Stanford Exploration Project, Report 61, January 13, 1998, pages 1–112

Stanford Exploration Project, Report 61, January 13, 1998, pages 1–112

Iterative L^1 deconvolution

Gilles Darche[1]

keywords: *deconvolution, algorithm, noise, wavelet*

ABSTRACT

An alternative to the classical Wiener deconvolution is to minimize the L^1 norm of the residuals rather than their L^2 norm. This minimization is achieved by using an iteratively reweighted least-squares algorithm, which finds L^1 minima, starting from L^2 minima. The solution found by this algorithm is still close to the result of the L^2 deconvolution, and does not resolve the indeterminacy of the phase of the seismic wavelet. However, this algorithm shows the efficiency of L^1 deconvolution in presence of noise bursts in the seismic data, and could be used for more general noise problems.

INTRODUCTION

For about ten years, iterative algorithms called IRLS (Iterative Reweighted Least-Squares) algorithms have been developed to solve L^p-norm minimization problems, for $1 \leq p \leq 2$, so lying between the least-absolute-values problem and the classical least-squares problem. Their main advantage is to provide an easy way to compute a L^1 solution; previously, L^1 solutions were found by linear programming techniques (Taylor et al., 1979), which need a large quantity of computer memory. L^1 solutions are known to be more "robust" than L^2 solutions, being less sensitive to spiky high-amplitude noise (Claerbout and Muir, 1973).

The applications of IRLS algorithms to geophysical problems have been limited up to now to travel-time tomography (Scales et al., 1988), 1-D acoustic inversion (Gersztenkorn et al.,1986). A simple example of deconvolution on a synthetic trace was given by Yarlagadda et al. (1985).

Each iteration of the IRLS algorithm consists of a weighted least-squares minimization, which is usually solved by a conjugate-gradient method. In many cases, such as tomography, this method is very efficient; if the linear systems involved are sparse, this method is also computationally fast.

However, deconvolution processes don't offer the same guarantees of computa-

[1]**email:** not available

tional efficiency, because they are *by essence* ill-conditioned and require many computations. Consequently, this algorithm demands some care before it can be implemented for deconvolution problems.

I will show the efficiency of the L^1-norm algorithm in presence of spiky noise using a simple synthetic example. This algorithm will in fact sharpen the L^2 residuals, as I will illustrate for predictive deconvolution; however, in that case, it does not improve the estimation of the seismic wavelet. This illustration will be completed by an example on a marine CMP gather.

Finally, I suggest the extension of this algorithm to use a mixed $L^1 - L^2$ norm, for more general noise problems.

L^P MINIMIZATION WITH THE IRLS ALGORITHM

Normal equations

The iterative reweighted least-squares (IRLS) algorithm provides a mean by which linear systems can be solved by minimizing the L^p norm of the residuals ($1 \leq p \leq 2$). Let us suppose that we try to solve the set of equations:

$$A.x = y \qquad\qquad x \in \mathcal{R}^m,\ y \in \mathcal{R}^n,\ A \in \mathcal{R}^{n\times m}\ .$$

To find the L^p solution of this system we minimize the L^p norm of the residuals:

$$\mathcal{N}_p(x) = (\sum_{i=1}^{n} |y_i - \sum_{j=1}^{m} A_{ij}x_j|^p)^{\frac{1}{p}} = \mathcal{L}_p(y - A.x)\ .$$

$\mathcal{L}_p$ is not a norm for $p < 1$, so, for the moment, I will only consider $p \geq 1$. I also omit the exponent $1/p$. To establish the normal equations, we set:

$$\frac{\partial \mathcal{N}_p}{\partial x_k} = 0 \qquad\qquad \text{for k=1 to m}\ .$$

The partial derivatives can be expressed as follows:

$$\frac{\partial \mathcal{N}_p}{\partial x_k} = -p\sum_{i=1}^{n} A_{ik}\text{sign}(r(i))\,|r(i)|^{p-1}\ .$$

where $r(i) = y_i - \sum_{j=1}^{m} A_{ij}x_j$ are the residuals. Then:

$$\frac{\partial \mathcal{N}_p}{\partial x_k} = -p\sum_{i=1}^{n} A_{ik}r(i)|r(i)|^{p-2}\ . \tag{1}$$

To form the normal equations we can now replace $r(i)$ by its expression:

$$\sum_{i=1}^{n} A_{ik}|r(i)|^{p-2}(\sum_{j=1}^{m} A_{ij}x_j) = \sum_{i=1}^{n} A_{ik}|r(i)|^{p-2}y_i$$

or, after some reorganization:

$$\sum_{j=1}^{m}(\sum_{i=1}^{n} A_{ik}|r(i)|^{p-2}A_{ij})x_j = \sum_{i=1}^{n} A_{ik}|r(i)|^{p-2}y_i \ . \tag{2}$$

There are m such equations ($k = 1$ to m). They can be expressed shortly in a matrix formulation:

$$W(i) = |r(i)|^{p-2} \qquad \text{for i=1 to n} , \tag{3}$$
$$W = \mathrm{diag}(W(1), \cdots, W(n)) ,$$

and finally:

$$A^T W A.x = A^T W.y \ . \tag{4}$$

This formulation is implicit and non-linear since the residuals $r(i)$, and so the matrix W, depend on the unknown vector x. However, for the classical least-squares inversion (p=2), W is the identity matrix, and equation (4) is the usual least-squares equation:

$$A^T A.x = A^T.y \ . \tag{5}$$

IRLS algorithm

At the iteration $k+1$, the algorithm solves:

$$A^T W^k A.x^{k+1} = A^T W^k.y \tag{6}$$

by taking:

- $W^0 = I_n$ (Identity matrix), at the first iteration,
- W^k formed with the residuals of iteration k ($r^k = y - Ax^k$), at the iteration $k+1$.

Byrd and Payne (1979) showed that this algorithm is convergent under two conditions:

- $W(i)$ must be non-increasing in $|r(i)|$,
- $W(i)$ must be bounded for all i.

The first condition is satisfied for $p \leq 2$; to assure the second condition, Huber (1981) suggested replacing equation (3) with:

$$W(i) = \begin{cases} |r(i)|^{p-2} & \text{if } |r(i)| > \varepsilon \\ \varepsilon^{p-2} & \text{if } |r(i)| \leq \varepsilon \end{cases} \tag{7}$$

for a small positive constant ε.
Equations (6) are the normal equations of the least-squares minimization of:

$$\mathcal{N}_2^k(x) = (y - Ax)^T W^k (y - Ax) \ .$$

W^k has the role of a weighting matrix, the inverse of a covariance matrix. By giving the large residuals small weights (or large variances), it reduces their role; on the contrary, well predicted values will be given large weights (small variances). Aberrant values, receiving low weights, will have a small influence: this makes the algorithm robust, less sensitive to noise bursts in the data. This is especially true for the extreme case $p = 1$, as it had been predicted by Claerbout and Muir (1973).

However, the L^1 norm is not strictly convex; so, the L^1 minimization problem might have non-unique solutions. The IRLS algorithm will lead us to one of these solutions. The fact that the first iteration ($W^0 = I_n$) is a least-squares inversion makes me think that this algorithm will lead us to a solution close to the L^2 solution. The choice of $W^0(i) = |y(i)|^{p-2}$ is not reasonable: during the algorithm, x stays close to 0, and the algorithm converges (if it does) to a vector x_∞ also close to zero, the residuals staying close to the original vector y.

Resolution of the normal weighted equations

We still have to solve the normal equations (6), at each step of the general algorithm. I omit the indexes k and $k+1$ in what follows.

The matrix A^TWA is symmetric and positive. However, it does not have a particular structure (such as the Toeplitz structure of A^TA in deconvolution). The computations of A^TWA and of its inverse are very expensive, in time and in memory. That is why several authors (Yarlagadda et al., 1985; Gersztenkorn et al., 1986; Scales et al., 1988) prefer iterative methods, especially conjugate gradient ("CG") methods, to direct inversion methods, like singular-value decomposition or orthogonalization by Givens rotations.

The advantages of conjugate-gradient methods are numerous. The matrix A^TWA does not need to be computed and stored. The algorithm "only" requires the computation of the gradient $g = A^TW.e$ (e: residuals) and of $WA.h$ (h: conjugate direction). The vectors e and h are updated at each iteration of the CG algorithm. W being diagonal, the product between W and a vector is trivial. Moreover, if the matrix A is sparse, its product by a vector can also be easily computed. If the problem is ill-conditioned, limiting the number of iterations is supposed to have an effect similar to that produced by adding a damping factor.

Finally, the convergence of the CG algorithm is controlled by three factors:

- the condition number of the problem;
- the choice of an initial estimate x^0 of x; the problem is modified into the resolution of: $(y - Ax^0) - A.\delta x = 0$.
- the structure of the eigenspectrum of A^TWA itself: the more gathered the eigenvalues are, the quicker the algorithm will converge.

In theory, if the matrix A^TWA has N different non-zero eigenvalues, the CG algorithm should converge without round-off errors in N iterations.

I will use the conjugate-gradient method in the IRLS algorithm applied to deconvolution. However, as I will show, its efficiency will not be optimal, unless some precautions are taken. For what follows, I mainly consider the case of the L^1 deconvolution, with $p = 1$.

IRLS ALGORITHM APPLIED TO L^1 DECONVOLUTION

Formulation of the problem

We want to solve the deconvolution problem:

$$y = a * x, \tag{8}$$

y being the data, x the unknowns, a the convolution filter. It can be written in the matricial form $A.x = y$, with:

$$x = (x_1, \ldots, x_{n_x})^T \; , \qquad y = (y_1, \ldots, y_{n_y})^T \; , \qquad n_y = n_a + n_x - 1 \; ,$$

$$A = \begin{pmatrix} a_1 & & & \\ a_2 & a_1 & 0 & \\ \vdots & a_2 & \ddots & \\ \vdots & & \ddots & \ddots \\ a_{n_a} & & & \ddots & a_1 \\ & a_{n_a} & & & a_2 \\ & & \ddots & & \vdots \\ & 0 & & \ddots & \vdots \\ & & & & a_{n_a} \end{pmatrix} \qquad (n_y \times n_x \text{ matrix}) \tag{9}$$

I will consider two kinds of deconvolution:

- predictive deconvolution: $(y_i) = (y_{i-1}) * f$, where f is the unknown predictive filter;
- deconvolution by a known wavelet: $y = w * r$, w being this wavelet, r unknown "reflection coefficients".

Ill-conditioning of L^2 deconvolution processes

To assure the convergence of the IRLS algorithm, the numerical result of each iteration must be close to the numerical limit of the corresponding CG algorithm, because it will be used through W as input for the next iteration. Moreover, the convolution matrices A and $W^{\frac{1}{2}}A$ may not be sparse (equation (9)), especially when $n_x \ll n_y$, as in predictive deconvolution. Then many operations are required to compute the convolutions and correlations in the conjugate-gradient algorithms, so the influence of round-off errors increases. A study of the convergence of the conjugate-gradient algorithm in this particular case is necessary.

The first iteration of the IRLS algorithm is a least-squares inversion. Then we can use some results about L^2 deconvolution problems. According to Szegö's theorem (Ekstrom, 1973), since A^TA is a Toeplitz symmetric positive matrix, its eigenvalues $(\lambda_1^2, \cdots, \lambda_{n_a}^2)$ can be expressed relatively to the Fourier transform $\hat{a}$ of the filter a. Especially, if (a_i) doesn't suffer from aliasing:

$$\lambda_i^2 \approx |\hat{a}(\omega_i)|^2 , \qquad \omega_i = \frac{2(i-1)\pi}{(n_a - 1)\Delta t} \qquad (i = 1 \text{ to } n_a) ,$$

where Δt is the time sampling of the vectors y, a and x (Figure **??**). It implies that, if the power spectrum of the filter a has some amplitudes near 0, for example if a is band-limited, the problem should be ill-conditioned. This is the case in predictive deconvolution, where the filter a is the seismic trace itself. Moreover, by oversampling the problem, we would remove the Nyquist frequency from the last frequency of the filter, and create smaller eigenvalues; we are thus increasing again the condition number of the problem (Figure **??**). Intuitively, we come closer to an infinite-dimension problem, where A^TA has an infinite set of positive eigenvalues, which decrease to 0 (Hilbert-Schmidt theorem for compact self-adjoint operators); this limit value causes the ill-conditioning of the problem for an infinite dimension.

In fact, we must be careful with Szegö's theorem. Milinazzo et al. (1987) have shown that, even if the power spectrum of the filter a has some 0 values, the minimum eigenvalue λ_{min} of A^TA might not vanish. They use an asymptotic development of λ_{min} versus n_x, which effectively goes to 0 when $n_x \to \infty$. Small-dimensioned problems might be well-conditioned even if there are zeros in the power spectrum of the convolution filter a.

Finally, two other remarks. First, adding some white noise increases the value of the minimum eigenvalue, and decreases the condition number, as the maximum eigenvalue is hardly modified. Secondly, if we want to apply CG algorithms to least-squares deconvolution, as in the first step of the IRLS algorithm, the convergence will be accelerated if the eigenvalues are gathered in groups: this will not be true with smooth spectra, or very irregular spectra.

Stability of the IRLS algorithm in deconvolution

The results concerning the conditioning and the eigenvalues of the equations (6) are unfortunately less accurate. The matrix A^TWA is no longer Toeplitz. To compute its eigenvalues would require us to know the singular-value decomposition of A, which increases the cost of the computations.

I could only observe that A^TA and A^TWA have more or less the same condition number during the IRLS algorithm; consequently, the weighted least-squares problems are also ill-conditioned if the original power spectrum of the filter a is band-limited. However, I have no general results concerning the repartition of the eigenvalues of A^TWA.

The choice of ε is not completely free. The synthetic examples I used showed that the more ill-conditioned the initial matrix A^TA is, the larger ε should be taken, otherwise the IRLS algorithm does not converge. For example, with a condition number equal to 1000, I took $\varepsilon = \text{Max}|y|/100$, and the convergence (rate 1/10000) was reached in 10 iterations; with a condition number equal to 10000, I took $\varepsilon = \text{Max}|y|/10$, even with double precision. Increasing ε makes W closer to the identity matrix, in which case the problem corresponds more to a L^2 minimization: we would be solving a mixed $L^1 - L^2$ problem, as I will explain later.

In conclusion, I may expect the convergence of the CG algorithms to be slow if the power spectrum of the filter a contains some small values. Despite the suggestion to limit the number of iterations in order to cope with the ill-conditioning, I prefer to increase their number, to be closer to the numerical limits. However, the speed of convergence is greatly increased if it is possible to give a first estimate of the unknown x: that is the case in predictive deconvolution, where the first estimate would be the L^2-Wiener filter.

SYNTHETIC EXAMPLE

I will illustrate the properties of the L^1 deconvolution on a synthetic trace, first without noise ("pure trace"), then with noise (several random spikes). Two kinds of deconvolution will be considered: deconvolution with a known wavelet, and predictive deconvolution. Though spiky noise is not usual in real data, it might be found for some kinds of acquisition (with a bad marine streamer for example) and is representative of the robustness of L^1-norm processes.

Figure **??** is a plot of all the synthetic inputs I used. The time sampling is dt=4 msec; the traces contain 512 samples. The first trace represents the synthetic wavelet; I chose it minimum-phase, in order to optimize the result of the standard L^2-Wiener predictive deconvolution, which I will compare to the L^1 deconvolution. Its frequency band is 10-70 Hz. The second trace represents a spiky sequence of reflection coefficients. Next comes the convolution between the synthetic wavelet and this spiky sequence. Finally, the fourth trace represents the noisy trace, formed with five high-amplitude spikes added to the pure trace.

Deconvolution with a known wavelet

Figure **??** shows the results of the L^2 and L^1 deconvolutions for the pure trace. Because the original wavelet lacked very low and very high frequencies, I used a damping factor for both deconvolutions: 1/1000 of the autocorrelation $\sum_i |w(i)|^2$ in the L^2 algorithm, and 1/1000 of the weighted autocorrelation $\sum_i W(i)|w(i)|^2$ at each iteration of the L^1 algorithm. The results are similar, as no noise was introduced. The ringing around the main peaks comes from the lack of high frequencies in the data, which forces the output to be convolved with a "sinc" function (impulse response of a high-cut filter).

Then I did the same process with the noisy trace. The results are presented on Figure **??**. Even with a damping factor, the L^2 deconvolution cannot avoid the influence of the noise, because a damping factor is adapted to gaussian noise. By giving the same weights to all the residuals ($W = Identity\ matrix$), it overestimates the importance of the noise bursts, and damages the output around these bursts. On the contrary, as expected, the L^1 deconvolution is insensitive to this noise.

Predictive deconvolution

The problem is now to estimate a predictive filter f with the input traces. As the input traces have 512 samples, I took the length of the filter equal to 50 samples.

The residuals of L^2 and L^1 deconvolution on the pure trace are similar (Figure **??**). As the input wavelet is minimum-phase, the L^2 deconvolution is efficient. However, the L^1 deconvolution has still improved the result. Supposing the a priori statistical distribution of the residuals is *exponential* (and not Gaussian as in L^2 deconvolution), it has forced the smallest values of the L^2 residuals to come closer to 0, and has increased the highest values.

In this case, the IRLS algorithm converged quickly, because the initial condition number was low (≈ 1000): with real data, we can expect this number to be larger than 10^4. I took the initial estimate of the filter equal to the L^2 filter. In Figure **??**, I plotted the matrices $A^T A$, $A^T W^1 A$, $A^T W^{30} A$; W^1 and W^{30} were respectively computed with the L^2 residuals (after the first iteration of the IRLS algorithm) and the L^1 residuals (after the 30^{th} iteration of the IRLS algorithm). The non-Toeplitz structure of $A^T W^1 A$ is evident. However, $A^T W^{30} A$ is close to a Toeplitz structure, because W is closer to the Identity matrix (to the factor ε): as the algorithm proceeds, more and more residuals become smaller than ε, and their weights get all equal to $1/\varepsilon$.

Finally, I applied L^2 and L^1 predictive deconvolution to the noisy trace. The residuals are represented in Figure **??**; I also plotted the difference between these residuals and the residuals of the pure trace. In both cases, the residuals still contain the original noise bursts. This is not surprising, because the predictive filters could not predict them. It is more interesting to study the influence of the noise bursts on the rest of the trace. The two plots of the perturbation of the residuals after introduction of the noise show that the L^2 residuals are much more perturbed than the L^1 residuals. This means that the L^2 predictive filter has tried to remove these noise bursts, and its modification has of course transformed the rest of the trace. On the contrary, the L^1 filter, being insensitive to the spikes, has hardly been changed: its residuals are not very different from the initial ones, except for the noise bursts. Thus, the L^1 deconvolution is more reliable than the L^2 deconvolution.

Other families of noise

I tried the same processing with exponential and gaussian noise. Unfortunately, the L^1 minimization is no more efficient. The noise samples now have various amplitudes, about the same repartitions as the original trace, and cannot be isolated from the seismic samples by the original criterion of small weight: they have all kinds of weighting. The L^1 norm itself does not improve the result of the L^2 deconvolution; the most usual way to suppress the noise is to use large damping factors, which is not a typical characteristic of the L^1 norm. Moreover, even if the noise n is gaussian, white, and uncorrelated with the seismic data, the L^1 normal equations cannot use these properties, because the correlations terms $(n^T n,\ y^T n)$ will be weighted by the matrix W, and replaced by $n^T W n$ and $y^T W n$ (which may not vanish). I will suggest in the last part a method to cope simultaneously with two kinds of noise, for example gaussian and spiky noise, using L^1 and L^2 norms simultaneously.

L^1 norm and non minimum-phase wavelets

A predictive L^2 deconvolution produces a minimum-phase inverse filter, independently of the phase of the wavelet w. Thus it would be interesting to study the effect of the L^1 deconvolution on non minimum-phase inputs. I used the same input reflection sequence as before, but changed the input wavelet into a constant phase (60^o) wavelet. I applied L^2 and L^1 deconvolution on the new synthetic trace. I initialized the L^1 algorithm first with the L^2 filter, then without the input filter. The results are shown on Figure **??**.

It appears that the L^1 deconvolution does not improve the phase properties of the output. In one case (starting from L^2 filter), the filter is from the start close to the L^2 filter; in the other case, the first iteration of the IRLS algorithm forces the filter to be close to the L^2 filter. In the first case, the L^1 deconvolution results in a sharpening of the L^2 result, again because the residuals have an a priori exponential distribution.

The L^1 deconvolution can be related to the variable norm deconvolution (Gray, 1979), in the particular case when the processing tries to draw the statistical distribution of the time series from a gaussian one (seismic trace) to an exponential one (residuals). However, L^1 and Gray's deconvolutions are slightly different, because the IRLS algorithm does not suppose any a-priori distribution for the inputs; the normal equations of the variable norm deconvolution are in fact a linear combination of the L^2 normal equations and of the L^1 normal equations.

In conclusion, the IRLS algorithm in predictive deconvolution essentially sharpens the residuals. If an initial estimate is used (Wiener filter, Burg filter, or any minimum-entropy deconvolution filter), the residuals of *this initial filter* will be sharpened; otherwise, the result will be a sharpened version of the L^2 residuals. It would be interesting in fact to modify the first iteration of the algorithm, but would it still guarantee the convergence?

EXAMPLE ON REAL DATA

I applied L^1 deconvolution on a marine CMP gather from BP Alaska. This gather, recorded in South California, is composed of 36 traces; the near-offset is 65.7 m, the intertrace distance 33.3 m. The time window I used is 0.7-3.0 sec; the time sampling is 4 msec. The CMP gather is presented in Figure **??**, along with the residuals of the L^2 deconvolution. For graphics convenience, I only present a time window between 0.7 and 2 sec. The residuals were computed with the same filter for all the traces, composed of 50 samples; I used a small prewhitening factor (5 %), in accordance to the small level of noise before the first arrival. It appears on the first arrival that the seismic wavelet has a precursor, making it clearly non minimum-phase; the source was composed of two water-guns.

I applied L^1 deconvolution on this gather, with the L^2 predictive filter as initial estimate; I computed one 50-sample filter for all traces. The residuals of this L^1 deconvolution are presented in Figure **??**, with also the difference between the L^2 residuals and the L^1 residuals. As I expected, the L^1 residuals are a sharpened version of the L^2 residuals: the largest amplitudes have been increased, the smallest have been decreased. This is not really obvious on the sections, but it appeared clearly when I computed the kurtosis of the traces in the central part of the gather (between 1.5 and 2.5 sec, to avoid the strong water-bottom reflections): the kurtosis of the L^1 residuals were larger than those of the L^2 residuals. It also appears on the difference section when we look at the water-bottom reflection. However, the estimation of the phase of the seismic wavelet has not been improved, and especially the precursor has not been removed, but accentuated.

This sharpening of the residuals presents the advantage of increasing the contrast between reflectors, and stressing the main reflectors. Moreover, as this gather presents some lateral variations of amplitude, these variations will be enhanced by the sharpening of the residuals.

I also tried a L^p deconvolution with $p = 0.1$, though the objective function is no longer a norm. However, with the L^2 filter as initial estimate, the algorithm converged. As expected, the result was once again a sharpening of the L^2 residuals, but it was very similar to the result of the L^1 deconvolution.

MIXED $L^1 - L^2$ NORM MINIMIZATIONS

Though the theoretical problem is a L^p norm minimization, the introduction of the parameter ε changes the theoretical resolution. In fact, the limit x_{lim} of the algorithm will solve the normal equations:

$$A^T W_\varepsilon A.x_{lim} = A^T W_\varepsilon.y \tag{10}$$

where:

$$r(i) = (y - A.x)(i)$$
$$W_\varepsilon(i) = \begin{cases} |r(i)|^{p-2} & \text{if } |r(i)| > \varepsilon \\ \varepsilon^{p-2} & \text{if } |r(i)| \leq \varepsilon \end{cases} \tag{11}$$

These equations are the normal equations of the next minimization problem:

$$\begin{cases} \text{Min} \quad (\varepsilon^{p-2}/2) \sum_{i/|r(i)|<\varepsilon} |r(i)|^2 + \frac{1}{p} \sum_{i/|r(i)|>\varepsilon} |r(i)|^p \\ r(i) = (y - A.x)(i) \ . \end{cases} \tag{12}$$

Effectively, if $|r(i)| < \varepsilon$, a small perturbation of x will maintain this inequality, because of the continuity of the matricial product. This property also holds for the other inequality: $|r(i)| > \varepsilon$. It is no longer true if we take large inequalities ($\leq$ or $\geq$). So, we can calculate the gradient of the objective function in equation (12) and keep the same inequalities; if I call this function $\mathcal{N}_{mix}$, then:

$$\frac{\partial \mathcal{N}_{mix}}{\partial x_k} = -\varepsilon^{p-2} \sum_{i/|r(i)|<\varepsilon} A_{ik} r(i) - \sum_{i/|r(i)|>\varepsilon} A_{ik} r(i) |r(i)|^{p-2} \ . \tag{13}$$

Finally, replacing $r(i)$ by $(y - A.x)(i)$, we find the normal equations (10). This result is true if the objective function $\mathcal{N}_{mix}$ is minimized for a vector x such that none of the residuals $|r(i)|$ is equal to ε.

So, the IRLS algorithm minimizes what would be called a *mixed* function. But this objective function $\mathcal{N}_{mix}$ is not convex: the summation of any two vectors x and x' can completely modify the repartition between residuals larger or smaller than ε if these vectors are not close to each other. In fact, this function is strictly convex inside the regions limited by the $2 * n_y$ hyperplanes:

$$r(i) = \varepsilon \qquad \text{and} \qquad r(i) = -\varepsilon \ ,$$

and it is not continuous on these hyperplanes. On Figure **??**, I plotted for example the function:

$$\mathcal{N}_{mix}(x_1, x_2) = \frac{1}{4} \sum_{i/|x_i|<2} |x_i|^2 + \sum_{i/|x_i|>2} |x_i| \ ,$$

for $|x_i| < 4$, obtained with p=1 and ε=2. This kind of function has one global minimum, and no other local minimum. However, if we cross the hyperplanes to reach the region of the global minimum, there might be problems of computation of the gradient near these hyperplanes, and so problems of convergence. This problem does not appear with the IRLS algorithm and the L^1 norm, because ε is small: the central region is smaller, and the discontinuity (whose jump is proportional to ε) becomes negligible. In fact, the fewer hyperplanes are crossed during the algorithm, the more efficient the convergence will be. Thus, it is better to have from the beginning as many residuals as possible smaller than ε; that is why the use of a first estimate of the filter enhances the convergence, and too small an ε hinders it.

There is in fact another reason which explains the slowness of the convergence of the IRLS algorithm for a small ε. A *theoretical* n-component L^1 solution must produce at least n residuals equal to zero. But during the algorithm, if any of the residuals gets smaller than ε, it creates a weight ε^{p-2} for the next iteration. If ε is small, this weight will be relatively large, and will force the same residual, at the next iteration, to be smaller than ε. Thus, the algorithm tries to keep this residual small, independently of whether or not it belongs to the set of theoretical zero residuals, and we might need many iterations to force the algorithm to "forget" this residual.

The objective function $\mathcal{N}_{mix}$ could be interesting for a mixed $L^1 - L^2$ problem, for example, if ε were taken large enough, in the minimization problem:

$$\begin{cases} \text{Min} \quad (1/2\varepsilon)\sum_{i/|r(i)|<\varepsilon}|r(i)|^2 + \sum_{i/|r(i)|>\varepsilon}|r(i)| + \alpha\sum_i |x(i)|^2 \\ r(i) = (y - A.x)(i)\ . \end{cases} \tag{14}$$

The convergence would be ensured by the proximity of the L^2 minimization problem. Problems of computation of the gradient on the hyperplanes could be solved by smoothing the objective function near these hyperplanes. Such a smoothing could be assured by taking:

$$W(i) = (|r(i)|^{2-p} + \varepsilon)^{-1}\ ,$$

but this choice would not actually correspond to the same minimization problem. An intermediate choice would be to assure only the continuity of the objective function $\mathcal{N}_{mix}$ by taking:

$$\mathcal{N}_{mix}(x) = \frac{1}{2}\sum_{i/|r(i)|\leq\varepsilon}(|r(i)|^2/\varepsilon + \varepsilon) + \sum_{i/|r(i)|>\varepsilon}|r(i)|\ .$$

The *mixed* function of the problem (14) could handle two kinds of noise at the same time: spiky high-amplitude noise with the L^1 norm, and gaussian noise with the L^2 norm and the damping factor α.

CONCLUSIONS

L^1 deconvolutions with IRLS algorithm are especially efficient and robust in the presence of high-amplitude noise bursts. In the general case, their residuals are a sharpened version of the L^2 residuals, as in predictive deconvolution: the largest amplitudes are increased, the smallest are decreased, and thus the contrast between small and large events is heightened. However, we cannot expect many improvements in the estimation of the phase of the seismic wavelet. Finally, this algorithm could be used for minimizations with mixed $L^1 - L^2$ norms, and it would thus cope with the simultaneous presence of different kinds of noise.

ACKNOWLEDGEMENTS

I would like to thank Clement Kostov and Bill Harlan for many fruitful discussions and suggestions about deconvolution processes. I am also grateful to SNEA(P) for supporting my study at Stanford.

REFERENCES

Byrd, R.A., Payne, D.A., 1979, Convergence of the IRLS algorithm for robust regression: Technical Report 313, John Hopkins University (unpublished)

Claerbout, J.F., Muir, F., 1973, Robust modeling with erratic data: Geophysics, **38**, pp. 826-844.

Ekstrom, M.A., 1973, A spectral characterization of the ill-conditioning in numerical deconvolution: IEEE Trans. Audio Electroacoust., vol. AU-**21**, pp. 344-348.

Gersztenkorn, A., Bednar, J.B., Lines, L.R., 1986, Robust iterative inversion for the one-dimensional acoustic wave equation: Geophysics, **51**, pp. 357-368.

Gray, W.C., 1979, Variable norm deconvolution: SEP **19**, Ph.D. Thesis, Stanford University.

Milinazzo, F., Zala, C., Barrodale, I., 1987, On the rate of growth of condition numbers for convolution matrices: IEEE Trans. on Acoustics, Speech, and Signal Processing, ASSP-**35**, pp. 471-475.

Scales, J.A., Gersztenkorn, A., Treitel, S., 1988, Fast Lp solution of large, sparse, linear systems - Application to seismic travel time tomography: Journal of Computational Physics, **75**, pp. 314-333.

Taylor, H.L., Banks, S.C., McCoy, J.F., 1979, Deconvolution with the L1 norm: Geophysics, **44**, pp. 39-52.

Yarlagadda, R., Bednar, J.B., Watt, T.L., 1985, Fast algorithms for Lp deconvolution: IEEE Trans. on Acoustics, Speech, and Signal Processing, ASSP-**33**, pp. 174-182.