Newton trace balancing

Stewart A. Levin

INTRODUCTION

In the last SEP report (SEP-41) I began a study of surface-consistent deconvolution in which I opted not to follow the classical route of transforming measurements to a log-linear model. Instead I'm fitting measurements directly to the convolutional or multiplicative model from which the log-linear model was derived. The reason for this is to bypass the problems of unwrapping complex phase and the distortion of error statistics one runs into when taking logarithms.

The prototype I'll tackle in this article is the trace balancing problem. Given a number representing, say, the rms amplitude of each trace, I want to decompose these amplitudes in a surface-consistent fashion. This was done by Taner and Koehler (1981) with the log-linear model.

NEWTON FACTOR ANALYSIS

The simplest trace balancing problem is factor analysis or analysis of variance (ANOVA). This may be posed as follows.

Given a rectangular array a_{ij}, $i=1,m$, $j=1,n$ I want to find vectors s_i and g_j to minimize

$$\| A - SG^T \|_F \tag{1}$$

where the Froebenius norm $\| \quad \|_F$ of a matrix is defined by

$$\| M \|_F = \left[\sum_i \sum_j m_{ij}^2 \right]^{1/2} . \tag{2}$$

To save some factors of –1 and 2, I'll rewrite the problem as

$$\max \ J = -1/2 \sum_{ij} (a_{ij} - s_i g_j)^2 \tag{3}$$

and set derivatives of J equal to zero:

$$0 = \frac{\partial J}{\partial s_i} = \sum_j (a_{ij} - s_i g_j) g_j \tag{4}$$

$$0 = \frac{\partial J}{\partial g_j} = \sum_i (a_{ij} - s_i g_j) s_i . \tag{5}$$

For future reference define vector functions

$$J_S \equiv \frac{\partial J}{\partial s_i} \tag{6}$$

$$J_G \equiv \frac{\partial J}{\partial g_j} \quad . \tag{7}$$

From (4) and (5) the answer to this rectangular fit satisfies

$$AG = \lambda_g S \tag{8}$$

and

$$A^T S = \lambda_s G \tag{9}$$

with $\lambda_g = G^T G$ and $\lambda_s = S^T S$. Applying A^T to (8) and substituting in (9) yields

$$A^T AG = \lambda_s \lambda_g G \quad . \tag{10}$$

Similarly

$$AA^T S = \lambda_s \lambda_g S \quad . \tag{11}$$

Thus G and S are eigenvectors (with the same eigenvalue) of $A^T A$ and AA^T respectively. Employing the singular value decomposition (SVD)

$$A = P \Lambda Q^T \quad , \tag{12}$$

the solution is to choose S as the column of P corresponding to the largest singular value on the diagonal of Λ and G as the corresponding column of Q.

Assume now that we didn't recognize the structure of (4) and (5). We could still try to solve this using Newton's root-search method to find zero(s) of the vector function $F = (J_S, J_G)$. To apply Newton's method we'll need the (symmetric) derivative of this function:

$$DF = \begin{pmatrix} -G^T G \, I_m & A - 2SG^T \\ A^T - 2GS^T & -S^T S \, I_n \end{pmatrix} \tag{13}$$

Newton's iteration takes a starting guess S_o, G_o and successively computes increments δS, δG according to

$$DF \begin{pmatrix} \delta S \\ \delta G \end{pmatrix} = -F \tag{14}$$

until convergence is reached.

A difficulty arises in performing (14) for this problem because the matrix DF is *singular* at the desired solution. This is a direct consequence of the ambiguity in S and G where S may be multiplied by an arbitrary constant and G divided by the same constant without changing the value of the objective function J. To handle this we'll need a constraint. Requiring

$$\|S\| \quad = \quad \|G\| \tag{15}$$

turns out to be a good choice. (More on this below.)

Let's turn now to finding a good starting guess S_0 and G_0. This is a necessity because Newton's method both converges rapidly (second order) near the root and diverges rapidly when too far from the root. To avoid the latter disaster, I'll use a continuation scheme devised by Herbert Keller of CalTech (Keller 1978). The idea is to concoct synthetic data for which we do have a good starting guess and slowly perturb this synthetic and the starting guess until it agrees with the data we're really interested in.

So I start with an easier problem for which I know an answer. If I replace A by a constant matrix $\overline{A}$, say the average of the elements of A, such an answer is immediate:

$$\begin{aligned} s_0 \quad &= \quad \pm\sqrt[4]{\overline{a}^2 n\,/m} \\ g_0 \quad &= \quad \pm\sqrt[4]{\overline{a}^2 m\,/n} \end{aligned} \qquad . \tag{16}$$

Now I'll introduce an auxiliary parameter $0 \leq \sigma \leq 1$, called a *homotopy* parameter, and try to fit S and G to the linear combination

$$\sigma A \;+\; (1-\sigma)\overline{A} \qquad . \tag{17}$$

Then the partial derivatives of F with respect to σ are given by

$$\frac{\partial J_S}{\partial \sigma} \quad = \quad \sum_j \,(a_{ij} \;-\; \overline{a})g_j \tag{18}$$

and

$$\frac{\partial J_G}{\partial \sigma} \quad = \quad \sum_i \,(a_{ij} \;-\; \overline{a})s_i \qquad . \tag{19}$$

At $\sigma \;=\; 0$ expand F about S_0, G_0

$$F \quad = \quad 0 \;+\; F_\sigma\,\delta\sigma \;+\; DF\begin{pmatrix} \delta S_0 \\ \delta G_0 \end{pmatrix} \;+\; \dots \tag{20}$$

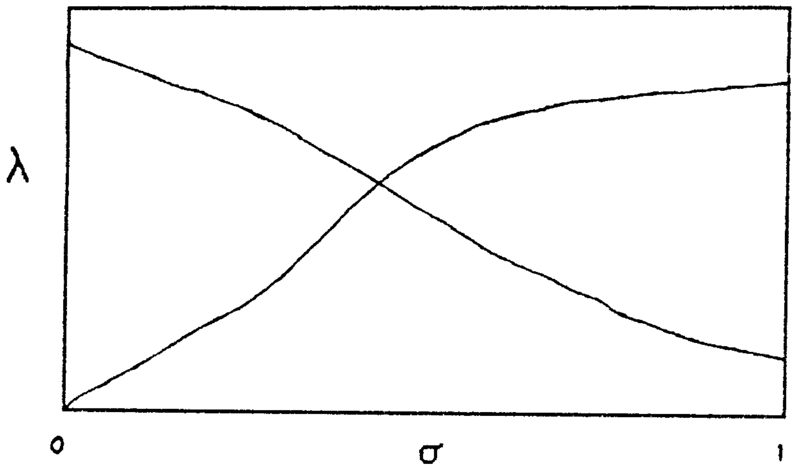
to determine that an appropriate correction, or *homotopy step,* is given by

$$DF\begin{pmatrix} \delta S_0 \\ \delta G_0 \end{pmatrix} \quad = \quad -F_\sigma\,\delta\sigma \qquad . \tag{21}$$

In practice one starts with $\delta\sigma \;=\; 1$ and decreases it only if Newton's method doesn't converge in a few iterations.

What problems do we face? I've already mentioned one: the ambiguity in S and G. This implies that DF (13) is rank deficient (by one) at the desired solution. For this reason a modified inverse, incorporating constraint (15), will be needed.

Another problem is brought to light by our initial eigenvalue analysis - there are multiple local maxima of J, one for each eigenvalue, with the global maximum associated with the largest eigenvalue in the SVD decomposition (12). Fortunately our choice of simple problem $\overline{A}$ has at least started us out with the largest eigenvalue solution. Indeed all other eigenvalues are zero. However as we increase σ we might encounter the situation depicted below



where the largest eigenvalue at $\sigma = 1$ corresponds to one of the eigenvalues at $\sigma = 0$. C'est la vie. (It's possible, though expensive, to track all of the zero eigenvalue solutions from $\sigma = 0$ as well.)

There are a number of ways to deal with the rank deficiency problem. The way I've chosen is to augment the problem $F = 0$ with the additional constraint (normalization) condition (15). Writing the normalization function as

$$N(S,G) = 0 \quad , \tag{22}$$

we get the augmented derivative matrix

$$\begin{bmatrix} F_S & F_G \\ N_S & N_G \end{bmatrix} \tag{23}$$

where for constraint (15) the new bottom row is

$$\begin{aligned} N_S &= S^T \\ N_G &= -G^T \end{aligned} \tag{24}$$

and the right hand sides of (14) are (21) augmented with a corresponding zero in the last position. Keller gives conditions under which this augmented problem has a solution. Basically they say that the constraint should be consistent with the unconstrained equations; more technically, the RHS must satisfy a solvability condition.

To understand the nature of this solvability condition, recall that the homotopy step (21) is really a question of tracking an eigenvector/eigenvalue away from $\sigma = 0$. Let's review how this is done using perturbations methods.

Suppose we have

$$\Gamma(\sigma)\, X(\sigma) \quad = \quad \lambda(\sigma)\, X(\sigma) \tag{25}$$

with Γ symmetric and some known solution at $\sigma = 0$. We want to estimate $X(\sigma)$ and $\lambda(\sigma)$ to first order in σ. Write

$$\begin{aligned} \Gamma(\sigma) &= \Gamma(0) + \sigma\Gamma'(0) + O(\sigma^2) && (26)\\ X(\sigma) &= X(0) + \sigma X'(0) + O(\sigma^2) && (27)\\ \lambda(\sigma) &= \lambda(0) + \sigma\lambda'(0) + O(\sigma^2) && (28) \end{aligned}$$

to first order. Substituting these into (25), one can cancel the constant terms and divide out σ to get

$$\Gamma' X + \Gamma X' \quad = \quad \lambda X' + \lambda' X \tag{29}$$

or

$$(\Gamma - \lambda)\, X' \quad = \quad -(\Gamma' - \lambda')\, X \quad . \tag{30}$$

Since λ is an eigenvalue of Γ and X is the corresponding eigenvector, X^T times the left hand side of (30) is identically zero, and so $(\Gamma - \lambda)X'$ must be orthogonal to X. Thus we have a solvability condition that

$$X^T(\Gamma' - \lambda')X \quad = \quad 0 \tag{31}$$

or

$$X^T\Gamma' X \quad = \quad \lambda' X^T X \tag{32}$$

whence

$$\lambda' \quad = \quad X^T\Gamma' X / X^T X \quad . \tag{33}$$

Using this value of λ', X' is determined only up to some multiple of X. To fix this multiple we impose a constraint condition such as $X^T X = 1$ which makes

$$X^T X' \quad = \quad 0 \tag{34}$$

and there will be no component of X in X'.

Let's look now for equivalent solvability condition(s) for our problem. Recall equation (21)

$$DF\begin{pmatrix}\delta S_o\\ \delta G_o\end{pmatrix} = -F_\sigma\,\delta\sigma \qquad . \tag{21}$$

For this equation to have a solution, the right hand side will have to be orthogonal to the null space of DF (technically DF transpose.) For this factor analysis problem there is only free scaling parameter and so in this case the null space has dimension 1. Let Q be an element of this null space. Then our solvability condition is

$$Q^T F_\sigma = 0 \qquad . \tag{35}$$

What is Q? Here I'll make use of the nature of the ambiguity in the solution. This ambiguity says that if (S, G) is a solution so is $(\gamma S, \gamma^{-1} G)$. Linearizing about $\gamma = 1$, we find

$$Q = \begin{pmatrix} S\\ -G\end{pmatrix} \qquad . \tag{36}$$

So to solve (21) we need

$$F_\sigma^T\begin{pmatrix} S\\ -G\end{pmatrix} = 0 \qquad . \tag{37}$$

From (18) and (19) this becomes

$$0 = \sum_{ij}(a_{ij} - \overline{a})g_j s_i - \sum_{ij}(a_{ij} - \overline{a})s_i g_j \qquad , \tag{38}$$

which is an identity. Solvability condition (37) is thus verified for the homotopy step (21) and the step is determined up to a multiple of $(S, -G)$. Differentiating constraint condition (15) produces

$$S^T \delta S - G^T \delta G = 0 \tag{39}$$

which is (34) with $X \equiv Q$ and $X' \equiv (\delta S, \delta G)$. Therefore my choice of constraint condition (15) makes the aforementioned multiple zero (i.e. a minimum norm step) which is why I said it was a good choice.

The vector $(S, -G)$ is also orthogonal to the RHS of (14) for

$$\begin{aligned} &S^T(AG - G^T GS) - G^T(AS - S^T SG)\\ &= S^T AG - G^T GS^T S - G^T AS + S^T SG^T G\\ &= 0 \end{aligned} \tag{40}$$

since all terms are scalars and so equal to their transposes.

In fact, DF is not necessarily rank deficient when computing the Newton step with (14). The deficiency appears only at a solution of (4) and (5). Numerically, however, we must consider DF singular in a neighborhood of the desired solution. Steps taken to handle the singularity in the homotopy step (21) should be equally applicable to the Newton iteration (14).

So, how do I actually solve (21) or (14)? With the constraint method previously outlined, I set up an $m+n+1$ by $m+n$ DF matrix with the bottom row Q^T. I also append a zero to the bottom of the RHS vectors as well. These are then input to subroutine GOLUB (QR factorization, see FGDP p. 116-119) to get the desired minimum norm, least squares solution. To be conservative, I then renormalize $S+\delta S$ and $G+\delta G$ by scaling them inversely so that (15) holds.

Using Cleve Moler's interactive mathematical program (matlab), I ran a test case with a 4 by 7 array of random numbers uniformly distributed between 0 and 1 as input. After one homotopy step with $\delta\sigma = 1$, the first Newton step matched the correct (SVD) answer to better than five significant figures.

TRACE BALANCING: WHEN THE DATA AIN'T THERE

A typical seismic survey doesn't record a trace for every shot and receiver combination. For trace balancing this means we have only a sparse subset of the full matrix A from which to estimate the factors S and G. So suppose we're given amplitudes a_{ij} for $(i,j) \in \Omega \subset [1,m]\times[1,n]$. As before I want to solve the least squares problem

$$\max J = -1/2 \sum_{\Omega} (a_{ij} - s_i g_j)^2 \quad . \tag{41}$$

Define Ω_i as the set of (i,j) in Ω for i fixed and similarly for Ω_j. The partial derivatives of J are

$$\frac{\partial J}{\partial s_i} = \sum_{j \in \Omega_i} (a_{ij} - s_i g_j) g_j \tag{42}$$

and

$$\frac{\partial J}{\partial g_j} = \sum_{i \in \Omega_j} (a_{ij} - s_i g_j) s_i \quad . \tag{43}$$

Again I'll use Newton's method to search for the zero of this vector function $F = (J_S, J_G)$ and so I'll want the derivative of F as well. This may be written in the block matrix form as

$$DF = \begin{pmatrix} \operatorname{diag}(-\sum_{\Omega_i} g_j^2) & (A - 2SG^T)|_{\Omega} \\ (A^T - 2GS^T)|_{\Omega} & \operatorname{diag}(-\sum_{\Omega_j} s_i^2) \end{pmatrix} \tag{44}$$

where the restriction to Ω means to use zero when $(i,j) \notin \Omega$.

S and G are still at best determined only up to a scale factor. I'll resolve the scaling by again requiring

$$\|S\| = \|G\| \quad , \qquad (45)$$

or in differential form,

$$\left(S^T, -G^T\right) \begin{pmatrix} \delta S \\ \delta G \end{pmatrix} = 0 \quad . \qquad (46)$$

Using the same solution method discussed earlier, I'll take homotopy steps

$$DF \begin{pmatrix} \delta S \\ \delta G \end{pmatrix} = -\sigma F_\sigma \qquad (47)$$

to generate starting guesses for Newton iterations

$$DF \begin{pmatrix} \delta S \\ \delta G \end{pmatrix} = -F \quad . \qquad (48)$$

As earlier indicated, in a typical problem DF should be quite sparse with perhaps one element in forty being nonzero. Therefore I'll employ least-squares equation solver LSQR (Paige and Saunders, 1982) to solve (46)+(47) and (46)+(48) taking advantage of this sparsity. LSQR is an iterative (conjugate gradient) method of solving $MX \approx Y$ that requires user subroutines to compute $V+MW$ and $W+M^T V$ for arbitrary vectors V and W.

To test this method I used the same test as for factor analysis: a 4 by 7 matrix of random numbers uniformly distributed between 0 and 1. To get 6 digit accuracy (using double precision arithmetic) required over 3,000 (tiny) homotopy steps and about 40 times that number of Newton iterations and about one hour of CPU time on a VAX-780. Clearly something was wrong. The only difference between this example and the factor analysis example was the choice of equation solver; in the earlier study I used Golub's QR method.

To better define the problem, I did two sets of experiments. In the first I added a bias to the uniform random number to push them farther away from zero. With a bias of about .25 or greater, the iteration converged in one homotopy step. A bias of .125 or lower produced the disappointing behavior we saw with the original zero bias. The second experiment was to lower the 6 digit accuracy requirement to 5 digits. In that case only one homotopy step was needed.

COPING

As the LSQR method had trouble imposing norm constraint (45) indirectly via differential constraint (46), I reformulated the problem. I did this by modifying the original least squares problem (41). My new problem is

$$\max \quad \hat{J} = -1/2 \sum_{ij} (a_{ij} - s_i g_j)^2 - \gamma/4 \left(\sum_i s_i^2 - \sum_j g_j^2 \right)^2 \qquad (49)$$

for some, to be chosen, constant γ. This is a penalty weight, *not* a Lagrange multiplier. Setting partial derivatives to zero now gives the equations

$$0 = \hat{J}_S = J_S - \gamma(S^T S - G^T G)S \tag{50}$$

and

$$0 = \hat{J}_G = J_G + \gamma(S^T S - G^T G)G \quad . \tag{51}$$

When $\gamma = 0$ we have the original equations (42) and (43). For factor analysis, where Ω is $[1, m] \times [1, n]$, choosing $\gamma = 1$ (why else did I divide by 4?) simplifies the equations to

$$AG = (S^T S)\, S \tag{52}$$

and

$$A^T S = (G^T G)\, G \quad , \tag{53}$$

nearly the same form as the original factor analysis equations (8) and (9). The difference is that these latter imply constraint (45) automatically. To see this directly, multiply (52) by S^T and (53) by G^T and notice the left hand sides are transposes of the same scalar. Therefore one can expect for our trace balancing problem that the modified DF matrix that corresponds to (44) will now be of full rank. This new second derivative matrix is

$$\hat{DF} = DF + (S^T S - G^T G)\begin{pmatrix} I_m & 0 \\ 0 & -I_n \end{pmatrix} - 2QQ^T \tag{54}$$

where Q is the vector $(S, -G)$ we saw earlier in factor analysis. Notice that while $\hat{DF}$ is no longer sparse, it is almost as cheap to apply to a vector as the sparse DF of equation (44), an important consideration for the iterative LSQR algorithm.

I tested this approach with the same 4 by 7 synthetic as before. This time it required one homotopy step and two Newton steps to get 10+ digit accuracy. When the dimensions of the problem were increased to 35 by 72, it took one more Newton step and the final accuracy was about 9 digits.

Next I applied the method using a simulated recording geometry of 32 shots and 7 geophones per shot. With the smaller bandwidth, three homotopy steps and a total of 19 Newton iterations were required for 5 digit accuracy. From experience with the surface consistent static model this was not totally unexpected. In the limiting case of a bandwidth of 1 the answer has many degrees of freedom, even with the norm constraint. Indeed the components of S and G then completely decouple allowing arbitrary scalars to be multiplied onto each component separately without affecting the (perfect) fit. In the small bandwidth case I therefore expect to have poorly constrained solutions with the components at one end of the line only mildly affecting the answer at the other end.

A PAUSE FOR REFLECTION

In last year's report (Levin 1984), I tried my hand at surface-consistent deconvolution by iteratively designing and applying a decon filter for each shot followed by a decon filter for each receiver, and so on. This was an analog of the Gauss-Seidel method for decomposing statics in a surface-consistent manner. The Gauss-Seidel method also has a direct equivalent for the trace balancing problem. By setting the derivatives in equations (42) and (43) to zero, I can rewrite the trace balancing problem as finding a solution to the equations

$$s_i \sum_{j \in \Omega_i} {g_j}^2 = \sum_{j \in \Omega_i} a_{ij}\, g_j \tag{55}$$

and

$$g_j \sum_{i \in \Omega_j} {s_i}^2 = \sum_{i \in \Omega_j} a_{ij}\, s_i \quad . \tag{56}$$

Fixing the g_j's, new s_i's are directly given by (55). With these new s_i's, (56) produces new g_j's and so on. One should also impose the constraint (45) at each step for stability. This is no hardship for inspection of (56) shows that scaling the s terms by a constant factor inversely scales the g's, which is what the constraint wants us to do anyhow. For the homotopy step I'll still use the linearized extrapolation from equation (47). To see how this compares with the Newton method I've been using I reran the test with 32 shots and 7 geophones per shot. Five digit accuracy was produced with one homotopy step and 129 "Gauss-Seidel" iterations. This is about six times greater than the number of Newton steps taken in the previous solution. However, the computer time required was almost six times *less* than before because of the time required to solve the linear systems (55) and (56) was small compared to the time LSQR required to solve the linear system (48). Clearly some more study is needed.

This "Gauss-Seidel" method suggests a different and more sensible way to do surface-consistent deconvolution than in last year's report. There I simply iterated by deconvolving each shot gather, then each receiver gather, then, say, again over shots and receivers, etc. until the results appeared unchanged from iteration to iteration. This left the deconvolution filter length pretty much uncontrolled – the more iterations the more filters cascaded together, and the longer the effective filter length. In the new method I'm working on, one stores the deconvolution filters rather than than the deconvolved gathers and thereby can fix the total filter length. Jon Claerbout, in a recent discussion, suggested a good framework in which to describe the essentials of both the Newton and the Gauss-Seidel iterative methods. If I may safely paraphrase the discussion, consider the dereverberation model

$$(1 - S_i) * (1 - G_j) * Data_{ij} \quad \approx \quad Random \tag{57}$$

where S_i and G_j are prediction (dereverberation) filters to be determined. Let us take initial guesses $\overline{S}_i$ and $\overline{G}_j$ for these unknown filters. A nonlinear approach, similar to the method just described, is to first fix all the G_j's to get the smaller (linear) problem of designing a single filter of a given fixed length best deconvolving each shot gather after the G filter have been applied to it. Claerbout (1985) in this volume discusses a family of methods for doing this. Note that the effective filter length stays fixed. It is the filters, rather than the filtered data, that are passed from one iteration to the next.

In a method such as the Newton scheme, we would calculate perturbations δS_i and δG_j that fit the linearized model

$$\Big[(1 - \delta G_j) * (1 - \overline{S}_i) \;+\; (1 - \delta S_i) * (1 - \overline{G}_j) \tag{58}$$
$$+ \; (1 - \overline{S}_i) * (1 - \overline{G}_j) \Big] * Data \quad \approx \quad Random * Data \qquad .$$

New values of $\overline{S}_i$ and $\overline{G}_j$ are formed by adding these perturbations and the process is repeated several times until convergence is reached. Again the filter length does not grow as the iterations proceed.

WHERE TO?

There's lots here to work on. First, as discussed in Luenberger (1984), there are a number of variations on Newton's method designed to accelerate convergence such as overrelaxation as well as conjugate-gradient like alternatives like PARTAN. These might substantially reduce both the number of iterations and the computational cost per iteration. Also I need to apply this to field data and estimate what accuracy is really needed to produce acceptable trace-balancing. The various generalizations to surface-consistent deconvolution I've discussed (as well as others I haven't) have yet to be seriously studied.

ACKNOWLEDGMENTS

My thanks go to Jon Claerbout, Gene Golub, John Toldi, Shuki Ronen, and Dan Rothman for their critical interest and suggestions.

REFERENCES

Claerbout, J.F., 1985, Simultaneous pre- and post- NMO deconvolution: This volume.

Keller, H.B., 1978, Global Homotopies and Newton Methods: *in* de Boor, C. and Golub, G.H. eds, Recent Advances in Numerical Analysis, Academic Press.

Levin, S.A., 1984, Surface-consistent deconvolution: SEP-41, p 1-26.

Luenberger, D.G., 1984, Linear and nonlinear programming: Addison-Wesley.

Paige, C.C., and Saunders, M.A., 1982, LSQR: An algorithm for sparse linear equations and sparse least squares: ACM Trans. Math. Soft., v 8, p 43-71.