Sparsity-Assisted Signal Smoothing (Revisited)

Ivan Selesnick

Electrical and Computer Engineering
Tandon School of Engineering
New York University
Brooklyn, New York

March 2017
Sparsity-Assisted Signal Smoothing (SASS)

We model the signal to be estimated as

\[ x = x_1 + x_2 \]

\[ x, \ x_1, \ x_2 \in \mathbb{R}^N \]

---

Signal Model

We model the signal to be estimated as

\[ x = x_1 + x_2, \quad x, x_1, x_2 \in \mathbb{R}^N, \]

where

- \( D x_1 \) is sparse where \( D \) is differentiation of order \( K \), i.e., a suitable regularizer for \( x_1 \) is

\[ \| D x_1 \|_1 \]

- \( x_2 \) is a low-frequency signal, i.e., for a zero-phase low-pass filter \( H \),

\[ x_2 = H(x_2) \]

\[ x_2 \approx H(x_2 + w) \]

where \( w \) is white Gaussian noise.
Signal Estimation

Goal: Estimate unknown signal $x$ from noisy signal $y$.

Signal in additive white Gaussian noise (AWGN):

$$y = x + w$$
$$y = x_1 + x_2 + w$$
$$y - x_1 = x_2 + w$$

Hence,

$$x_2 \approx H(x_2 + w) \implies x_2 \approx H(y - x_1).$$

If $x_1$ were known, then we could estimate $x_2$ by low-pass filtering $y - x_1$.

Thus, we estimate $x_2$ as

$$\hat{x}_2 = H(y - \hat{x}_1).$$
Signal Estimation

We estimate $x$ as

\[
\hat{x} = \hat{x}_1 + \hat{x}_2 \\
\hat{x} = \hat{x}_1 + H(y - \hat{x}_1) \\
\hat{x} = (I - H)\hat{x}_1 + Hy \\
\hat{x} = G\hat{x}_1 + Hy
\]

where $G$ is a zero-phase high-pass filter

\[
G = I - H.
\]
Signal Estimation

We assume $G$ is a high-pass filter that admits the factorization

$$G = RD$$

where $D$ is differentiation of order $K$.

$G$ could be a Butterworth or Chebyshev-II filter. Then

$$\hat{x} = RD\hat{x}_1 + Hy$$

i.e.,

$$\hat{x} = R\hat{u} + Hy$$

where

$$\hat{u} := D\hat{x}_1$$

is sparse.
SASS Optimization Problem

Given

\[ y = x + w, \]

to estimate \( x \) as

\[ \hat{x} = R\hat{u} + Hy \]

where \( u \) is sparse, we use sparse-regularized least squares

\[ \hat{u} = \arg \min_{u} \left\{ \frac{1}{2} \| y - (Ru + Hy) \|_2^2 + \lambda \| u \|_1 \right\} \]

where

\[ \| x \|_2^2 := \sum_n x(n)^2, \quad \| x \|_1 := \sum_n |x(n)|, \quad \lambda > 0. \]
SASS Optimization Problem

\[
\hat{u} = \arg \min_u \left\{ \frac{1}{2} \| y - (Ru + Hy) \|_2^2 + \lambda \| u \|_1 \right\}
\]

i.e.,

\[
\hat{u} = \arg \min_u \left\{ \frac{1}{2} \| (I - H)y - Ru \|_2^2 + \lambda \| u \|_1 \right\}
\]

can be solved via forward-backward splitting (FBS), ISTA, FISTA, etc.

Filters as Matrices

A banded Toeplitz matrix $P$

\[
P = \begin{bmatrix}
p_2 & p_1 & p_0 \\
p_2 & p_1 & p_0 & & \\
& & \ddots & \ddots \\
p_2 & p_1 & p_0
\end{bmatrix}
\]

represents an LTI system.

Convolution:

\[
(Px)_n = (p \ast x)(n)
\]

Transfer function:

\[
P(z) = \sum_{n} p_n z^{-n}
\]

Frequency response:

\[
P(e^{j\omega}) = \sum_{n} p_n e^{-jn\omega}
\]
Filters as Matrices

Consider cost function

\[
J(x) = \| Q(y - x) \|_2^2 + \alpha \| Px \|_2^2, \quad \alpha > 0
\]

where \( P \) and \( Q \) are banded Toeplitz matrices as above.

The function \( J \) is minimized by

\[
x = Hy
\]

where

\[
H := (Q^TQ + \alpha P^TP)^{-1} Q^TQ,
\]

i.e.,

\[
H := A^{-1} Q^TQ
\]

where

\[
A := Q^TQ + \alpha P^TP.
\]

\( A \) is banded.
Filters as Matrices

Matrix $H$ represents an LTI system with transfer function

$$H(z) = \frac{Q(z)Q(1/z)}{Q(z)Q(1/z) + \alpha P(z)P(1/z)}$$

and frequency response

$$H(e^{j\omega}) = \frac{|Q(e^{j\omega})|^2}{|Q(e^{j\omega})|^2 + \alpha|P(e^{j\omega})|^2}$$

Note that $H(e^{j\omega})$ is zero-phase (i.e., real-valued).
Butterworth Low-pass Filter

Some classical filters have transfer functions of the form

$$H(z) = \frac{Q(z)Q(1/z)}{Q(z)Q(1/z) + \alpha P(z)P(1/z)}.$$

For example, the Butterworth low-pass filter has

$$P(z) = (1 - z^{-1})^d$$
$$Q(z) = (1 + z^{-1})^d.$$  

When $d = 3$,

$$P = \begin{bmatrix} -1 & 3 & -3 & 1 \\ -1 & 3 & -3 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ -1 & 3 & -3 & 1 \end{bmatrix},$$
$$Q = \begin{bmatrix} 1 & 3 & 3 & 1 \\ 1 & 3 & 3 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 3 & 3 & 1 \end{bmatrix}.$$
Butterworth Low-pass Filter

Zero-phase Butterworth filter for finite-length data. (The dashed line is the old formulation which has end-point transients.)
High-pass Filter

If $H$ is a zero-phase low-pass filter, then $G = I - H$ is a zero-phase high-pass filter

$$G(z) = 1 - H(z) = \frac{\alpha P(z)P(1/z)}{Q(z)Q(1/z) + \alpha P(z)P(1/z)}.$$ 

and

$$G = I - H$$

$$= I - A^{-1}Q^TQ$$

$$= A^{-1}(A - Q^TQ)$$

$$= \alpha A^{-1}P^TP$$

where we used $A = Q^TQ + \alpha P^TP$ from above.
Factorization

Let $D$ be the $K$-order difference

$$D(z) = (1 - z^{-1})^K$$

When $K = 2$,

$$D = \begin{bmatrix} 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots \\ & & & 1 & -2 & 1 \end{bmatrix}$$

If $1 \leq K \leq d$, then

$$P(z) = (1 - z^{-1})^d$$
$$= (1 - z^{-1})^{d-K} (1 - z^{-1})^K$$
$$= P_1(z) \cdot D(z)$$

$$P = P_1D$$
Filters as Matrices

Summarizing:

\[ A = Q^TQ + \alpha P^TP \]
\[ H = A^{-1}Q \]
\[ P = P_1D \]
\[ G = I - H \]
\[ = \alpha A^{-1}P^TP \]
\[ = \alpha A^{-1}P^TP_1D \]
\[ = RD \]

where

\[ R = \alpha A^{-1}P^TP_1 \]
SASS Optimization Problem

The SASS cost function

\[ J(u) = \frac{1}{2} \| (I - H)y - Ru \|_2^2 + \lambda \| u \|_1 \]

where

\[ I - H = RD \]

can then be written

\[ J(u) = \frac{1}{2} \| \alpha A^{-1} P^T P y - \alpha A^{-1} P^T P_1 u \|_2^2 + \lambda \| u \|_1 \]

or

\[ J(u) = \frac{1}{2} \| \alpha A^{-1} P^T (P y - P_1) u \|_2^2 + \lambda \| u \|_1 \]

which can be solved using proximal algorithms.
Example

(a) Noisy data ($\sigma = 0.20$)

(b) Low-pass filtering ($d = 2$, $f_c = 0.030$)

RMSE = 0.117

(c) SASS ($K = 2$, $\lambda = 1.52$, $d = 2$, $f_c = 0.030$)

RMSE = 0.066

(d) Sparse signal $u$

Time ($n$)
Conclusion

Numerous signals can be modeled as the sum of two component signals: (1) signal with a sparse $K$-order derivative. (2) low-frequency signal and LTI filters over-smooth discontinuities (e.g., ‘corners’ of a signal).

Sparsity-assisted signal smoothing (SASS) combines and unifies LTI low-pass filtering and generalized total-variation denoising.

The SASS algorithm formulates the denoising problem as a sparse deconvolution problem which can be solved via proximal algorithms.

For the formulation and efficient implementation of SASS, we formulate zero-phase recursive filtering of finite-length input signals in terms of banded Toeplitz matrices.