Stanford Exploration Project, Report 82, May 11, 2001, pages 1–183

Short Note

A simple example of a null space and how to modify it

Dave Nichols[1]

WHAT IS A NULL SPACE?

In many situations in geophysics we seek a solution to a problem of the form

$$\mathbf{Bm} = \mathbf{d},$$

where $\mathbf{B}$ is a linear operator that relates a model, $\mathbf{m}$, to the data, $\mathbf{d}$. The null space of $\mathbf{B}$ is the space, $\mathbf{N}$ for which any vector $\mathbf{n} \in \mathbf{N}$ satisfies

$$\mathbf{Bn} = \mathbf{0}.$$

If the problem is under-determined it will have a non-empty null space. When a vector $\hat{\mathbf{m}}$ is a valid solution to our problem we can add an arbitrary amount of the null space to the vector and it will be a equally valid solution.

$$\mathbf{B}(\hat{\mathbf{m}} + \alpha \mathbf{n}) = \mathbf{d} \qquad ; \ \mathbf{n} \in \mathbf{N}$$

The problem that we want to answer is "how much of the null space should be added to the solution".

The SVD explanation

The properties of the null space can be explored by examining the singular value decomposition of the operator. The SVD of $\mathbf{B}$ is given by

$$\mathbf{B} = \mathbf{U}\Lambda\mathbf{V}^{\dagger}.$$

The matrix Λ is a diagonal matrix whose entries are the singular values of $\mathbf{B}$. The columns of the matrices $\mathbf{U}$ and $\mathbf{V}$ form orthonormal bases for the data and model spaces respectively. The null space of $\mathbf{B}$ corresponds to the space spanned by the vectors associated with singular values of zero. An important property of the adjoint operator $\mathbf{B}^{\dagger}$ is that, when applied to any

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

vector, it will always produce an output that has no component in the null space. This is easily shown by considering the SVD of $\mathbf{B}$. The data vector $\mathbf{d}$ can be written as a linear combination of the basis vectors in the data space.

$$\mathbf{d} = \sum \alpha_i \mathbf{u}_i$$

When the adjoint operator is applied to this vector we obtain

$$\mathbf{g} = \mathbf{B}^\dagger \mathbf{d} = \sum \alpha_i \lambda_i \mathbf{v}_i,$$

i.e. the basis vectors in the model space are scaled by their singular value. Any basis vector associated with a singular value of zero will have no component in the resulting vector.

An example problem

Consider the trivial under-determined problem

$$(1 \quad 1)\begin{pmatrix} m_1 \\ m_2 \end{pmatrix} = (1).$$

This operator has two model space vectors one of which is in the null space. The first vector with singular value of one is,

$$\begin{pmatrix} \sqrt{2} \\ \sqrt{2} \end{pmatrix}.$$

The second vector with a singular value of zero, and thus in the null space is

$$\begin{pmatrix} -\sqrt{2} \\ \sqrt{2} \end{pmatrix}.$$

One possible solution to the problem is made purely of the first vector,

$$(1 \quad 1)\begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix} = (1)$$

An arbitrary amount of the second vector can be added to the solution

$$(1 \quad 1)\left[\begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix} + \alpha \begin{pmatrix} -1 \\ 1 \end{pmatrix}\right] = (1)$$

MINIMUM LENGTH SOLUTIONS AND ITERATIVE SOLVERS

The minimum length solution is the the solution with no null space component. These solutions are prefered by some people because they contain no information about the solution that is not a linear function of the data. Any solution created by an iterative solver can be expressed in terms of a series expansion:

$$\mathbf{m} = \mathbf{m_0} + \alpha_0 \mathbf{B}^\dagger \mathbf{r_0} + \alpha_1 \mathbf{B}^\dagger \mathbf{B} \mathbf{B}^\dagger \mathbf{r_0} + \cdots + \alpha_n (\mathbf{B}^\dagger \mathbf{B})^n \mathbf{B}^\dagger \mathbf{r_0} + \cdots,$$

Where $\mathbf{r_0}$ is the initial residual calculated from the initial solution $\mathbf{m_0}$.

$$\mathbf{r_0} = \mathbf{d} - \mathbf{B}\mathbf{m_0}$$

The coefficients are calculated in a manner that depends on the algorithm. Since the last operation in each component of this solution is always the adjoint operator, we can immediately see that this type of solver will never add any information to the null space. If the initial solution is zero, then the final solution will have no component in the null space and it will be the minimum length solution.

Iterative solution of the model problem

The model problem is solved in one iteration by almost any algorithm since the operator is of rank one. I will illustrate the properties of the solution using a steepest descent algorithm. The first step applies the adjoint operator to the residual to get a gradient vector. Assuming an initial solution of zero, the residual is the same as the data.

$$\mathbf{r_0} = \mathbf{d} - \mathbf{B}\mathbf{m_0} = (\,1\,) - (\,1 \quad 1\,)\begin{pmatrix} 0 \\ 0 \end{pmatrix} = (\,1\,).$$

Then the gradient is given by,

$$\mathbf{g} = \mathbf{B}^{\dagger}\mathbf{r} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}(\,1\,) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

The step length is chosen to minimize the residual

$$\min_{\alpha} |\alpha \mathbf{B}\mathbf{g} - \mathbf{d}| = \min_{\alpha} |\alpha \times 2 - 1|.$$

Thus a step length of $1/2$ gives the solution

$$\mathbf{m_1} = \mathbf{m_0} + \alpha\mathbf{g} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix}.$$

The answer may be different if the initial solution is not zero. If the initial solution has energy in the null space, then that energy will remain in the final solution. An initial solution chosen arbitrarily is,

$$\mathbf{m_0} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \sqrt{2}\begin{pmatrix} \sqrt{2} \\ \sqrt{2} \end{pmatrix} - \sqrt{2}\begin{pmatrix} -\sqrt{2} \\ \sqrt{2} \end{pmatrix}$$

This initial solution has half its energy in the null space. Since this solution has a residual of zero, it is the final solution as well.

LEFT PRECONDITIONERS

Many methods for solving this problem introduce a premultiplying operator to the problem

$$\mathbf{M_L}\mathbf{B}\mathbf{m} = \mathbf{M_L}\mathbf{d}$$

This operator has different interpretations:

1. It can be thought of as a preconditioner dependent on the operator.

2. Tarantola defines a “data covariance” operator that is used in this way. It expresses the reliability and the correlation of the different data measurements.

3. In iteratively reweighted least squares (for the L_p norm), it is a data variance measure that is estimated from the current residual.

4. Claerbout calls it a “change of regression”.

However, whatever the derivation, it does not change the fact that an iterative solver will put no new energy in the null space of the model. The adjoint operation is $\mathbf{B}^{\dagger}\mathbf{M_L}^{\dagger}$. The last operator applied is $\mathbf{B}^{\dagger}$ so there is no null space energy in the result.

RIGHT PRECONDITIONERS

In contrast a right preconditioner can change the energy in the null space of the solution. Now the problem to be solved is,

$$\mathbf{B}\mathbf{M_R}\hat{\mathbf{m}} = \mathbf{d}$$

followed by

$$\mathbf{m} = \mathbf{M_R}\hat{\mathbf{m}}.$$

Again there are many reasons given for using this type of operator

1. A right preconditioner dependent on the operator.

2. A “quelling” operator to stabilize the solution.

3. Claerbout calls it “a change of variable in the model”. E.G. “FROB”.

When the adjoint of this operator is applied it may produce energy in the null space of $\mathbf{B}$. The adjoint operation is $\mathbf{M_R}^{\dagger}\mathbf{B}^{\dagger}$. When this operator is used in an iterative solver the solution will have no energy in the null space of the composite operator ($\mathbf{B}\mathbf{M_R}$). The length that is minimized is the norm of the solution in the transformed space, not the norm in the original space.

$$\min_{\hat{\mathbf{m}}} |\hat{\mathbf{m}}| = \min_{\mathbf{m}} |\mathbf{M_R}^{+}\mathbf{m}|$$

$\mathbf{M_R}^{+}$ is a pseudo inverse of $\mathbf{M_R}$. Since $\mathbf{M_R}$ should always have rank of the length of $\mathbf{m}$, a well behaved pseudo-inverse will always exist.

Right preconditioning of the example problem

The example problem can be solved with a diagonal right preconditioner. For no particular reason I have decided that I prefer to have more energy in the first unknown so I choose the preconditioner

$$\mathbf{M_R} = \begin{pmatrix} 10 & 0 \\ 0 & 1 \end{pmatrix}.$$

Starting with the same initial solution as before, zero, I find the gradient,

$$\mathbf{g} = \begin{pmatrix} 10 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} (\,1\,) = \begin{pmatrix} 10 \\ 1 \end{pmatrix}.$$

The step length is chosen to minimize the residual

$$\min_{\alpha} |\alpha \times 101 - 1|.$$

A step length of $1/101$ gives the solution

$$\hat{\mathbf{m}}_{\mathbf{1}} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \frac{1}{101} \begin{pmatrix} 10 \\ 1 \end{pmatrix} = \begin{pmatrix} 10/101 \\ 1/101 \end{pmatrix}.$$

This solution is projected back to the original model space to give

$$\mathbf{m_1} = \begin{pmatrix} 10 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 10/101 \\ 1/101 \end{pmatrix} = \begin{pmatrix} 100/101 \\ 1/101 \end{pmatrix}.$$

The solution has lots of energy in the null space of $\mathbf{B}$ but no energy in the null space of $\mathbf{BM_R}$

MODEL PENALTY FUNCTIONS

An alternative strategy for dealing with the null space is to augment the problem with a model penalty operator, $\mathbf{P}$. We now seek the solution to the problem

$$\min_{\mathbf{m}} |\mathbf{Pm}|$$

and

$$\min_{\mathbf{m}} |\mathbf{Bm} - \mathbf{d}|$$

The penalty operator can be used to turn an underdetermined problem into an overdetermined problem.

$$\begin{pmatrix} \mathbf{B} \\ \epsilon \mathbf{P} \end{pmatrix} \mathbf{m} \approx \begin{pmatrix} \mathbf{d} \\ \mathbf{0} \end{pmatrix}$$

Tarantola views these penalty as “inverse model covariance operators”. I view them more generally; they describe something that I don’t like about the model (e.g. roughness, smoothness, curvature, energy in one dip) that I want to minimize. Claerbout would like to estimate PEF’s to use as penalty operators at the same time as he estimates the solution to the problem.

The sample problem with a model penalty operator

I can choose a simple penalty function that will minimize the energy in the second unknown

$$\mathbf{P} = \begin{pmatrix} 1 & 0 \\ 0 & 5 \end{pmatrix}.$$

Then the augmented problem, with $\epsilon = .1$, is

$$\begin{pmatrix} (1 \quad 1) \\ \begin{pmatrix} .1 & 0 \\ 0 & .5 \end{pmatrix} \end{pmatrix} \begin{pmatrix} m_1 \\ m_2 \end{pmatrix} = \begin{pmatrix} (1) \\ \begin{pmatrix} 0 \\ 0 \end{pmatrix} \end{pmatrix}.$$

This can be rewritten as

$$\begin{pmatrix} 1 & 1 \\ .1 & 0 \\ 0 & .5 \end{pmatrix} \begin{pmatrix} m_1 \\ m_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}.$$

The solution to this problem is

$$\mathbf{m} = \begin{pmatrix} .9524 \\ .0381 \end{pmatrix}.$$

By a suitable choice of penalty function we have reduced the energy in the second model parameter. However, the scaling of my penalty function was somewhat arbitrary. I would have obtained a different result if I had used scaled my operator P differently.

DISCUSSION

What is the best thing to do? I'm not sure. A good initial guess is obviously a good thing to have, but you had better be sure that you like the null-space component of your initial guess, as it isn't going to change. The preconditioning approach has the attractive feature that you don't need to choose a scale factor. It has the limitation that the preconditioner must be full rank; you must not remove possible energy from the model. The penalty function approach is much more general. It allows you to use any operator as a penalty function, but you have the added burden of choosing a good scale factor for that function.