Streaming TV-IID with missing data

Jon Claerbout

ABSTRACT

A PEF can be updated as new data arrives. This notion can be merged with a multidimensional helix to produce a nonstationary multidimensional PEF.

INTRODUCTION

Here I sketch an alternative approach to TV-IID decon. It builds the TV-PEF while data streams through. Multidimensional data is handled via a helix. Random numbers can be divided by this PEF to make synthetic data. Restoration of missing data suggests (but does not require) streaming both forward and backward and merging the results.

METHOD

The averaging region is not the familiar triangle. The averaging window is something more like a causal (or anticausal) damped exponential. (Most likely it can be compounded with its time reverse thereby yielding a more time-symmetical averaging of past and future data.) Here is how it goes:

Suppose we have a PEF that represents all previous moments in time. Call it $\bar{\mathbf{a}} = (1, \bar{a}_1, \bar{a}_2, \bar{a}_3, \cdots)$. Say that $\bar{\mathbf{a}}$ represents data values $(d_1, d_2, d_3, \cdots, d_{98})$. We seek to define the $\mathbf{a}$ that represents that data with an appended data value, d_{99}.

Consider the regression

$$\begin{bmatrix} d_{99} & d_{98} & d_{97} & d_{96} \\ \hline \epsilon & \cdot & \cdot & \cdot \\ \cdot & \epsilon & \cdot & \cdot \\ \cdot & \cdot & \epsilon & \cdot \\ \cdot & \cdot & \cdot & \epsilon \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} \quad \approx \quad \epsilon \begin{bmatrix} 0 \\ \hline 1 \\ \bar{a}_1 \\ \bar{a}_2 \\ \bar{a}_3 \end{bmatrix} \tag{1}$$

The numerical value of ϵ would be set large (not small!) because the filter should change slowly with each added data value. As usual, toss out all those regression equations involving missing data.

I believe the regression (1) will have a simple analytical solution in terms of a couple dot products. If not, I'm pretty sure steepest decent will give me an excellent approximation in the usual case where $\mathbf{a} \approx \bar{\mathbf{a}}$. It looks like conjugate-gradient iteration is not required(!?). The code should be far simpler than our summer's code.

This is likely the most rudimentary form of a Kalman filter. Stew is digging up some classic papers.

Missing data estimation

Wherever the PEF first encounters a missing data value you define it to be the predicted value. This can be done recursively scanning forward in time or scanning backward, or both. (Technicality: When you extrapolate into a big hole, there likely is a discontinuity when reaching real data again on the other side. Going both forward and backward enables tapering that assures continuity.)

3-D

The helix would be used in 2-D and 3-D. Notice that each physical dimension may require a different ϵ. I have hardly given this any thought.

Increased stability

We can build in a stationarity assumption. In big holes and off ends, we might see instability. This is reduced by asking the filter to work backward in time as well as forward. Thus the regression:

$$\begin{bmatrix} d_{99} & d_{98} & d_{97} & d_{96} \\ d_{96} & d_{97} & d_{98} & d_{99} \\ \hline \epsilon & \cdot & \cdot & \cdot \\ \cdot & \epsilon & \cdot & \cdot \\ \cdot & \cdot & \epsilon & \cdot \\ \cdot & \cdot & \cdot & \epsilon \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} \quad \approx \quad \epsilon \begin{bmatrix} 0 \\ 0 \\ \hline 1 \\ \bar{a}_1 \\ \bar{a}_2 \\ \bar{a}_3 \end{bmatrix} \tag{2}$$

Most likely we'd want a stationarity assumption on space, but not always on time. (Echos weaken in time.)

Is this going anywhere?

Your comments are invited.

Would someone please solve regression (1) for me?