Robust and stable velocity analysis using the Huber function

Antoine Guitton, Stanford Exploration Project, William W. Symes, The Rice Inversion Project, Rice University*

SUMMARY

The Huber function is an hybrid l^1-l^2 misfit measure. We demonstrate, on velocity analysis examples, that the Huber function is more robust to outlier effects, and more stable with respect to the number of iterations than the l^2 norm. We also show that the Huber threshold ϵ which controls the transition between the l^1 and the l^2 norm may be chosen within a certain range of values without damaging the final result.
At this point, the l^2 norm with conjugate gradient is twice as fast as the Huber function with a nonlinear optimization method. The comparison may be altered in favor of the nonlinear scheme by algorithm tuning, however. A more natural data dependent expression for the threshold ϵ might be also useful. These results encourage use of the Huber function whenever the data are contaminated with noise and, as a robust and stable function, to replace the l^2 norm in other geophysical applications.

INTRODUCTION

Robust error measures such as the l^1 norm have found a number of uses in geophysics. As measures of data misfit, they show considerably less sensitivity to large measurement errors than does the mean square (l^2) measure. As almost all geophysical inverse problems are ill-posed, relatively noise insensitive misfit measures can yield far more stable estimates of Earth parameters (Claerbout and Muir, 1973; Taylor et al., 1979; Chapman and Barrodale, 1983; Scales and Gersztenkorn, 1988; Scales et al., 1988). This insensitivity to large noise has a statistical interpretation: robust measures are related to long-tailed density functions in the same way that the mean square is related to the (short-tailed) Gaussian (Tarantola, 1987).

The best known of the robust measures is the l^1 norm: denoting the residual (misfit) components by r_i, $i = 1, ..., N$, its l^1 norm is $\sum_{i=1}^{N} |r_i|$. This function is not smooth: it is singular at zero residual. As a result, numerical minimization is difficult. Various approaches based for example on a linear programming viewpoint (Barrodale and Roberts, 1980) or iterative smoothing (Scales et al., 1988), have been used with success but require considerable tuning. Moreover, the singularity implies that small residuals are "taken as seriously" as large residuals, which may not be appropriate in all circumstances.

These drawbacks of the l^1 norm have led to various proposals which combine robust treatment of large residuals with Gaussian treatment of small residuals. In the work reported here, we use a hybrid l^1-l^2 error measure proposed by Huber (Huber, 1973):

$$M_\epsilon(r) = \left\{ \begin{array}{ll} \frac{r^2}{2\epsilon}, & 0 \leq |r| \leq \epsilon \\ |r - \frac{\epsilon}{2}|, & \epsilon < |r| \end{array} \right.$$

We will call $\sum_{i=1}^{N} M_\epsilon(r_i)$ the *Huber misfit function*, or Huber function for short (Figure 1). Note that the Huber function is smooth near zero residual, and weights small residuals by mean square. It is reasonable to suppose that the Huber function is easier to minimize than l^1 while still robust against large residuals.

This paper describes the application of the Huber misfit function to velocity analysis. Estimation of RMS velocity (or slowness) can be posed as a linear inverse problem, through the *velocity transform*, described in the next section. Definition of the misfit *via* the Huber function (or any other robust error measure) results in a nonlinear optimization problem for the velocity model. This nonlinearity would seem to compare unfavorably with the least squares (l^2) treatment of the same problem, which leads to a linear system (the normal equation) and so can be solved by efficient iterative methods such as conjugate gradient. We show that use of an appropriate nonlinear optimization method gives a Huber-based solution with computation effort comparable to that of conjugate gradient least squares solution. Thus the excellent noise rejection properties of the Huber misfit function come at no appreciable premium in computational effort.

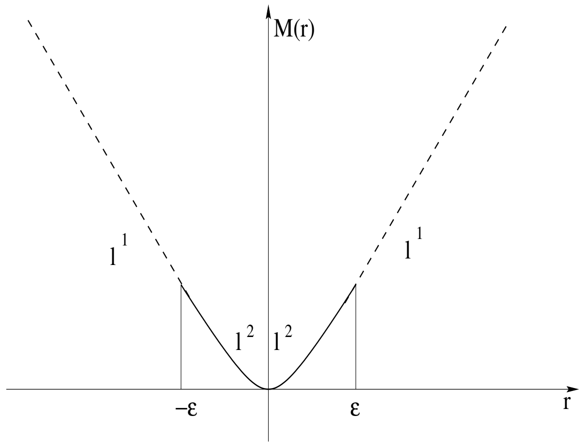

Figure 1: Error measure proposed by Huber (Huber, 1973). The upper part above ϵ is the l^1 norm while the lower part is the l^2 norm.

In the work reported here we have used a version of the Limited Memory BFGS algorithm (Nocedal, 1980) as implemented in the Hilbert Class Library (Gockenbach et al., 1999). Other nonlinear iterative optimizers could be used - we have treated the same examples with nonlinear conjugate gradients (Fletcher, 1980) with somewhat reduced efficiency but comparable results.

The second section of our paper explains the velocity transform and formulates a linear inverse problem for velocity analysis. The third and fourth sections present examples based on synthetic and field data respectively. In the final section we briefly summarize our conclusions.

THE GEOPHYSICAL PROBLEM: VELOCITY ANALYSIS

The *velocity domain* representation of seismic data is an alternative to the standard CMP presentation. Transformation of CMP data into the velocity domain (producing a velocity *model* or *panel* of the data) exhibits clearly the moveout inherent in the data and therefore, forms a convenient basis for velocity analysis as a linear inverse problem. The velocity transform $\mathbf{A}$ from the model space (velocity domain) into the data space (CMP gathers) stretches the velocities back in the offset plane (superposition of hyperbolas) whereas the adjoint operation ($\mathbf{A}'$) squeezes the data (summation over hyperbolas):

$$\mathbf{A}m(t,x) = \sum_s \frac{t}{\tau} w(s,x,\tau) m(\tau,s) \Big|_{\tau=\sqrt{t^2-s^2x^2}}$$

$$\mathbf{A}'d(\tau,s) = \sum_x w(s,x,\tau) d(t,x) \Big|_{t=\sqrt{\tau^2+s^2x^2}}$$

with $w(s,x,\tau)$ a weighting function that we define later. $\mathbf{A}$ is related to the velocity stack as defined by Taner and Koehler (1969). The problem is: given a CMP gather can we find a velocity panel which synthesizes it *via* $\mathbf{A}$? In equations, given data $\mathbf{d}$, we want to solve for model $\mathbf{m}$:

$$\mathbf{Am} \approx \mathbf{d}$$

A simple way to solve this problem, which makes sense whether you can solve this equation exactly or not is: find the model $\mathbf{m}$ to make the *mean square* misfit

$$(\mathbf{Am} - \mathbf{d})'(\mathbf{Am} - \mathbf{d})$$

as small as possible. This optimization problem is equivalent to the linear system ("normal equations")

$$\mathbf{A}'\mathbf{Am} = \mathbf{A}'\mathbf{d}$$

This system is easy to solve if $\mathbf{A}'\mathbf{A} \approx \mathbf{I}$, i.e if $\mathbf{A}$ is close to unitary: then $\mathbf{m} = \mathbf{A}'\mathbf{d}$. In fact, $\mathbf{A}$ is far from an unitary operator for many reasons. In particular, we have to define weightings to compensate for geometry spreading, angle-dependent effects. These corrections require the use of a weighting function (Claerbout and Black, 1997),

$$w(s, x, \tau) = \frac{1}{\sqrt{(\tau^2 + s^2x^2)^{1/2}}} \frac{\tau}{\sqrt{\tau^2 + s^2x^2}} \sqrt{xs}$$

Moreover, the summation in the velocity space boosts low frequencies (Claerbout and Black, 1997) so we need to apply a half derivative operator ($\sqrt{i\omega}$). Introducing $\sqrt{i\omega}$ and $w(s, x, \tau)$ brings $\mathbf{A}$ closer to being a unitary operator, but a lot of problems remain. First, when hyperbolas cross in a CMP gather, the relation between t and τ is no longer a one to one transformation, since one point in the data belongs to different locations in the model. Second, the theory does not predict non-hyperbolic events linked to dipping beds or diffractions, affecting the model space. The last problem is the presence of random noise that spreads out in the velocity panel.

Since the data is noisy, the modeling operator is not unitary and the numbers of equations and unknowns may be large, an iterative data-fitting approach seem reasonable:

$$\min_{\mathbf{m}}(E(\mathbf{Am} - \mathbf{d}))$$

where $\mathbf{m}$ is the model, $\mathbf{d}$, the data we want to fit, $\mathbf{A}$ the modeling operator, and E a misfit measurement function we have to choose. We have already presented one possibility, namely that E is the l^2 norm (least squares inversion). A convenient iterative method in this case is to solve the normal equation using conjugate gradient iteration. We refer to this approach as "CG" or "l^2". An alternative approach is E equals the Huber function introduced in the first section. With the Huber misfit measure, the velocity transform inverse problem is no longer equivalent to a linear system. We choose to solve it using a general-purpose nonlinear optimizer, as mentioned above, rather than one of several special-purpose methods invented for this type of problem. We refer to this approach as"Huber" or the "Huber solver".

The next two parts of this paper will show some velocity analysis using both CG and Huber solver for different gathers. We implemented the weight $w(s, x, \tau)$, and the half derivative function $\sqrt{i\omega}$ in our program for $\mathbf{A}$ to be a pseudo-unitary operator.

SYNTHETIC DATA TESTS

To compare the Huber function with the least square measure, we generated a synthetic CMP gather that we perturb by introducing a low velocity aliased plane wave. This data set constitutes the first input for the iterative schemes. Figure 2 displays the model space on the left, and the data space on the right. The high amplitude low velocity noise does not contaminate the Huber result. However, the CG result demonstrates the l^2 norm's inability to cope with this inconsistent data. The huber function demonstrates its robustness since it is not affected by the noisy data.

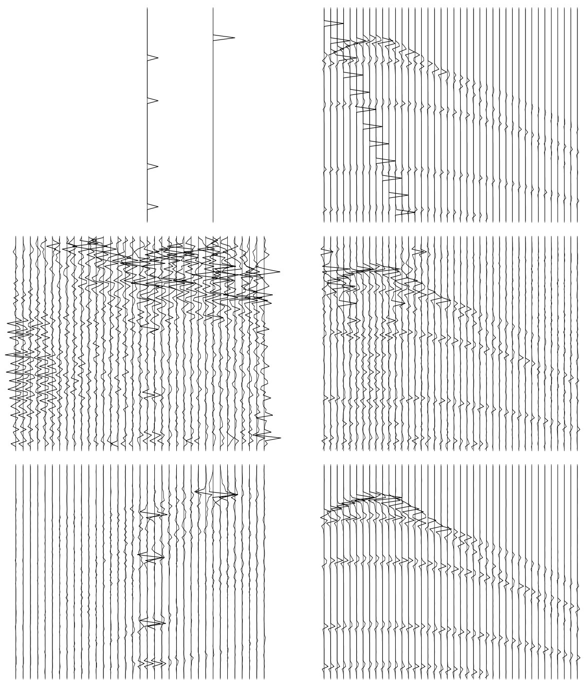

Figure 2: l^2 and Huber results with a slow plane wave: the top left panel shows the ideal velocity scan, the top right panel is the input data for the iterative scheme; the middle left displays the CG output, the middle right are the modeled data after the CG velocity transform; the bottom left displays Huber output, the bottom right are the modeled data after the velocity transform with Huber.

FIELD DATA EXAMPLES

In this section we compare CG and the Huber function for a gather acquired in a relatively flat environment. We divide this section in three: first, we show the inverted velocity panels and modeled data obtained for both CG and Huber solver. The second and third section will treat stability problems and computational efficiency.

This section shows that the Huber solver not only works well in our velocity analysis problem, but also is stable with respect to the number of iterations. In addition, the Huber function allows different threshold values without damaging the final model, and runs fast enough to make it a serious rival to CG.

Velocity analysis on field data

Figure 4 (the same clip has been applied within each figure) display different interesting features like amplitude anomalies, dead traces, ground roll etc... The velocities are computed after 5 iterations only. Figure 3 shows the velocity analysis using Huber and CG on a first data set. It appears that the CG result has artifacts at times above 1.5 sec. We see some horizontal stripes that make reliable interpretation difficult. In contrast, the Huber result displays a focused velocity corridor. Some low frequency events do appear in the upper right part of the panel however but they do not interfere with the main fairway. It interesting to notice that Huber separates the low frequency/low velocity noise from the signal whereas the l^2 measure spreads it out along the velocity axis. If we now model those results back into the data-space, we obtain Figure 4. We notice that CG does not do a good job in estimating the upper part. Furthermore, some high frequency noise appears. The Huber result however looks close to the original data. In particular, the upper part is well estimated and no harmful artifacts are visible. The abnormally high amplitude first trace does not affect the final results either. This result is quite encouraging and gives us a flavor of what could be accomplished with the Huber norm. The same conclusion applies on the field data than on the synthetics, the Huber function is robust.

Is Huber a stable solver?

For Huber to be a serious competitor to the l^2 norm and CG, it needs to be stable with respect to the number of iterations and to the Huber threshold, ϵ. This is what we test in this section. The first test deals with the number of iterations, the second with the threshold. Figure 5 displays two velocity panels using 70 iterations for both CG and Huber solvers. The left panel shows one of the well known characteristic of conjugate gradients: we basically start to invert noise in the data space, making the model space particularly messy and uninterpretable. In contrast, the Huber result is fairly stable since we can pick velocity functions without bothering with the noise after 70 iterations. It looks amazing that we get this velocity corridor when we see what happened with the conjugate gradient.

Testing the Huber response to different thresholds is another important issue. Remember that this threshold determines the border between the l^1 and l^2 norms. Ideally, we would like not to have to specify this parameter *a priori*, but rather have the algorithm adaptively estimates it from the data. It's interesting, however, to test the system's sensitivity to the threshold. Figure 6 shows a comparison for ϵ ranging from 0.001 to 1 for 20 iterations. The last panel is the CG result. As expected, Huber starts to behave like an l^2 norm for large ϵ. Nonetheless, for $\epsilon = 0.001$ to $\epsilon = 0.1$, the velocity panels are fairly comparable giving us a wide range of possibility. An automatic computation of this threshold would be preferred, however.

The Huber solver looks fairly stable with respect to two major choices: the number of iterations and ϵ. The best result arises when we increase the number of iterations: as the l^2 result explodes, the Huber result looks the same. In addition, we may choose a threshold without any problems since it does not damage the velocity panel (once ϵ is small enough, 0.1 in this case). The Huber misfit function seems to hold many interesting properties to solve geophysical problems. We did not apply any regularization on the least square method: it would make l^2 less noise sensitive but requires either a regularization weight or a noise level estimate and results are rather sensitive to these. The Huber function requires a parameter too (ϵ) but the results seem not to depend strongly on its choice.

Computational efficiency

The Limited Memory BFGS algorithm implementation gives fast convergence for the nonlinear optimization method. Nonetheless, each iteration is twice slower using the Huber function than the l^2 norm with conjugate gradient. The comparison might be altered in favor of the nonlinear method by algorithm tuning.

CONCLUSION

Since geophysical inverse problems are often ill-posed due to the presence of inconsistent data, high amplitude anomalies and outlier effects, relatively noise insensitive misfit measures can yield far more stable estimates of Earth parameters. The Huber function is an ideal compromise between l^1 and l^2 norms for misfit measure, boasting robustness in the presence of noise and outlier effects like an l^1 measure, but also the stability and rapid convergence characteristics of l^2 measures. The transition between the two norms is a governed by a free parameter, the Huber threshold ϵ.

A comparison of the Huber function against an l^2 norm shows that the Huber function is stable with regard to both the number of iterations and the threshold ϵ , provided the latter is small enough. In addition, the Huber function gives better results than the l^2 when applied to velocity analysis showing its robustness to outlier effects. In an initial performance comparison, we determined that conjugate gradient is twice as fast as the nonlinear optimization method using the Huber function. The comparison might be altered in favor of the nonlinear method by algorithm tuning. A data-dependent criterion for choosing the Huber threshold may prove fruitful, i.e., "treat $X\%$ of the data as Gaussian in the small residuals", where X is specified interactively by the end-user. These results encourage use of the Huber function whenever the data are contaminated with noise and, as a robust and stable measure, to replace the l^2 norm in many geophysical applications.

ACKNOWLEDGEMENTS

The authors would like to thank CGG, the sponsors of the Stanford Exploration Project and of The Rice Inversion Project. The authors are also grateful to Louis Vaillant and Jon Claerbout for their help and fruitful suggestions.

REFERENCES

Barrodale, I., and Roberts, F. D. K., 1980, Algorithm 552 : Solution of the constrained l1 linear approximation problem: ACM Transactions on Mathematical Software, **6**, 231–235.

Chapman, N. R., and Barrodale, I., 1983, Deconvolution of marine seismic data using the l1 norm: Geophys. J. Roy. Astr. Soc., **72**, 93–100.

Claerbout, J. F., and Black, J. L., 1997, Basic earth imaging: Class notes, http://sepwww.stanford.edu/sep/prof/index.html.

Claerbout, J. F., and Muir, F., 1973, Robust modeling with erratic data: Geophysics, **38**, 820–844.

Fletcher, R., 1980, Practical methods of optimization, I:Unconstrained Optimization: Wiley & Sons, New York.

Gockenbach, M. S., Petro, M., and Symes, W. W., 1999, C++ classes for linking optimization with complex simulations: ACM Transactions on Mathematical Software, in press.

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

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

Scales, J. A., and Gersztenkorn, A., 1988, Robust methods in inverse theory: Inverse Problems, **4**, 1071–1091.

Scales, J. A., Gersztenkorn, A., and Treitel, S., 1988, Fast lp solution of large, sparse, linear systems: application to seismic travel time tomography: J. Comp. Phys., **75**, 314–333.

Taner, M., and Koehler, F., 1969, Velocity spectra: digital computer derivation and application of velocity functions: Geophysics, **34**, 859–881.

Tarantola, A., 1987, Inverse problem theory: Elsevier.

Taylor, H. L., Banks, S. C., and McCoy, J. F., 1979, Deconvolution with the l1 norm: Geophysics, **44**, 39–52.

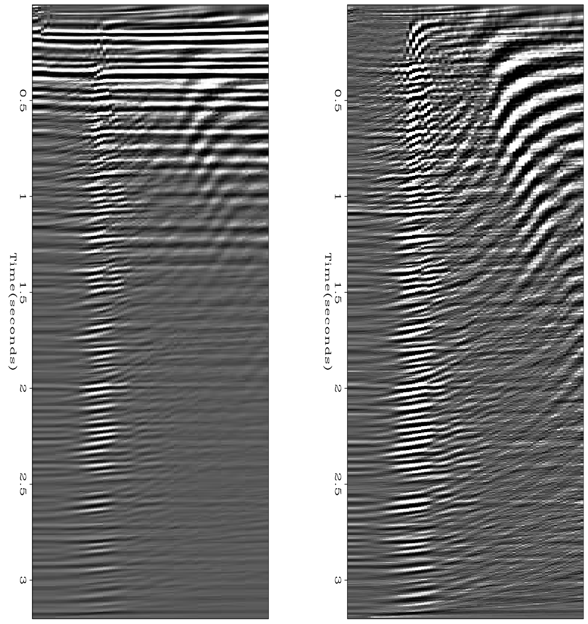

Figure 3: Velocity panel using 5 iterations for l^2 (left) and Huber (right).

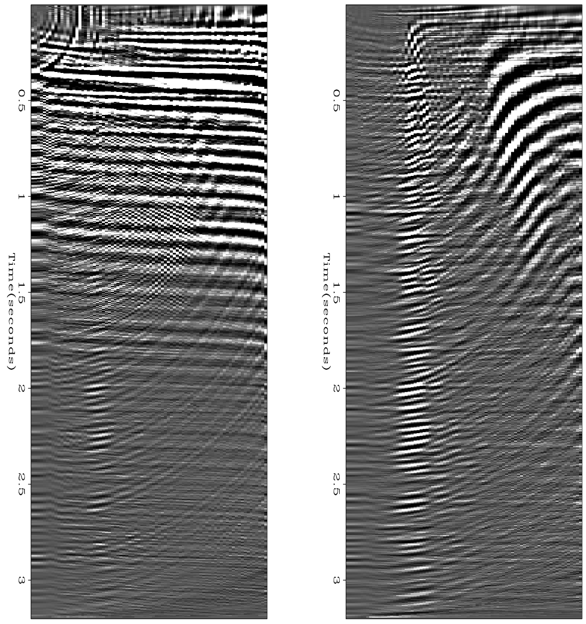

Figure 5: Velocity panel using 70 iterations for l^2 (left) and Huber (right).

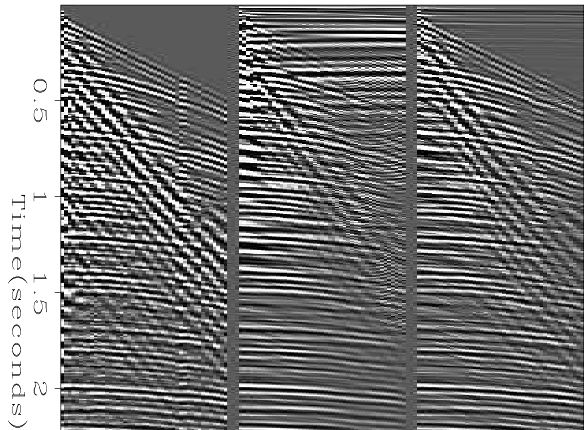

Figure 4: Data modeling after iteratively improved velocity model: input data on the left, CG output in the middle, Huber output on the right.

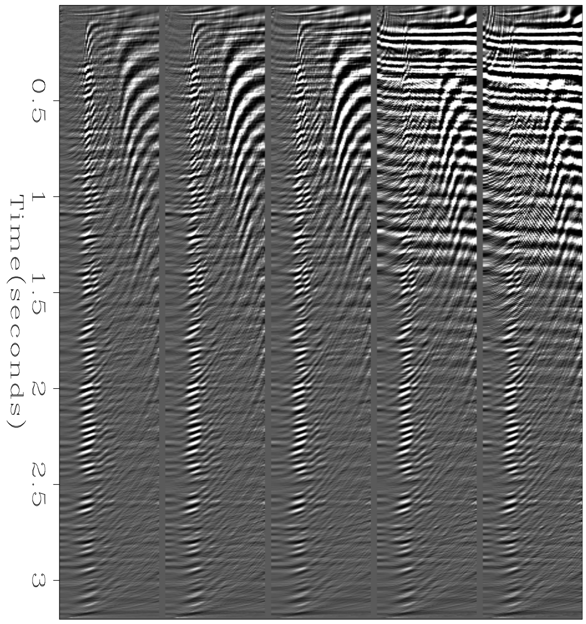

Figure 6: Velocity panel for different ϵ. From left to right $\epsilon = 0.001, 0.01, 0.1, 1$. The last panel shows the CG result.