# Streaming nonstationary prediction error (II)

*Sergey Fomel, Jon Claerbout, Stewart A. Levin, and Rahul Sarkar*

## ABSTRACT

A PEF can be updated as each new data value arrives. The update costs only one dot product of vectors the size of the PEF. Typical PEF applications do not require storing the PEF; it can be used on the fly. This streaming notion could be merged with a multidimensional helix to produce a nonstationary multidimensional PEF.

## INTRODUCTION

Experience shows we have a great need to deal with time and space variable spectra. Now we are progressing rapidly. In summer 2015 Ruan et al. (2015) developed time variable deconvolution code by storing the prediction-error filters (PEF)s on a coarse mesh. At summer's end Jon proposed an alternate method using streaming. His PEFs would not be stored, but would be used on the fly. They would be used for achieving IID residuals in data fitting, accelerating iterative data fitting, and estimating missing data.

But, Jon was unable to perform a key algebraic step. So, he circulated his idea among friends. He abandoned that approach and went to another streaming approach. It takes a fixed block of input and performs one step of steepest descent before moving the block forward to the next time point. Results are shown earlier in this report in an article by Claerbout (2016).

Meanwhile Sergey solved the algebraic problem that Jon was unable to solve. Sergey's solution is exact. After verifying Sergey's solution (with help from Stew), Jon studied it with the goal in mind of coherently explaining for practical purposes the numerical the choice of an epsilon (now variously gamma and lambda). This study led him to a simpler and cruder approach: Simply find a gradient and move a fixed distance down it. It's not surprising that Jon's approach is simpler. What is surprising is that it turns out to be identical to Sergey's analytical solution! How can that be? The problem Sergey solved exactly includes an unspecified constant gamma. This unspecified constant amounts to Jon's fixed distance.

Crude guesses at this parameter work surprisingly well, very well indeed. Left for the future is **deep learning** a deeper philosophical understanding of this parameter enabling automatic optimal choice for where spectra change rapidly or change slowly.

# JON DEFINES THE PROBLEM

Here I sketch an alternative to the Clapp summer group named TV-IID decon. It builds the time variable PEF (TV-PEF) while data streams through. Multidimensional data may be handled via a helix. Random numbers can be divided by this PEF to model 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. Here's 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:

$$\left[\begin{array}{cccc} d_{99} & d_{98} & d_{97} & d_{96} \\ \hline \gamma & \cdot & \cdot & \cdot \\ \cdot & \gamma & \cdot & \cdot \\ \cdot & \cdot & \gamma & \cdot \\ \cdot & \cdot & \cdot & \gamma \end{array}\right] \left[\begin{array}{c} 1 \\ a_1 \\ a_2 \\ a_3 \end{array}\right] \quad \approx \quad \gamma \left[\begin{array}{c} 0 \\ \hline 1 \\ \bar{a}_1 \\ \bar{a}_2 \\ \bar{a}_3 \end{array}\right] \tag{1}$$

[Jon wrote] I believe the regression (1) will have a simple analytical solution in terms of a couple dot products. It looks like conjugate-gradient iteration is not required(!?). I say this because I seem to remember (from Gene Golub?) that it is easy to append another regression to a family. But I cannot find such an analytic solution! Would someone please solve regression (1) for me?

# SERGEY BEGINS

[Sergey wrote] We can rewrite Jon's equation (1) as

$$\left[\begin{array}{ccc} d_n & d_{n-1} & d_{n-2} \\ \gamma & 0 & 0 \\ 0 & \gamma & 0 \\ 0 & 0 & \gamma \end{array}\right] \left[\begin{array}{c} a_1 \\ a_2 \\ a_3 \end{array}\right] \quad \approx \quad \left[\begin{array}{c} -d_{n+1} \\ \gamma\,\bar{a}_1 \\ \gamma\,\bar{a}_2 \\ \gamma\,\bar{a}_3 \end{array}\right] \tag{2}$$

or, in a shortened block-matrix notation, we have the residual to minimize

$$\mathbf{0} \quad \approx \quad \mathbf{r} \quad = \quad \left[\begin{array}{c} \mathbf{d}^T \\ \gamma\,\mathbf{I} \end{array}\right] \mathbf{a} \; - \; \left[\begin{array}{c} -d_{n+1} \\ \gamma\,\bar{\mathbf{a}} \end{array}\right] \;, \tag{3}$$

where

$$\mathbf{d} = \begin{bmatrix} d_n \\ d_{n-1} \\ d_{n-2} \end{bmatrix} , \quad \mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} ,$$

and $\mathbf{I}$ is the identity matrix.

## JON'S OLD MEMORIES

After Sergey shows Jon his analytic solution (equation (13) coming up), Jon remembers an old trail. Old timers attacked problems like this by defining the quadratic form and finding its gradient. Then repeatedly making small steps down it. How big are the small steps? Experience will teach us. The quadratic form is $\mathbf{r}^T\mathbf{r}$. We take its derivative to find the search direction.

$$\Delta\mathbf{a} \quad = \quad - \text{ (some constant) } \left. \frac{\partial}{\partial \mathbf{a}^T} \right|_{\mathbf{a}=\bar{\mathbf{a}}} \mathbf{r}^T\mathbf{r} \tag{4}$$

Form the transpose of the residual (3) and then differentiate by $\mathbf{a}^T$. (By $\mathbf{a}^T$ we mean the complex conjugate transpose of $\mathbf{a}$.)

$$\frac{\partial \mathbf{r}^T}{\partial \mathbf{a}^T} \quad = \quad \frac{\partial}{\partial \mathbf{a}^T} \left\{ \mathbf{a}^T [\mathbf{d} \quad \gamma\mathbf{I}] \; - \; [-d_{n+1} \quad \gamma\bar{\mathbf{a}}] \right\} \quad = \quad [\mathbf{d} \quad \gamma\mathbf{I}] \tag{5}$$

and multiply that onto $\mathbf{r}$ from (3) keeping in mind that $\mathbf{d}^T\bar{\mathbf{a}}$ is a scalar.

$$\begin{aligned} \frac{\partial \mathbf{r}^T}{\partial \mathbf{a}^T} \quad \mathbf{r} \quad &= \quad [\mathbf{d} \quad \gamma\mathbf{I}] \left\{ \begin{bmatrix} \mathbf{d}^T \\ \gamma\,\mathbf{I} \end{bmatrix} \mathbf{a} \; - \; \begin{bmatrix} -d_{n+1} \\ \gamma\,\bar{\mathbf{a}} \end{bmatrix} \right\} & (6) \\ &= \quad \mathbf{d}(\mathbf{d}^T\mathbf{a}) + \gamma^2\mathbf{a} + \mathbf{d}d_{n+1} - \gamma^2\bar{\mathbf{a}} & (7) \\ \left. \frac{\partial \mathbf{r}^T}{\partial \mathbf{a}^T} \right|_{\mathbf{a}=\bar{\mathbf{a}}} \mathbf{r} \quad &= \quad (\mathbf{d}^T\bar{\mathbf{a}} + d_{n+1})\mathbf{d} & (8) \end{aligned}$$

Now the old timer has to think about what scale factor he will put on this gradient. The expression $(\mathbf{d}^T\bar{\mathbf{a}} + d_{n+1})$ is the prediction error. The prediction filter $\mathbf{a}$ takes data to predicted data, so it has no physical units. But the gradient (8) has units of data squared. So, the gradient needs a normalizing factor of the same units, say a local average of data squared which we could call an estimated data variance $\hat{\sigma}_d^2$. Finally the adjustable constant $\lambda$ that the old timer will learn from experience.

$$\mathbf{a} \quad = \quad \bar{\mathbf{a}} \; - \; \lambda \left( \frac{\text{prediction error}}{\hat{\sigma}_d^2} \right) \mathbf{d} \tag{9}$$

## SERGEY SOLVES THE PROBLEM ANALYTICALLY

Sergey rewrites Jon's equation (1) without the unity constraint getting equation (3). Notice the matrix in (3) that multiplies the unknown $\mathbf{a}$. Premultiply by its transpose thus obtaining the *formal solution,* the normal equations.

$$\mathbf{a} = \left(\mathbf{d}\,\mathbf{d}^T + \gamma^2\,\mathbf{I}\right)^{-1}\,\left(-d_{n+1}\,\mathbf{d} + \gamma^2\,\bar{\mathbf{a}}\right)\ . \tag{10}$$

Next, we can use the Sherman-Morrison formula[1] to transform the inverse matrix in equation (10) as follows:

$$\left(\mathbf{d}\,\mathbf{d}^T + \gamma^2\,\mathbf{I}\right)^{-1} = \frac{1}{\gamma^2}\,\left(\mathbf{I} - \frac{\mathbf{d}\,\mathbf{d}^T}{\gamma^2 + \mathbf{d}^T\,\mathbf{d}}\right)\ . \tag{11}$$

This Sherman-Morrison formula (11) is established in an appendix. Substituting equation (11) into (10) we have the least squares *actual solution*

$$\mathbf{a} \;=\; \frac{1}{\gamma^2}\,\left(\mathbf{I} - \frac{\mathbf{d}\,\mathbf{d}^T}{\gamma^2 + \mathbf{d}^T\,\mathbf{d}}\right)\,\left(-d_{n+1}\,\mathbf{d} + \gamma^2\,\bar{\mathbf{a}}\right) \tag{12}$$

which Sergey simplifies to the final result:

$$\mathbf{a} \;=\; \bar{\mathbf{a}} - \left(\frac{d_{n+1} + \mathbf{d}^T\,\bar{\mathbf{a}}}{\gamma^2 + \mathbf{d}^T\,\mathbf{d}}\right)\,\mathbf{d}\ . \tag{13}$$

In an appendix Stew Levin fills in the steps of Sergey's simplification.

Sergey writes, "Equation (13) confirms Jon's assertion that,

> 'It looks like conjugate-gradient iteration is not required(!?)' "

Indeed, computing $\mathbf{a}$ from $\bar{\mathbf{a}}$ and $d_{n+1}$ according to this method requires only elementary algebraic operations (vector dot-products) and no iterations.

## Choice of gamma

Practitioners hate adjustable parameters without given defaults. The dreadful situation we often see is someone reporting a numerical value for their $\epsilon$ where they fail to mention that the $\epsilon$ has physical units, and they don't tell us what those units are.

A practitioner will need to choose a numerical value for $\gamma$. It is our obligation to supply a default value. We do this by study of equations (1) and (13). Substituting $\gamma = \lambda\sigma_d$ where $\sigma_d^2$ is the data variance, we now have a dimensionless parameter $\lambda$ that we seek to choose so that the bottom block in regression (1) dominates the top, meaning that all of history should dominate the most recent single data value. Understanding that the numerical choice of $\lambda$ is subjective, we can choose to merge the two denominator terms in equation (13). We may think of $\gamma^2 + \mathbf{d}^T\mathbf{d}$ as $(N_d + N_a)\sigma_d^2$ where $N_a$ is the number of coefficients in the PEF and $N_d$ is the quantity of history

[1] `https://en.wikipedia.org/wiki/Sherman-Morrison_formula`

we wish to invoke. We may define $\hat{\sigma}_t^2$ as a leaky integration averaging of recent values of $d_t^2$.

Notice that the numerator in equation (13) is simply the prediction error. In summary, equation (13) is saying

$$\mathbf{a} \quad = \quad \bar{\mathbf{a}} - \left( \frac{\text{prediction error}}{(N_d + N_a)\ \hat{\sigma}_d^2} \right) \mathbf{d} \tag{14}$$

which says little chunks of recent data are simply added into or subtracted from the prediction filter in proportion to then current prediction error. The practitioner need supply only $N_d$ and $N_a$. The cost of updating the filter matches the cost of applying it. A bargain!

The challenge for the next generation is learning the scale factor, the adaptation rate, from the data itself. Instead of relying on a data analyst to play around in each case and choose a constant for it we should learn it from the residuals so we can adapt to spectral changes as they occur. **Deep Learning** we may call this.

# THE BIG PICTURE

In data modeling we have two main goals, a data fitting goal and a model styling goal. The data fitting goal requires a PEF and its adjoint. We've got those. The model styling goal soon leads us to preconditioning where an inverse PEF is needed.

The inverses to our PEFs are not assuredly stable. We've had that issue for years, but now having adaptive filters will tempt us to weaker statistical control. Luckily the exact inverse is not required for our main applications. The preconditioning application is a transformation to speed convergence of iterative data fitting. For that application, any stable variation of the estimated PEF should be satisfactory. Missing data filling is a related case. A divergent inverse PEF should not be a problem if we do not extrapolate too far.

## Averaging region

In Claerbout (2016) in this report the region of statistical averaging is very clearly defined as a 2-D patch of size `(b1,b2)`. Sergey's method intrinsically gives us a one dimensional patch (on a helix!) that is two dimensional only in the sense that adaptation rate can be chosen very small so multiple loops of the helix are included. But that ruins the notion of nonstationarity on the first axis! Sarkar (2016) in this report investigates path dependence and seems yet not to overcome this problem. This is a serious problem. We had first imagined Sergey's idea would speed things by a factor of `b1` × `b2`. Now we fear we have lost this feature.

Now 28 hours after this progress report deadline Jon feels he has a solution. We need two adjustable $\lambda$ parameters to manage the size and shape of a two-dimensional averaging region. All we need is to define a new $\bar{\mathbf{a}}$. Let $\bar{\mathbf{a}} = \lambda_t \mathbf{a}_{t-1,x} + \lambda_x \mathbf{a}_{t,x-1}$. This change requires storing one column of previously calculated PEFs, not a memory burden.

## Rahul identifies the adjoint of the TV-PEF

Those of us with a long history of filtering think of a filter adjoint as running the filter backwards. That view arises with recursive filters whose adjoint must indeed run backwards. With nonrecursive filters, such as prediction error, there is a more basic view. In a (nonrecursive) linear operator program, the inputs and outputs can be exchanged to produce the adjoint, namely

```
do iy = 1, ny {
do ix = 1, nx {
          if operator itself
                    y(iy) = y(iy) + b(iy,ix) × x(ix)
          if adjoint
                    x(ix) = x(ix) + b(iy,ix) × y(iy)
          }}
```

Visually, the adjoint PEF is simply the backwards PEF. We commonly need to apply a PEF (to a residual) and then immediately apply the adjoint PEF. There is no need to store the PEF coefficients between the two stages. We simply recompute them the second time when they are needed again. This situation arises while computing a model perturbation $\Delta\mathbf{m}$. A residual $\mathbf{Fm} - \mathbf{d}$ is first filtered with a PEF $\mathbf{A}$, and then hit with its adjoint $\Delta\mathbf{m} = \mathbf{F}^T\mathbf{A}^T\mathbf{A}(\mathbf{Fm} - \mathbf{d})$. On the second pass we would have two data inputs, the first input $\mathbf{Fm} - \mathbf{d}$ for again finding the PEF, the second input for applying its adjoint $\mathbf{A}^T$ to $\mathbf{A}(\mathbf{Fm} - \mathbf{d})$.

# CONCLUSION

We are panting in anticipation. What we have long wanted is coming into view. Nonstationarity! Wall Street collapsed because its wizards assumed spectral statistics are time invariant. My GIEE interpolation of Madagascar assumed topography was stationary where clearly it is not. Seismic velocity analyses show truncation effects at both inner and outer offsets. Soon we should be able to get a far more realistic grip on these problems.

These new tools are fast. The cost of finding and applying the new nonstationary PEFs merely doubles that of simple static stationary filters. These are small filters, and they will easily be multidimensional. Wow! Hooray!

Conceptual opportunities arise from the lightning speed. We'll easily be able to afford multiple realizations. Such realizations should enable us to learn about optimal parameter choice. Data analysts, often not deep in understanding the theory, need to chose operational parameters. The astonishing new speed will allow **deep learning**, where the process itself contributes to the choosing of its parameters. Those parameters too can be nonstationary. Meanwhile, all this is dreams, so for now what we must do is push forward with test cases of increasing diversity and complexity.

It is not clear how much in the way of applications we will be able to pull together by the SEP report deadline, but it is our goal to merge our preliminary papers. This on-going work is not on Claerbout's website (yet) and will not be until maybe after the coming SEP sponsor meeting.

## REFERENCES

Claerbout, J., 2016, Streaming nonstationary prediction error (I) : SEP-Report, **163**, 265–269.

Ruan, K., J. Jennings, E. Biondi, R. G. Clapp, S. A. Levin, and J. Claerbout, 2015, Industrial scale high-performance adaptive filtering with pef applications: SEP-Report, **160**, 177–188.

Sarkar, R., 2016, TV-PEF Design : Algorithmic Aspects, A Fast Update Scheme And 2D Path Dependence : SEP-Report, **163**, 279–296.

## JON CHECKS THE SHERMAN-MORRISON FORMULA

Sergey states the Sherman-Morrison formula (11). It's far from obvious.

$$\left(\mathbf{d}\,\mathbf{d}^T + \gamma^2\,\mathbf{I}\right)^{-1} = \frac{1}{\gamma^2}\left(\mathbf{I} - \frac{\mathbf{d}\,\mathbf{d}^T}{\gamma^2 + \mathbf{d}^T\,\mathbf{d}}\right) \ . \tag{A-1}$$

We rewrite it in product form preparing to expand and cancel terms.

$$\gamma^2\,\mathbf{I} \quad = \quad \left(\mathbf{d}\,\mathbf{d}^T + \gamma^2\,\mathbf{I}\right)\left(\mathbf{I} - \frac{\mathbf{d}\,\mathbf{d}^T}{\gamma^2 + \mathbf{d}^T\,\mathbf{d}}\right) \tag{A-2}$$

$$\gamma^2\,\mathbf{I} \quad = \quad \left(\mathbf{d}\,\mathbf{d}^T + \gamma^2\,\mathbf{I}\right) \ - \ \left(\mathbf{d}\,\mathbf{d}^T + \gamma^2\,\mathbf{I}\right)\left(\frac{\mathbf{d}\,\mathbf{d}^T}{\gamma^2 + \mathbf{d}^T\,\mathbf{d}}\right) \tag{A-3}$$

$$\gamma^2\,\mathbf{I} \quad = \quad \left(\mathbf{d}\,\mathbf{d}^T + \gamma^2\,\mathbf{I}\right) \ - \ \left(\frac{(\mathbf{d}\,\mathbf{d}^T + \gamma^2\,\mathbf{I})\mathbf{d}\,\mathbf{d}^T}{\gamma^2 + \mathbf{d}^T\,\mathbf{d}}\right) \tag{A-4}$$

$$\gamma^2\,\mathbf{I} \quad = \quad \left(\mathbf{d}\,\mathbf{d}^T + \gamma^2\,\mathbf{I}\right) \ - \ \left(\frac{\mathbf{d}\,\mathbf{d}^T\mathbf{d}\,\mathbf{d}^T + \gamma^2\,\mathbf{d}\,\mathbf{d}^T}{\gamma^2 + \mathbf{d}^T\,\mathbf{d}}\right) \tag{A-5}$$

$\mathbf{d}^T\mathbf{d}$ is a scalar, so it escapes the confinement inside $\mathbf{d}(\mathbf{d}^T\mathbf{d})\mathbf{d}^T$.

$$\gamma^2\,\mathbf{I} \quad = \quad \left(\mathbf{d}\,\mathbf{d}^T + \gamma^2\,\mathbf{I}\right) \;-\; \mathbf{d}\,\mathbf{d}^T \left(\frac{\mathbf{d}^T\,\mathbf{d} + \gamma^2}{\gamma^2 + \mathbf{d}^T\,\mathbf{d}}\right) \tag{A-6}$$

$$\gamma^2\,\mathbf{I} \quad = \quad \gamma^2\mathbf{I} \tag{A-7}$$

## STEW CHECKS SERGEY'S SIMPLIFICATION

From the least squares solution (12)

$$\mathbf{a} \;=\; \frac{1}{\gamma^2}\left(\mathbf{I} - \frac{\mathbf{d}\mathbf{d}^T}{\gamma^2 + \mathbf{d}^T\mathbf{d}}\right)\left(-d_{n+1}\mathbf{d} + \gamma^2\bar{\mathbf{a}}\right) \tag{B-1}$$

$$\mathbf{a} \;=\; \frac{1}{\gamma^2}\left(-d_{n+1}\mathbf{d} + \gamma^2\bar{\mathbf{a}} + \frac{d_{n+1}\mathbf{d}\mathbf{d}^T\mathbf{d} - \mathbf{d}\mathbf{d}^T\gamma^2\bar{\mathbf{a}}}{\gamma^2 + \mathbf{d}^T\mathbf{d}}\right) \tag{B-2}$$

$$\mathbf{a} \;=\; \bar{\mathbf{a}} \;-\; \frac{1}{\gamma^2}\;\frac{d_{n+1}\mathbf{d}(\gamma^2 + \mathbf{d}^T\mathbf{d}) + \mathbf{d}\mathbf{d}^T\gamma^2\bar{\mathbf{a}} - d_{n+1}\mathbf{d}\mathbf{d}^T\mathbf{d}}{\gamma^2 + \mathbf{d}^T\mathbf{d}} \tag{B-3}$$

$$\mathbf{a} \;=\; \bar{\mathbf{a}} \;-\; \frac{1}{\gamma^2}\;\frac{d_{n+1}\mathbf{d}\gamma^2 + d_{n+1}\mathbf{d}\mathbf{d}^T\mathbf{d} + \mathbf{d}\mathbf{d}^T\gamma^2\bar{\mathbf{a}} - d_{n+1}\mathbf{d}\mathbf{d}^T\mathbf{d}}{\gamma^2 + \mathbf{d}^T\mathbf{d}} \tag{B-4}$$

$$\mathbf{a} \;=\; \bar{\mathbf{a}} \;-\; \frac{1}{\gamma^2}\;\frac{d_{n+1}\mathbf{d}\gamma^2 + \mathbf{d}\mathbf{d}^T\gamma^2\bar{\mathbf{a}}}{\gamma^2 + \mathbf{d}^T\mathbf{d}} \tag{B-5}$$

$$\mathbf{a} \;=\; \bar{\mathbf{a}} \;-\; \frac{d_{n+1}\mathbf{d} \;+\; \mathbf{d}(\mathbf{d}^T\bar{\mathbf{a}})}{\gamma^2 + \mathbf{d}^T\mathbf{d}} \tag{B-6}$$

Since $\mathbf{d}^T\bar{\mathbf{a}}$ is a scalar, we can interchange it with $\mathbf{d}$.

$$\mathbf{a} \;=\; \bar{\mathbf{a}} \;-\; \frac{d_{n+1}\mathbf{d} \;+\; (\mathbf{d}^T\bar{\mathbf{a}})\mathbf{d}}{\gamma^2 + \mathbf{d}^T\mathbf{d}} \tag{B-7}$$

$$\mathbf{a} \;=\; \bar{\mathbf{a}} \;-\; \left(\frac{d_{n+1} + \mathbf{d}^T\bar{\mathbf{a}}}{\gamma^2 + \mathbf{d}^T\mathbf{d}}\right)\mathbf{d} \tag{B-8}$$

which is Sergey's final result (13).