Time variable prediction without mathematics

Jon Claerbout

ABSTRACT
With almost no math, a quick trick leads to $\ell_1$ norm nonstationary decon.

SUMMARY AND CONCLUSION
Start with data (a signal of thousands of values). Start with any filter $f$ of maybe ten lags. The filter will change as we move it along the data. At time $t$ set the filter down on the data. The ten data values under the filter are designated $d$. Take $\epsilon$ to be a tiny scalar (for example $\epsilon = 1/(200 \|d\|)$).

The filter $f$ has output ($f \cdot d$) that we may choose to be a prediction of anything, for example, a prediction of the next data value to slide under the range of $d$. The augmented filter $f + \epsilon d$ offers us the two predictions ($f + \epsilon d \cdot d = (f \cdot d) + \epsilon (d \cdot d)$). Comparing these predictions to the actual incoming data value reveals which sign for $\epsilon$ better improves the prediction. Update the filter $f$. Move to time $t + \Delta t$. Update $d$. Repeat indefinitely. The filter adapts to best make your prediction.

THEORY AND POTENTIAL APPLICATIONS
The above idea can be based on conventional mathematics. The gradient of $\ell_2$ normed prediction error turns out to be $d \times \text{PredictionError}(d)$. The above algorithm marches with $d \times \text{Signum}(\text{PredictionError}(d))$ which smells like $\ell_1$ norm decon. Wow!

Taking many small steps down a gradient has two advantages over analytic solutions: (1) it allows nonstationarity, and (2) it streams data (saving memory).

As with deconvolution on a helix, this method extends naturally to higher dimensional spaces (such as $(t,x)$-space).

I’m always trying to convince students to use PEFs to make their residuals IID.

I’ve explained how this method extends to multichannel data (vector-valued data), but it’s not yet been tried. http://sep.stanford.edu/sep/jon/VectorDecon.pdf