Stanford Exploration Project, Report 82, May 11, 2001, pages 1–187

Short Note

Spectral preconditioning

Jon Claerbout and Dave Nichols[1]

INTRODUCTION

Industry uses many scaled adjoint operators. We'd like to define the best scaling functions. It is not clear how to define the "best" scaling but one way is to define it as "that which causes CG inversion to go the fastest". The rationale for this choice is that inversion is presumeably the goal, and the scaled adjoint is the first step.

Here I will briefly describe conventional wisdom about scaling and then go on to show how spectral scaling of Kirchoff and other operators can be done using multidimensional prediction-error filters.

CONVENTIONAL SCALING

Iterative methods like conjugate gradients (CG) can sometimes be accelerated by a change of variables. Say we are solving $\mathbf{Bm} \approx \mathbf{d}$. We implicitly define new variables $\mathbf{x}$ by a "trial solution" $\mathbf{m} = \mathbf{Cx}$ (where $\mathbf{C}$ is any matrix of rank equal the dimension of $\mathbf{m}$) getting $(\mathbf{BC})\mathbf{x} \approx \mathbf{d}$. After solving this system for $\mathbf{x}$, we merely substitute into $\mathbf{m} = \mathbf{Cx}$ to get the solution to the original problem. The question is whether this change of variables has saved any effort. On the surface, things seem worse. Instead of iterative applications of $\mathbf{B}$ and $\mathbf{B}'$ we have introduced iterative applications of $\mathbf{BC}$ and $\mathbf{C}'\mathbf{B}$. This is not a problem if the operator $\mathbf{C}$ is quicker than $\mathbf{B}$. Our big hope is that we have chosen $\mathbf{C}$ so that the number of iterations decreases. We have little experience to guide the choice of $\mathbf{C}$ to cause the number of iterations to decrease other than that "columns of $(\mathbf{BC})$ should have equal scales" so $\mathbf{C}$ could be a diagonal matrix scaling columns of $\mathbf{B}$. The preconditioning matrix $\mathbf{C}$ need not even be square.

The use of a preconditioner does not change the final solution unless the operator $\mathbf{B}$ has a null space. In that case, iterating with $\mathbf{B}$ leads to a solution with no component in the null space of $\mathbf{B}$. On the other hand, iterating with $\mathbf{BC}$ leads to a solution with no component in the null space of $\mathbf{BC}$. We will not pause for a proof since no application comes directly to mind.

[1]**email:** jon@sep.stanford.edu, dave

Scaling the adjoint

Given the usual linearized regression between data space and model space, $\mathbf{d} \approx \mathbf{B}\mathbf{m}$, the simplest image of the model space results from application of the adjoint operator $\hat{\mathbf{m}} = \mathbf{B}'\mathbf{d}$. Unless $\mathbf{B}$ has no physical units, however, the physical units of $\hat{\mathbf{m}}$ do not match those of $\mathbf{m}$ so we need a scaling factor. The theoretical solution $\mathbf{m}_{\mathrm{theor}} = (\mathbf{B}'\mathbf{B})^{-1}\mathbf{B}'\mathbf{d}$ suggests the scaling units should be those of $(\mathbf{B}'\mathbf{B})^{-1}$. We could probe the operator $\mathbf{B}$ or its adjoint with white noise or a zero frequency input. Bill Symes suggests we probe with the data $\mathbf{d}$ because it has the spectrum of interest. He proposes we make our image with $\hat{\mathbf{m}} = \mathbf{W}^2\mathbf{B}'\mathbf{d}$ where we choose the weighting function to be:

$$\mathbf{W}^2 \quad = \quad \mathbf{diag}\left(\frac{\mathbf{B}'\mathbf{d}}{\mathbf{B}'\mathbf{B}\mathbf{B}'\mathbf{d}}\right) \tag{1}$$

which obviously has the correct physical units. The weight $\mathbf{W}^2$ can be thought of as a diagonal matrix containing the ratio of two images. A problem with the choice (1) is that the denominator might vanish or might even be negative. The way to stabilize any ratio is to revise it by changing $\mathbf{W}^2 = \mathbf{diag}(a/b)$ to

$$\mathbf{W}^2 \quad = \quad \mathbf{diag}\left(\frac{< ab >}{< b^2+\epsilon^2 >}\right) \tag{2}$$

where ϵ is a parameter to be chosen, and the angle braces indicate the possible need for local smoothing.

Since a scaled adjoint is a guess at the solution to the regression, it is logical to choose values for ϵ and the smoothing parameters that give fastest convergence of the conjugate-gradient regression. To go beyond the scaled adjoint we can use $\mathbf{W}$ as a preconditioner.

To use $\mathbf{W}$ as a preconditioner we define implicitly a new set of variables $\mathbf{x}$ by the substitution $\mathbf{m} = \mathbf{W}\mathbf{x}$. Then $\mathbf{d} \approx \mathbf{B}\mathbf{m} = \mathbf{B}\mathbf{W}\mathbf{x}$. To find $\mathbf{x}$ instead of $\mathbf{m}$ we do CG iteration with the operator $\mathbf{B}\mathbf{W}$ instead of with $\mathbf{B}$. At the end we convert from $\mathbf{x}$ back to $\mathbf{m}$ with $\mathbf{m} = \mathbf{W}\mathbf{x}$.

By (1), $\mathbf{W}$ has physical units inverse to $\mathbf{B}$. Thus the transformation $\mathbf{B}\mathbf{W}$ has no units so the $\mathbf{x}$ variables have physical units of data space. Note that after one iteration $\hat{\mathbf{m}} = \mathbf{W}(\mathbf{B}\mathbf{W})'\mathbf{d} = \mathbf{W}^2\mathbf{B}'\mathbf{d}$ we have Symes scaling. Sometimes it might be more natural to view the solution with data units $\mathbf{x}$ than with proper model units $\mathbf{m}$.

SPECTRAL PRECONDITIONING

In the regression $\mathbf{d} \approx \mathbf{B}\mathbf{m}$ we often think of the model as white, meaning that it is defined up to the Nyquist frequency on the computational mesh, and we think of the data as red, meaning that it is sampled significantly more densely than the Nyquist frequency. Fitting $\mathbf{d} \approx \mathbf{B}\mathbf{m}$, we can probe the operator $\mathbf{B}$ with random numbers $\mathbf{r}_m$ in model space and we can probe the operator $\mathbf{B}'$ with random numbers $\mathbf{r}_d$ in data space getting a red model image and a red synthetic data space

$$\mathbf{m}_r \quad = \quad \mathbf{B}'\mathbf{r}_d \tag{3}$$

$$\mathbf{d}_r \quad = \quad \mathbf{B}\mathbf{r}_m \tag{4}$$

since both $\mathbf{B}$ and $\mathbf{B}'$ turn white into red. Given $\mathbf{m}_r$ we can define a whitening operator $\mathbf{A}_m$ in model space and given $\mathbf{d}_r$ we can define a whitening operator $\mathbf{A}_d$ in data space.

Red-space iteration

First we use $\mathbf{A}_m$ to implicitly define a new model space $\tilde{\mathbf{m}}$ (which I will later justify calling the "data-colored" model) by the substitution $\mathbf{m} = \mathbf{A}_m\tilde{\mathbf{m}}$. Substituting into $\mathbf{d} \approx \mathbf{Bm}$ gives $\mathbf{d} \approx \mathbf{BA}_m\tilde{\mathbf{m}}$. We could solve this by CG iteration of $\mathbf{BA}_m$. The first step is the adjoint. After this step, the data-colored model estimate $\hat{\tilde{\mathbf{m}}}$ and model estimate $\hat{\mathbf{m}}$ are

$$\begin{array}{rclclcl} \hat{\tilde{\mathbf{m}}} & = & \mathbf{A}_m'\mathbf{B}'\mathbf{d} & = & \text{blue} \times \text{red} \times \text{red} & = & \text{red} \\ \hat{\mathbf{m}} & = & \mathbf{A}_m\mathbf{A}_m'\mathbf{B}'\mathbf{d} & = & \text{blue} \times \text{blue} \times \text{red} \times \text{red} & = & \text{white} \end{array} \tag{5}$$

which suggests naming $\hat{\tilde{\mathbf{m}}}$ the *data-colored* model.

White-space iteration

Next we try the data-space whitener $\mathbf{A}_d$ which we obtained by pouring a random-number model into $\mathbf{B}$ and designing a decon filter $\mathbf{A}_d$ on the data out. The regression $\mathbf{0} \approx \mathbf{d} - \mathbf{Bm}$ is red, so it needs a leveling weighting function to whiten it. Using $\mathbf{A}_d$ for the weighting function gives

$$\mathbf{A}_d\mathbf{d} \approx \mathbf{A}_d\mathbf{Bm} \quad = \quad \text{white} \tag{6}$$

so CG iterations are done with the operator $\mathbf{A}_d\mathbf{B}$. The first iteration is the adjoint, namely,

$$\hat{\mathbf{m}} \quad = \quad \mathbf{B}'\mathbf{A}_d'\mathbf{A}_d\mathbf{d} \quad = \quad \text{white} \tag{7}$$

Residual-whitening iteration

Gauss says the noise $\mathbf{d} - \mathbf{Bm}$ should be whitened. The data whitener $\mathbf{A}_d$ might do it, but really we should be using the residual whitener. Thus, after some number of iterations by any method we have a residual and we can find a whitener for it, say $\mathbf{A}_n$. Thus the fitting we should do is

$$\mathbf{A}_n\mathbf{d} \quad \approx \quad \mathbf{A}_n\mathbf{Bm} \tag{8}$$

As previously, pouring random numbers in data space $\mathbf{r}_d$ into the operator $(\mathbf{A}_n\mathbf{B})'$ gives a model space $\mathbf{m}_r = \mathbf{B}'\mathbf{A}_n'\mathbf{r}_d$ from which we can derive a whitener $\mathbf{A}_m$ to implicitly define a new model space $\tilde{\mathbf{m}}$ by $\mathbf{m} = \mathbf{A}_m\tilde{\mathbf{m}}$. Converting the regression from $\mathbf{m}$ to $\tilde{\mathbf{m}}$ gives

$$\mathbf{A}_n\mathbf{d} \quad \approx \quad \mathbf{A}_n\mathbf{BA}_m\tilde{\mathbf{m}} \tag{9}$$

The first iteration is now

$$\begin{array}{rcl} \hat{\tilde{\mathbf{m}}} & = & \mathbf{A}_m'\mathbf{B}'\mathbf{A}_n'\mathbf{A}_n\mathbf{d} \\ \hat{\mathbf{m}} & = & \mathbf{A}_m\mathbf{A}_m'\mathbf{B}'\mathbf{A}_n'\mathbf{A}_n\mathbf{d} \end{array} \tag{10}$$

This proliferation of interesting methodologies is why we need C++!

APPLICATION: DECONVOLVING VELOCITY SPECTRA

Where might spectral scaling be useful? First I should describe a similar example with a surprising outcome. With Kirchoff operators in PVI and BEI I proposed integrating the velocity transform with the rho filter $\sqrt{-i\omega}$. Dave Nichols tested this idea and found that the rho filter accelerates the early steps, but retards all the others! This suggests that deconvolution might suffer the same fate. On the other hand, the slowed convergence with $\sqrt{-i\omega}$ might simply be a consequence of Fourier methods spreading signals long distances with wraparounds that would not be part of a decon approach. In any event, improving the first iteration is a worthwhile goal.

The practical goal is not just to handle the simple rho-filter effect, but to cope with truncations and irregular data sampling. The model space is a regular mesh (although the data might be on an irregular one) so deconvolution there always makes sense. To manufacture a striking example, I would consider velocity transformation with a short, wide-offset marine cable, say extending from 1.5 to 2.0 km. The velocity space would be strongly "dipped" so a simple 5×2 filter should be able to flatten its dip spectrum.