On Computing a Least Square Smoothing Filter

Németh, Á.,Vörös, I., Gergó, L.

Submitted to HEJ. Manuscript no.: ANM-010128-A

This version is for internal use only!

Abstract

In this article a modification of the algorithm in [1] is given. The point of algorithm is that introducing new variables the equations to solve will be significantly simplified, and the main part of the computation can be done in a preliminary phase and be applied on new functions to filter, which can be parallel processed considerably accelerating it.

1 Introduction

For in many application the measured variable is corrupted by random noise, so as to reconstruct the underlying smooth function, a smoothing filter is desirable to be applied. We may assume that the noise is not influenced by the observed variable and obeys a normal distribution with a mean of zero and a standard deviation of δ.

In this article such a numeric algorithm is discussed that is for observations that are measured at fixed number of equally spaced (time)points.

In section 2 the theory of the original least square filter and the introduction of the new variables are discussed. With reference to [2] in section 3 we interpret a method for calculating eigenvectors of a kind of Toeplitz matrix.

In section 4 a MATLAB 5.3 program is given with its results.

2 Theoretical background

Let f be the vector of the measured variables, g the desired vector of smoothed data. Instead of the

$$\int_{x_0}^{x_n} g''(x)^2 dx$$

integral we consider the minimization of the

$$\sum_{i=2}^{n-1} (g_{i+1} - 2g_i + g_{i-1})^2$$

sum using the popular approximation of the second derivative of $g(x)$ by a finite difference scheme:

$$g''(x) \approx \frac{g_{i+1} - 2g_i + g_{i-1}}{\Delta},$$

whereas

$$\alpha^2 := n\delta^2 \geq \sum_{i=1}^{n} (g_i - f_i)^2. \quad \text{(The variance criterion)}$$

Let

$$A = \begin{bmatrix} 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \end{bmatrix}_{(n-2)\times n}.$$

So the problem is

$$\begin{aligned} \|Ag\|_2 &= \min \\ \|g-f\|_2^2 &\leq \alpha^2. \end{aligned}$$

If there are elements in the null space we require the unique nearest one, that is $f - A^T z$, where z is the solution of $AA^T z = Af$ equation.

Therefore let us assume that there is no element from the null space, i.e. for all $g \in \mathcal{N}(A)$ $\|g-f\|_2 > \alpha$. Now let us introduce the

$$L(g,l,s) = g^T A^T A g + l(\|g-f\|_2^2 + s^2 - \alpha^2)$$

Lagrange function, where $s^2 = \alpha^2 - \|g-f\|_2^2$ slack variable. With reference to the conditional extreme value theorem

$$\begin{aligned} \frac{\partial L}{\partial g} &= & (A^T A + lI)g - lf = 0 \\ \frac{\partial L}{\partial l} &= & \|g-f\|_2^2 + s^2 - \alpha^2 = 0 \\ \frac{\partial L}{\partial s} &= & ls = 0. \end{aligned}$$

If $l = 0$, then for A^T has full rank, $Ag = 0$. Therefore, using the assumption, $s = 0$. Let $z = \frac{1}{l} Ag$. Thus

$$\begin{aligned} (A^T A + lI)g = lf \quad (\Longrightarrow g - f = -A^T z) \quad &\Longrightarrow (AA^T + lI)z = Af \\ &\|g-f\|_2 = \|A^T z\|_2 = \alpha \end{aligned}$$

Let UDU^T be the Schur-decomposition of $C = AA^T$.

$$\begin{aligned} (UDU^T + lI)z &= Af \\ (DU^T + lU^T)z &= U^T Af \\ (D + lI)\underbrace{U^T z}_{z'} &= \underbrace{U^T Af}_{f'} \end{aligned}$$

Substituting this

$$\begin{aligned} z^T AA^T z &= \alpha^2 \\ z^T UDU^T z &= \alpha^2 \\ z'^T D z' &= \alpha^2 \\ \sum_{i=1}^{n} \frac{d_i (f_i')^2}{(d_i + l)^2} &= \alpha^2 \end{aligned}$$

So the root of

$$\sum_{i=1}^{n} \frac{d_i (f_i')^2}{(d_i + l)^2} - \alpha^2$$

is being searched.

As the derivative

$$-2\sum_{i=1}^{n}\frac{d_i(f_i')^2}{(d_i+l)^3}$$

is less than zero for positive numbers, there can be only unique positive solution. From the Gersgorin theorem $d_i < 16$. This way we know an l, where this function is less then zero:

$$\sum_{i=1}^{n}\frac{d_i(f_i')^2}{(d_i+l)^2}-\alpha^2<0$$
$$\frac{16}{l^2}\sum(f_i')^2<\alpha^2$$
$$l>\frac{4}{\alpha}\|f'\|_2$$

The Newton iteration proved to be unstable, but the bisection method is fast enough for this relatively short interval.

After this computation only the

$$(AA^T+lI)z=Af$$
$$g=f-A^Tz$$

system is to be solved.

3 Schur decomposition

x is an eigenvector of

$$C=\begin{bmatrix}6&-4&1&&&\\-4&6&-4&1&&\\1&-4&6&4&1&\\&\ddots&\ddots&\ddots&\ddots&\ddots\end{bmatrix}_{(}n-2)\times(n-2)$$

of λ eigenvalue, iff

$$x_{i-2}-4x_{i-1}+(6-\lambda)x_i-4x_{i+1}+x_{i+2}=0\quad\forall i\in[1..n-2],$$

assuming $x_0=x_{-1}=x_{n-1}=x_n=0$. This is a constant coefficient recursive sequence with a boundary condition above. If $w=z+\frac{1}{z}$,

$$P(z)=1-4z+(6-\lambda)z^2-4z^3+z^4=z^2(w^2-4w+(4-\lambda)).$$

The roots of this polynomial are

$$w_1=2+\sqrt{\lambda},\qquad w_2=2-\sqrt{\lambda},$$

so

$$z_i=\frac{w_i+\sqrt{w_i^2-4}}{2},$$

this way

$$x_h=c_1z_1^h+c_3z_1^{-h}+c_2z_2^h+c_4z_2^{-h},$$

as $z_1, z_1^{-1}, z_2, z_2^{-1}$ are different by pairs. Let

$$J = \left[a_{ij} = \begin{cases} 1 & \text{if } i+j=n+1 \\ 0 & \text{else} \end{cases} \right]_{(n-2)\times(n-2)}.$$

For A is symmetrical and diagonal, $AJ = JA$. Consequently, if x is an eigenvector, then so does Jx. Since λ is single, the eigensubspace of λ is one-dimensional: $Jx = \varepsilon x \Rightarrow x = Ix = JJx = \varepsilon^2 x$. I.e. x is either symmetrical or antisymmetrical: $x = \varepsilon x$. Therefore

$$0 = x_{n-1-h} - \varepsilon x_h = (c_3 z_1^{-(n-1)} - \varepsilon c_1) z_1^h + (c_1 z_1^{n-1} - \varepsilon c_3) z_1^{-h} + (c_4 z_2^{-(n-1)} - \varepsilon c_2) z_2^h + (c_2 z_2^{n-1} - \varepsilon c_4) z_2^{-h}.$$

For $z_i^{\pm 1}$ are different, the expressions in parenthesis are zeros, namely

$$c_3 = \varepsilon c_1 z_1^{n-1}, \qquad c_4 = \varepsilon c_2 z_2^{n-1}.$$

Thus

$$x_h = c_1(z_1^h + \varepsilon z_1^{n-1-h}) + c_2(z_2^h + \varepsilon z_2^{n-1-h}).$$

This is solvable for a known eigenvalue, and a fixed coefficient. The matrix of normalized eigenvectors is the conjugator matrix of the singular decomposition of C.

4 The program

In the following example we smoothed the

$$e^{-100(x-1/5)^2} + e^{-500(x-2/5)^2} + e^{-2500(x-3/5)^2} + e^{-12500(x-4/5)^2}$$

function.

Elapsed times:

n	1,000	5,000	10,000	20,000
Storing	24”	8’49”	32’20”	1h34’
Smoothing	.94”	18.9”	55”	2’43”

5 Reference

1. *Walter Gander and Jiří Hřebíček* , Solving Problems in Scientific Computing Using Maple and MATLAB, Springer, 1995.

2. *Kátai, I.* , Numerikus Analízis, ELTE, 1983 (in Hungarian)

Figure 1: Corrupted function (n=5000)

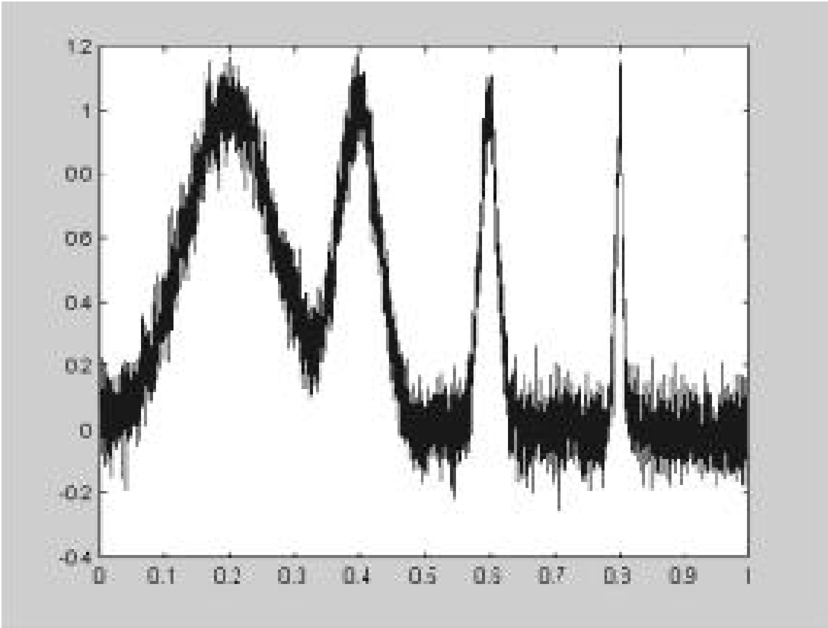

Figure 2: Smoothed function