Anti-crosstalk

Jon Claerbout

ABSTRACT

Inverse theory can never be wrong because it's just theory. Where problems arise and opportunities are overlooked is where practitioners grab and use inverse theory without recognizing that alternate assumptions might be better. Here I formulate *anti-crosstalk* operators to supplement (or replace) familiar regularizations in image estimation. Applications appear abundant.

Theoretical example

First let us consider an extremely simple theoretical example with an issue of "cross talk". Notice that word itself is absent from linear inverse theory. (Anyway, it's absent from the index of Tarantola's book.) Normally we think of "high frequencies" as being orthogonal to "low frequencies". In this example that will be true from a theoretical viewpoint, but no practical person would consider it to be true.

Consider a signal that is an impulse which is to be split into its low and high frequencies. We might write this as $0 = l + h - d$, namely, zero equals low plus high minus data. Let us extract low frequencies from the data with a step function in the Fourier domain or convolving with a sinc function in the time domain. Clearly the high frequency component in the Fourier domain is the constant function minus the step. In the time domain it is a delta function minus a sinc (since high plus low is a delta function). Theoretically everything is fine. Usually a vanishing cross product means a sum of terms vanishes. Here every term vanishes in the Fourier domain when we multiply the step times one minus the step. That is powerful orthogonality! In the time domain the convolution of the two, $sinc * (1 - sinc) = sinc - sinc * sinc = sinc - sinc = 0$ vanishes not just at zero lag, but at all lags. That is powerfully strong orthogonality too. Theoretically, high and low frequency components of the data are orthogonal at every frequency and at every lag. What would the experimentalist have to say? The experimentalist would look in the time domain at the low frequency function and at the high frequency function and say, "Everywhere I look at these two functions they are the same. They can't be orthogonal. They have a massive amount of *crosstalk*." Of course the two functions are not exactly the same. They have opposite polarities, and they are not the same at the origin point. But everywhere else they look the same. We don't like it. These signals should not be coherent but they are.

I first encountered this crosstalk issue in a serious geophysical application of a very simple nature. The lack of a good place for crosstalk in the theoretical framework was blatantly obvious.

Lake example

A depth sounding survey was made of a lake. A boat with a depth sounder sailed a gridwork of passes on the lake. Upon analysis the final image contained obvious evidence of the survey grid. Oops! We should always hide our data acquisition footprint. The footprint is not the geography or geology we wish to show. How did this happen? We guessed the water level changed during the survey. Perhaps it rained or perhaps the water was used for agriculture. Perhaps the wind caused the lake to "pendulum" (seiche). Perhaps the operator sat in the front of the boat, or sometimes its back, or ran it at various speeds giving the depth measurement a different bias. Our data measured the difference between the top and the bottom of the lake; yet we had no idea how to model the top. Eventually we modeled the top as an "arbitrary low frequency function" in data space (a one dimensional function following the boat). This got rid of the tracks in the model space (map space) but led to much embarrassment. It was embarrassing to discover that the geography (as seen in data space) was correlated with the lake surface (rain and drain).

Let us express these ideas mathematically: $\mathbf{d}$ is the data, depth along the survey coordinate $d(s)$. Here s is a parameter like time. It increases steadily whether the boat is sailing north-south or east-west or turning inbetween. The model space is the depth $h(x, y)$. There will be a regularization on the depth, perhaps $0 \approx \nabla h(x, y)$.

For the top of the lake with the ship we need some slowly variable function of location s. It's embarrassing for us to need to specify it because we have no good model for it. So, we specify a slowly variable function $\mathbf{u} = u(s)$ by asking a random noise function $\mathbf{n} = n(s)$ to run thru a low frequency filter, say $\mathbf{L}$. We are not comfortable also about needing to choose $\mathbf{L}$. We call the function $\mathbf{u} = \mathbf{Ln}$ the rain and drain function. We take the regularization for the unknown $\mathbf{n}$ to be $\mathbf{0} \approx \mathbf{n}$. (Least squares will tend to drive components of $\mathbf{n}$ to similar values, and under some conditions likewise the spectrum of $\mathbf{n}$ will tend to white, so we expect (and often find) the spectrum of $\mathbf{u}$ comes out that of $\mathbf{L}$.)

The operator we do understand very clearly is the geography operator $\mathbf{G}$. Given we wish to make a theoretical data point (water depth), the geography operator $\mathbf{G}$ tells us where to go on the map to get it. Of course each of the two regularizations $0 \approx \nabla h(x, y)$ and $\mathbf{0} \approx \mathbf{n}$ has its own epsilon which is annoying because we need to specify those too. With all these definitions our unknowns are the geography $\mathbf{h}$ and the noise $\mathbf{n}$ that builds us a drift function. Our data fitting goal says the data should be the separation of the top and bottom of the lake.

$$\mathbf{0} \quad \approx \quad \mathbf{Gh} + \mathbf{Ln} - \mathbf{d} \tag{1}$$

In my free on-line textbook GEE all this seemed rather conventional and rather fine. Our embarrassment came when we compared the geographically modeled part of the data $\mathbf{Gh}$ to the drift (rain and drain) modeled part of the data $\mathbf{Ln}$. They were visibly correlated. This is crazy! The boat being in deep water should not correlate with rain (or drain). We needed to add an ingredient to the formulation saying $\mathbf{u} = \mathbf{Ln}$ should be orthogonal to $\mathbf{Gh}$ (which is practically the same as $\mathbf{d}$) in some generalized sense. Let us see how this might be done.

A regression to minimize crosstalk

Observing the geographically modeled data $\mathbf{Gh}$ correlating with the data drift $\mathbf{u} = \mathbf{Ln}$ we wish to articulate a regression that says they should not correlate. Since the drift $\mathbf{u}$ is a small correction to the data $\mathbf{d}$, in other words $\mathbf{Gh} \approx \mathbf{d}$, we can simplify the goal by asking that the dot product of $\mathbf{d}$ with $\mathbf{u}$ should vanish, vanish not necessarily over the entire data set; but that it should vanish under many triangular weighed windows.

Let us define $\mathbf{D}$ as a diagonal matrix with $\mathbf{d}$ on the diagonal. This may be a little unfamiliar. Often we see positive weighting functions on the diagonal. Here we see data (possibly with both polarities) on the diagonal. Additionally, let us define a matrix $\mathbf{T}$ of convolution with a triangle. Columns of $\mathbf{T}$ contain shifted triangle functions, likewise do rows. Take $\mathbf{t}'$ to be any row of $\mathbf{T}$. Then $\mathbf{t}'\mathbf{D}$ is a row vector of triangle weighted data. We want the regression $\mathbf{0} \approx \mathbf{t}'\mathbf{Du}$ for all shifts of the triangle function. The way to express this is:

$$\mathbf{0} \approx (\mathbf{TD})\mathbf{u} \tag{2}$$
$$\mathbf{0} \approx (\mathbf{TD})\mathbf{Ln} \tag{3}$$

Hooray! Now we know what coding to do! But first, to better understand the regression (2) imagine instead that $\mathbf{T}$ is a square matrix of all ones, say $\mathbf{1}$. That would be like super wide triangular windows. Then every component of the vector $\mathbf{1Du}$ contains the same dot product $\mathbf{d} \cdot \mathbf{u}$. Using $\mathbf{T}$ instead of $\mathbf{1}$ gives us those dot products under a triangle weight, each final vector component having a shifted triangle.

What is a good name for $\mathbf{TD}$? It measures the similarity of $\mathbf{d}$ and $\mathbf{u}$. It might be called the "data similarity" operator. What is a good name for its adjoint $\mathbf{DT}$? Assuming whatever comes out of $\mathbf{T}$ is a smooth positive function, then $\mathbf{DT}$ is a data gaining operator (its input being a gain function). Do we have any geophysical problems where the unknown is the gain?

Crosstalk in a more general context

We seemed to escape nonlinearity in the lake depth sounding example above, but that was a lucky accident. Since the data there was mostly explained by geography with a small perturbation by rain and drain, the crosstalk while fundamentally nonlinear was practically linear. More generally anti-crosstalk strategies seem little (if ever!) developed because they lead us directly into nonlinear regression. Let us work through the general case, the nonlinear theory.

Consider data $\mathbf{d}$ a shot gather or CMP gather. We might choose to model it as reflections (hyperbolas) $\mathbf{d}_1$ plus linear events $\mathbf{d}_2$ (noises or head waves). We might thus set up the regression

$$\mathbf{0} \approx \mathbf{d}_1 + \mathbf{d}_2 - \mathbf{d} \tag{4}$$

$$\mathbf{0} \approx \mathbf{F}_1\mathbf{m}_1 + \mathbf{F}_2\mathbf{m}_2 - \mathbf{d} \tag{5}$$

Of course we need some damping regularization on $\mathbf{m}_1$ and $\mathbf{m}_2$ which for simplicity of exposition I will take to be $\mathbf{0} \approx \mathbf{m}_1$ and $\mathbf{0} \approx \mathbf{m}_2$. Is that all there is to this problem? Not necessarily. We'll be annoyed if we discover a lot of cross talk between $\mathbf{d}_1$ and $\mathbf{d}_2$. Two different physical mechanisms are supposed to have created our data. We'll be annoyed to discover they both make the same contribution or that they make opposite contributions. The regularization should reduce (or prevent) the contributions from coming out opposite. If they are opposites, it could represent our lack of analytic skills in formulating the regularization, or it could represent our need for the anti-crosstalk methodology being proposed here.

We'd like that the modeled data parts $\mathbf{d}_1$ and $\mathbf{d}_2$ do not "look like" each other. It's not enough that the dot product $\mathbf{d}_1 \cdot \mathbf{d}_2$ vanish. That dot product should be small under all shifted (say triangular) weighting windows. Since $\mathbf{d}_1$ is a linear function of the model $\mathbf{m}_1$ and likewise for $\mathbf{d}_2$, the orthogonality we seek involves the product of $\mathbf{m}_1$ with $\mathbf{m}_2$ so our goals are a non-linear function of our unknowns. Never fear. We have done non-linear problems before. They don't turn out badly when we are able to define a good starting location (which we do by solving the linearized non-linear problem first).

The full non-linear derivation

For warm up we linearize in the simplest possible way. Suppose we allow only $\mathbf{m}_1$ to vary keeping $\mathbf{m}_2$ fixed. We put $\mathbf{d}_2$ on the diagonal of a matrix, say $\mathbf{D}_2$. The regression for anti-crosstalk is now

$$\mathbf{0} \approx \mathbf{T}\mathbf{D}_2\mathbf{d}_1 \tag{6}$$

$$\mathbf{0} \approx \mathbf{T}\mathbf{D}_2\mathbf{F}_1\mathbf{m}_1 \tag{7}$$

Define the element-by-element cross product of $\mathbf{d}_2$ times $\mathbf{d}_1$ to be $\mathbf{d}_1 \times \mathbf{d}_2$. Now let us linearize the full non-linear anti-crosstalk regularization. Let a single element of $\mathbf{d}_1 \times \mathbf{d}_2$ be decomposed as a base plus a perturbation $d = \bar{d} + \tilde{d}$. A single component of the vector $\mathbf{d}_1 \times \mathbf{d}_2$ is $(\bar{d}_1 + \tilde{d}_1)(\bar{d}_2 + \tilde{d}_2)$. Linearizing the product (neglecting the product of the perturbations) gives

$$\bar{d}_2\tilde{d}_1 + \bar{d}_1\tilde{d}_2 + \bar{d}_2\bar{d}_1 \tag{8}$$

This is one component. We seek an expression for all. It will be a vector which is a product of a matrix with a vector. We want no unknowns in matrices; we want them all in vectors so we will know how to solve for them.

$$\bar{\mathbf{D}}_2\tilde{\mathbf{d}}_1 + \bar{\mathbf{D}}_1\tilde{\mathbf{d}}_2 + \bar{\mathbf{D}}_1\bar{\mathbf{d}}_2 \tag{9}$$

Express the perturbation parts of the vectors as functions of the model space

$$\bar{\mathbf{D}}_2\mathbf{F}_1\tilde{\mathbf{m}}_1 + \bar{\mathbf{D}}_1\mathbf{F}_2\tilde{\mathbf{m}}_2 + \bar{\mathbf{D}}_1\bar{\mathbf{d}}_2 \tag{10}$$

This vector should be viewed under many windows (triangle shaped, for example). Under each window we hope to see the product have a small value. The desired anti-crosstalk regression is to minimize the length of the vector below by variation of the model parameters $\tilde{\mathbf{m}}_1$ and $\tilde{\mathbf{m}}_2$.

$$\mathbf{0} \quad \approx \quad \mathbf{T}(\bar{\mathbf{D}}_2\mathbf{F}_1\tilde{\mathbf{m}}_1 + \bar{\mathbf{D}}_1\mathbf{F}_2\tilde{\mathbf{m}}_2 + \bar{\mathbf{D}}_1\bar{\mathbf{d}}_2) \tag{11}$$

This regression augments our usual regularizations. Perhaps it partially or significantly supplants them. Unfortunately, it requires yet another epsilon.

Upon finding $\tilde{\mathbf{m}}_1$ and $\tilde{\mathbf{m}}_2$ we update the base model $\bar{\mathbf{m}} \leftarrow \bar{\mathbf{m}} + \tilde{\mathbf{m}}$ and iterate.

Outlook

Many examples suggest themselves.

1. We might model reflection data as a superposition of primaries and multiples. We might model it as a superposition of pressure waves and shear waves.

2. In tomography we might model event flatness as a superposition of shallow and deep slownesses. The shallow and deep slownesses have wholly different causes separated by millions of years. They should not show crosstalk.

3. The problem of segregating signal and noise offers many examples. We'd like to see signal containing no evident noise and vice versa.

4. In time-lapse seismology we would like to see the the image change unpolluted by the original image. Unfortunately, the methodology proposed here does not allow for time-shifted crosstalk.

Getting started

As the concepts here are quite new to us, the first thing we should do is cook up some super simple synthetic examples. With working synthetic codes in hand we should see if we can go ahead and repair the Galilee survey. Hopefully we'll recognize we have built some reusable software to facilitate other projects.

Getting started will not be easy. Most commonly we have a simple synthetic example under control and struggle to find an appropriate real data set. Here we have a suitable beginners' data set (Galilee) but we need to find a synthetic data set to provide examples that give clarity to the whole process. Just one issue is dealing with the relative scaling of the three regularizations. We'd like meaningful examples where only one or two of the regularizations are actually required.

There are many paths to explore with anti-crosstalk technology. Besides the many potential applications one can hope that the anti-crosstalk regularization eliminates (or reduces) the need for the usual regularizations. That would be nice if true. The need to specify a suitable regularization is often what makes it difficult to automate data analysis based on inversion.

I'm worried about the job as I defined it for the first-year SEP students. Given the linearization I suggested to them, did they know how to measure success?