Evolutionary and Deterministic Methods for Design,
Optimization and Control with Applications to Industrial and Societal Problems
EUROGEN 2005
R. Schilling, W.Haase, J. Periaux, H. Baier, G. Bugeda (Eds)

ON COMPUTATION OF SPATIAL MEDIAN FOR ROBUST DATA MINING

T. Kärkkäinen and S. Äyrämö*

Department of Mathematical Information Technology
University of Jyväskylä, Jyväskylä, Finland
P.O.Box 35, FIN-40014 University of Jyväskylä
e-mail: tka@mit.jyu.fi and sami.ayramo@mit.jyu.fi

Key words: Data mining, clustering, robust estimation, spatial median.

Abstract. *Data mining (DM) is a process for extracting unexpected and novel information from very large databases. From the computational point of view, most data mining methods are based on statistical estimation which, in many cases, can be treated as an optimization problem. Because the sample mean, which is based on the least squares fitting, is very sensitive to extreme and missing values, robust and efficient multivariate location estimators are needed. In this paper, we discuss different formulations and techniques for solving the optimization problem underlying one particular robust estimate - the spatial median. Numerical comparison of the different methods on simulated data are presented and analyzed.*

1 INTRODUCTION

Clustering is one of the core tasks in data mining (DM) and knowledge discovery (KD). The idea of clustering is to search groups of similar observations from a multivariate dataset. Several different classifications for clustering methods have been introduced, such as, partitioning, hierarchical and density-based algorithms (see, e.g., [6] and [7]).

The context of this work is on the partitioning clustering methods (see, e.g., [9]) that are usually built on two basic steps: 1) assigment of each data point to its closest cluster prototype and 2) update of the cluster prototypes. In statistics, the second step is actually known as a location estimation problem. The cluster prototypes are supposed to be the most representative points of the data clusters and they are used to sketch the whole data set. Hence, in this paper, we focus on the computation of a particular prototype – the spatial median.

2 THE SPATIAL MEDIAN AS A NONSMOOTH OPTIMIZATION PROBLEM

Let $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be a sample of multivariate random variable $\mathbf{x}$ in the p-dimensional real Euclidean space $\mathbb{R}^p$. Throughout the paper, we denote by $(\mathbf{v})_i$ the ith component of a vector $\mathbf{v} \in \mathbb{R}^n$. Without parenthesis, $\mathbf{v}_i$ represents one element in the set of vectors $\{\mathbf{v}_i\}$. The Euclidean norm of a vector $\mathbf{v}$ is denoted by $\|\mathbf{v}\|$ and the maximum norm by $\|\mathbf{v}\|_\infty$.

For a statistician the spatial median is member of a class of so-called M-estimators [8]. M-estimators are defined by an optimization problem whose optimality condition is, in some cases, characterized by a nonsmooth function. Since the nonsmoothness is usually omitted in statistics, we define the optimality condition, in a mathematically correct way, using subdifferentials [10]. We compare the accuracy and efficiency of several algorithms using smooth reformulations of the problem. Considering the data mining problems, efficient algorithms are needed because of the large size of the data sets.

The problem of the spatial median is defined by

$$\min_{\mathbf{u}\in\mathbb{R}^p} \mathcal{J}_2^1(\mathbf{u}), \quad \text{for } \mathcal{J}_2^1(\mathbf{u}) = \sum_{i=1}^{n} \|\mathbf{u} - \mathbf{x}_i\|. \tag{1}$$

Notice, that in the univariate case (1) is equivalent to the coordinatewise sample median.

The gradient of the convex cost function $\mathbf{f}(\mathbf{u}, \mathbf{x}_i) = \|\mathbf{u} - \mathbf{x}_i\|_2$ is well-defined and unique for all $\mathbf{u} \neq \mathbf{x}_i$. However, case $\mathbf{u} = \mathbf{x}_i$ leads to the use of the subgradient, which is characterized by condition $\|\xi\|_2 \leq 1$. Thus, the (local) extremity of (1) is characterized by means of a subgradient, which reads as

$$\partial\mathcal{J}_2^1(\mathbf{u}) = \sum_{i=1}^{n} \xi_i, \quad \text{with } \begin{cases} (\xi_i)_j = \frac{(\mathbf{u}-\mathbf{x}_i)_j}{\|\mathbf{u}-\mathbf{x}_i\|}, & \text{for } \|\mathbf{u} - \mathbf{x}_i\| \neq 0, \\ \|\xi_i\| \leq 1, & \text{for } \|\mathbf{u} - \mathbf{x}_i\| = 0. \end{cases} \tag{2}$$

As pointed out in [10], problem (1) is a nonsmooth optimization problem [17], which means that it can not be described by using the classical (C^1) differential calculus.

2.1 Reformulation of the problem

In order to solve problem (1) using gradient-based optimization or iterative methods, smooth approximations are used. In the following, two smoothed approximating reformulations are proposed.

2.1.1 Modified gradient

This approach is based on a modified gradient. Since problem (1) is potentially nonsmooth only at a finite number of points, well-definiteness can simply be attained by replacing the denominator of the subgradient by a smoothing constant ε (a very small positive real number, e.g., 10^{-8}) whenever a data point coincide with a solution. Hence, the modified gradient reads as

$$\nabla \mathcal{J}_2^1(\mathbf{u}) = \sum_{i=1}^{N} \xi_i, \quad \text{with } (\xi_i)_j = \frac{(\mathbf{u}-\mathbf{x}_i)_j}{\max\{\|\mathbf{u}-\mathbf{x}_i\|, \varepsilon\}} \quad \text{for all} \quad \mathbf{x}_i. \tag{3}$$

By using the convention $\frac{0}{0} = 0$, this reformulation actually corresponds with a treatment where the coincident points are left out from computation (cf. the extended definition of the gradient by Kuhn [13]). This can be done since an overlapping data point does not effect on the value of the cost function (the distance from such a point to a solution equals to zero). Hence, by using the modified gradient (3) for the determination of the search direction, for example, CG method can be assumed to solve the nonsmooth problem (1).

2.1.2 ε-approximating problem

This approach smooths the original problem (1) by adding a small positive real number ε (e.g., $\varepsilon = 10^{-8}$) to the denominator [23],[16],[19] and [11]. The smooth perturbed ϵ-approximating problem is given by

$$\min_{\mathbf{u}\in\mathbb{R}^n} \mathcal{J}_\varepsilon(\mathbf{u}), \quad \text{for } \mathcal{J}_\varepsilon(\mathbf{u}) = \sum_{i=1}^{N} \sqrt{\|\mathbf{u}-\mathbf{x}_i\|^2 + \varepsilon}, \tag{4}$$

According to [19], the approximating cost function is uniformly convergent to the original cost function as the smoothing constant approaches zero. The gradient is again well-defined everywhere and reads as

$$\nabla \mathcal{J}_\varepsilon(\mathbf{u}) = \sum_{i=1}^{N} \xi_i, \quad \text{where } (\xi_i)_j = \frac{(\mathbf{u}-\mathbf{x}_i)_j}{\sqrt{\|\mathbf{u}-\mathbf{x}_i\|^2 + \varepsilon}}. \tag{5}$$

The smooth problem can be solved using a gradient-based optimization method or Weiszfeld-type of iterative algorithms. This reformulation with Weiszfeld algorithm has been applied, for example, to single and multifacility location problems (see [19] and [23]).

3 ITERATIVE METHODS FOR COMPUTING THE SPATIAL MEDIAN

The best-known iterative algorithm for solving the problem of the spatial median is the Weiszfeld algorithm[1] (see, e.g., [13]). It is based on the first-order necessary conditions for a stationary point of the cost function in (1), which provides the following iterative scheme:

$$\mathbf{u}^{k+1} = \begin{cases} \frac{\sum_{i=1}^{m} w_i \mathbf{x}_i / \|\mathbf{u}^k - \mathbf{x}_i\|}{\sum_{i=1}^{m} w_i / \|\mathbf{u}^k - \mathbf{x}_i\|}, & \text{if} \quad \mathbf{u}^k \notin \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}, \\ \mathbf{u}^k & \text{if} \quad \mathbf{u}^k = \mathbf{x}_i \quad \text{for some} \quad i = 1, \ldots, n. \end{cases} \tag{6}$$

In this paper, we restrict to $w_i = 1$ for all $i = 1, \ldots, n$, since we treat all data points with equal weights and hence diminish the required amount of prior information in data mining context. Defining $\mathbf{u}^{k+1} = \mathbf{u}^k$ the scheme becomes well-defined (continuous) for all $\mathbf{x} \in \mathbb{R}^p$. The discussion about convergence properties has evolved through the years, see, e.g., in [13],[12] [4],[1], [3] and [22].

A recent modification of the Weiszfeld algorithm is proposed by Vardi et al. [24] with a proof of monotonic convergence to the spatial median from any starting point in $\mathbb{R}^p$. In order to represent this modified Weiszfeld (MW) iteration, the following definitions are given

$$\eta(\mathbf{u}) = \begin{cases} 1, & \text{if} \quad \mathbf{u} \in \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}, \\ 0, & \text{otherwise.} \end{cases}$$

$$r(\mathbf{u}) = \left\| \sum_{\mathbf{x}_i \neq \mathbf{u}} \frac{\mathbf{x}_i - \mathbf{u}}{\|\mathbf{x}_i - \mathbf{u}\|} \right\|$$

The spatial median is attained by the following iterative process

$$\mathbf{u}^{k+1} = \max\left(0, 1 - \frac{\eta(\mathbf{u}^k)}{r(\mathbf{u}^k)}\right) \tilde{T}(\mathbf{u}^k) + \min\left(1, \frac{\eta(\mathbf{u}^k)}{r(\mathbf{u}^k)}\right) \mathbf{u}^k, \tag{7}$$

where $\tilde{T}(\mathbf{u}) = \left\{ \sum_{\mathbf{x}_i \neq \mathbf{u}} \frac{1}{\|\mathbf{u} - \mathbf{x}_i\|} \right\}^{-1} \sum_{\mathbf{x}_i \neq \mathbf{u}} \frac{\mathbf{x}_i}{\|\mathbf{u} - \mathbf{x}_i\|}$.

In [19],[18] and [23] modifications and convergence of a generalized Fermat-Weber problem, utilizing an approximating cost function as in (4), are studied.

[1]The original paper of the Weiszfeld [26] was not available to the authors of this paper, but a plenty of references can be found.

3.1 SOR accelerated Weiszfeld algorithm with perturbed problem reformulation

A number of acceleration methods, such as Steffensen's iteration and Aitken's δ^2 process, are given. See, for example, in [12], [5], [2], [25] and [15].

Here, a new accelerated iterative algorithm for solving (1) is proposed. The modified cost function from (4) is applied, which protects from the ill-defined gradient of the original problem (1). The basic iteration is based on the first-order necessary conditions for a stationary point of the cost function of the perturbed problem (4):

$$\sum_{i=1}^{n} \frac{\mathbf{u} - \mathbf{x}_i}{\sqrt{\|\mathbf{u} - \mathbf{x}_i\|^2 + \varepsilon}} = 0. \tag{8}$$

First, (8) is "linearized" by defining explicit weights using the denominator:

$$\alpha_i^k = \frac{1}{\sqrt{\|\mathbf{u}^k - \mathbf{x}_i\|^2 + \varepsilon}}. \tag{9}$$

Assuming that sample $\mathbf{X}$ does not contain empty columns (corresponds to a variable without any observations), the candidate solution $\mathbf{v}$ is solved, by combining (8) and (9), from

$$\sum_{i=1}^{n} \alpha_i^k (\mathbf{v} - \mathbf{x}_i) = 0 \Leftrightarrow \mathbf{v} = (\sum_{i=1}^{n} \alpha_i^k)^{-1} \sum_{i=1}^{n} \alpha_i^k \mathbf{x}_i \tag{10}$$

The obtained solution is then accelerated using the SOR type of stepsize factor as follows

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \omega(\mathbf{v} - \mathbf{u}^k), \quad \omega \in [0, 2], \tag{11}$$

where ω is the stepsize factor, $(\mathbf{v} - \mathbf{u}^k)$ is the search direction, and $\mathbf{v}$ is an approximate solution to (1) as defined in (10).

The overall algorithm is given as:

Algorithm 3.1 SOR

Step 1. Initialize $\mathbf{u}$ and set ω.

Step 2. Solve α_i^k:s for $i \in \{1, .., n\}$ using (9).

Step 3. Solve the basic iterate $\mathbf{v}$ from (10).

Step 4. Accelerate $\mathbf{u}$ using (11).

Step 5. If the stopping criterion is satisfied then stop, else return to step 2.

3.1.1 Convergence analysis

The properties of Step 3. of Algorithm 3.1 are addressed in the following theorem by adapting the ideas from [13]. Notice that for $\omega = 1$ Algorithm 3.1 coincides with the perturbed Weiszfeld algorithm [19] and [23].

Theorem 3.1 *If* $\mathbf{v} \neq \mathbf{u}^k$, *then* $\mathcal{J}_\varepsilon(\mathbf{v}) < \mathcal{J}_\varepsilon(\mathbf{u}^k)$.

Proof 3.1 *As* $\mathbf{v}$ *satisfies* (10), *it is the unique minimizer of the strictly convex cost function*

$$\mathcal{J}(\tilde{\mathbf{v}}) = \sum_{i=1}^{n} \frac{\|\tilde{\mathbf{v}} - \mathbf{x}_i\|^2 + \varepsilon}{\sqrt{\|\mathbf{u}^k - \mathbf{x}_i\|^2 + \varepsilon}}. \tag{12}$$

Let us denote $e_i(\mathbf{u}) = \sqrt{\|\mathbf{u} - \mathbf{x}_i\|^2 + \varepsilon}$. *Since* $\mathbf{u}^k \neq \mathbf{v}$, *it follows that*

$$\tilde{\mathcal{J}}(\mathbf{v}) = \sum_{i=1}^{n} \frac{e_i^2(\mathbf{v})}{e_i(\mathbf{u}^k)} < \tilde{\mathcal{J}}(\mathbf{u}^k) = \sum_{i=1}^{n} \frac{e_i^2(\mathbf{u}^k)}{e_i(\mathbf{u}^k)} = \sum_{i=1}^{n} e_i(\mathbf{u}^k) = \mathcal{J}(\mathbf{u}^k).$$

On the other hand,

$$\begin{aligned}
\tilde{\mathcal{J}}(\mathbf{v}) &= \sum_{i}^{n} \frac{\{e_i(\mathbf{u}^k) + [e_i(\mathbf{v}) - e_i(\mathbf{u}^k)]\}^2}{e_i(\mathbf{u}^k)} \\
&= \mathcal{J}(\mathbf{u}^k) + 2\mathcal{J}(\mathbf{v}) - 2\mathcal{J}(\mathbf{u}^k) + \sum_{i=1}^{n} \frac{[e_i(\mathbf{v}) - e_i(\mathbf{u}^k)]^2}{e_i(\mathbf{u}^k)} \\
\Leftrightarrow \tilde{\mathcal{J}}(\mathbf{v}) = \mathcal{J}(\mathbf{u}^k) + 2\mathcal{J}(\mathbf{v}) - 2\mathcal{J}(\mathbf{u}^k) + \sum_{i=1}^{n} \frac{[e_i(\mathbf{v}) - e_i(\mathbf{u}^k)]^2}{e_i(\mathbf{u}^k)} &< \mathcal{J}(\mathbf{u}^k) \\
\Leftrightarrow 2\mathcal{J}(\mathbf{v}) + \sum_{i=1}^{n} \frac{[e_i(\mathbf{v}) - e_i(\mathbf{u}^k)]^2}{e_i(\mathbf{u}^k)} &< 2\mathcal{J}(\mathbf{u}^k)
\end{aligned}$$

Since $\sum_{i=1}^{n} \frac{[e_i(\mathbf{v}) - e_i(\mathbf{u}^k)]^2}{e_i(\mathbf{u}^k)} \geq 0$, *it follows that* $2\mathcal{J}(\mathbf{v}) < 2\mathcal{J}(\mathbf{u}^k) \Leftrightarrow \mathcal{J}(\mathbf{v}) < \mathcal{J}(\mathbf{u}^k)$.

3.2 SOR accelerated Weiszfeld algorithm with removing of nonsmooth points

The following algorithm, called ASSOR (Active Set SOR, cf. [11] and articles therein), corresponds with the one presented in Section 3.1, but is replaced by the original gradient of (1) by restricting the nonsmooth data points. In order to realize the idea, neighborhood ϕ is defined and all data points in $B(\mathbf{u}, \phi)$, which defines an open ball with center $\mathbf{u}$ and radius ϕ, are discarded from the computation. The modified problem corresponding to (8) is now given as

$$\sum_{i:\mathbf{x}_i \notin B(\mathbf{u},\phi)} \frac{\mathbf{u} - \mathbf{x}_i}{\sqrt{\|\mathbf{u} - \mathbf{x}_i\|^2}} = 0. \tag{13}$$

ASSOR algorithm is following:

Algorithm 3.2 ASSOR

Step 1. Initialize $\mathbf{u}$ and set ϕ and ω.

Step 2. Discard all $\mathbf{x}_i$s for which $\mathbf{x}_i \in B(\mathbf{u}, \phi)$ (The number of remaining data points is denoted by m).

Step 3. Solve α_i^ks for $i \in \{1, .., m\}$ using (9).

Step 4. Solve the basic iterate $\mathbf{v}$ using the remaining data and (10).

Step 5. Accelerate $\mathbf{u}$ using (11).

Step 6. If the stopping criterion is satisfied then stop, else return to step 2.

4 NUMERICAL EXPERIMENTS

Accuracy and computational requirements of different algorithms and formulations are compared through the numerical experiments. To ensure the extensiveness of the results, a large number of simulated data sets, containing different numbers of data points, dimensions and shapes, were used in the experiments. In order to evaluate the sensitivity of the different algorithms with respect to initial conditions, several different starting points for each algorithms were also used. The main goal of the experiments is to find out which of the proposed methods and formulations solve the problem of the spatial median (1) accurately and efficiently. Scalability of the SOR-based algorithms to high dimensional problems is also investigated.

Due to absence of a closed-form solution and consequent lack of an exact analytical solution for the spatial median, reference values for the experiments were computed using Nelder-Mead (NM) algorithm (see, e.g., [20] and [14]) and extremely strict stopping criterion

$$\max_{i \in \{2,\ldots,n+1\}} \|\mathbf{x}_1 - \mathbf{x}_i\|_\infty < 10^{-12}.$$

Because the global convergence for NM can not be guaranteed, NM was initialized by conjugate gradient (CG) method (see, e.g., [21]). The combination is later referred to as CGNM.

4.1 Implementation of the algorithms and test settings

All numerical experiments were realized on MATLAB environment. All algorithms, excluding Nelder-Mead where MATLAB Toolbox implementation was utilized, were self-implemented. Polak-Ribiere approach [21] was applied in the update of CG search direction. Golden section (GS) method was used in the line search.

The following settings were used during the computation:

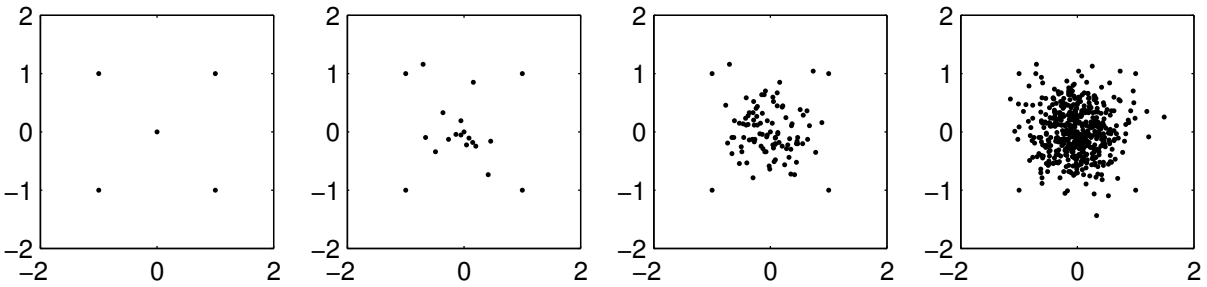

Figure 1: 2D plots of the test data sets 1-4.

CGNM Length of the GS search interval: $s_{int} = 1$. The stopping criteria of GS/CG: $\|\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}\|_\infty < 10^{-3}$/ $\|\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}\|_\infty < 10^{-2}$. The maximum number of iterations for GS/CG: 50000/2000. The smoothing parameter of (3): $\varepsilon = 1.49 \times 10^{-8}$. The stopping criterion of NM: $\max_{i \in \{2,\ldots,n+1\}} \|\mathbf{x}_1 - \mathbf{x}_i\|_\infty < 10^{-6}$.

NM The same settings as for NM in CGNM combination.

CG Length of the search intervals as in CGNM. The stopping criteria of GS/CG: $\|\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}\|_\infty < 10^{-8}$/ $|\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}\|_\infty < 10^{-6}$. The maximum number of iterations for GS/CG: 50000/1000. The smoothing parameter of (3): $\varepsilon = 1.49 \times 10^{-8}$.

SOR/ASSOR The value of the overrelaxation parameter ω is based on preliminary results. A compromising value 1.5 was chosen. As a stopping criteria we used $\|\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}\|_\infty < 10^{-6}$. The maximum number of iterations was 500.

MW The stopping criteria: $\|\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}\|_\infty < 10^{-6}$. The maximum number of iterations: 500.

The obtained results are compared to the reference values. Error of the cost function at the minimizing solution is computed by $e(\mathcal{J}(\mathbf{u}^*)) = \mathcal{J}(\mathbf{u}^*) - \mathcal{J}(\mathbf{u}^{ref})$, where $\mathbf{u}^*$ is a solution of a candicate method and $\mathbf{u}^{ref}$ is the reference solution. Correspondingly, misplacement error of a minimizing solution is computed by $e(\mathbf{u}^*) = \|\mathbf{u}^* - \mathbf{u}^{ref}\|_\infty$.

4.2 Simulated datasets

Computations were accomplished on eight two-dimensional data sets (see Figures 1 and 2). The data sets were generated manually or by random sampling from the normal or Laplace distribution. In data set 1, location of the spatial median is trivial, which enables accurate comparison of the results.

4.2.1 Selection of starting points

Let $\mathbf{X}$ be a given experiment data set. For thorough evaluation of the algorithms, four different kind of starting points were used: 1. The sample mean of the data set, 2. Any

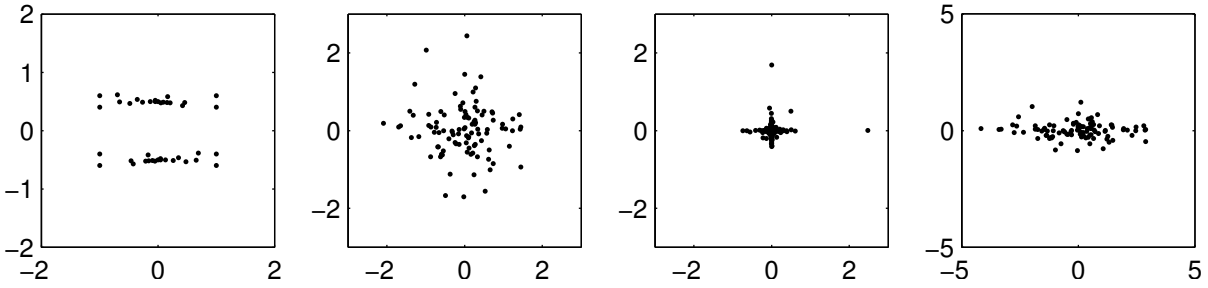

Figure 2: 2D plots of the test data sets 5-8.

data point $\mathbf{x}_i$ such that $\mathbf{x}_i \in \mathbf{X}$, 3. An arbitrary point $\mathbf{x}$ inside of the convex hull of $\mathbf{X}$ and 4. An arbitrary point $\mathbf{x}$ outside of the convex hull of $\mathbf{X}$.

	NM			CGNM					CG		
	$e(\mathcal{J}(\mathbf{u}^*))$	total	$e(\mathbf{u}^*)$	$e(\mathcal{J}(\mathbf{u}^*))$	#CG	#NM	total	$e(\mathbf{u}^*)$	$e(\mathcal{J}(\mathbf{u}^*))$	total	$e(\mathbf{u}^*)$
min	0.00e+00	37	0.00e+00	0.00e+00	4	37	41	0.00e+00	0.00e+00	4	0.00e+00
max	4.35e-04	112	4.34e-03	5.37e-07	158	76	211	1.29e-06	4.23e-07	1583	1.63e-06
mean	1.36e-05	104	1.36e-04	7.45e-08	95	52	147	3.78e-07	3.31e-08	583	1.77e-07
median	1.48e-11	111	2.82e-07	1.34e-11	96	51	147	2.98e-07	2.35e-13	477	3.42e-08

Table 1: Summary results from the experiments on NM, CGNM and CG. "#CG", "#NM" and "total" are the numbers of function evaluations taken by CG, NM and the complete algorithms, respectively.

	MW			SOR			ASSOR		
data	$e(\mathcal{J}(\mathbf{u}^*))$	it	$e(\mathbf{u}^*)$	$e(\mathcal{J}(\mathbf{u}^*))$	it	$e(\mathbf{u}^*)$	$e(\mathcal{J}(\mathbf{u}^*))$	it	$e(\mathbf{u}^*)$
min	0.00e+00	1	0.00e+00	0.00e+00	1	0.00e+00	0.00e+00	1	0.00e+00
max	7.36e-07	41	3.08e-06	3.42e-05	26	1.42e-04	4.57e-07	26	1.49e-06
mean	8.93e-08	21	1.31e-06	4.31e-06	14	1.84e-05	6.61e-08	14	5.76e-07
median	9.93e-11	19	1.16e-06	1.51e-11	13	3.88e-07	1.81e-11	12	3.99e-07

Table 2: Results from the experiments on the modified Weiszfeld, SOR and ASSOR. "it" is the number of iterations taken by an algorithm.

4.3 Accuracy of algorithms on bivariate data sets

The numerical experiments show that accuracy of the results by CGNM, CG, MW, SOR and ASSOR is well-comparable to the reference results (see Tables 1 and 2). Error to the location of a minimum solution was, at worst, approximately 10^{-5}, which is clearly a sufficient level in practice. Data set 2 is the most difficult case for each algorithm. We point out that also a combination of the perturbed problem reformulation (4) and conjugate gradient method was tested, but the results were poor.

NM managed also well except on a single run on "splitted" data set 5 (Figure 2). The weak result seems to result from initialization, since CGNM solves the same problem

better than NM alone. This strengthens the prior assumptions that NM may have weak convergence when the starting point is not close enough to the optimal point and, on the other hand, it may solve the same problem accurately and relative fast when initialized well. This case provides a practical example about the high sensitivity of the simplex methods to the initial conditions.

Notice that data set 5 contains two quite clearly separated batches of data. Providing the rest of the data lies far away from this splitted subset, it may also be considered as a single coherent cluster.

Alltogether, the results show that MW, SOR, ASSOR, CGNM, CG and also NM, with a proper initialization, are able to solve the problem of the spatial median accurately.

4.4 Comparison of computational efforts

Due to the different nature of the algorithms, only approximate analysis of the computational effort can be done. Therefore, estimation of CPU time is based on the number of function evaluations (CGNM and CG) and iterations (iterative methods: SOR, ASSOR, Weiszfeld).

It is known that CG and NM need more vector operations $O(p)$ ($\mathbf{u} \in \mathbb{R}^p$) than the iterative methods. However, here we concentrate from the data mining point of view on a more practical case when $n > p$ and iterations over the data dominate the computational cost. Hence, it is sufficient to estimate CPU time using the number of iterations and function evaluations.

The function evaluation in (1) needs one pass through the data, which means $O(n)$ worst case time complexity. On the other hand, one SOR iteration examines the data set twice ((9) and (10)), which leads to $O(2n)$ time complexity. Time complexity of one ASSOR iteration is approximately $O(3n)$. In comparison to SOR one more iteration is needed for finding the nonsmooth data points. This corresponds to three cost function evaluations.

The experiments show that it takes an average of almost four times more function evaluations for CG than CGNM to solve the problem (cf. Table 1). This accelerating effect of NM underpins the usual conception about good convergence of NM in the neighborhood of an optimal solution. One may notice that there are very small differences with respect to accuracy of a solution. This may be due to different stopping criteria applied in the simplex and gradient based methods: size of the simplex and the maximum norm of change of the solution, respectively. Perhaps a slightly looser limit of stopping criterion could have been chosen for CG still having comparable solutions to CGNM and a reduced number of function evaluations. On the other hand, the results on CGNM show that the number of function evaluations is very high for CG even if very loose stopping criteria is used. Bearing in mind the inherent properties of NM for nonsmooth problems and reduced number of function evaluations, CGNM seems to be the best choice of the compared classic optimization methods.

When CGNM is compared to iterative methods, the average cost seems to be about six

times higher than for SOR and about four times higher than for ASSOR (after the aforementioned ratio factors were applied to the results). Hence, the iterative overrelaxation methods seem to be superior to classical methods in terms of computation cost taken by the problem of the spatial median.

SOR and ASSOR algorithms are also compared to the MW algorithm (see Table 2). MW takes approximately one and a half times more iterations than SOR and ASSOR. Notice that one MW iteration is more expensive than the very simple SOR iteration. On the other hand, it is possible to accelerate MW as well. See results of the accelerated Weiszfeld-like algorithm, e.g., in [25] and [5]. The papers do not provide results in high dimensions ($p > 2$), which is of interest by the data miners.

4.4.1 Discussion about SOR and ASSOR

It is obvious that the worst case cost of ASSOR is higher than of SOR due to the pruning of overlapping nonsmooth data points. The worst case for ASSOR iteration exist when there are no overlapping data points. But, when a large amount of data points can be pruned away, it follows that (9) and (10) are solved in shorter time. Thereby, it means that on certain kind of data sets ASSOR may need less CPU time than SOR.

As we know, data mining tasks usually concentrate on large data sets that do not fit into the fast main memory. Hence, handling of such large data sets requires frequent accessing to the hard-disk. The pruning of data points on each iteration may reduce the time required for computing (9) and (10). But, on the other hand, data mining considers usually also high-dimensional data sets that tend to be sparse, which makes the overlapping of the data points and a solution unlike and, thereby, diminish the advantage of the pruning. This issue depends also on the dimension that is used during the computation. For example, use of projection techniques brings the data points closer to each other. To summarize, it is difficult to make a general choice between SOR and ASSOR.

4.5 Scalability of algorithms to high dimensions

Scalability of the algorithms with respect to the number of dimensions was investigated on eight high dimensional data sets ($\mathbb{R}^8$,$\mathbb{R}^{16}$,$\mathbb{R}^{32}$ and $\mathbb{R}^{64}$), which were generated by copying data sets 5 and 6 (see Figure 2). The stopping criteria were the same as in $\mathbb{R}^2$ experiments. Sample results are given in Figure 3. We focused on comparing CGNM and SOR method, but as a footnote, ASSOR and MW gave similar results to SOR.

CGNM provides quite precise results, but the number of function evaluations increases significantly along the number of dimensions. The growth in the number of function evaluations was mainly due to the NM part of the algorithm. In overall, CG showed better scalability to high-dimensions than NM.

The experiments show that the iterative SOR-based algorithms are superior to the classical optimization approach. The precision of the results is comparable to and efficiency is superior to any of the classical optimization methods. The most salient thing from

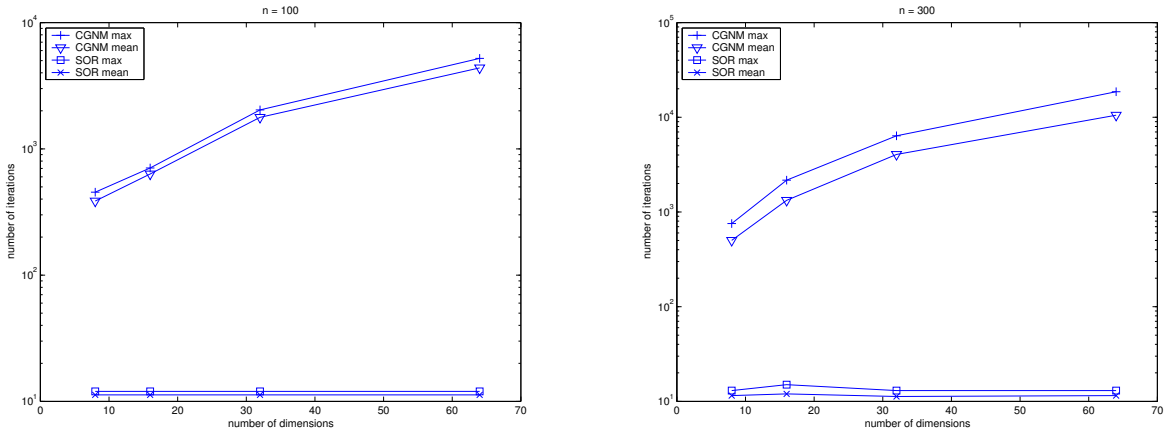

Figure 3: Scaling of the CGNM and SOR methods to high dimensional problems.

the data mining point of view is that the number of iterations on SOR-based methods is independent on the number of dimensions. As well the experiments showed that the over-relaxation parameter ω is independent on the number of dimensions. These results are very significant results for the development of reliable data mining tools.

Acknowledgement

This work was partially supported by National Technology Agency of Finland under the project no. 40280/05.

REFERENCES

[1] J. Brimberg, *The fermat-weber location problem revisited*, Mathematical Programming, 71 (1995), pp. 71–76.

[2] J. Brimberg, R. Chen, and D. Chen, *Accelerating convergence in the fermat-weber location problem*, Operations Research Letters, 22 (1998), pp. 151–157.

[3] L. Cánovas, R. Cañavate, and A. Marín, *On the convergence of the Weiszfeld algorithm*, Math. Program., 93 (2002), pp. 327–330.

[4] R. Chandrasekaran and A. Tamir, *Open questions concerning Weiszfeld's algorithm for the Fermat-Weber location problem*, Math. Programming, 44 (1989), pp. 293–295.

[5] Z. Drezner, *A note on accelerating the weiszfeld procedure*, Location Science, 3 (1995), pp. 275–279.

[6] B. S. Everitt, S. Landau, and M. Leese, *Cluster analysis*, Arnolds, a member of the Hodder Headline Group, 2001.

[7] J. Han and M. Kamber, *Data mining: concepts and techniques*, Morgan Kaufmann Publishers, Inc., 2001.

[8] P. Huber, *Robust statistics*, John Wiley & Sons, 1981.

[9] T. Kärkkäinen and S. Äyrämö, *Robust clustering methods for incomplete and erroneous data*, in Proceedings of the Fifth Conference on Data Mining, Wessex Institute of Technology, WIT Press, 2004, pp. 101–112.

[10] T. Kärkkäinen and E. Heikkola, *Robust formulations for training multilayer perceptrons*, Neural Computation, 16 (2004), pp. 837–862.

[11] T. Kärkkäinen, K. Majava, and M. M. Mäkelä, *Comparison of formulations and solution methods for image restoration problems*, Inverse Problems, 17 (2001), pp. 1977–1995.

[12] I. N. Katz, *Local convergence in fermat's problem*, Mathematical Programming, 6 (1974), pp. 89–104.

[13] H. W. Kuhn, *A note on fermat's problem*, Mathematical programming, 4 (1973), pp. 98–107.

[14] J. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, *Convergence properties of the nelder-mead simplex method in low dimensions*, SIAM Journal of Optimization, 9 (1998), pp. 112–147.

[15] Y. Li, *A newton acceleration of the weiszfeld algorithm for minimizing the sum of euclidean distances*, Computational Optimization and Applications, 10 (1998), pp. 219–242.

[16] R. Love, J. Morris, and G. Wesolowsky, *Facilities Location. Models and Methods*, North Holland Publishing Company, 1988.

[17] M. Mäkelä and P. Neittaanmäki, *Nonsmooth Optimization; Analysis and Algorithms with Applications to Optimal Control*, World Scientific, Singapore, 1992.

[18] J. G. Morris, *Convergence of the weiszfeld algorithm for weber problems using a generalized "distance" function*, Operations Research, 29 (1981), pp. 37–48.

[19] J. G. Morris and W. A. Verdini, *Minisum l_p distance location problems solved via a perturbed problem and Weiszfeld's algorithm*, Operations Research, 27 (1979), pp. 1180–1188.

[20] J. Nelder and R. Mead, *A simplex method for function minimization*, Computer Journal, 7 (1965), pp. 308–313.

[21] J. Nocedal and S. Wright, *Numerical Optimization*, Springer-Verlag, 1999.

[22] D. Rautenbach, M. Struzyna, C. Szegedy, and J. Vygen, *Weiszfeld"s algorithm revisited once again*, tech. rep., Research Institute for Discrete Mathematics, University of Bonn, 2004.

[23] J. B. Rosen and G. L. Xue, *On the convergence of a hyperboloid approximation procedure for the perturbed Euclidean multifacility location problem*, Operations Research, 41 (1993), pp. 1164–1171.

[24] Y. Vardi and C.-H. Zhang, *A modified Weiszfeld algorithm for the Fermat-Weber location problem*, Mathematical Programming, 90 (2001), pp. 559–566.

[25] B. S. Verkhovsky and Y. S. Polyakov, *Feedback algorithm for the single-facility minisum problem*, Annals of the European Academy of Sciences, 1 (2003), pp. 127–136.

[26] E. Weiszfeld, *Sur le point pour lequel les sommes des distances de n points donnés et minimum*, Tôhoku Mathematical Journal, 43 (1937), pp. 355–386.