Stanford Exploration Project, Report 103, April 27, 2000, pages 205–285

Short Note

Implementation of a nonlinear solver for minimizing the Huber norm

Antoine Guitton[1]

INTRODUCTION

The Huber norm (Huber, 1973) is an alternative to Iteratively Reweighted Least Square programs for solving the hybrid l^2-l^1 problem. In this note, I detail a method for minimizing the Huber norm. Because the Huber norm gives rise to a non-linear problem with non-twice continuously differentiable objective functions, its use is quite challenging. Claerbout (1996) implemented a Huber regression based on conjugate-gradient descents. However, the final results were not satisfying. Here I propose to solve the Huber problem using a quasi-Newton update of the solution with the computation of an approximated Hessian (second derivative of the objective function). This strategy is innovative in seismic processing and merits some explanation.

In this paper I first provide general definitions plus sufficient conditions to solve the optimization problem. Then, I present the quasi-Newton method and the complete algorithm used to solve the Huber problem.

DEFINITIONS AND CONDITIONS FOR OPTIMALITY

This part follows closely Kelley 's *Iterative Method for Optimization* (Kelley, 1999). We start here with a series of definitions:

1. $\mathbf{A}$ is positive definite if $\mathbf{x^T A x} > 0$ for all $\mathbf{x} \in \Re^N$
2. $\mathbf{A}$ is *spd* if $\mathbf{A}$ is positive definite and symmetric
3. $\mathbf{x}^* \in U$ $(U \subset \Re^N)$ is a global minimizer if $f(\mathbf{x}^*) \leq f(\mathbf{x})$ for all $\mathbf{x} \in U$

The Euclidian norm is also defined as

$$\| \mathbf{x} \| = \sqrt{\sum_{i=1}^{N} (x_i)^2}.$$

[1] **email:** antoine@sep.stanford.edu

Now, I give sufficient conditions that a minimizer $\mathbf{x}^*$ exists for a function *f*.

Theorem

Let f be twice continuously differentiable in a neighborhood of $\mathbf{x}^*$. Assume that $\nabla f(\mathbf{x}^*) = 0$ and that $\nabla^2 f(\mathbf{x}^*)$ is positive definite, then $\mathbf{x}^*$ is a local minimizer of f.

Proof

Let $\mathbf{u} \in \Re^N$ with $\mathbf{u} \neq 0$. For sufficiently small t we have

$$f(\mathbf{x}^* + t\mathbf{u}) = f(\mathbf{x}^*) + t\nabla f(\mathbf{x}^*)^T\mathbf{u} + \frac{t^2}{2}\mathbf{u}^T\nabla^2 f(\mathbf{x}^*)\mathbf{u} + o(t^2).$$

But $\nabla f(\mathbf{x}^*) = 0$ giving

$$f(\mathbf{x}^* + t\mathbf{u}) = f(\mathbf{x}^*) + \frac{t^2}{2}\mathbf{u}^T\nabla^2 f(\mathbf{x}^*)\mathbf{u} + o(t^2).$$

If $\nabla^2 f(\mathbf{x}^*)$ is positive definite, its smallest eigenvalue λ obeys $\lambda > 0$. So we have

$$f(\mathbf{x}^* + t\mathbf{u}) - f(\mathbf{x}^*) \geq \frac{\lambda}{2} \parallel t\mathbf{u} \parallel^2 + o(t^2) > 0.$$

Then, $\mathbf{x}^*$ is a local minimizer for f.

We see that a sufficient condition for a local minimizer is $\nabla f(\mathbf{x}^*) = 0$ and $\nabla^2 f(\mathbf{x}^*)$ (Hessian) is positive definite. These conditions are very important and should guide us in the choice of an optimization strategy.

Quadratic functions form the basis for most of the algorithms in optimization, in particular for the quasi-Newton method detailed in this paper. It is then important to discuss some issues involved with these functions. Now, if we pose a quadratic objective function

$$f(\mathbf{x}) = -\mathbf{x}^T\mathbf{b} + \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x},$$

we see that we want to solve

$$\nabla f(\mathbf{x}) = -\mathbf{b} + \mathbf{H}\mathbf{x} = 0.$$

We may assume that the Hessian $\mathbf{H}$ is symmetric because

$$\mathbf{x^T H x} = \mathbf{x^T}\frac{\mathbf{H^T} + \mathbf{H}}{2}\mathbf{x}.$$

So, the unique global minimizer is the solution of the system above if $\mathbf{H}$ (the Hessian) is *spd*.

A QUASI-NEWTON METHOD FOR UNCONSTRAINED OPTIMIZATION

We will assume that f and $\mathbf{x}^*$ satisfy the following assumptions:

1. f is twice continuously differentiable
2. $\nabla f(\mathbf{x}^*) = 0$
3. $\nabla^2 f(\mathbf{x}^*)$ is symmetric positive definite

The Newton methods update the current iteration $\mathbf{x}_n$ by the formula

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \lambda_n \mathbf{H}_n^{-1} \nabla f(\mathbf{x}_n), \tag{1}$$

where λ_n is given by a line search that ensures sufficient decrease. Quasi-Newton methods update an approximation of the Hessian $\mathbf{H}_n^{-1}$ as the iterations progress. A possible update is the BFGS method (Broyden, 1969; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970), which overcomes some limitations of the earlier Broyden's method (Broyden, 1965). In particular, the Broyden's update does not keep the *spd* structure of the Hessian. This structure not only ensures the existence of a local minimizer but also allows the convergence of the updated solution $\mathbf{x}_{n+1}$ to the minimum (Kelley, 1999). The BFGS update is a rank-two update given by

$$\mathbf{H}_{n+1} = \mathbf{H}_n + \frac{\mathbf{y}\mathbf{y}^T}{\mathbf{y}^T\mathbf{s}} - \frac{(\mathbf{H}_n\mathbf{s})(\mathbf{H}_n\mathbf{s})^T}{\mathbf{s}^T\mathbf{H}_n\mathbf{s}}, \tag{2}$$

with $\mathbf{s} = \mathbf{x}_{n+1} - \mathbf{x}_n$ and $\mathbf{y} = \nabla f(\mathbf{x}_{n+1}) - \nabla f(\mathbf{x}_n)$. In practice, it is very useful to express the previous equation in terms of the inverse matrices. We then have

$$\mathbf{H}_{n+1}^{-1} = \left(\mathbf{I} - \frac{\mathbf{s}\mathbf{y}^T}{\mathbf{y}^T\mathbf{s}}\right)\mathbf{H}_n^{-1}\left(\mathbf{I} - \frac{\mathbf{y}\mathbf{s}^T}{\mathbf{y}^T\mathbf{s}}\right) + \frac{\mathbf{s}\mathbf{s}^T}{\mathbf{y}^T\mathbf{s}}. \tag{3}$$

Lemma

Let $\mathbf{H}_n$ be spd, $\mathbf{y}^T\mathbf{s} > 0$, and $\mathbf{H}_{n+1}$ given in equation (2). Then $\mathbf{H}_{n+1}$ is spd.

Proof

Starting from equation (2), we can write for all $\mathbf{z} \neq 0$ and $\mathbf{y}^T\mathbf{s} > 0$,

$$\mathbf{z}^T\mathbf{H}_{n+1}\mathbf{z} = \mathbf{z}^T\mathbf{H}_n\mathbf{z} + \frac{(\mathbf{z}^T\mathbf{y})^2}{\mathbf{y}^T\mathbf{s}} - \frac{(\mathbf{z}^T\mathbf{H}_n\mathbf{s})^2}{(\mathbf{s}^T\mathbf{H}_n\mathbf{s})}.$$

Since $\mathbf{H}_n$ is *spd*, we have

$$(\mathbf{z}^T\mathbf{H}_n\mathbf{s})^2 \leq (\mathbf{s}^T\mathbf{H}_n\mathbf{s})(\mathbf{z}^T\mathbf{H}_n\mathbf{z})$$

with equality only if $\mathbf{z} = 0$ or $\mathbf{s} = 0$. But we have $\mathbf{z} \neq 0$ and $\mathbf{y}^T\mathbf{s} > 0$ so that

$$\mathbf{z}^T\mathbf{H}_{n+1}\mathbf{z} > \frac{(\mathbf{z}^T\mathbf{y})^2}{\mathbf{y}^T\mathbf{s}} \geq 0.$$

Then $\mathbf{H}_{n+1}$ is spd. If during the iterations we have $\mathbf{y}^T\mathbf{s} \leq 0$, then the update is a failure.

The previous lemma is very important since it shows that starting from an initial *spd* Hessian $\mathbf{H}_0$, the next approximation of the Hessian is *spd* (given that $\mathbf{y}^T\mathbf{s} > 0$). This guarantees the existence of a minimizer for the function f (the inverse $\mathbf{H}^{-1}$ is also *spd*). It can be shown (Kelley, 1999) that given some assumptions, the BFGS iterates are defined and converge *q-superlinearly* [2] to the local minimizer $\mathbf{x}^*$. In practice, the storage needed to compute the update and the possibility that $\mathbf{y}^T\mathbf{s} \leq 0$ are important issues. The updated Hessian is computed at each iteration recursively. For this, we need to store a solution step vector $\mathbf{s}$ and a gradient step vector $\mathbf{y}$ after each iteration. If for small problems this storage is not an issue, it may become critical for large-scale problems. Unfortunately, these large-scale problems occur in geophysics, and we need to find a better storage solution. Nocedal (1980) gives an interesting answer to this problem. Instead of keeping all the $\mathbf{s}$ and $\mathbf{y}$ from the past iterations, we update the Hessian using the information from the m previous iterations, where m is given by the end user. This means that when the number of iterations is smaller than m, we have a "real" BFGS update, and when it is larger than m, we have a Limited-memory BFGS (L-BFGS) update.

L-BFGS update

For the sake of completeness, I give the updating formulas of the Hessian as presented by Nocedal (1980). We define first

$$\rho_i = 1/\mathbf{y}_i^T\mathbf{s}_i \text{ and } \mathbf{v}_i = (I - \rho_i\mathbf{y}_i\mathbf{s}_i^T).$$

In addition, we pose $\mathbf{H}^{-1} = \mathbf{B}$. As described above, when k, the number of iterations, obeys $k+1 \leq m$, where m is the storage limit, we have the usual BFGS update

$$\begin{aligned}
\mathbf{B}_{k+1} \quad = \quad & \mathbf{v}_k^T\mathbf{v}_{k-1}^T\cdots\mathbf{v}_0^T\mathbf{B}_0\mathbf{v}_0\cdots\mathbf{v}_{k-1}\mathbf{v}_k \\
& +\mathbf{v}_k^T\cdots\mathbf{v}_1^T\rho_0\mathbf{s}_0\mathbf{s}_0^T\mathbf{v}_1\cdots\mathbf{v}_k \\
& . \\
& . \\
& . \\
& +\mathbf{v}_k^T\rho_{k-1}\mathbf{s}_{k-1}\mathbf{s}_{k-1}^T\mathbf{v}_k \\
& +\rho_k\mathbf{s}_k\mathbf{s}_k^T.
\end{aligned} \tag{4}$$

[2] $\mathbf{x}_n \rightarrow \mathbf{x}^*$ q-superlinearly if

$$lim_{n\to\infty}\frac{\parallel \mathbf{x}_{n+1} - \mathbf{x}^* \parallel}{\parallel \mathbf{x}_n - \mathbf{x}^* \parallel} = 0.$$

For $k+1>m$ we have the special limited-memory update

$$\begin{aligned}
\mathbf{B}_{k+1} &= \mathbf{v}_k^T\mathbf{v}_{k-1}^T\cdots\mathbf{v}_{k-m+1}^T\mathbf{B}_0\mathbf{v}_{k-m+1}\cdots\mathbf{v}_{k-1}\mathbf{v}_k \\
&\quad+\mathbf{v}_k^T\cdots\mathbf{v}_{k-m+2}^T\rho_{k-m+1}\mathbf{s}_{k-m+1}\mathbf{s}_{k-m+1}^T\mathbf{v}_{k-m+2}\cdots\mathbf{v}_k \\
&\quad. \\
&\quad. \\
&\quad. \\
&\quad+\mathbf{v}_k^T\rho_{k-1}\mathbf{s}_{k-1}\mathbf{s}_{k-1}^T\mathbf{v}_k \\
&\quad+\rho_k\mathbf{s}_k\mathbf{s}_k^T.
\end{aligned} \tag{5}$$

It is easy to show that the special updated Hessian is also spd. The L-BFGS algorithm is then

Algorithm 1

1. Choose $\mathbf{x}_0$, m, $0<\mu<1$, $\mu<\nu<1$ and a symmetric positive definite $\mathbf{B}_0$. Set $k=0$

2. Compute

$$\begin{aligned}
\mathbf{d}_k &= -\mathbf{B}_k\nabla f(\mathbf{x}_k) \qquad (6)\\
\mathbf{x}_{k+1} &= \mathbf{x}_k+\lambda_k\mathbf{d}_k, \qquad (7)
\end{aligned}$$

   where λ_k verifies the Wolfe conditions (More and Thuente, 1994):

$$\begin{aligned}
f(\mathbf{x}_k+\lambda_k\mathbf{d}_k) &\leq f(\mathbf{x}_k)+\mu\lambda_k\nabla f(\mathbf{x}_k)^T\mathbf{d}_k, \qquad (8)\\
|\nabla f(\mathbf{x}_k+\lambda_k\mathbf{d}_k)^T\mathbf{d}_k| &\geq \nu|\nabla f(\mathbf{x}_k)^T\mathbf{d}_k|. \qquad (9)
\end{aligned}$$

   We always try steplength $\lambda_k=1$ first.

3. Let $\hat{m}$=min$\{k$,$m-1\}$. Check if $\mathbf{y}_k^T\mathbf{s}_k>0$.

   - If no: $\mathbf{B}_{k+1}=\mathbf{I}$ (steepest descent step) and delete the pairs $\{\mathbf{y}_i,\mathbf{s}_i\}_{j=k-\hat{m}}^k$.
   - If yes: Update $\mathbf{B}_0$ $\hat{m}+1$ times using the pairs $\{\mathbf{y}_i,\mathbf{s}_i\}_{j=k-\hat{m}}^k$, i.e., let

$$\begin{aligned}
\mathbf{B}_{k+1} &= \mathbf{v}_k^T\mathbf{v}_{k-1}^T\cdots\mathbf{v}_{k-\hat{m}}^T\mathbf{B}_0\mathbf{v}_{k-\hat{m}}\cdots\mathbf{v}_{k-1}\mathbf{v}_k \\
&\quad+\mathbf{v}_k^T\cdots\mathbf{v}_{k-\hat{m}+1}^T\rho_{k-\hat{m}}\mathbf{s}_{k-\hat{m}}\mathbf{s}_{k-\hat{m}}^T\mathbf{v}_{k-\hat{m}+1}\cdots\mathbf{v}_k \\
&\quad. \\
&\quad. \\
&\quad. \\
&\quad+\mathbf{v}_k^T\rho_{k-1}\mathbf{s}_{k-1}\mathbf{s}_{k-1}^T\mathbf{v}_k \\
&\quad+\rho_k\mathbf{s}_k\mathbf{s}_k^T.
\end{aligned} \tag{10}$$

4. Set $k:=k+1$ and go to 2.

The update $\mathbf{B}_{k+1}$ is not formed explicitly; instead, we compute $\mathbf{d}_k = -\mathbf{B}_k \nabla f(\mathbf{x}_k)$ with an iterative formula (Nocedal, 1980). Liu and Nocedal (1989) propose that we scale the initial symmetric positive definite $\mathbf{B}_0$ at each iteration:

$$\mathbf{B}_k^0 = \frac{\mathbf{y}_k^T \mathbf{s}_k}{\parallel \mathbf{y}_k \parallel^2} \mathbf{B}_0. \tag{11}$$

This scaling greatly improves the performances of the method. Liu and Nocedal (1989) show that the storage limit for large-scale problems has little effect on the method's performances. A common choice for m is $m = 5$ (this is the default in my implementation as well). Conditions (8) and (9) are satisfied if we use an appropriate line search algorithm. I programmed a MoreThuente line search algorithm (More and Thuente, 1994), which ensures sufficient decrease of the objective function (equation 8) and obeys the curvature condition given in equation (9). We do not describe this program here. In practice, the initial guess $\mathbf{B}_0$ for the Hessian can be the identity matrix $\mathbf{I}$; then it might be scaled as proposed above. Liu and Nocedal (1989) prove that the L-BFGS algorithm converges to the local minimizer $\mathbf{x}^*$ and that the family of solutions $\{\mathbf{x}_k\}$ converges *R-linearly* [3] (remember that the usual BFGS gives *q-superlinear* convergence, which is better).

SOLVING THE HUBER PROBLEM WITH A QUASI-NEWTON METHOD

The Huber norm (Huber, 1973, 1981) is a hybrid l^1-l^2 measure. We expect to find the minimum of the function using a quasi-Newton method with a L-BFGS update of the Hessian (Guitton and Symes, 1999). The Huber norm is

$$\begin{aligned} f(\mathbf{x}) &= |\mathbf{Ax}-\mathbf{m}|_{Huber}, \\ &= |\mathbf{r}|_{Huber}, \\ &= \sum_{i=1}^{N} M_\epsilon(r_i), \end{aligned} \tag{12}$$

where

$$M_\epsilon(r) = \begin{cases} \frac{r^2}{2\epsilon}, & 0 \le |r| \le \epsilon \\ |r| - \frac{\epsilon}{2}, & \epsilon < |r|. \end{cases} \tag{13}$$

ϵ commands the limit between an l^1 or l^2 treatment of the residual; we call it the Huber threshold and it must be given by the user. The gradient of the objective function is given by

$$\nabla f(\mathbf{x}) = \mathbf{A}^T (\mathbf{Ax}-\mathbf{m})_{-\epsilon}^{\epsilon}, \tag{14}$$

[3] $\mathbf{x}_n \to \mathbf{x}^*$ R-linearly if there is a constant $0 \le r < 1$ such that

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \le r^k [f(\mathbf{x}_0) - f(\mathbf{x}^*)].$$

where $\mathbf{z}^{\epsilon}_{-\epsilon}$ is the vector whose ith component is

$$z_i = max\{-\epsilon, min\{\epsilon, z_i\}\}.$$

Because the Huber function is not twice continuously differentiable, it does not satisfy the three necessary conditions that guarantee the convergence to a minimum. However, we only need to compute the gradient for the BFGS update of the Hessian. Furthermore, given that the approximated Hessian is certainly a vague approximation of the real one (Symes, 1999, Personal communication), the violation of the initial conditions is mild. In addition, results (Guitton, 2000) show that this method converges to a minimum. Li (1995) shows that the Huber function has a unique minimizer for any meaningful choice of ϵ. Indeed, if the l^1 problem $f(\mathbf{x}) = |\mathbf{Ax} - \mathbf{m}|_1$ has multiple solutions (Tarantola, 1987), then the Huber problem, provided that ϵ is small enough, also has multiple solutions. This is annoying since we want to find a global minimum for the problem using quasi-Newton updates. In practice, however, it seems that

$$\epsilon = \frac{max|\mathbf{d}|}{100}$$

is a good choice for the threshold (Darche, 1989). The threshold being set properly, the Huber function has mathematical properties that allow the use of quasi-Newton methods. We can now define an efficient algorithm in order to solve the Huber problem:

Algorithm 2

1. Choose $\mathbf{x}_0$ and the threshold ϵ. Set $k = 0$
2. Compute $\nabla f(\mathbf{x}_k)$ using equation 14
3. Compute $\mathbf{d}_k = -\mathbf{B}_k \nabla f(\mathbf{x}_k)$ using a L-BFGS update (Algorithm 1, step 3)
4. Compute the step λ_k using a MoreThuente line search ($\lambda_k = 1$ tried first)
5. Update the solution $\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k$
6. Go to step 2

This algorithm will converge to the minimizer $\mathbf{x}^*$, as proven by Liu and Nocedal (1989).

CONCLUSION

Given an adequate threshold ϵ, the Huber problem may be solved using a quasi-Newton solver. The Limited memory BFGS method, a quasi-Newton update, has interesting storage properties that lead to efficient convergence to the local minimum of any convex function. In this paper, I proposed an algorithm to solve the Huber problem using the L-BFGS solver and a MoreThuente line search. This algorithm is then supposed to give a *R-linear* convergence to the desired solution.

REFERENCES

Broyden, C. G., 1965, A class of methods for solving nonlinear simultaneous equations: Math. Comp., **19**, 577–593.

Broyden, C. G., 1969, A new double-rank minimization algorithm: AMS Notices, **16**, 670.

Claerbout, J., 1996, Conjugate-direction Huber regression: SEP–**92**, 229–235.

Darche, G., 1989, Iterative l1 deconvolution: SEP–**61**, 281–301.

Fletcher, R., 1970, A new approach to variable metric methods: Comput. J., **13**, 317–322.

Goldfarb, D., 1970, A family of variable metric methods derived by variational means: Math. Comp., **24**, 23–26.

Guitton, A., and Symes, W. W., 1999, Robust and stable velocity analysis using the Huber function: 69th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1166–1169.

Guitton, A., 2000, Huber solver versus IRLS algorithm for quasi L1 inversion: SEP–**103**, 255–271.

Huber, P. J., 1973, Robust regression: Asymptotics, conjectures, and Monte Carlo: Ann. Statist., **1**, 799–821.

Huber, P. J., 1981, Robust statistics: Wiley series in Probability and Mathematical statistics.

Kelley, C. T., 1999, Iterative methods for optimization: SIAM in applied mathematics.

Li, W., 1995, Numerical algorithms for the Huber M-estimator problem: Approximation Theory, **8**, 1–10.

Liu, D. C., and Nocedal, J., 1989, On the limited memory BFGS method for large scale optimization: Mathematical Programming, **45**, 503–528.

More, J. J., and Thuente, J., 1994, Line search algorithms with guaranteed sufficient decrease: ACM Transactions on Mathematical Software, **20**, 286–307.

Nocedal, J., 1980, Updating quasi-Newton matrices with limited storage: Mathematics of Computation, **95**, 339–353.

Shanno, D. F., 1970, Conditioning of quasi-Newton methods for function minimization: Math. Comp., **24**, 647–657.

Tarantola, A., 1987, Inverse Problem Theory: methods for data fitting and model parameter estimation: Elsevier.