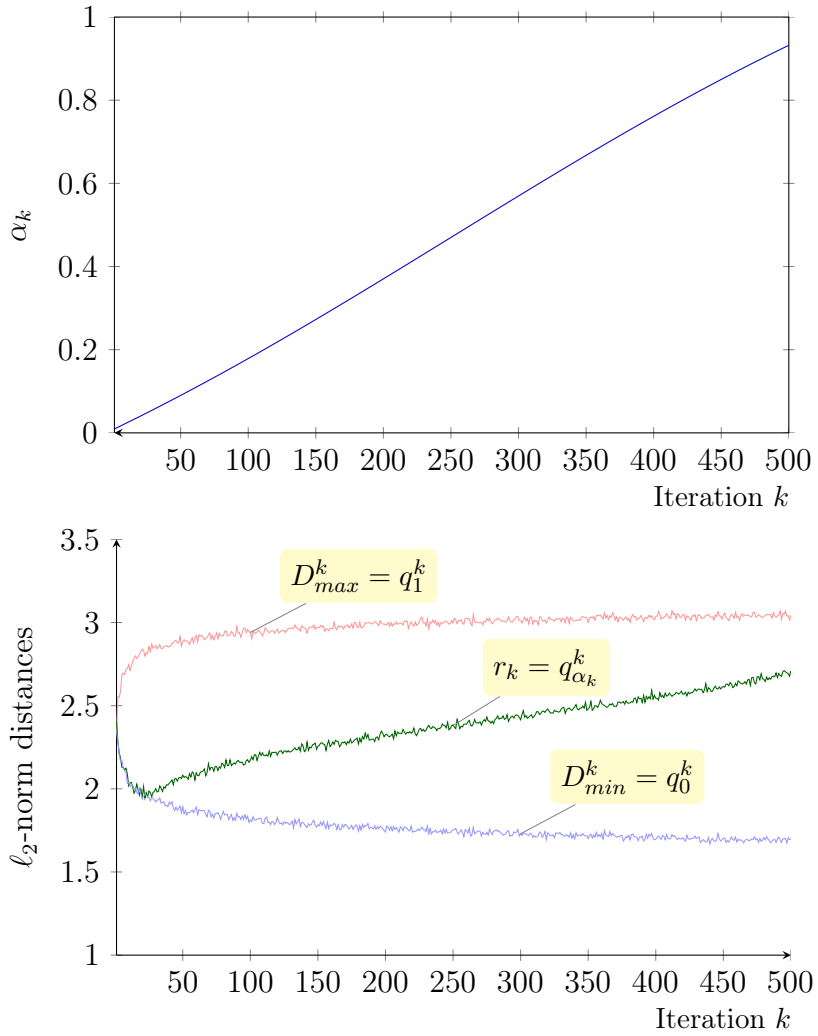

Figure I.3.6: Above: a sequence of quantile orders α_k; below: radii obtained from the sequence α_k.

also to use local information during the first iterations and more and more global information as the number of iterations increases. This behavior has as consequence that showing that the probability of launching a local search tends to 0 is quite easy, since asymptotically QGO behaves as BS. However, obtaining the rate at which this convergence occurs is much harder than for methods using a threshold that goes to 0, which use local techniques such as Taylor expansions for asymptotic analysis (see eg. Rinnooy Kan and Timmer, 1987a, Lemma 7).

Conclusion

In this first part of the thesis, we analyzed some possible approaches to the efficient approximation of τ-estimators. The bibliographic review, both on global optimization and on computational robust statistics, suggested that two-phase stochastic algorithms coupled with appropriate stopping conditions, could help improving the efficiency of state-of-the-art algorithms.

Our numerical tests confirmed that stopping conditions are a valuable tool, permitting to shorten the computing time of existing algorithms while keeping the quality of the solutions thereof obtained.

Clustering techniques proved to be useful in low dimension, but their efficacy rapidly deteriorate in middling and high dimension. A deeper study of clustering global optimization in higher dimensions shed some light on the problem. The approach of quantiles besides the advantage of being adaptive to the dimension and the current sampled point, represents also a first step towards non-uniform sampling.

However, further research would be necessary to get more concluding results. There may be many other factors influencing the efficiency of clustering methods in high dimension. Just to mention one, let us cite a result of Ledoux and Talagrand (1991) about concentration of measures: let λ denote the uniform measure on the cube, and σ a Lipschitz function with Lipschitz constant L and median M. Then

$$\lambda(|\sigma - M| > t) \leq ce^{-t^2/2s^2}$$

where c is a constant, and $s^2 = (2\pi)^{1/2}L^2$. This could suggest that uniform sampling is not a good strategy, since sampling far from the median, thus eventually close to minima, is unlikely. This fact has been noticed by Schoen (1998), who proposed to replace uniform random sampling by deterministic, well distributed, sample points. Nevertheless, the simplicity of the uniform distribution permits to have many explicit results, particularly useful for obtaining usable stopping conditions. The theoretically ideal sampling distribution is given by (I.1.1) as $T \to 0$. Unfortunately, implementable version of this sampling strategy, notably simulated annealing, does not perform better in practice than two-phase algorithms based on uniform sampling.

All the experience collected from numerical experiments seems to indicate that stochastic algorithms are able to provide "good" solutions to global optimization problems. Nevertheless, the quality of the outcome is quite variable and difficult

to foresee even for particular problem classes with loose computational-time constraints.

Part II

Deterministic algorithms for mixed-integer bilinear programs.

Chapter II.1

Introduction

The second part of this dissertation is devoted to the study of deterministic algorithms for exact calculation of robust regression estimators. In particular, we shall focus on robust regression estimators defined through L-estimates of scale.

We begin by recalling the definition of L-scales: Let $|r(\beta)|_{(1)} \leq ... \leq |r(\beta)|_{(n)}$ be the ordered absolute values of residuals (see model on page 3). $L-$estimates of scale are defined as weighted ℓ_1 or ℓ_2 norms of $|r(\beta)|_{(i)}$,

$$\sum_{i=1}^{n} a_i |r|_{(i)}, \qquad \text{or} \qquad \left(\sum_{i=1}^{n} a_i |r|^2_{(i)} \right)^{1/2}, \tag{II.1.1}$$

where a_i are nonnegative constants.

The restriction to this class of estimators is motivated by the fact that the rigidity of scales (II.1.1) gives to the problem a combinatorial structure which permits to solve it to optimality. Besides the exhaustive enumeration approach, that is feasible only for very small problems, there exist many more sophisticated techniques for solving global optimization problems; some of them are suited to a particular subclass of problems, and others are applicable to a wide range of problems. The method to be used here strongly depends on the characteristics of the particular problem under consideration. Therefore, the first step will be to examine the problem of minimization of L-scales from an optimization point of view.

Let us concentrate on the problem

$$\min_{\beta \in \mathbb{R}^d} \sum_{i=1}^{n} a_i |r(\beta)|^p_{(i)} \tag{II.1.2}$$

for p equal to 1 or 2, and where the a_is are such that $a_i = 1$ if $i \leq h$, and $a_i = 0$ if $i > h$ for a certain $h \in \{1, ..., n\}$, as it is the case for the LTS and the LTA estimators (see pages 7 and 8).

Problem (II.1.2) can be written as a mixed integer nonlinear program using the fact that for arbitrary $r \in \mathbb{R}^n$, if $r_{(1)} \leq r_{(2)} \leq \cdots \leq r_{(n)}$ denote its ordered components, then

$$\sum_{i=1}^{h} r_{(i)} = \min \left\{ \sum_{i=1}^{n} w_i r_i \,\middle|\, w \in \{0,1\}^d, \sum_{i=1}^{n} w_i = h \right\}, \tag{II.1.3}$$

and

$$\min_{\beta\in\mathbb{R}^d} \sum_{i=1}^{h} |r(\beta)|^p_{(i)} = \min_{\beta\in\mathbb{R}^d} \min_{w\in\tilde{\mathcal{C}}_h} \sum_{i=1}^{n} w_i |r(\beta)|^p_i, \tag{II.1.4}$$

where $\tilde{\mathcal{C}}_h = \{w \in \{0,1\}^d, \sum_{i=1}^n w_i = h\}$.

It is not difficult to check that we have an analogous formulation for the sum of the largest residuals, replacing minimum by maximum.

$$\begin{aligned}\sum_{i=n-h}^{n} r_{(i)} &= \max\left\{ \sum_{i=1}^{n} w_i r_i \,\middle|\, w\in\{0,1\}^d, \sum_{i=1}^{n} w_i = h \right\} \\ &= -\min\left\{ -\sum_{i=1}^{n} w_i r_i \,\middle|\, w\in\{0,1\}^d, \sum_{i=1}^{n} w_i = h \right\},\end{aligned}$$

which for $h = 1$ amounts to minimizing the ℓ_∞-norm of the residuals.

Hereafter we consider the problem of computing the LTS estimator,

$$\begin{array}{ll} \min & \sum_{i=1}^{n} w_i r_i(\beta)^2, \\ \beta\in\mathbb{R}^d, & \\ w\in\tilde{\mathcal{C}}_h, & \end{array} \tag{II.1.5}$$

which is a MINLP, quadratic in β and linear in w.

Since the constraint $w \in \{0,1\}^n$ can be equivalently expressed as $w_i^2 - w_i = 0$, problem (II.1.5) fits into the polynomial programming framework. Lasserre (2001) introduced an innovative approach to polynomial global optimization that consists in approximating the problem by a hierarchy of efficiently solvable relaxations. The relaxation is exact for a sufficiently high order. Though, the large size of the relaxations for moderate-sized problems urged researchers on to exploit sparsity in order to diminish the size of the relaxed problems. Lasserre (2006) and Waki et al. (2006) went in that direction by putting forward lightweight relaxations under structured sparsity assumptions. Nevertheless, even if sparsity is present, the relaxation of order r of problem (II.1.5) involves $(d+1)^2 n\binom{3+2r}{2r}$ variables; so increasing by one the number of observations will result in $(d+1)^2\binom{3+2r}{2r}$ new variables added to the relaxation of order r. To get an idea of the sizes, in dimension $d = 10$ adding one observation would increase the number of variables of the first 5 relaxations by 1210, 4235, 10164, 19965 and 34606 respectively. The methodology of Lasserre (2001) is intended to deal with a very wide class of problems; moreover it is a generalization of the «lifting »procedure. Our objective will be to gain efficiency by exploiting the characteristics of our particular problem.

Let us start by cutting problem (II.1.5) off as

$$\min\{\ v(w)\ \mid w\in\tilde{\mathcal{C}}_h\}, \tag{II.1.6}$$

where $v(w)$ is the marginal value of problem (II.1.5), obtained by minimization over $\beta\in\mathbb{R}^d$ for fixed $w\in\tilde{\mathcal{C}}_h$,

$$v(w) = \inf_{\beta\in\mathbb{R}^d} \sum_{i=1}^{n} w_i r_i(\beta)^2. \tag{II.1.7}$$

Problem (II.1.6) is a combinatorial optimization problem over the discrete set $\tilde{\mathcal{C}}_h$. In order to avoid the exhaustive enumeration of all of its elements, Agulló (2001) proposed a branch-and-bound (BB) algorithm. A BB algorithm consists in enumerating in a tree the elements of $\tilde{\mathcal{C}}_h$, disregarding unpromising ones. In Figure II.1.1 we depict the BB tree for a small example with the LTS estimator keeping 3 observations out of 6.

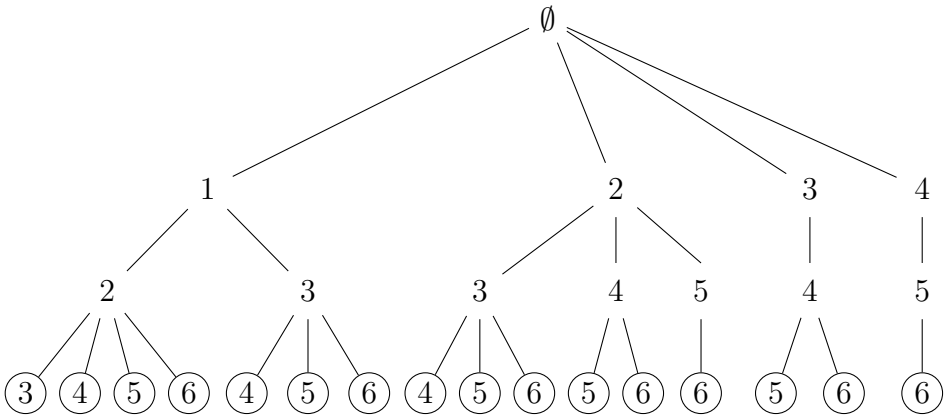

Figure II.1.1: The BB tree for $n = 6$ and $h = 3$.

The circled nodes are called leaves. Each leaf represents a subset of 3 observations, which is obtained adding recursively the parent of each node until the root $\emptyset$ is reached. For example, in the branch of Figure II.1.2 there are two leaves, the leaf

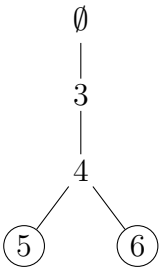

Figure II.1.2: A sample branch of the tree of Figure II.1.1.

at the right is associated to the subset of observations $6, 4$ and 3, and that at left to observations $5, 4$ and 3. In terms of the optimization variable w, they are associated to the points $(0, 0, 1, 1, 0, 1)'$ and $(0, 0, 1, 1, 1, 0)'$ respectively.

The BB algorithm proceeds by examining the branches from right to the left, from top to the bottom; keeping track of the best solution obtained up to that moment, called incumbent solution and denoted by $\bar{w}$ in the sequel. At each node the algorithm estimates the least function value that can be attained by a leaf belonging to the subtree rooted at that node. If this lower bound exceeds the function value of the incumbent solution, then we can safely discard that subtree, since we are sure that it will not improve the incumbent solution.

A subtree rooted at a node in the BB tree is associated to a set of partially prescribed points of $\tilde{\mathcal{C}}_h$. For instance, the branch in Figure II.1.2, rooted at the

node $(0,0,1,0,0,0)'$, consists of all the points in $\{0,1\}^6$ with $(0,0,1)'$ at the first coordinates. Let us write these node constraints as $Nw = \psi$. Note that N is a projection operator; in our example $Nw = (w_1; w_2; w_3)$ and $\psi = (0,0,1)'$. Therefore, that branch can be ignored if the value of the problem

$$\begin{array}{ll} \min & v(w) \\ s.t & \\ & Nw = \psi, \\ & \sum_{j=1}^{n} w_j = h, \\ & w \in \{0,1\}^n, \end{array} \tag{II.1.8}$$

is greater than the function value of the incumbent solution, which is an upper bound for the optimal value of Problem (II.1.6). However, obtaining lower bounds for a problem like (II.1.5) is not easy, because of the integer variables and the nonlinearity/nonconvexity. In practice, the value of problem (II.1.8) is essentially as difficult to compute as the value of the original problem, this is why we only settle for lower bounds. The efficiency of a BB algorithm is intimately related to the tightness of the lower bounds.

The algorithm of Agulló (2001) is based on a monotonicity property of the LTS problem: in the LTS tree (see Figure II.1.1) every node has greater function value than its father; this is because adding observations to the ordinary least-squares regression can only increase the sum of the squared residuals. Thus the function value at a node serves as lower bound for the whole subtree rooted at it. At each node the algorithm compares the function value of its children to that of the incumbent solution, hereafter denoted UB, and decides whether to prune it or not. If w^+ is one of those child nodes, and $|I^+|$ is the number of nonzero entries in w^+, then:

1. If $|I^+| < d$, since in dimension d we can always find an hyperplane that passes through $|I^+|$ points, then $v(w^+) = 0 < UB$. Thus, it cannot be discarded and the analysis continues with its children.

2. If $|I^+| \geq d$, evaluate $v(w^+)$.

   (a) If $v(w^+) < UB$ and w^+ is a leaf, then a new incumbent solution have been found; set $UB = v(w^+)$ and $\bar{w} = w^+$. Otherwise repeat the analysis with its children.

   (b) If $v(w^+) > UB$ discard w^+ and all its children.

The monotonicity lower bound presents two flaws:

- It does not provide lower bounds for nodes at the d highest levels.

- It does not look ahead. At each node the increase in function value due to the inclusion of the children is exactly computed, but the contribution of the rest of the observations added in the path to the leaves is underestimated by 0. For instance, if $n = 10$ and $h = 6$ it says "the sum of the squared residuals of a regression over six observations comprising observations $2, 4, 5$

and 8 will be greater than the sum of the squared residuals of the regression over observations $2, 4, 5$ and 8 themselves" without quantifying the minimum increase in sum of squared residuals possible due to the incorporation of two more observations.

The objective of the next two chapters is to obtain alternative lower bounds for problem (II.1.8) when far from the leaves, in particular at the d upper levels. In Chapter II.2 we introduce *cuts* for reducing the feasible domain of problem (II.1.8), thus tightening the lower bounds. In fact, since the objective function in (II.1.7) is linear in w, the function v is the pointwise minimum of linear functions

$$v(\cdot) = \min_{\beta \in \mathbb{R}^d} l'_\beta \cdot, \quad \text{with } l_\beta = (r_1(\beta)^2, r_2(\beta)^2, ..., r_n(\beta)^2),$$

and consequently it is a closed concave function (Hiriart-Urruty and Lemaréchal, 1993a, Prop. IV.2.1.2). A first consequence of the concavity of the objective function v is that minimization over $\tilde{\mathcal{C}}_h = \{w \in \{0,1\}^n, e'w = h\}$ is equivalent to minimizing over the polytope $\mathcal{C}_h = \{w \in [0,1]^n, e'w = h\}$. The reason is that the global minimum of a concave function over a convex set is always attained at an extreme point; in the case of a polytope, at a vertex. The cuts obtained in Chapter II.2 are based on the concavity of the function v, and moreover, they are called concavity cuts. If we represent the cuts under the form of a linear system $Cw \leq b$, then the value of the problem (II.1.6) is the same as the value of

$$\begin{array}{ll} \min & v(w) \\ s.t & \\ & Cw \leq b, \\ & \sum_{j=1}^n w_j = h, \\ & w \in [0,1]^n, \end{array} \tag{II.1.9}$$

and the least value at a BB node given by problem (II.1.8) is the same as

$$\begin{array}{ll} \min & v(w) \\ s.t & \\ & Nw = \psi, \\ & Cw \leq b, \\ & \sum_{j=1}^n w_j = h, \\ & w \in [0,1]^n. \end{array} \tag{II.1.10}$$

The reason is that concavity cuts eliminate portions of the feasible domain that cannot host a global minimum. Note that the values of problems (II.1.6) and (II.1.8) equal the value of problems (II.1.9) and (II.1.10) respectively, but relaxations of problems (II.1.9) and (II.1.10) will have in general a higher value than relaxations of (II.1.6) and (II.1.8).

Our approach for obtaining lower bounds for (II.1.9) is to cast it as a *multi-quadratic problem*, in order to take advantage of techniques specifically conceived for quadratic problems. Using a by-now classic technique, we lift the problem into a space of matrices, to obtain a linear problem with second order cone constraints, which is a relaxation of (II.1.9). By the way, under the same formulation and using

a surrogate of a rank constraint that can be be decomposed as the difference of two convex functions, we shall obtain global optimality conditions for problem (II.1.9).

Notations

In the sequel, we shall always use lowercase letters to name vectors and uppercase letters to denote matrices. Vertical concatenation will be indicated by ';' and horizontal concatenation by ','. For instance, $(0;1)$ is a column vector in $\mathbb{R}^2$, and $(0,1) = (0;1)'$ is the row vector with the same entries. Similarly, if $v1$ and $v2$ are two column vectors of the same length, then $(v1, v2)$ is the matrix containing the vectors $v1$ and $v2$ as columns, and $(v1'; v2')$ is the transpose of that matrix, containing $v1'$ and $v2'$ as rows. The column vector of length n with all components equal to 1 will be denoted as e, it will be used notably for expressing the linear constraint $\sum_{j=1}^{n} w_j = h$ in matrix form as $e'w = h$. For $1 \leq p \leq \infty$, B_p denote the unit ball for the p-norm. The space of real matrices of size $s_1 \times s_2$ will be denoted by $\mathbb{R}^{s_1 \times s_2}$, and the space of square symmetric matrices of size $s \times s$ by $\mathbb{S}_s$. The space $\mathbb{R}^{s_1 \times s_2}$ will be equipped with the inner product $\langle A, B \rangle = \text{trace}(B'A)$, where trace denotes the sum of the diagonal elements, as usual. Given a vector v, $D(v)$ will denote the diagonal matrix with diagonal terms $D_{ii} = v_i$.; reciprocally, for a matrix A, $d(A)$ is the vector whose components are the diagonal elements of A, $d(A)_i = A_{ii}$. A matrix of zeros of size $s_1 \times s_2$ will be denoted by $0_{s_1 \times s_2}$; if $s_1 = s_2$ we just write 0_s. We will make use later of two well-known objects from convex analysis. For a compact set E, $\delta(x \mid E) = \sup_{y \in E} x'y$ is the support function of the set E. The indicator function of an arbitrary set E, denoted as I_E, is the function defined as

$$I_E(x) = \begin{cases} 0 & \text{if } x \in E \\ +\infty & \text{if } x \notin E. \end{cases}$$

For a function f finite at a point x, and $\varepsilon \geq 0$, the ε-subgradient of f at x, denoted $\partial_\varepsilon f(x)$, is the set of vectors s such that

$$f(y) \geq f(x) + s'(y - x) - \varepsilon \quad \text{for all } y.$$

For the particular case $\varepsilon = 0$ we shall use the notation $\partial f(x)$ rather than $\partial_0 f(x)$.

Chapter II.2

Concavity cuts for LTS

II.2.1 Tuy's cuts for concave minimization over polytopes

Concavity cuts or Tuy's cuts, introduced by Tuy (1964), are a domain-reduction technique for concave minimization over a polytope $\mathcal{C}$,

$$\min_{w \in \mathcal{C}} v(w). \tag{II.2.1}$$

Given a vertex of the polytope and a level γ, the concavity cut thereof obtained discards a portion of $\mathcal{C}$ with a function value not better than γ. Lower values of γ give raise to deeper concavity cuts, therefore γ is usually set as low as possible in such a way that the concavity cut does not discard vertices with lower function value than the incumbent solution.

Figure II.2.1 illustrates the construction of a concavity cut. The first step consists in finding a sub-optimal vertex w_0 to eliminate, and a level γ such that $v(w_0) > \gamma$ (Fig. II.2.1-a). Then we find the hyperplane defining the concavity cut as follows. Let w_0 be the vertex of $\mathcal{C}$ to be eliminated and let us denote by $u_1, u_2, ..., u_n \in \mathbb{R}^n$ the directions of the edges of $\mathcal{C}$ emanating from w_0, and by $\text{cone}(u_1, ..., u_n)$ their conical envelope $\{\sum_{i=1}^n \lambda_i u_i, \lambda_i \geq 0\}$. Then,

$$K(w_0) := w_0 + \text{cone}(u_1, ..., u_n)$$

is the smallest cone vertexed at w_0 which contains $\mathcal{C}$. To derive a concavity cut we consider the level set $L(\gamma) := \{w \in \mathbb{R}^n | v(w) \geq \gamma\}$, which is commonly assumed for the sake of simplicity to be closed and bounded. Note that since v is concave $L(\gamma)$ is convex (Fig. II.2.1-a). Let $z_i \in \mathcal{C}$ and $\bar{\tau}_i$ be such that $z_i = w_0 + \bar{\tau}_i u_i$ are the intersection points of the cone edges $E_i(\tau_i) := x_0 + \tau_i u_i,\ \tau_i \geq 0$ with the boundary of $L(\gamma)$ (Fig. II.2.1-b). Then we determine the hyperplane described by the equation $c'(w - w_0) = 1$, which intersects the edges of $K(w_0)$ in $L(\gamma)$, and contains at least n of the points z_i (Fig. II.2.1-c). This hyperplane is given by

$$c' = \left(\frac{1}{\tau_1}, \frac{1}{\tau_2}, ..., \frac{1}{\tau_n}\right)' U^{-1} \tag{II.2.2}$$

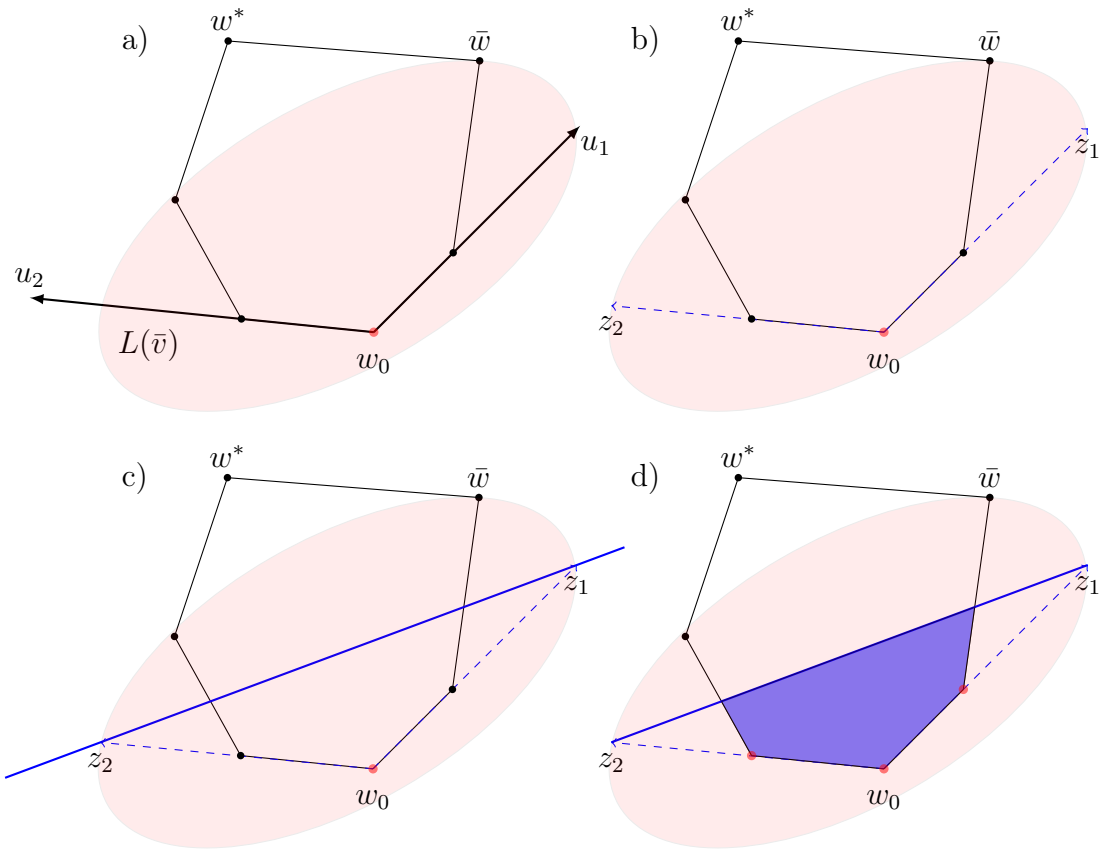

Figure II.2.1: Construction of a concavity cut.

where $U = (u_1, u_2, \cdots, u_n)$ is the matrix containing the directions u_i as columns. By construction, $\mathcal{C} \cap \{c'(w - w_0) \leq 1\} \subseteq L(\gamma)$, thus $\mathcal{C} \cap \{c'(w - w_0) \leq 1\}$, shaded in Fig. II.2.1-d), can be safely ignored from further consideration, without discarding any point with better function value than the incumbent solution.

II.2.2 Concavity cuts for the LTS problem

In this section we discuss the construction of concavity cuts for minimization of the function v defined in (II.1.7) over the polytope $\mathcal{C}_h = \{w \in [0,1]^n, e'w = h\}$. There are two issues that hinder obtaining concavity cuts for problem (II.2.1),

1. The polytope $\mathcal{C}_h$ is degenerate.

2. The level sets of v are unbounded.

The polytope $\mathcal{C}$ for $n = 3$ and $h = 2$ is depicted in Figure II.2.2, where each vertex corresponds to a subset of 2 observations. In general, the polytope $\mathcal{C}$ lies on the hyperplane H defined by the constraint $e'w = h$, and any vertex has $(n - h)h$ neighbors. Instead of making the effort to obtain $n-1$ directions to define a flat cone on the hyperplane H, we will ignore the constraint $e'w = h$ and we will construct a concavity cut on the unit cube. The resulting cut constructed on the cube will

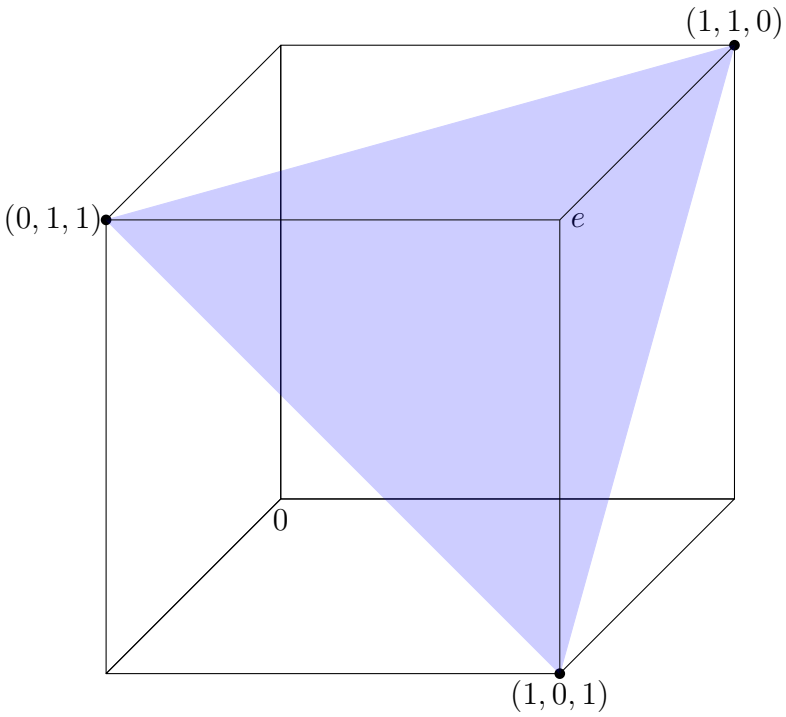

Figure II.2.2: The LTS polytope for $n = 3$ and $h = 2$.

intersect H unless H and the cut are parallel, which is impossible as we will see later. Incidentally, the BB tree nodes, other than the leaves, correspond to vertices of the unit cube not belonging to the hyperplane $e'w = h$. Let w_0 be an extreme point of $[0, 1]^n$ and $I_0 = \{j \mid (w_0)_j = 1\}$ be the index set of the components equal to one.

Let us define the directions $u_1, ..., u_n$ as

$$u_j = \begin{cases} -e_j & \text{if } j \in I_0 \\ e_j & \text{otherwise,} \end{cases} \tag{II.2.3}$$

where e_j is the jth euclidean unit vector (whose jth component is equal to 1 and 0 otherwise). We have that $w_0 + \text{cone}(u_1, ..., u_n)$ is the smallest cone pointed at w_0 that contains the unit cube. In Figure II.2.3 we see an example of cut on the unit cube for the situation of Figure II.2.2 with $n = 3$ and $h = 2$. The next step to construct the concavity cut on the cube is to determine the intersection point of the rays $w_0 + \tau_j u_j,\ j = 1, ..., n$ with the boundary of $L(\gamma)$, for $\gamma < v(w_0)$. In other words, we look for $\bar{\tau}_j,\ j = 1, ..., n$ such that $v(w_0 + \bar{\tau}_j u_j) = \gamma$.

For $j \in I_0$, by moving along the direction $-\tau_j e_j$ we will downweight observation j , and we will eventually meet the boundary of $L(\gamma)$. On the contrary, if $j \notin I_0$ we will move along the direction $\tau_j e_j$, which amounts to add an observation with weight τ_j, and we will always increase the objective value, for any $\tau_j > 0$. This means that the set $L(\gamma)$ in unbounded in the direction u_j for $j \notin I_0$; in this case we will take $\tau_j = +\infty$. In fact, we are looking for *the largest* conical set vertexed at w_0, completely contained in $L(\gamma)$ and possessing a concise description. Of course,

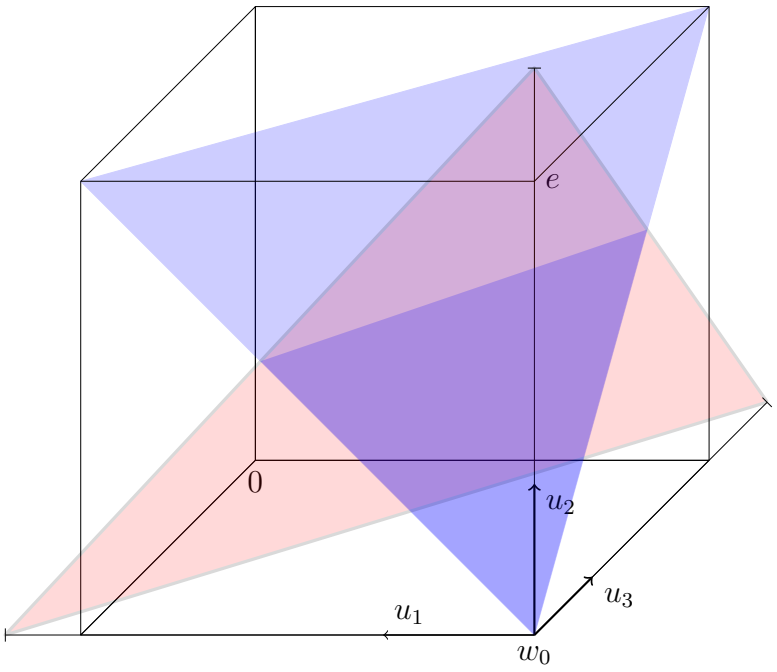

Figure II.2.3: A concavity cut on the unit cube.

when the level sets are bounded, such a polyhedron can be obtained as described hereabove.

The remaining computations in (II.2.2) are:

- Obtaining the 'step sizes' $\bar{\tau}_j$ for $j \in I_0$.
- Inverting the matrix U that contains the directions u_j as columns.

As we work on the unit cube, U is simply proportional to the identity matrix, thus inverting that matrix is not an issue. So we turn next to the computation of $\bar{\tau}_j$ for $j \in I_0$.

Given that the function v is defined in (II.1.7) as the value of the convex minimization problem

$$\inf_{\beta \in \mathbb{R}^d} r(\beta)' D(w) r(\beta),$$

then we have, when $\det(X'D(w)X) \neq 0$,

$$v(w) = r(\beta_w)' D(w) r(\beta_w),$$

for the unique β_w satisfying

$$X'D(w)X\beta_w = X'D(w)y. \tag{II.2.4}$$

Since $r(\beta) = y - X\beta$,

$$\begin{aligned} v(w) =& r(\beta_w)' D(w) r(\beta_w) \\ =& (y - X\beta_w)' D(w) r(\beta_w) \\ =& y' D(w) r(\beta_w) - \beta_w' (X' D(w) r(\beta_w)), \end{aligned}$$

and rewriting (II.2.4) as $X'D(w)r(\beta_w) = 0$ we conclude that

$$v(w) = y'D(w)r(\beta_w). \tag{II.2.5}$$

Furthermore, as $\beta_w = (X'D(w)X)^{-1}X'D(w)y$,

$$\begin{aligned} v(w) &= y'D(w)(y - X\beta_w) \\ &= y'D(w)(y - X(X'D(w)X)^{-1}X'D(w)y), \end{aligned}$$

which motivates the introduction of the matrices

$$M(w) = X'D(w)X$$

and

$$\tilde{M}(w) = \begin{pmatrix} X'D(w)X & X'D(w)y \\ y'D(w)X & y'D(w)y \end{pmatrix} = \begin{pmatrix} M(w) & X'D(w)y \\ y'D(w)X & y'D(w)y \end{pmatrix};$$

then by the *Schur complement* formula for determinants of block matrices we obtain

$$\begin{aligned} \det(\tilde{M}(w)) &= \det(M(w))\det(y'D(w)y - (y'D(w)X)M(w)^{-1}(X'D(w)y)) \\ &= \det(M(w))\det(y'D(w)[y - XM(w)^{-1}X'D(w)y]) \\ &= \det(M(w))\det(y'D(w)[y - X\beta_w]) \\ &= \det(M(w))y'D(w)r(\beta_w) \\ &= \det(M(w))v(w), \end{aligned} \tag{II.2.6}$$

which supplies us, provided that $\det(M(w)) \neq 0$, with a convenient expression for the objective function v,

$$v(w) = \frac{\det(\tilde{M}(w))}{\det(M(w))}.$$

The next step is to give a formula for $v(w_0 + \tau_j u_j)$, $j \in I_0$. Let us compute $v(w - \tau_j e_j)$. From the definition of matrices M and $\tilde{M}$ we have that

$$M(w - \tau_j e_j) = M(w) - \tau_j x_j x_j', \quad \tilde{M}(w - \bar{\tau}_j e_j) = \tilde{M}(w) - \tau a_j a_j',$$

where $a_j = (x_j; y_j) \in \mathbb{R}^{n+1}$. Then,

$$v(w - \tau_j e_j) = \frac{\det(\tilde{M}(w))(1 - \tau a_j'\tilde{M}(w)^{-1}a_j)}{\det(M(w))(1 - \tau x_j'M(w)^{-1}x_j)} = v(w)\frac{1 - \tau a_j'\tilde{M}(w)^{-1}a_j}{1 - \tau x_j'M(w)^{-1}x_j}$$

and using that (Agulló, 2001, p. 429)

$$a_j'\tilde{M}(w)^{-1}a_j = r_j(w)^2/v(w) + x_j'M(w)^{-1}x_j,$$

where $r_j(w)^2 = (y_j - x_j'\beta_w)^2$ is the square of the jth residual, we obtain

$$v(w_0 - \tau_j e_j) - v(w_0) = \frac{-\tau_j r_j(w_0)^2}{1 - \tau_j x_j'M(w_0)^{-1}x_j}. \tag{II.2.7}$$

Finally, using the notation $\Delta \triangleq \gamma - v(w_0)$, we obtain an explicit formula for the step size $\bar{\tau}_j$ such that $v(w_0 - \bar{\tau}_j e_j) = \gamma$

$$\bar{\tau}_j = \frac{-\Delta}{\Delta x_j' M(w_0)^{-1} x_j - r_j(w_0)^2}. \tag{II.2.8}$$

Combined with the general theory presented in Section II.2.1, these calculations yield to the following result:

Proposition 1 *Let γ be a positive real number, and let w_0 be a point in $\mathcal{C}_h$ such that $\gamma < v(w_0)$ and $\det(X'D(w_0)X) \neq 0$. Then the vector $c \in \mathbb{R}^n$ defined as*

$$c_j = \frac{r_j(w_0)^2 - (\gamma - v(w_0)) x_j' (X'D(w_0)X)^{-1} x_j}{\gamma - v(w_0)} \tag{II.2.9}$$

if $(w_0)_j = 1$, and $c_j = 0$ otherwise, defines a valid γ-cut for the function v over $\mathcal{C}_h$, i.e

$$\mathcal{C}_h \cap \{c'(w - w_0) \leq 1\} \subseteq \{v \geq \gamma\}.$$

We would like to stress the fact that a closed-form expression for the concavity cut as in (II.2.9) is quite unusual. It is crucial in our calculations to work on the unit cube, firstly because we do not need to inverse the matrix U, and also because for obtaining the step-sizes $\bar{\tau}_j$ we have used the Sherman-Morrison formula for the inverse of a rank-one perturbation of a matrix. We see from (II.2.9) that the only possibility to obtain a cut parallel to the hyperplane $e'w = h$ is to take $w_0 = e$, which will never be the case in a BB algorithm, since all nodes have at most h non-zero components. In general the situation will be that of Figure (II.2.3).

II.2.3 Numerical experiments

In order to get a first idea of the efficiency of the concavity cuts for the LTS problem, we perform the following experiment. We randomly generate a standard normal sample X of size n in $\mathbb{R}^2$, and a response $y_i = \theta_0 + X\theta + \varepsilon$ with $\theta_0 = 1,\ \theta = (1; 1)$ and ε following a standard normal distribution. The number n is chosen small enough to be able to generate all the vertices of $\mathcal{C}_h$, corresponding to the number of subsamples of size h. Then, we derive some cuts and keep track of the percentage of vertices that have been eliminated so far.

In our implementation, we store the cuts as a linear system by defining

$$C = \begin{pmatrix} -c_1' \\ \vdots \\ -c_{nc}' \end{pmatrix}, \quad b = \begin{pmatrix} -(1 + c_1' w_1) \\ \vdots \\ -(1 + c_{nc}' w_{nc}) \end{pmatrix},$$

where $w_1, ..., w_{nc}$ are the vertices used to construct the cuts. With this notation, points $w \in \mathcal{C}$ such that $Cw > b$ are eliminated by concavity cuts.

First of all, we need a "good" initial solution, whose function value will be used as γ level for generating concavity cuts. In order to obtain such a "good" initial point, we randomly generate a number of candidates and apply to each of them a "concentration step", that consists in repeating iteratively the following steps (Rousseeuw and Driessen, 2006):

n	# subsamples	1 cut	2 cuts	3 cuts	4 cuts	5 cuts
21	5985	39.1	56.2	67.9	75.75	85.3
22	26334	47.5	70.5	83.9	90.8	94.2
23	100947	31.8	52.5	65.1	75.8	80.0
24	346104	28.3	40.7	55.8	59.0	70.9
25	1081575	11.7	26.1	40.1	51.0	57.1

Table II.2.1: Percentage of the vertices of $\mathcal{C}$ ruled out by the concavity cuts, with $h = 17$.

Given a candidate w:

- Compute $\hat{\beta}$ as the LS fit to the observations selected by w, the residuals vector $\mathbf{r}$ and the ordered residuals $|r_{\pi(1)}| \leq |r_{\pi(2)}| \leq \cdots \leq |r_{\pi(n)}|$, where π is the ordering, $|r_{\pi(i)}| = |r|_{(i)}$.
- Set $\bar{w}_{\pi(1)} = \bar{w}_{\pi(2)} = \cdots = \bar{w}_{\pi(h)} = 1$ and 0 otherwise.

The best of the generated candidates after the previous procedure becomes the incumbent solution $\bar{w}$. Alternative methods for obtaining the improved candidate are presented in Agulló (2001) and Schyns et al. (2010). Here, we described the simplest one.

The cuts are generated by Procedure 4.

Procedure 4 Procedure for generating concavity cuts.

1. Draw an initial point w_0 randomly, apply the concentration steps described above to obtain the incumbent solution $\bar{w}$. Set $w_k = w_0$, $\gamma = v(\bar{w})$ and $C, b = []$.
2. Iterate
   - Derive a concavity cut from w_k, add it to C, b.
   - Let s_C be the row vector of the sums of each column of C. Set the component j of w_k equal to 1 if j is one of the h smallest components of s_C.
   - Test whether $Cw_k \leq b$, if so iterate, otherwise break.

The results are shown in Tables II.2.1 and II.2.2. In Table II.2.1 we show the results obtained when fixing $h = 17$ and varying the number of observations from $n = 21$ to $n = 25$. In Table II.2.2 the parameter h changes as a function of n as $h = \text{round}(3n/4)$, for n in the same range. These results show that for the tested problem the bundle of cuts obtained from the very simple Procedure 4 rules out in all cases more than a half of the subsamples. This evidence suggests that the feasible domain of problem II.1.10 (with cuts) would be significantly narrower than that of problem II.1.8 (without cuts).

n	h	# subsamples	1 cut	2 cuts	3 cuts	4 cuts
21	16	20349	41.4	65.1	71.8	83.2
22	17	26334	36.2	68.2	82.4	89.4
23	17	100947	20.1	37.8	54.7	61.1
24	18	134596	38.1	60.4	73.9	85.8
25	19	177100	21.1	54.5	65.8	72.8

Table II.2.2: Percentage of the vertices of $\mathcal{C}$ ruled out by the concavity cuts, with $h = 3n/4$.

II.2.4 Some final comments

In the context of a BB algorithm, concavity cuts will be derived from the leaves that are visited and from nodes that are pruned by the lower bound criterion. The distance from a vertex w_0 to the hyperplane $c'(w - w_0) = 1$ equals $1/\|c\|$, and is often used as a measure of the depth of the cut. Therefore, "sparse" vertices would result in deeper cuts and it is expected that cuts from pruned nodes would yield deeper cuts than cuts from leaves.

In large trees, adding one cut each time a node is pruned or a leaf is explored can result in a very large number of cuts. Numerical experiments reported by Alarie et al. (2001) suggest that better results are obtained by keeping only a fraction of the cuts. It is suggested to drop cuts that are nearly collinear to existing ones.

Chapter II.3

The bilinear formulation and SDP-type relaxations

In this chapter we reformulate the MINLP (II.1.9), whose optimal value equals the sum of trimmed squares of the residuals of the LTS estimator, with its domain reduced by concavity cuts, as a multi-quadratic problem. The objective of those relaxations is to obtain lower bounds for the optimal value of (II.1.9), or lower bounds for the least value of a branch in the BB tree when they are complemented with node constraints.

Recall from (II.2.5) that, for w such that $\det(X'D(w)X) \neq 0$,

$$v(w) = y'D(w)r(\beta_w),$$

for the unique β_w satisfying $X'D(w)r(\beta_w) = 0$. As a consequence, the function v can be equivalently expressed, using that $y'D(w) = w'D(y)$, as

$$\begin{array}{rll} v(w) = & \min\limits_{\beta\in\mathbb{R}^d} & w'D(y)r(\beta) \\ & s.t & \\ & & X'D(w)r(\beta) = 0, \end{array} \tag{II.3.1}$$

Therefore, problem (II.1.9) is equivalent to the quadratically constrained quadratic problem (also called multi-quadratic problem, or even Q^2P)

$$\begin{array}{ll} \min\limits_{w,\beta} & w'D(y)r(\beta) \\ \text{s.t} & \\ & X'D(w)r(\beta) = 0, \\ & Cw \leq b, \\ & e'w = h, \\ & w \in \{0,1\}^n, \\ & \beta \in \mathbb{R}^d. \end{array} \tag{II.3.2}$$

Problem (II.3.2) involves only linear terms in w and bilinear terms of the form $w_k\beta_j$. It is possible, using a technique dating back to Glover (1975), to linearize crossed products of integer and continuous variables by introducing a new variable

R_{kj} along with a set of linear constraints to enforce that R_{kj} equals $w_k\beta_j$ at all binary realizations of w. This technique was used in Adams and Sherali (1993) to treat mixed-integer bilinear programs under decoupled constraints $w \in P_1 \cap \{0,1\}^n, \beta \in P_2$ for two bounded polyhedral sets P_1 and P_2. Nevertheless, those linearizations ask the user to provide explicit bounds on β, which are then incorporated in the linearized problem. If such bounds are not available, which is the case of general regression problems, the problem can be solved with a "big box" constraint, but the introduction of such large numbers results in weaker continuous relaxations (c.f. Adams et al., 2004). Note that in our problem, obtaining such a bound before computing the robust regression coefficient would mean that we can obtain reliable information about the "true" or "clean" regression coefficient β from the eventually contaminated data X, y. For these reasons, in the next section we explore relaxations that suit better the structure of our problem.

II.3.1 SDP and SOC relaxations

In semidefinite programming (SDP) one minimizes a linear function, possibly under linear constraints, over the cone of positive semidefinite matrices. Although semidefinite programs are a pretty general class of problems, they can be solved very efficiently by interior-point methods. Semidefinite programming has drawn a lot of attention in the last years, mainly because of its success in obtaining tight lower bounds for hard combinatorial optimization problems. Vandenberghe and Boyd (1996) give an excellent survey on the theory and applications of semidefinite programming. In this section we analyze the SDP relaxation of problem (II.3.2), and then we shall propose a derived relaxation which is better adapted to our problem.

We start by putting problem (II.3.2) in quadratic form. The objective function $w'D(y)r(\beta)$ and the constraint $X'D(w)r(\beta) = 0$ can be written as $z'\tilde{Q}_0 z + q_0'w$, and

$$z'\tilde{Q}_l z + q_l'w = 0 \quad l = 1, .., m,$$

respectively, with $z = (\beta, w) \in \mathbb{R}^\zeta$, $\zeta := n + d$; and using, for $l = 0, 1, ..., d$, the block matrix

$$\tilde{Q}_l = \begin{pmatrix} 0_d & \boxed{M_l'} \\ \boxed{M_l} & 0_n \end{pmatrix},$$

where the $n \times d$ blocks M_l and the $n \times 1$ vectors q_l equals

$$M_0 = -\frac{1}{2}D(y)X, \quad q_0 = y^2,$$

for $l = 0$, and for $l = 1, ..., d$,

$$(M_l)_{ij} = X_{il}X_{ij}, \quad (q_l)_i = y_i X_{il}.$$

For problem (II.3.2) in the form:

$$\begin{array}{ll} \min\limits_{w,z} & z'\tilde{Q}_0 z + q_0' w \\ \text{s.t} & \\ & z'\tilde{Q}_l z + q_l' w = 0, \quad l = 1,..,m, \\ & Cw \leq b, \\ & e'w = h, \\ & w_j^2 = w_j \quad j = 1,...,n, \\ & z \in \mathbb{R}^\zeta, \end{array} \tag{II.3.3}$$

the SDP relaxation is obtained by lifting problem (II.3.3) in a matrix space, using the fact that

$$z'Qz = \text{trace}(z'Qz) = \text{trace}(Qzz') = \langle Q, zz' \rangle.$$

Note that since the variable z is partitioned as $z = (\beta, w)$, then the rank-one matrix zz' has the following block structure:

$$zz' = \begin{pmatrix} \beta\beta' & \beta w' \\ w\beta' & ww' \end{pmatrix}. \tag{II.3.4}$$

Next we replace the product of vector variables zz' by the matrix variable $\tilde{Z} \in \mathbb{S}_\zeta$,

$$\tilde{Z} = \begin{pmatrix} B & R' \\ R & W \end{pmatrix} \tag{II.3.5}$$

with $B \in \mathbb{S}_d, R \in \mathbb{R}^{n\times d}$ and $W \in \mathbb{S}_n$, and we add the constraint $\tilde{Z} = zz'$. The constraints $w_j^2 = w_j$ become $W_{jj} = w_j$, which can be written compactly as $d(W) = w$. Thus, the multi quadratic problem with cuts (II.3.3) is equivalent to

$$\begin{array}{ll} \min\limits_{w\in\mathbb{R}^n,\ \tilde{Z}\in\mathbb{R}^{n\times d}} & \langle \tilde{Q}_0, \tilde{Z} \rangle + q_0' w \\ s.t & \\ & \langle \tilde{Q}_l, \tilde{Z} \rangle + q_l' w = 0, \quad l = 1,..,m, \\ & Cw \leq b, \\ & e'w = h, \\ & d(W) = w, \\ & \tilde{Z} - zz' = 0. \end{array} \tag{II.3.6}$$

Let us introduce, for $l = 0, 1, ..., d$, the matrices

$$Q_l = \begin{pmatrix} 0 & q_l'/2 \\ q_l/2 & \tilde{Q}_l \end{pmatrix} = \begin{pmatrix} 0_{d+1} & \begin{array}{|c|} \hline q_l' \\ \hline M_l' \\ \hline \end{array} \\ \begin{array}{|c|c|} \hline q_l & M_l \\ \hline \end{array} & 0_n \end{pmatrix}.$$

The SDP relaxation (which coincides with the Lagrangian relaxation) is obtained by relaxing the non-convex constraint $\tilde{Z} - zz' = 0$ to $\tilde{Z} - zz' \succcurlyeq 0$, which can be

expressed equivalently as $Z \succcurlyeq 0$, with a matrix $Z \in \mathbb{S}_{\zeta+1}$ structured as

$$Z = \begin{pmatrix} 1 & \beta' & w' \\ \beta & B & R' \\ w & R & W \end{pmatrix}.$$

In this way we end up with the SDP relaxation of problem (II.3.2):

$$(SDP) \begin{cases} \min\limits_{w \in \mathbb{R}^n,\ Z \in \mathbb{S}_{\zeta+1}} & \langle Q_0, Z \rangle \\ s.t & \\ & \langle Q_l, Z \rangle = 0, \quad l = 1, .., m, \\ & Cw \leq b, \\ & e'w = h, \\ & d(W) = w, \\ & Z \succcurlyeq 0. \end{cases} \tag{II.3.7}$$

Since the data matrices Q_l involved in problem (II.3.7) are sparse, the $d \times d$ block B and the off-diagonal elements of the $n \times n$ block W appears only in the semidefinite constraint. Thus, the solution to problem (II.3.7) cannot be unique, since for a given feasible pair (w, Z), adding a positive-definite $d \times d$ matrix to the block B of Z will keep the pair feasible without changing the objective value of the pair. Besides of introducing a number of unnecessary variables, the non-uniqueness is undesirable from a computational point of view.

Kim et al. (2003) introduced a transparent approach for dealing with sparsity. It consists in replacing the SDP constraint by second order cone (SOC) constraints that are implied by the SDP constraint. The second order cone $\mathcal{K}_u \subseteq \mathbb{R}^u$ is defined as $\mathcal{K}_u = \{z \in \mathbb{R}^u : z_1 \geq \|(z_2; z_3; ...; z_u)\|\}$; a constraint of the type $z \in \mathcal{K}_u$ is called a SOC constraint. The condition $Z \succcurlyeq 0$ implies that any principal submatrix of Z is positive semidefinite. Saying that each 1×1 principal submatrix is positive semidefinite amounts to imposing the positivity constraints $Z_{ii} \geq 0,\ i = 1, ..., \zeta$. The 2×2 principal submatrices are positive semidefinite if and only if $(Z_{kj})^2 \leq Z_{kk} Z_{jj}$ for any pair of indices (k, j). Together, they are equivalent to the SOC constraints

$$\left\| \begin{pmatrix} Z_{kk} - Z_{jj} \\ 2Z_{kj} \end{pmatrix} \right\| \leq Z_{kk} + Z_{jj}, \text{or } \begin{pmatrix} Z_{kk} + Z_{jj} \\ Z_{kk} - Z_{jj} \\ 2Z_{kj} \end{pmatrix} \in \mathcal{K}_3.$$

In our problem, for $k = 1, d + 2 \leq j \leq \zeta + 1$ we obtain $w_j^2 \leq W_{jj}$, which, combined with the constraint $d(W) = w$, will imply that $w_j^2 \leq w_j$, then $0 \leq w_j, W_{jj} \leq 1$. Note that the introduction of the matrix variable W becomes useless, since it only appears due to the constraint $w_i^2 - w_i = 0$, which will be relaxed to $w_i \in [0, 1]$. If we choose $2 \leq k \leq d + 1$ and $d + 2 \leq j \leq \zeta + 1$, we obtain

$$R_{kj}^2 \leq B_{kk} W_{jj}.$$

This constraint expresses the logical implication $W_{jj} = 0 \Rightarrow R_{kj} = 0$, bringing into play the diagonal of the matrix variable B. Ignoring inequalities involving the off-diagonal elements of the blocks B and W, we obtain the SOC relaxation of problem (II.3.2):

$$(SOC)\begin{cases}\min & 2\langle M_0, R\rangle + q_0' w \\ s.t & \\ & 2\langle M_l, R\rangle + q_l' w = 0, & l = 1,..,m, \\ & Cw \leq b, \\ & e'w = h, \\ & \|(\varpi_k - w_j; 2R_{kj})\| \leq \varpi_k + w_j, & k = 1,...,d;\ j = 1,...,n, \\ & w_j \leq 1, & j = 1,...,n, \\ & w \in \mathbb{R}^n,\ R \in \mathbb{R}^{n\times d},\ \varpi \in \mathbb{R}^d. \end{cases} \tag{II.3.8}$$

This time we keep only the diagonal of the block B in the vector variable $\varpi \in \mathbb{R}^d$; we completely drop the block W, and we introduce directly the constraint $w_j \in [0,1]$ instead.

Problem (II.3.8) is a convex problem, and can be efficiently solved using interior points methods (see Alizadeh and Goldfarb, 2003, for a comprehensive exposition of SOC programming). The only annoying point with it is the non-uniqueness of the variable ϖ. This inconvenient can be eliminated by adding a term of the form $\mu e'\varpi$, for small $\mu > 0$ to the objective function in order to achieve uniqueness of ϖ without perturbing the solution of the problem. Note that if we had a bound for the variable β, say $|\beta_k| \leq M_k$, we could eliminate the variable ϖ replacing the SOC constraint $R_{kj}^2 \leq b_k w_j$ by $R_{kj}^2 \leq M_k^2 w_j$. This is in fact one of the linear inequalities added by the above-mentioned linearization proposed in Adams and Sherali (1993).

The SOC relaxation can be seen as a compromise between the SDP relaxation that gives tight bounds at a strong computational cost, and linear programming relaxations, which are cheaper from a computational point of view but give much weaker bounds.

II.3.2 Global optimality conditions

The objective of this section is to describe a reformulation of problem (II.3.2) that keeps only the relevant crossed terms $w_j\beta_k$ in bilinear programs. It is based on a convenient writing of the condition $R = w\beta'$ as a rank-one constraint, followed by a formulation of that constraint as the difference of two convex functions.

Let us reconsider problem (II.3.8), with the SOC constraints replaced by the exact definition of R as $w\beta'$,

$$\begin{array}{lll}\min & 2\langle M_0, R\rangle + q_0' w & \\ s.t & & \\ & 2\langle M_l, R\rangle + q_l' w = 0, & l = 1,..,m, \\ & Cw \leq b, & \\ & e'w = h, & \\ & R = w\beta', & \\ & 0 \leq w_j \leq 1, & j = 1,...,n, \\ & w \in \mathbb{R}^n, \quad R \in \mathbb{R}^{n\times d}, & \beta \in \mathbb{R}^d. \end{array} \tag{II.3.9}$$

Then we use the fact that

$$R = w\beta' \quad \text{if and only if} \quad \text{rank}\begin{pmatrix} 1 & \beta' \\ w & R \end{pmatrix} = 1,$$

to obtain:

$$\begin{array}{ll} \min & \langle A_0, Z\rangle \\ s.t & \\ & \langle A_l, Z\rangle = 0,\ l = 1, ..., m, \\ & CZ(2 : n+1, 1) \leq b, \\ & \sum_{j=2}^{n+1} Z_{j1} = h, \\ & Z_{11} = 1, \\ & 0 \leq Z_{j1} \leq 1,\ j = 2, ..., n+1, \\ & \text{rank}(Z) = 1, \\ & Z \in \mathbb{R}^{(n+1)\times(d+1)}, \end{array}$$

where $A_l = \begin{pmatrix} 0 & 0_{1\times d} \\ q_l & 2M_l \end{pmatrix}$, $l = 0, 1, ..., m$, and $Z(2 : n+1, 1)$ denotes the n last elements of the first column of the matrix Z. We introduce two additional matrices A_{m+1} and A_{m+2} to express the constraints $\sum_{j=2}^{n+1} Z_{j1} = h$ and $Z_{11} = 1$ respectively,

$$A_{m+1} = \begin{pmatrix} 0 & 0_{1\times d} \\ e & 0_{n\times d} \end{pmatrix}, \quad A_{m+2} = \begin{pmatrix} 1 & 0 \\ 0 & 0, \end{pmatrix},$$

and the matrices $C_i,\ i = 1, ..., nc$, to express the constraint $CZ(2 : n+1, 1) \leq b$ as $\langle C_i, Z\rangle \leq b_i,\ i = 1, ..., nc$. Each matrix C_i has only zeros, except for its first column that equals $(0; c_i)$, where c_i is the ith concavity cut (stored in the matrix C).

Let $\mathcal{A}_1 : \mathbb{R}^{(n+1)\times(d+1)} \to \mathbb{R}^{m+2}$ and $\mathcal{A}_2 : \mathbb{R}^{(n+1)\times(d+1)} \to \mathbb{R}^{nc}$ be defined as

$$\begin{aligned} \mathcal{A}_1 Z =& (\langle A_1, Z\rangle; ...; \langle A_m, Z\rangle; \langle A_{m+1}, Z\rangle; \langle A_{m+2}, Z\rangle), \\ \mathcal{A}_2 Z =& (\langle C_1, Z\rangle; ...; \langle C_{nc}, Z\rangle). \end{aligned}$$

and $f \in \mathbb{R}^{m+2}$ be the column vector defined by $f_i = 0,\ i = 1, ..., m$ and $f_{m+1} = f_{m+2} = 1$.

Then we can rewrite problem (II.3.9) in compact form as:

$$\left\{ \begin{array}{ll} \min & \langle A_0, Z\rangle \\ s.t & \\ & \mathcal{A}_1 Z = f, \\ & \text{rank}(Z) = 1, \\ & Z \in \Delta, \end{array} \right. \tag{II.3.10}$$

where $\Delta \subseteq \mathbb{R}^{(n+1)\times(d+1)}$ is the convex set

$$\Delta := \{Z \mid \mathcal{A}_2 Z \leq b,\ 0 \leq Z_{j1} \leq 1 \text{ for } j = 2, ..., n+1\}.$$

Note that problem (II.3.10) is an exact reformulation of problem (II.3.2) with the constraint $w_j \in \{0,1\}$ relaxed to $w_j \in [0,1]$. It is a linear problem in the matrix variable Z except for the rank-one constraint. The rank function is non-convex and not even continuous; in fact it is integer-valued and constant along rays. For these reasons, the study of optimization problems involving the rank function is quite recent. The Eckart-Young Theorem is possibly the only 'old' optimization result involving the rank function. It gives an explicit solution to the problem of minimizing the distance from an arbitrary matrix to the set of matrices of a given rank. A significant breakthrough in the study of the rank function has been made by Fazel (2002), by computing the convex envelope of the rank function restricted to a ball. This result is of great importance for problems involving the rank in the objective function. In the remaining of this section we shall focus rather on relaxations of optimization problems under rank constraints, paying particular attention to the rank-one constraint.

II.3.2.1 The rank-one constraint as the difference of convex functions

Let a given $n \times d$ matrix A have the singular value decomposition

$$A = UD(\varsigma)V',$$

where U and V are orthogonal matrices of size $n \times d$ and $d \times d$ respectively, and $\varsigma \in \mathbb{R}^d$ contains the singular values in decreassing order:

$$\varsigma_1 \geq \cdots \geq \varsigma_d \geq 0.$$

Let ϕ be a norm in $\mathbb{R}^d$, then we can define an associated matrix norm in $\mathbb{R}^{n\times d}$ as

$$\|A\|_\phi = \phi(\varsigma). \tag{II.3.11}$$

By putting $\phi = \|\cdot\|_2$ we obtain the norm associated to the Frobenius inner product

$$\|A\|_F^2 = trace(A'A) = \sum_{i=1}^{d} \varsigma_i^2.$$

A very important case in the sequel will be $\phi = \|\cdot\|_1$, which originates the nuclear norm

$$\|A\|_* = \sum_{i=1}^{d} |\varsigma_i|.$$

The following function will play an important role in the study of the rank-one constraint:

Definition 1 *Let us define the continuous surrogate of the rank as the function* $g : \mathbb{R}^{n\times d} \to \mathbb{R}$ *given by*

$$g(A) = \begin{cases} \frac{\|A\|_*^2}{\|A\|_F^2}, & \text{if } A \neq 0 \\ 0, & \text{if } A = 0. \end{cases} \tag{II.3.12}$$

This function has a close relationship with the rank function, as the following proposition shows:

Proposition 2 *The continuous surrogate of the rank, defined in* (II.3.12) *satisfies the following properties:*

G1. $g(\alpha A) = g(A), \quad \forall A \in \mathbb{R}^{n\times d}, \alpha \neq 0.$

G2. $1 \leq g(A) \leq \operatorname{rank}(A), \quad \forall A \in \mathbb{R}^{n\times d} \setminus \{0\}.$

G3. If all the non-zero singular values of A *are equal, then* $g(A) = \operatorname{rank}(A)$.

G4. $\operatorname{rank}(A) = 1$ *if and only if* $g(A) = 1$.

Proof. Property *G1* follows directly from the homogeneity of norms. As before, let $\varsigma_1 \geq \cdots \geq \varsigma_d \geq 0$ be the singular values of A. By the definition of $\|A\|_*$ and $\|A\|_F$, $g(A) = \|\varsigma\|_1^2/\|\varsigma\|_2^2$ provided that $A \neq 0$. In that case $\varsigma_1 > 0$, thus scaling by $\alpha = 1/\varsigma_1$ if necessary we can suppose that $\varsigma_1 = 1$. Then

$$1 \leq \|\varsigma\|_2 = \sqrt{\sum_{i=1}^{d} \varsigma_i^2} \leq \sum_{i=1}^{d} \varsigma_i^2 \leq \sum \varsigma_i, \tag{II.3.13}$$

and since all the singular values are non-negative $\|\varsigma\|_1 = \sum_{i=1}^{d} \varsigma_i$. As a consequence $\|\varsigma\|_2 \leq \|\varsigma\|_1$ and $g(A) \geq 1$ for any $A \neq 0$. Note also that equality holds in (II.3.13) if and only if ς has at most one non-zero component, which is precisely what *G4* states. For the upper bound we use that $\operatorname{rank}(A) = k$ if and only if $\varsigma_i > 0$ for $1 \leq i \leq k$ and $\varsigma_i = 0$ for $k+1 \leq i \leq d$. Then $\|\varsigma\|_1 = \sum_{i=1}^{k} \varsigma_i = e_k'\varsigma$, where e_k is the vector with ones in the first k entries and zeros elsewhere. Then, by the Cauchy-Schwarz inequality, $\|\varsigma\|_1 \leq \|e_k\|_2\|\varsigma\|_2 = \sqrt{k}\|\varsigma\|_2$, hence $g(A) \leq k = \operatorname{rank}(A)$ and *G2* is proved. For proving *G3* it suffices to consider the case $\varsigma_1 = \cdots = \varsigma_k = 1$. Then $\|\varsigma\|_1^2 = k^2$, $\|\varsigma\|_2^2 = k$ and $g(A) = k = \operatorname{rank}(A)$. □

Properties G1 and G2 show that the function g is a minorant of the rank function with the same homogeneity. We will be mainly interested in property G4, which has been proved in Malick (2007, Theorem 1) for the particular case of square symmetric matrices with only ones on the diagonal, which is a corner of the $\|\cdot\|_*$-ball of radius n. It can be directly obtained from *G3* as well. Reciprocally, the problem with fixed Frobenius norm has been studied by Fazel (2002), where they proved that, for any $M > 0$, $\|\cdot\|_*/M$ is the convex envelope of the rank function on the set $\{A \in \mathbb{R}^{n\times d} \mid \|A\|_F \leq M\}$. This result has provided a theoretical background to the so-called trace heuristic, that consists in replacing the rank function by the nuclear norm in rank minimization problems. Properties G1 and G4 say that, for $A \neq 0$, the rank-one constraint can be expressed as a difference-of-convex (dc) functions constraint,

Proposition 3 *Let* $A \in \mathbb{R}^{n\times d} \setminus \{0\}$, *then*

$$\operatorname{rank}(A) = 1 \iff \|A\|_* - \|A\|_F \leq 0. \tag{II.3.14}$$

This result permits to put the rank-one constraint in the context of dc programming. In this way we have at hand global optimality conditions, local minimization algorithms that exploits that structure and, of course, the advantage of working with continuous, differentiable functions instead of the integer-valued rank function.

II.3.2.2 Rank constraints as a reverse-convex constraint

In this subsection we shall give yet another formulation of the rank constraint as a *reverse convex* constraint. We say that a constraint is reverse convex if it can be expressed as $g(A) \geq 0$ for a convex function g, or equivalently as $h(A) \leq 0$ for a concave function h. This formulation has the advantage of describing the set $\{A \mid \operatorname{rank}(A) \leq k\}$ for any $1 \leq k \leq d$. Its weak point is that it is primarily formulated for square matrices.

Let us suppose that A is a square positive semidefinite matrix, and let $\varsigma_1 \geq \cdots \geq \varsigma_d \geq 0$ be the eigenvalues of A. Then (Horn and Johnson, 1985, p. 191)

$$\varsigma_{k+1} + \cdots + \varsigma_d \leq \operatorname{trace}(U'AU)$$

for any $U \in \mathcal{U}_k$, where $\mathcal{U}_k$ denotes the set of $n \times (n-k)$ matrices such that $U'U = I_n$. Moreover, equality holds if and only if the columns of U form an orthonormal basis of eigenvectors of the matrix A. In other words, we have

$$\varsigma_{k+1} + \cdots + \varsigma_d = \min_{U \in \mathcal{U}_k} \operatorname{trace}(U'AU). \tag{II.3.15}$$

As the function $A \mapsto \operatorname{trace}(U'AU)$ is linear for any $U \in \mathcal{U}_k$, then the function

$$h_k(A) = \min_{U \in \mathcal{U}_k} \operatorname{trace}(U'AU)$$

is concave, and by virtue of (II.3.15), we have proved

Proposition 4 *Let A be a $d \times d$ symmetric positive semidefinite matrix. Then*

$$\operatorname{rank}(A) \leq k \iff h_k(A) \leq 0.$$

Note that the function $\varsigma \mapsto \varsigma_{k+1} + \cdots + \varsigma_d$ is concave (use (II.1.3) to write it as a minimum of linear functions) as a function of the singular values, but the function $A \mapsto \varsigma_{k+1}(A) + \cdots + \varsigma_d(A)$, where $\varsigma(A)$ denote the vector of singular values of A, is not necessarily a concave function of the matrix A, even for square A (cf. Dacorogna and Marcellini, 1999, Theorem 7.8). For an optimal control problem, Kim and Moon (2006) use the characterization above to devise a penalty method for the rank constraint, since if $U \in \mathcal{U}_k$ then $\operatorname{rank}(A) > k \Rightarrow \operatorname{trace}(U'AU) > 0$, thus a term of the form $\alpha \operatorname{trace}(U'AU)$ for $\alpha > 0$ and $U \in \mathcal{U}_k$ acts as penalization of the rank constraint violation.

II.3.2.3 Global optimality conditions using ε-subdifferentials

In this section, using the dc formulation of the rank-one constraint (II.3.14), we obtain global optimality conditions for problem (II.3.10).

Let us dualize the rank-one constraint, introducing a multiplier $\alpha > 0$

$$\begin{array}{ll} \min & \langle A_0, Z\rangle + \alpha(\|Z\|_* - \|Z\|_F) \\ s.t & \mathcal{A}_1 Z = f, \\ & Z \in \Delta, \end{array}$$

which we rearrange as:

$$\begin{array}{ll} \min & (\langle A_0, Z\rangle + \alpha\|Z\|_*) - \alpha\|Z\|_F \\ s.t & \mathcal{A}_1 Z = f, \\ & Z \in \Delta. \end{array} \tag{II.3.16}$$

Then, for each $\alpha > 0$, the value of problem (II.3.16) gives a lower bound for the original problem. A matrix $Z \in \Delta$ is a local optimum of problem (II.3.16) if there exists vectors $\mu \in \mathbb{R}^{m+2}, \lambda \in \mathbb{R}^n$ and $\nu \in \mathbb{R}^{nc}$ such that

$$\begin{array}{c} \mathcal{A}_1 Z = f, \\ \mathcal{A}_1^*\mu + \mathcal{A}_2^*\nu + \Lambda + \alpha\partial\|Z\|_F \subset \{A_0\} + \alpha\partial\|Z\|_*, \end{array} \tag{II.3.17}$$

where Λ is a matrix of zeros, except for the first column that equals $(0;\lambda)$. The multipliers λ and ν must also satisfy the complementarity conditions

$$\lambda_i \in \begin{cases} \{0\} & \text{if } 0 < z_i < 1, \\ (-\infty, 0] & \text{if } z_i = 0, \\ [0, \infty) & \text{if } z_i = 1, \end{cases} \tag{II.3.18}$$

and

$$D(\nu)(b - \mathcal{A}_2 Z) \leq 0,$$

respectively. Needless to say, if the solution of problem (II.3.16) is a rank-one matrix, then we have obtained the global solution to problem (II.3.10).

Next we recall the characterization given in Watson (1992) for the subdifferential of norms defined as in (II.3.11), at a matrix $A = UD(\varsigma)V'$,

$$\partial\|A\|_\phi = \text{conv}\{UD(s)V', s \in \partial\phi(\varsigma)\}.$$

this result extends naturally to ε-subdifferentials

$$\partial_\varepsilon\|A\|_\phi = \text{conv}\{UD(s)V', s \in \partial_\varepsilon\phi(\varsigma)\}, \quad \varepsilon \geq 0. \tag{II.3.19}$$

Similarly, since each ℓ_q norm is the support function of the unit ball for the dual norm $\ell_{\bar{q}}$, with $1/q + 1/\bar{q} = 1$,

$$\|\cdot\|_1 = \delta(\cdot|B_\infty), \quad \|\cdot\|_2 = \delta(\cdot|B_2)$$

we have (see Hiriart-Urruty and Lemaréchal, 1993b, pp. 97)

$$\partial_\varepsilon\delta(d, C) = \{s \in C : \delta(d, C) \leq s'd + \varepsilon\}, \quad \varepsilon \geq 0.$$

Then, we obtain the expression for the ε-subgradient of a norm:

$$\partial_\varepsilon \|d\|_q = \{s \in B_{\bar{q}} : \|d\|_q \leq s'd + \varepsilon\}.$$

Note that by Hölder inequality $s'd \leq \|s\|_{\bar{q}}\|d\|_q$. Thus, for $s \in B_{\bar{q}}$, $s'd \leq \|d\|_q$, and in the particular case $\varepsilon = 0$ we have an equality:

$$\partial \|d\|_q = \{s \in B_{\bar{q}} : \|d\|_q = s'd\}.$$

An interesting fact about ε-subgradients of norms, emanating directly from the later characterization, is that, for any $\varepsilon \geq 2\|\varsigma\|_1$,

$$\partial_\varepsilon \|\varsigma\|_2 = B_2, \quad \text{and} \quad \partial_\varepsilon \|\varsigma\|_1 = B_\infty.$$

This will be useful since for dc. programs there exist global optimality conditions which are stated precisely in terms of ε-subgradients.

Proposition 5 *A feasible matrix $Z \in \Delta$ is a* global *optimum of problem* (II.3.16) *if, for every $0 \leq \varepsilon \leq 2\|\varsigma\|_1$, where ς is the vector of singular values of Z, there exist $\varepsilon_1, \varepsilon_2$ satisfying $\varepsilon_1 + \varepsilon_2 = \varepsilon$, and vectors $\lambda \in N_{\varepsilon_2}$, $\mu \in \mathbb{R}^{m+2}$ and $\nu \in \mathbb{R}^{nc}$ such that,*

$$\mathcal{A}Z = 0, \tag{II.3.20}$$

$$\mathcal{A}_1^*\mu + \mathcal{A}_2^*\nu + \Lambda + \alpha\partial_{\varepsilon/\alpha}\|Z\|_F \subset \{A_0\} + \alpha\partial_{\varepsilon_1/\alpha}\|Z\|_*, \tag{II.3.21}$$

$$D(\nu)(b - \mathcal{A}_2 Z) \leq \varepsilon_2, \tag{II.3.22}$$

where N_{ε_2} is the set of all vectors $\xi \in \mathbb{R}^n$ whose components verify:

$$\xi_j \in \begin{cases} \{0\} & \text{if } 0 < Z_{j1} < 1, \\ (-\infty, \varepsilon_2] & \text{if } Z_{j1} = 0, \\ [-\varepsilon_2, \infty) & \text{if } Z_{j1} = 1, \end{cases} \tag{II.3.23}$$

and Λ is as before.

Proof. Let us denote by H the kernel of the linear operator $\mathcal{A}_1$. Note that H is a subspace of $\mathbb{R}^{(n+1)\times(d+1)}$. Then, rewrite the constraints of problem (II.3.16) using indicator functions, defined as

$$I_E(x) = \begin{cases} 0 & \text{if } x \in E, \\ +\infty & \text{if } x \notin E, \end{cases}$$

our problem becomes

$$\min\left[\alpha\|Z\|_* + I_H(Z) + I_\Delta(Z)\right] - \left[\alpha\|Z\|_F + \langle -A_0, Z\rangle\right].$$

This is a dc program without constraints, whose global optimality conditions are

$$\partial_\varepsilon\left[\alpha\|Z\|_F + \langle -A_0, Z\rangle\right] \subseteq \partial_\varepsilon\left[\alpha\|Z\|_* + I_{\{f\}+H}(Z) + I_\Delta(Z)\right]$$

by using Hiriart-Urruty and Lemaréchal (1993b, Prop. XI.1.3.1.(ii) and (vi)) we see that the left-hand of the inclusion equals $-A_0 + \alpha\partial_{\varepsilon/\alpha}\|Z\|_F$. For the right-hand side, we have from Hiriart-Urruty and Lemaréchal (1993b, Prop. XI.1.3.2), that for $Z \in \{f\} + H$

$$\partial_\varepsilon(\|Z\|_* + I_{\{f\}+H}(Z)) = \partial_\varepsilon\|Z\|_* + H^\perp.$$

Then, since $\mathrm{dom}(\|\cdot\|_* + I_{\{f\}+H}) \cap \mathrm{dom}(I_\Delta) \neq \emptyset$, we can apply Hiriart-Urruty and Lemaréchal (1993b, Theo. XI.3.1.1.) to obtain,

$$\partial_\varepsilon\left[\alpha\|Z\|_* + I_H(Z) + I_\Delta(Z)\right] = \bigcup_{\varepsilon_1+\varepsilon_2=\varepsilon} \left(\alpha\partial_{\varepsilon_1/\alpha}\|Z\|_* + \partial_{\varepsilon_2} I_\Delta\right) + H^\perp.$$

The final form follows by using that $\ker(\mathcal{A}_1)^\perp = \mathrm{Im}(\mathcal{A}_1^*)$, and that $\partial_{\varepsilon_2} I_\Delta(Z) = \Lambda + \mathcal{A}_2^*\nu$.
□

Note that in obtaining such a "closed" form of the global optimality conditions we have greatly benefited from the linear structure of our problem, since ∂_ε is a singleton only for affine functions (cf. Hiriart-Urruty and Lemaréchal, 1993b, prop. 1.2.4).

Just to illustrate the mechanism, let us consider the toy example of pure rank minimization over non-zero matrices:

$$\min_{R\in\mathbb{R}^{n\times d}\setminus\{0\}} \|R\|_* - \|R\|_F,$$

whose solution set is obviously the ray of rank-one matrices. The global optimality condition reads:

$$\partial_\varepsilon\|Z\|_F \subset \partial_\varepsilon\|Z\|_*.$$

Let $Z = UD(\varsigma)V'$ be a non-optimal candidate, then there exists $\varepsilon > 0$ such that

$$\partial_\varepsilon\|Z\|_F \setminus \partial_\varepsilon\|Z\|_* \neq \emptyset,$$

this can occur only if

$$\partial_\varepsilon\|\varsigma\|_2 \setminus \partial_\varepsilon\|\varsigma\|_1 \neq \emptyset,$$

and

$$s \in \partial_\varepsilon\|\varsigma\|_2 \setminus \partial_\varepsilon\|\varsigma\|_1 \Longrightarrow s \in B_2,\ \|\varsigma\|_2 \leq s'\varsigma + \varepsilon \leq \|\varsigma\|_1$$

but we know that in general $\|\varsigma\|_2 \leq \|\varsigma\|_1$ with equality only if ς has at most one non-zero component.

Proposition 5 is not directly related to the original problem (II.1.5), since it gives a characterization of the global optimum of a transformation of the same problem with the integer constraint relaxed. Its interest comes mainly from a global optimization viewpoint, because it gives global optimality conditions for bilinear problems, for which the SDP relaxation does not apply directly.

Conclusions and perspectives

We have studied some global optimization problems involved in the computation of high breakdown point regression estimators. Due to their advantageous statistical properties, we consider robust estimators based on M-scales and robust estimators based on L-scales. Robust estimators based on M-scales, and more particularly τ-estimators of regression, can be tuned to have the desired breakdown point and also the desired efficiency, therefore they are preferable from a statistical point of view. A closer look shows that the efficiency of τ-estimators is obtained by giving a positive weight to every observation with small *scaled residuals*. In contrast, estimators based in L-scales trim a fraction of the observations independently of the value of their residuals. However, in practice the rigidity of the L-scales gives to the optimization problem a combinatorial structure that permits, time considerations aside, to find a global optimum. Concerning robustness, for the LTS estimator the number of leaves in the branch-and-bound tree equals $\binom{n}{h}$, and is maximal for $h \sim n/2$, which gives the maximal breakdown point. In general, better estimators involve more difficult global optimization problems.

The first part of the thesis is devoted to the approximation of τ-estimators for robust regression. Considering factors such as the non existence of a characterization for the global optimum of the associated optimization problem and the unboundedness of the feasible domain that impedes an exhaustive inspection, we decided to study two-steps stochastic algorithms based on random subsampling and local searches that permit to take advantage of the differentiability of the problem. In Flores (2010), after reviewing existing algorithms for approximating τ-estimators and observing that most of them can be seen as restricted versions of clustering global optimization algorithms, we investigate the impact of incorporating clustering techniques and stopping conditions. The main conclusions of these extensive numerical tests are that stopping conditions improve the efficiency of existing algorithms, while clustering techniques work well in low dimension, but they harm the efficacy of the algorithm in middling and high dimension. The disappointing behavior of clustering techniques in moderate dimension led us to further investigate the reasons thereof in Chapter I.3. Since most clustering global optimization algorithms rely in some way on the *nearest neighbor*, we payed particular attention to the *gap phenomenon* affecting the notion of nearest neighbor in high dimension. We show how the quantile order of distances to points in the sample acts as an index in $[0, 1]$, equal to 0 for multistart and equal to 1 for best start, and that the gap phenomenon pushes algorithms to take extreme values and behave therefore as MS or BS. We propose as an alternative to replace distance thresholds by quantiles. The main strength of this approach, that motivates further research, is that it is independent

of the dimension, the feasible domain and the sampling distribution.

Concerning this last point, it is worth to note that clustering global optimization algorithms performing a concentration step (that consists in applying a few iterations of a local minimization routine to each sampled point, see Törn and Žilinskas (1989)), that performed very well in our tests of Chapter I.2, have been very little studied; mainly because the sample after the concentration steps is no longer uniformly distributed.

The choice of the sequence of quantile orders that characterizes the QGO algorithm remains as an open question. Guidelines for choosing such sequences are needed; the main difficulty in doing so is that our proposal consists not only in replacing radius by quantiles, but also in using more and more global information as the number of iterations increases. As a consequence, local techniques such as Taylor expansions will not be useful for the asymptotic analysis of the algorithm.

The key points of the first part are:

- We position the existing algorithms in the context of clustering global optimization.
- We introduce inexact iterations for accelerating the search for local minima.
- We perform extensive numerical tests in order to evaluate the impact of stopping conditions and of clustering techniques in existing algorithms for approximating the τ-estimator.
- We investigate the consequences of the gap phenomenon in clustering global optimization algorithms.
- We prove the convergence of a new clustering-type algorithm intended to be independent of the dimension, the optimization domain and the sampling distribution.

The main research direction emanating from the work in this part is undoubtedly the search for some criteria permitting to find "good" quantile order sequences for the QGO algorithm. At a longer term, what is needed is a better understanding of the consequences of the interactions between sampling distribution and feasible domain, such as the concentration of measures mentioned at the end of Chapter I.3.

The second part of the thesis is devoted to deterministic algorithms for computing the least trimmed squares regression estimator, which is defined through a nonlinear mixed-integer program. Due to the combinatorial nature of this problem, we concentrated on obtaining lower bounds to be used in a branch-and-bound algorithm. It is known that the associated problem can be written as a concave minimization problem over a polyhedral domain. Under this formulation, we obtain a closed-form expression for concavity cuts, that can be derived at a negligible computational cost. We also perform some preliminary numerical tests showing that concavity cuts eliminate a considerable fraction of the vertices of the feasible polyhedron, which could considerably tighten lower bounds obtained from relaxations of

the problem. Then, we show that the same problem can also be cast as a bilinear program with bilinear constraints, and study the associated positive-semidefinite and second-order cone programming relaxations. The last section deviates a few from the original motivation, and is devoted to global optimality conditions for the bilinear problem lifted into a matrix space. Before that, we introduce and study a continuous minorant of the rank function that deserves attention by its own, and we derive alternative formulations of some constraints involving the rank function as difference-of-convex (dc) or reverse convex constraints. Finally, we use the dc formulation of the rank-one constraint to obtain global optimality conditions for the bilinear problem.

The key points of the second part are:

- We obtain *explicit* concavity cuts.
- We give a reformulation of the original mixed-integer nonlinear problem as a bilinear problem with bilinear constraints.
- We propose a second-order cone programming relaxation to obtain lower bounds to be used in a branch-and-bound algorithm, reinforced by concavity cuts.
- We conduct a study of rank constraints, obtaining dc and reverse convex formulations of them.
- We obtain global optimality conditions for the bilinear problem.

We would have loved to test the tightness of the second-order cone programming relaxation in practice through numerical experiments, but unfortunately this (time-consuming!) task will remain at the place number one of the ToDo list. We expect to continue with the research lines that led to the reformulations of rank constraints, as we hope to contribute to the rapidly growing topic of *rank optimization*.

Let us conclude by mentioning the possible extensions of the work in this thesis to the computation of robust estimators for multivariate analysis. The experience with two-steps algorithms of Chapter I.2 will be very valuable for devising approximation algorithms for τ-estimators of multivariate location and scatter. In fact, some advances in this direction have been made during a stay of the author at the Department of statistics of the University of British Columbia in Vancouver, Canada. That piece of work (still in progress), in collaboration with Matias Salibian-Barrera and Ruben Zamar, led us to identify some issues specific to the multivariate case.

Bibliography

Adams, W. P., Forrester, R. J., Glover, F. W., 2004. Comparisons and enhancement strategies for linearizing mixed 0-1 quadratic programs. Discrete optimization 1 (2), 99–120.

Adams, W. P., Sherali, H. D., 1993. Mixed-integer bilinear programming problems. Mathematical Programming 59 (3), 279–305.

Aggarwal, C., Hinneburg, A., Keim, D., 2001. On the surprising behavior of distance metrics in high dimensional space. In: Van den Bussche, J., Vianu, V. (Eds.), Database Theory — ICDT 2001. Vol. 1973 of Lecture Notes in Computer Science. Springer, pp. 420–434.

Agulló, J., 1997. Computación de estimadores con alto punto de ruptura. Ph.D. thesis, Universidad de Alicante, Spain.

Agulló, J., 2001. New algorithms for computing the least trimmed squares regression estimator. Computational Statistics & Data Analysis 36 (4), 425–439.

Alarie, S., Audet, C., Jaumard, B., Savard, G., 2001. Concavity cuts for disjoint bilinear programming. Mathematical Programming 90 (2), 373–398.

Alizadeh, F., Goldfarb, D., 2003. Second-order cone programming. Mathematical Programming 95 (1), 3–51.

Archetti, F., Schoen, F., 1984. A survey on the global optimization problem: General theory and computational approaches. Annals of Operations Research 1, 87–110.

Ben-Tal, A., Nemirovski, A., 2008. Selected topics in robust convex optimization. Mathematical Programming 112, 125–158.

Betrò, B., 1991. Bayesian methods in global optimization. Journal of Global Optimization 1, 1–14.

Beyer, K., Goldstein, J., Ramakrishnan, R., Shaft, U., 1999. When is "nearest neighbor" meaningful? In: Beeri, C., Buneman, P. (Eds.), Database Theory — ICDT'99. Vol. 1540 of Lecture Notes in Computer Science. Springer, pp. 217–235.

Björck, Å., 1996. Numerical methods for least squares problems. Society for Industrial and Applied Mathematics (SIAM).

Boender, C. G. E., Rinnooy Kan, A. H. G., 1987. Bayesian stopping rules for multistart global optimization methods. Mathematical Programming 37 (1), 59–80.

Boender, C. G. E., Rinnooy Kan, A. H. G., Timmer, G. T., Stougie, L., 1982. A stochastic method for global optimization. Mathematical Programming 22 (2), 125–140.

Candes, E. J., Tao, T., 2005. Decoding by linear programming. IEEE Transactions on Information Theory 51 (12), 4203–4215.

Clarke, F., 1983. Optimization and nonsmooth analysis. John Wiley & Sons.

Dacorogna, B., Marcellini, P., 1999. Implicit partial differential equations. Birkhauser.

Donoho, D., Huber, P. J., 1983. The notion of breakdown point. In: A Festschrift for Erich L. Lehmann. Wadsworth Statist./Probab. Ser. Wadsworth, pp. 157–184.

Fazel, M., 2002. Matrix rank minimization with applications. Ph.D. thesis, Stanford University.

Flores, S., 2010. On the efficient computation of robust regression estimators. Computational Statistics & Data Analysis 54 (12), 3044–3056.

Giloni, A., Padberg, M., 2002. Least trimmed squares regression, least median squares regression, and mathematical programming. Mathematical and Computer Modelling 35 (9-10), 1043 –1060.

Giloni, A., Padberg, M., 2004. The finite sample breakdown point of ℓ_1-regression. SIAM Journal on Optimization 14 (4), 1028–1042.

Glover, F., 1975. Improved linear integer programming formulations of nonlinear integer problems. Management Science 22 (4), 455–460.

Hawkins, D. M., Olive, D., 1999. Applications and algorithms for least trimmed sum of absolute deviations regression. Computational Statistics & Data Analysis 32 (2), 119 – 134.

He, X., Jureckova, J., Koenker, R., Portnoy, S., 1990. Tail behavior of regression estimators and their breakdown points. Econometrica 58 (5), 1195 –1214.

Hiriart-Urruty, J.-B., Lemaréchal, C., 1993a. Convex Analysis and Minimization Algorithms I: Fundamentals. Vol. 305 of Grundlehren der mathematischen Wissenschaften. Springer-Verlag.

Hiriart-Urruty, J.-B., Lemaréchal, C., 1993b. Convex Analysis and Minimization Algorithms II: Advanced Theory and Bundle Methods. Vol. 306 of Grundlehren der mathematischen Wissenschaften. Springer-Verlag.

Horn, R., Johnson, C., 1985. Matrix analysis. Cambridge University Press.

Huber, P. J., 1981. Robust statistics. John Wiley & Sons, wiley Series in Probability and Mathematical Statistics.

Kaufman, L., Rousseeuw, P. J., March 1990. Finding Groups in Data: An Introduction to Cluster Analysis, 9th Edition. Wiley-Interscience.

Kim, S., Kojima, M., Yamashita, M., 2003. Second order cone programming relaxation of a positive semidefinite constraint. Optimization Methods & Software 18 (5), 535–541.

Kim, S., Moon, Y., 2006. Structurally constrained H_2 and H_∞ control: A rank-constrained LMI approach. Automatica 42 (9), 1583–1588.

Lasserre, J. B., 2001. Global optimization with polynomials and the problem of moments. SIAM Journal on Optimization 11 (3), 796–817.

Lasserre, J. B., 2006. Convergent SDP-relaxations in polynomial optimization with sparsity. SIAM Journal on Optimization 17 (3), 822–843 (electronic).

Ledoux, M., Talagrand, M., 1991. Probability in Banach spaces : isoperimetry and processes. Springer-Verlag.

Locatelli, M., Schoen, F., 1996. Simple linkage: Analysis of a threshold-accepting global optimization method. Journal of Global Optimization 9, 95–111.

Locatelli, M., Schoen, F., 1999. Random Linkage: a family of acceptance/rejection algorithms for global optimisation. Mathematical Programming 85 (2), 379–396.

Malick, J., 2007. The spherical constraint in Boolean quadratic programs. Journal of Global Optimization 39 (4), 609–622.

Maronna, R. A., Martin, R. D., Yohai, V. J., 2006. Robust statistics. Wiley Series in Probability and Statistics. John Wiley & Sons.

Mizera, I., Müller, C., 2001. The influence of the design on the breakdown points of ℓ_1-type m-estimators. In: Atkinson, A., Hackl, P., Müller, W. (Eds.), MODA6 – Advances in Model-Oriented Design and Analysis. Contributions to Statistics. Physica-Verlag, pp. 193–200.

Nguyen, T., Welsch, R., 2010. Outlier detection and robust covariance estimation using mathematical programming. Advances in Data Analysis and Classification.

Nocedal, J., Wright, S. J., 2006. Numerical optimization, 2nd Edition. Springer Series in Operations Research and Financial Engineering. Springer.

Piccioni, M., Ramponi, A., 1990. Stopping rules for the multistart method when different local minima have different function values. Optimization 21 (5), 697–707.

Rinnooy Kan, A. H. G., Timmer, G. T., 1987a. Stochastic global optimization methods. I. Clustering methods. Mathematical Programming 39 (1), 27–56.

Rinnooy Kan, A. H. G., Timmer, G. T., 1987b. Stochastic global optimization methods. II. Multilevel methods. Mathematical Programming 39 (1), 57–78.

Rousseeuw, P., Driessen, K. V., 2006. Computing lts regression for large data sets. Data Mining and Knowledge Discovery (12), 29–45.

Rousseeuw, P. J., 1984. Least median of squares regression. Journal of the American Statistical Association 79 (388), 871–880.

Rousseeuw, P. J., Driessen, K. V., 1998. A fast algorithm for the minimum covariance determinant estimator. Technometrics 41, 212–223.

Rousseeuw, P. J., Leroy, A. M., 1987. Robust regression and outlier detection. John Wiley & Sons.

Ruppert, D., 1992. Computing s estimators for regression and multivariate location/dispersion. Journal of Computational and Graphical Statistics 1 (3), 253–270.

Salibian-Barrera, M., Willems, G., Zamar, R., 2008. The fast-τ estimator for regression. Journal of Computational and Graphical Statistics 17 (3), 659–682.

Salibian-Barrera, M., Yohai, V. J., 2006. A fast algorithm for S-regression estimates. Journal of Computational and Graphical Statistics 15 (2), 414–427.

Schoen, F., 1991. Stochastic techniques for global optimization: a survey of recent advances. Journal of Global Optimization 1 (3), 207–228.

Schoen, F., 1998. Random and quasi-random linkage methods in global optimization. Journal of Global optimization 13, 445–454.

Schoen, F., 1999. Global optimization methods for high-dimensional problems. European Journal of Operational Research 119 (2), 345–352.

Schyns, M., Haesbroeck, G., Critchley, F., 2010. Relaxmcd: Smooth optimisation for the minimum covariance determinant estimator. Computational Statistics & Data Analysis 54 (4), 843 – 857.

Törn, A., Žilinskas, A., 1989. Global optimization. Vol. 350 of Lecture Notes in Computer Science. Springer-Verlag.

Tuy, H., 1964. Concave programming under linear constraints. Soviet Mathematics 5 (1), 1437–1440.

Vandenberghe, L., Boyd, S., 1996. Semidefinite programming. SIAM Review 38 (1), 49–95.

Waki, H., Kim, S., Kojima, M., Muramatsu, M., 2006. Sums of squares and semidefinite program relaxations for polynomial optimization problems with structured sparsity. SIAM Journal on Optimization 17 (1), 218–242 (electronic).

Watson, G. A., 1992. Characterization of the subdifferential of some matrix norms. Linear Algebra and its Applications 170, 33–45.

Yohai, V. J., Zamar, R. H., 1988. High breakdown-point estimates of regression by means of the minimization of an efficient scale. Journal of the American Statistical Association 83 (402), 406–413.

Author: Salvador FLORES
Title: Global optimization problems in robust statistics

Abstract

Robust statistics is a branch of statistics dealing with the analysis of data containing contaminated observations. The robustness of an estimator is measured notably by means of the *breakdown point*. High-breakdown point estimators are usually defined as global minima of a non-convex scale of the errors, hence their computation is a challenging global optimization problem. The objective of this dissertation is to investigate the potential contributions of modern global optimization methods to this class of problems.

The first part of this thesis is devoted to the τ-estimator for linear regression, which is defined as a global minimum of a nonconvex differentiable function. We investigate the impact of incorporating clustering techniques and stopping conditions in existing stochastic algorithms. The consequences of some phenomena involving the nearest neighbor in high dimension on clustering global optimization algorithms is thoroughly discussed as well.

The second part is devoted to deterministic algorithms for computing the least trimmed squares regression estimator, which is defined through a nonlinear mixed-integer program. Due to the combinatorial nature of this problem, we concentrated on obtaining lower bounds to be used in a branch-and-bound algorithm. In particular, we propose a second-order cone relaxation that can be complemented with concavity cuts that we obtain explicitly. Global optimality conditions are also provided.

Keywords: Robust regression, breakdown point, global optimization, curse of dimensionality, SOCP relaxation, concavity cuts, global optimality conditions

Auteur: Salvador FLORES
Titre: Problèmes d'optimisation globale en statistique robuste
Directeur(s) de Thèse: Anne Ruiz-Gazen et Marcel Mongeau
Soutenance: Le 25 février 2011 à l' Université Paul Sabatier

Résumé

La statistique robuste est une branche de la statistique qui s'intéresse à l'analyse de données contenant une proportion significative d'observations contaminées avec des erreurs dont l'ampleur et la structure peuvent être arbitraires. Les estimateurs robustes au sens du point de rupture sont généralement définis comme le minimum global d'une certaine measure non-convexe des erreurs, leur calcul est donc un problème d'optimisation globale très coûteux. L'objectif de cette thèse est d'étudier les contributions possibles des méthodes d'optimisation globale modernes à l'étude de cette classe de problèmes.

La première partie de la thèse est consacrée au τ-estimateur pour la régression linéaire robuste, qui est défini comme étant un minimum global d'une fonction non-convexe et dérivable. Nous étudions l'impact des techniques d'agglomération et des conditions d'arrêt sur l'efficacité des algorithmes existants. Les conséquences de certains phénomènes liés au voisin le plus proche en grande dimension sur ces algorithmes agglomératifs d'optimisation globale sont aussi mises en évidence.

Dans la deuxième partie de la thèse, nous étudions des algorithmes déterministes pour le calcul de l'estimateur de moindres carrés tronqués, qui est défini à l'aide d'un programme en nombres entiers non linéaire. En raison de sa nature combinatoire, nous avons dirigé nos efforts vers l'obtention des bornes inférieures pouvant être utilisées dans un algorithme du type *branch-and-bound*. Plus précisément, nous proposons une relaxation par un programme sur le cône de deuxième ordre, qui peut être renforcée avec des coupes dont nous présentons l'expression explicite. Nous fournissons également des conditions d'optimalité globale.

Mot-clefs: Regression robuste, point de rupture, conditions d'arrêt, optimisation globale, fléau de la dimension, relaxation SOCP, coupes de concavité, conditions d'optimalité globale

Discipline: Mathématiques appliquées

Institut de Mathématiques de Toulouse, UMR CNRS 5219
UFR MIG - Université Paul Sabatier, Toulouse III
118 route de Narbonne
F-31062 Toulouse Cedex 9