Simultaneous Polynomial Approximation and Total Variation Denoising

Ivan Selesnick

Polytechnic Institute of New York University

Brooklyn, New York

selesi@poly.edu

Abstract

This paper addresses the problem of smoothing data with additive step discontinuities. The problem formulation is based on least square polynomial approximation and total variation denoising. In earlier work, an ADMM algorithm was proposed to minimize a suitably defined sparsity-promoting cost function. In this paper, an algorithm is derived using the majorization-minimization optimization procedure. The new algorithm converges faster and, unlike the ADMM algorithm, has no parameters that need to be set. The proposed algorithm is formulated so as to utilize fast solvers for banded systems for high computational efficiency. This paper also gives optimality conditions so that the optimality of a result produced by the numerical algorithm can be readily validated.

Introduction

Problem: estimate simultaneously a polynomial signal and an approximately piecewise constant signal from a noisy additive mixture.

The observed discrete-time data $y(n)$ is modeled as

$$y(n) = p(n) + x(n) + w(n), \quad n = 0, \ldots, N-1, \tag{1}$$

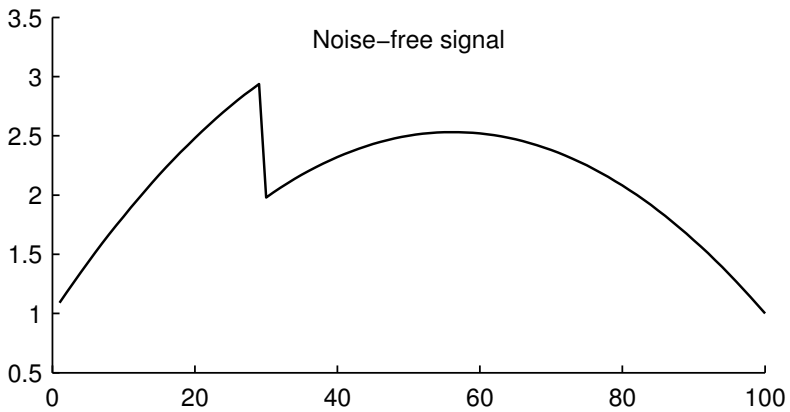
where:

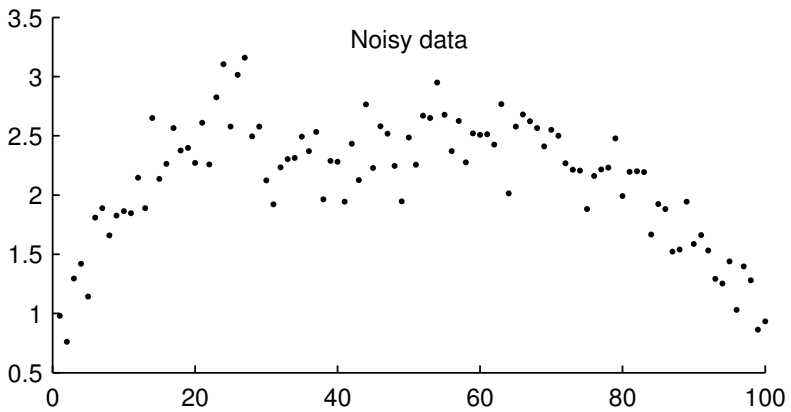
1. $p(n)$ is a low-order polynomial of order $d \ll N$.
2. $x(n)$ is approximately piecewise constant.
3. $w(n)$ is stationary white Gaussian noise.

Such data arises when a discrete event phenomenon is observed in the presence of a comparatively slow varying signal.

- Detection of virus particles using the Whispering Gallery Mode (WGM) biosensor (Arnold et al).
- Near infrared spectroscopic (NIRS) imaging systems (Graber et al).

MATLAB software available at `http://eeweb.poly.edu/iselesni/patv/`


Noise−free signal
3.5
3
2.5
2
1.5
1
0.5
0
20
40
60
80
100


Noisy data
3.5
3
2.5
2
1.5
1
0.5
0
20
40
60
80
100

Optimization Problem Statement

We pose the minimization problem[1]

$$(\mathbf{a}^*, \mathbf{x}^*) = \arg\min_{\mathbf{a},\mathbf{x}} \frac{1}{2} \sum_{n=0}^{N-1} |y(n) - p(n) - x(n)|^2 + \sum_{n=1}^{N-1} \phi(x(n) - x(n-1)) \tag{2}$$

where $p(n)$ is a polynomial,

$$p(n) = a_0 + a_1\, n + \cdots + a_d\, n^d, \tag{3}$$

and $\phi : \mathbb{R} \to \mathbb{R}$ is a sparsity-promoting penalty function, e.g.,

$$\phi(u) = \lambda |u| \quad \text{or} \quad \phi(u) = \frac{\lambda}{\alpha} \log(1 + \alpha |u|). \tag{4}$$

When $\phi(u) = \lambda|u|$, then the regularization term in (2) is $\lambda \|\mathbf{D}\mathbf{x}\|_1$, where $\|\cdot\|_1$ denotes the ℓ_1 norm.

We refer to (2) as the PATV problem. It is defined by $\{\mathbf{y}, d, \phi\}$.

[1] I. W. Selesnick, S. Arnold, and V. R. Dantham. "Polynomial smoothing of time series with additive step discontinuities". In: *IEEE Trans. Signal Process.* 60.12 (Dec. 2012), pp. 6305–6318. ISSN: 1053-587X. DOI: 10.1109/TSP.2012.2214219.

Notation

The N-point signal $\mathbf{x}$ is represented by the vector $\mathbf{x} = [x(0), \ldots, x(N-1)]^T$.

First-order difference of $\mathbf{x} \in \mathbb{R}^N$ is $\mathbf{Dx}$. $\mathbf{D}$ is $(N-1) \times N$.

$$\mathbf{D} = \begin{bmatrix} -1 & 1 & & & & \\ & -1 & 1 & & & \\ & & -1 & 1 & & \\ & & & -1 & 1 & \\ & & & & -1 & 1 \end{bmatrix}. \tag{5}$$

In MATLAB: $\mathbf{Dx}$ is `diff(x)`.

Cumulative sum of $\mathbf{u} \in \mathbb{R}^{N-1}$ is $\mathbf{Su}$. $\mathbf{S}$ is $N \times (N-1)$.

$$\mathbf{S} := \begin{bmatrix} 0 & & & & \\ 1 & & & & \\ 1 & 1 & & & \\ 1 & 1 & 1 & & \\ 1 & 1 & 1 & 1 & \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{6}$$

In MATLAB: $\mathbf{Sx}$ is `cumsum([0; x])`.

Note!

$$\boxed{\mathbf{DS} = \mathbf{I}} \tag{7}$$

Notation

Tall Vandermonde matrix, $\mathbf{V}$,

$$\mathbf{V} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 2 & & 2^d \\ 1 & 3 & & 3^d \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ 1 & N & \cdots & N^d \end{bmatrix}$$

$\mathbf{G}$: Orthogonalize columns of $\mathbf{V}$

$$\boxed{\mathbf{G}^T\mathbf{G} = \mathbf{I}}$$

$\mathbf{G}$ is $N \times (d+1)$.

Columns of $\mathbf{G}$ are an orthonormal basis for polynomials of order d on $\{0, \ldots, N-1\}$.

In MATLAB: `G = orth(bsxfun(@power,(0:N-1)',0:d))`

MM Optimization Algorithm[2]

Majorization-Minimization (MM) minimizes a convex function $F(\mathbf{x})$, using the iteration

$$\mathbf{x}^{(k+1)} = \arg\min_{\mathbf{x}} G(\mathbf{x},\, \mathbf{x}^{(k)}). \tag{8}$$

$G(\mathbf{x}, \mathbf{v})$ should be a convex majorizor of $F(\mathbf{x})$ that coincides with $F(\mathbf{x})$ at $\mathbf{x} = \mathbf{v}$.

That is,

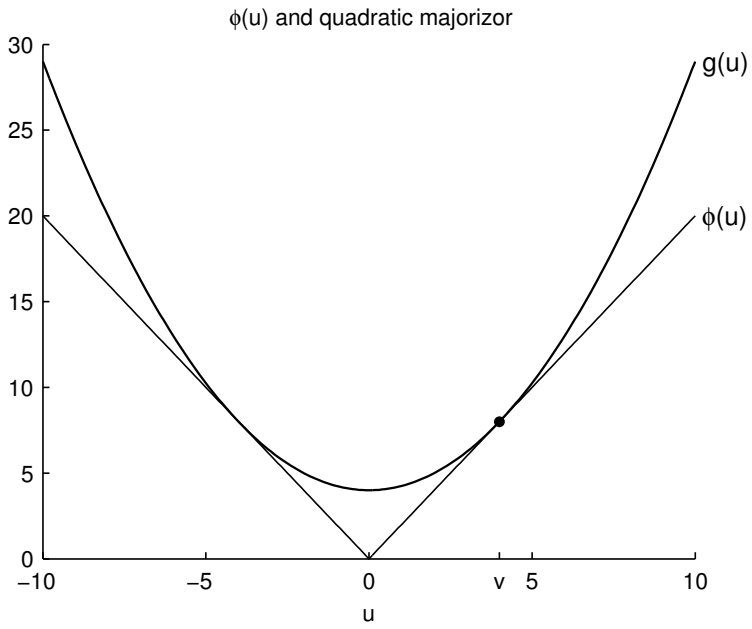
$$G(\mathbf{x}, \mathbf{v}) \geqslant F(\mathbf{x}),\ \forall \mathbf{x},$$

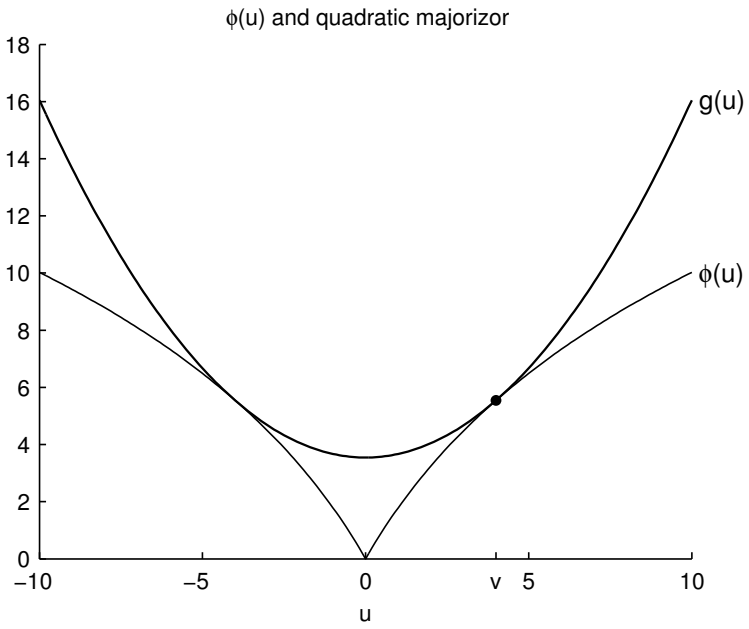
and

$$G(\mathbf{v}, \mathbf{v}) = F(\mathbf{v})$$

MM produces a sequence $\mathbf{x}^{(k)}$ converging to the minimizer of $F(\mathbf{x})$ under mild assumptions.

[2] M. Figueiredo, J. Bioucas-Dias, and R. Nowak. "Majorization-minimization algorithms for wavelet-based image restoration". In: *IEEE Trans. Image Process.* 16.12 (Dec. 2007), pp. 2980–2991.


ϕ(u) and quadratic majorizor
30
25
20
15
10
5
0
−10
−5
0
v
5
10
u
g(u)
ϕ(u)


ϕ(u) and quadratic majorizor
18
16
14
12
10
8
6
4
2
0
−10
−5
0
v
5
10
u
g(u)
ϕ(u)

Optimization Problem Simplification

The PATV problem (2) is equivalent to[3]

$$(\mathbf{a}^*, \mathbf{x}^*) = \arg\min_{\mathbf{a},\mathbf{x}} \ \frac{1}{2}\|\mathbf{y} - \mathbf{G}\mathbf{a} - \mathbf{x}\|_2^2 + \sum_n \phi([\mathbf{D}\mathbf{x}]_n) \tag{9}$$

where $\mathbf{G}$ is an orthogonal basis for degree-d polynomials (Vandermonde matrix, etc.)

Note that $\mathbf{a}^*$ can be expressed explicitly:

$$\mathbf{a}^* = \mathbf{G}^T(\mathbf{y} - \mathbf{x}). \tag{10}$$

Substituting (10) into (9), the PATV problem can be written as

$$\boxed{\mathbf{x}^* = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{H}(\mathbf{y} - \mathbf{x})\|_2^2 + \sum_n \phi([\mathbf{D}\mathbf{x}]_n)} \tag{11}$$

where $\mathbf{H}$ is given by

$$\mathbf{H} := \mathbf{I} - \mathbf{G}\mathbf{G}^T. \tag{12}$$

[3] In (9), polynomial p is represented using the orthonormal basis $\mathbf{G}$.

Algorithm

Adding a constant to $\mathbf{x}^*$ does not change the value of the cost function (11).

1. $\mathbf{D}$ annihilates constants.
2. $\mathbf{H}$ annihilates constants (a constant signal is exactly represented as a polynomial of degree zero).

The minimizer $\mathbf{x}^*$ is unique only up to an additive constant.

Therefore, we may write $\mathbf{x} = \mathbf{S}\mathbf{u}$ where $\mathbf{u} \in \mathbb{R}^{N-1}$.

Then by (7),

$$\mathbf{D}\mathbf{x} = \mathbf{D}\mathbf{S}\mathbf{u} = \mathbf{I}\mathbf{u} = \mathbf{u} \tag{13}$$

and (11) becomes

$$\boxed{\mathbf{u}^* = \arg\min_{\mathbf{u}} \Big\{ F(\mathbf{u}) = \frac{1}{2}\|\mathbf{H}(\mathbf{y} - \mathbf{S}\mathbf{u})\|_2^2 + \sum_n \phi(u(n)) \Big\}.} \tag{14}$$

To minimize F we use MM.

Let $g(u, v) : \mathbb{R}^2 \to \mathbb{R}$ be a quadratic majorizer of $\phi(u)$, defined as

$$g(u, v) = \frac{\phi'(v)}{2v}\, u^2 + \phi(v) - \frac{v}{2}\phi'(v). \tag{15}$$

The majorizor g can be used to obtain a majorizor for F. If $\mathbf{u}$ and $\mathbf{v}$ are vectors, then

$$\underbrace{\sum_n g(u(n), v(n))}_{\text{majorizor}} \geq \sum_n \phi(u(n)). \tag{16}$$

Note:

$$\sum_n g(u(n), v(n)) = \frac{1}{2}\mathbf{u}^T \mathbf{W}(\mathbf{v})\, \mathbf{u} + c \tag{17}$$

where $\mathbf{W}(\mathbf{v})$ is a diagonal matrix defined by

$$[\mathbf{W}(\mathbf{v})]_{n,n} = \frac{\phi'(v(n))}{v(n)}. \tag{18}$$

Therefore, a majorizor for $F(\mathbf{u})$ is given by

$$\boxed{G(\mathbf{u}, \mathbf{v}) = \frac{1}{2}\|\mathbf{Hy} - \mathbf{HSu}\|_2^2 + \frac{1}{2}\mathbf{u}^T \mathbf{W}(\mathbf{v})\, \mathbf{u} + c.} \tag{19}$$

$G(\mathbf{u}, \mathbf{v})$ is quadratic in $\mathbf{u}$. Hence, minimizing $G(\mathbf{u}, \mathbf{v})$ wrt $\mathbf{u}$ gives

$$\mathbf{u} = (\mathbf{S}^T \mathbf{H}^T \mathbf{H} \mathbf{S} + \mathbf{W}(\mathbf{v}))^{-1} \mathbf{S}^T \mathbf{H}^T \mathbf{H} \mathbf{y} \tag{20}$$

where $\mathbf{W}(\mathbf{v})$ depends on $\mathbf{v}$ per (18).

$\mathbf{H}^T \mathbf{H} = \mathbf{H}$, so

$$\mathbf{u} = (\mathbf{S}^T \mathbf{H} \mathbf{S} + \mathbf{W}(\mathbf{v}))^{-1} \mathbf{S}^T \mathbf{H} \mathbf{y}. \tag{21}$$

Therefore, the MM update produces the sequence

$$\boxed{\mathbf{u}^{(k+1)} = (\mathbf{S}^T \mathbf{H} \mathbf{S} + \mathbf{W}^{(k)})^{-1} \mathbf{S}^T \mathbf{H} \mathbf{y}} \tag{22}$$

where $\mathbf{W}^{(k)} := \mathbf{W}(\mathbf{u}^{(k)})$, i.e.,

$$[\mathbf{W}^{(k)}]_{n,n} = \frac{\phi'(u^{(k)}(n))}{u^{(k)}(n)}. \tag{23}$$

Two problems with (22):

1. High computational complexity: solution to a system of equations = order N^2.
2. Entries of $\mathbf{W}^{(k)}$ go to infinity as $\mathbf{u}$ becomes sparse (divide by zero).

Solution[4]: use the matrix inverse lemma (MIL).

Here, we need to use MIL twice.

First, write:

$$\begin{aligned}
\mathbf{S}^T\mathbf{H}\mathbf{S} + \mathbf{W}^{(k)} &= \mathbf{S}^T(\mathbf{I} - \mathbf{G}\mathbf{G}^T)\mathbf{S} + \mathbf{W}^{(k)} && (24)\\
&= \mathbf{W}^{(k)} + \mathbf{S}^T\mathbf{S} - \mathbf{S}^T\mathbf{G}\mathbf{G}^T\mathbf{S} && (25)\\
&= \mathbf{A}^{(k)} - \mathbf{B}\mathbf{B}^T && (26)
\end{aligned}$$

where

$$\mathbf{A}^{(k)} := \mathbf{W}^{(k)} + \mathbf{S}^T\mathbf{S}, \qquad \mathbf{B} := \mathbf{S}^T\mathbf{G}.$$

With this notation, by the matrix inverse lemma, we can write

$$\boxed{(\mathbf{S}^T\mathbf{H}\mathbf{S} + \mathbf{W}^{(k)})^{-1} = [\mathbf{A}^{(k)}]^{-1} + [\mathbf{A}^{(k)}]^{-1}\mathbf{B}\Big[\mathbf{I} - \mathbf{B}^T[\mathbf{A}^{(k)}]^{-1}\mathbf{B}\Big]^{-1}\mathbf{B}^T[\mathbf{A}^{(k)}]^{-1}} \quad (27)$$

[4] M. Figueiredo, J. Bioucas-Dias, and R. Nowak. "Majorization-minimization algorithms for wavelet-based image restoration". In: *IEEE Trans. Image Process.* 16.12 (Dec. 2007), pp. 2980–2991.

But now we need an efficient implementation of $[\mathbf{A}^{(k)}]^{-1}$ ...

Use the MIL again:

$$
\begin{aligned}
[\mathbf{A}^{(k)}]^{-1} &= [\mathbf{W}^{(k)} + \mathbf{S}^T\mathbf{S}]^{-1} && (28)\\
&= [\mathbf{W}^{(k)}]^{-1} - [\mathbf{W}^{(k)}]^{-1}\Big[(\mathbf{S}^T\mathbf{S})^{-1} + [\mathbf{W}^{(k)}]^{-1}\Big]^{-1}[\mathbf{W}^{(k)}]^{-1} && (29)\\
&= \mathbf{\Lambda}^{(k)} - \mathbf{\Lambda}^{(k)}\Big[(\mathbf{S}^T\mathbf{S})^{-1} + \mathbf{\Lambda}^{(k)}\Big]^{-1}\mathbf{\Lambda}^{(k)}. && (30)
\end{aligned}
$$

where $\mathbf{\Lambda}^{(k)} := [\mathbf{W}^{(k)}]^{-1}$ with

$$
[\mathbf{\Lambda}^{(k)}]_{n,n} = \frac{u^{(k)}(n)}{\phi'(u^{(k)}(n))}. \tag{31}
$$

$\mathbf{\Lambda}^{(k)}$ is <u>diagonal</u>.

Three observations:

1) As $u^{(k)}(n)$ goes to zero, the value $[\mathbf{\Lambda}^{(k)}]_{n,n}$ goes to zero, not infinity. ✓

$$
\phi(u) = \lambda|u| \qquad \Longrightarrow \qquad u/\phi'(u) = |u|/\lambda
$$

$$
\phi(u) = (\lambda/\alpha)\log(1+\alpha|u|) \qquad \Longrightarrow \qquad u/\phi'(u) = |u|(1+\alpha|u|)/\lambda
$$

2) The matrix $(\mathbf{S}^T\mathbf{S})^{-1}$ is *banded*

$$(\mathbf{S}^T\mathbf{S})^{-1} = \begin{bmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & -1 & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix}. \tag{32}$$

Defining

$$\mathbf{R}^{(k)} = (\mathbf{S}^T\mathbf{S})^{-1} + \mathbf{\Lambda}^{(k)}, \tag{33}$$

then (28) can be written as

$$[\mathbf{A}^{(k)}]^{-1} = \mathbf{\Lambda}^{(k)} \left(\mathbf{I} - [\mathbf{R}^{(k)}]^{-1}\mathbf{\Lambda}^{(k)}\right) \tag{34}$$

where $\mathbf{R}^{(k)}$ is banded. In fact, $\mathbf{R}^{(k)}$ a tridiagonal matrix.

So $[\mathbf{R}^{(k)}]^{-1}$ can be implemented with fast solvers for banded systems[5].

Hence $[\mathbf{A}^{(k)}]^{-1}$ has a fast implementation. ✓

[5] W. H. Press et al. *Numerical recipes in C: the art of scientific computing (2nd ed.)* Cambridge University Press, 1992. ISBN: 0-521-43108-5Sect 2.4.

3) Matrix (27) also involves

$$\mathbf{Q}^{(k)} := \mathbf{I} - \mathbf{B}^T[\mathbf{A}^{(k)}]^{-1}\mathbf{B}. \tag{35}$$

$\mathbf{Q}$ is small, $(d+1) \times (d+1)$, where d is the order of the low-order polynomial $p(n)$.

$\mathbf{B}$ is tall and narrow, $(N-1) \times (d+1)$, and need be computed once and saved.

So at each iteration, the matrix $\mathbf{Q}^{(k)}$ can be computed efficiently. ✓

Hence matrix (27) can be written as

$$\boxed{(\mathbf{S}^T\mathbf{H}\mathbf{S} + \mathbf{W}^{(k)})^{-1} = [\mathbf{A}^{(k)}]^{-1}\left(\mathbf{I} + \mathbf{B}\,[\mathbf{Q}^{(k)}]^{-1}\,\mathbf{B}^T\,[\mathbf{A}^{(k)}]^{-1}\right)} \tag{36}$$

and can be implemented efficiently as an operator with order Nd operations.

Therefore, the MM update (22) can be written as

$$\begin{aligned}
\mathbf{b} &= \mathbf{S}^T\mathbf{H}\mathbf{y} \\
\mathbf{u}^{(k+1)} &= [\mathbf{A}^{(k)}]^{-1}\left[\mathbf{b} + \mathbf{B}\,[\mathbf{Q}^{(k)}]^{-1}\,\mathbf{B}^T\,[\mathbf{A}^{(k)}]^{-1}\mathbf{b}\right]
\end{aligned}$$

This is a computationally stable and efficient algorithm to solve problem (14).

The algorithm is derived so as to take advantage of fast solvers for banded systems for computational efficiency.

MM algorithm for PATV

Input: $\mathbf{y} \in \mathbb{R}^N$, d, ϕ

1. $\mathbf{b} = \mathbf{S}^T \mathbf{H} \mathbf{y}$
2. $\mathbf{B} = \mathbf{S}^T \mathbf{G}$
3. $\mathbf{u} = \mathbf{1}$ (initialization)

repeat

4. $\mathbf{\Lambda}_{n,n} = \dfrac{u(n)}{\phi'(u(n))}$
5. $\mathbf{R} = (\mathbf{S}^T \mathbf{S})^{-1} + \mathbf{\Lambda}$
6. $\mathbf{A}^{-1} = \mathbf{v} \mapsto \mathbf{\Lambda}(\mathbf{v} - \mathbf{R}^{-1} \mathbf{\Lambda} \mathbf{v})$
7. $\mathbf{Q} = \mathbf{I} - \mathbf{B}^T \mathbf{A}^{-1}(\mathbf{B})$
8. $\mathbf{u} = \mathbf{A}^{-1}(\mathbf{b} + \mathbf{B}\,\mathbf{Q}^{-1} \mathbf{B}^T \mathbf{A}^{-1}(\mathbf{b}))$

until convergence

9. $\mathbf{x} = \mathbf{S}\mathbf{u}$
10. $\mathbf{a} = \mathbf{G}^T(\mathbf{y} - \mathbf{x})$

output: $\mathbf{x}$, $\mathbf{a}$

MATLAB program online at `http://eeweb.poly.edu/iselesni/patv/`

Optimality Conditions

When penalty ϕ is convex, then minimizer $\mathbf{u}^*$ must satisfy optimality conditions.

Useful to verify the optimality of a solution produced by a numerical algorithm.

Suppose ϕ is convex. If $\mathbf{u}$ solves (14), then $\mathbf{u}$ must satisfy:

$$[\mathbf{S}^T\mathbf{H}^T\mathbf{H}(\mathbf{y}-\mathbf{S}\mathbf{u})]_n \in \partial\phi(u(n)), \quad \forall n \tag{37}$$

where $[\mathbf{v}]_n$ denotes the n-th component of the vector $\mathbf{v}$ and $\partial\phi(\cdot)$ is the subdifferential, a *set*-valued generalization of the derivative.

Using $\mathbf{H}^T\mathbf{H} = \mathbf{H}$, $\mathbf{x} = \mathbf{S}\mathbf{u}$, and $\mathbf{u} = \mathbf{D}\mathbf{x}$, we may write

$$[\mathbf{S}^T\mathbf{H}(\mathbf{y}-\mathbf{x})]_n \in \partial\phi([\mathbf{D}\mathbf{x}]_n), \quad \forall n. \tag{38}$$

With $\phi(u) = \lambda|u|$, condition (38) becomes

$$\Big[\mathbf{S}^T\mathbf{H}(\mathbf{y}-\mathbf{x})\Big]_n \begin{cases} = \lambda, & u(n) > 0 \\ \in [-\lambda, \lambda], & u(n) = 0 \\ = -\lambda, & u(n) < 0 \end{cases} \tag{39}$$

The optimality condition is conveniently illustrated by a *scatter plot*.

Example: ℓ_1 norm penalty (convex)

Example of PATV (polynomial approximation / total variation denoising)

data = second order polynomial + additive step discontinuity + Gaussian noise

Set: $d = 2$, $\phi(u) = \lambda|u|$, and $\lambda = 1.5$,

The MM algorithm converges substantially faster than the ADMM algorithm[6].

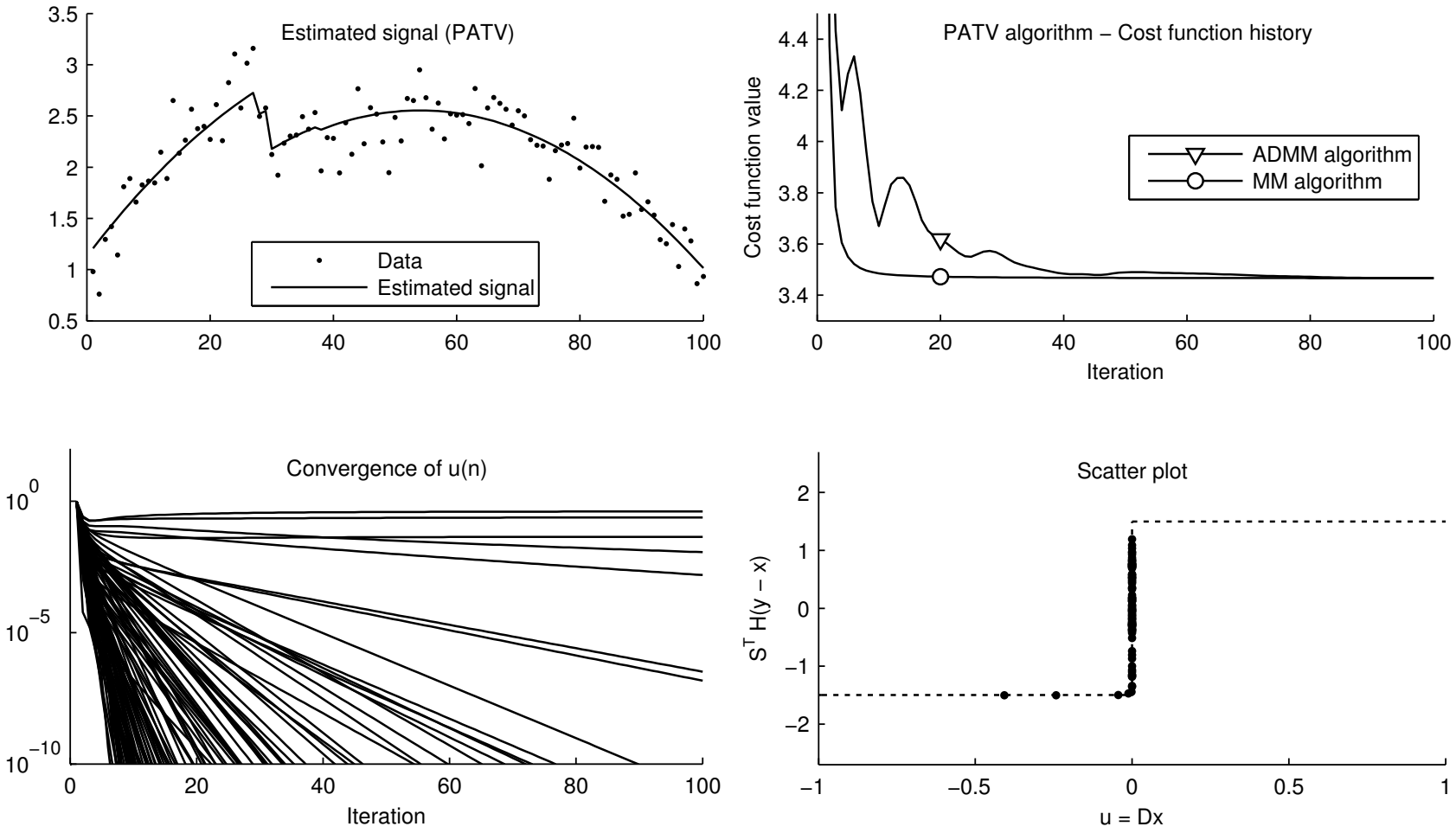
Moreover, the proposed MM algorithm requires no user specified parameter. The ADMM algorithm requires two parameters (μ_i) that must be carefully set so as to avoid slow convergence.

The convergence of the MM algorithm is further illustrated by showing the value $u^{(k)}(n)$ for each index n for 100 iterations. In this example, $u^{(0)}(n) = 1$ for all n. It can be seen that many $u(n)$ rapidly converge to zero.

The optimality of the obtained solution is validated in the scatter plot of $\mathbf{S}^T\mathbf{H}(\mathbf{y} - \mathbf{x})$ versus $\mathbf{Dx}$. According to (38), the solution is optimal if the points in the scatter plot lie on the dashed lines.

[6] I. W. Selesnick, S. Arnold, and V. R. Dantham. "Polynomial smoothing of time series with additive step discontinuities". In: *IEEE Trans. Signal Process.* 60.12 (Dec. 2012), pp. 6305–6318. ISSN: 1053-587X. DOI: 10.1109/TSP.2012.2214219.

Example: ℓ_1 norm penalty (convex)



Polynomial approximation of noisy polynomial data with an additive step discontinuity.

The penalty function is $\phi(u) = \lambda|u|$.

Example: logarithmic penalty (non-convex)

For enhances sparsity, this example uses

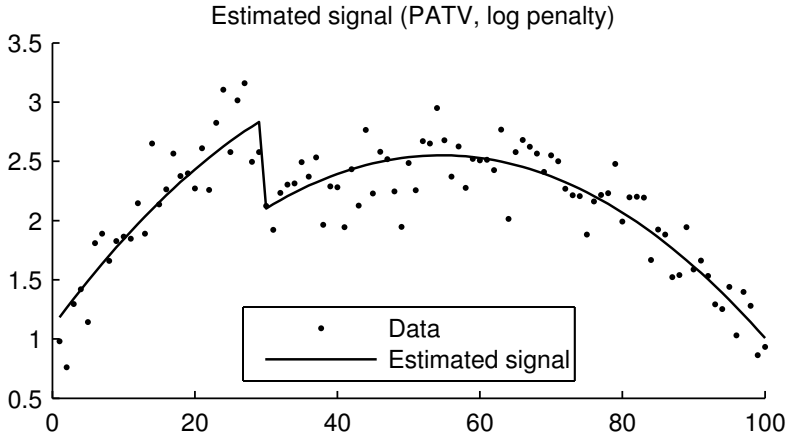
$$\phi(u) = \frac{\lambda}{\alpha} \log(1 + \alpha|u|)$$

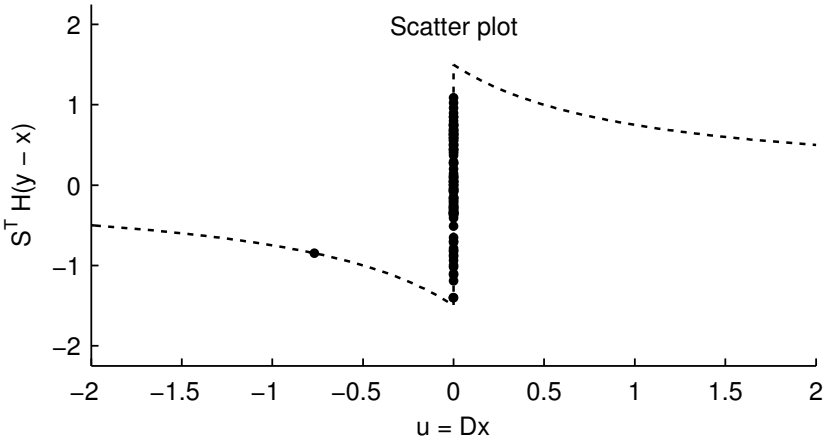
with $\lambda = 1.5$, $\alpha = 1.0$.

This non-convex function promotes sparsity more strongly than the ℓ_1 norm.

Now, $\mathbf{u}^*$ has only a single non-zero value, consistent with the simulated data.

The scatter plot of $\mathbf{S}^T\mathbf{H}(\mathbf{y} - \mathbf{x})$ versus $\mathbf{Dx}$ lies on $\phi'(u)$.





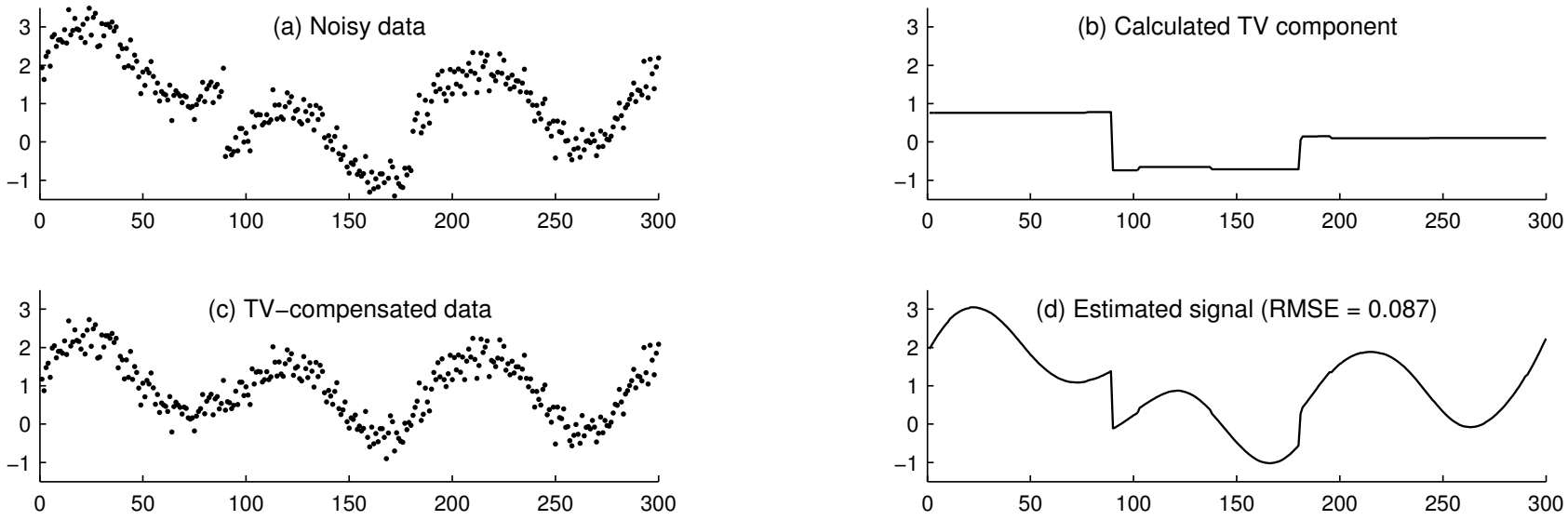
Conclusion

1. Signal model: smooth + additive step discontinuity + noise
2. Approach: polynomial approximation / total variation denoising (PATV)

New algorithm:

1. Fast convergence
2. No parameters
3. Uses fast solvers for banded systems for computational efficiency
4. Equally efficient for both ℓ_1 norm and non-convex penalties

Simultaneous Low-pass Filtering and Total Variation Densoising (LPF+TVD)

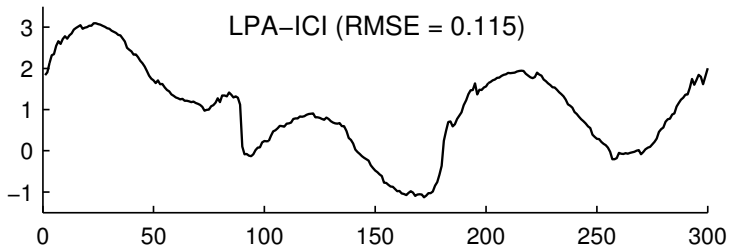


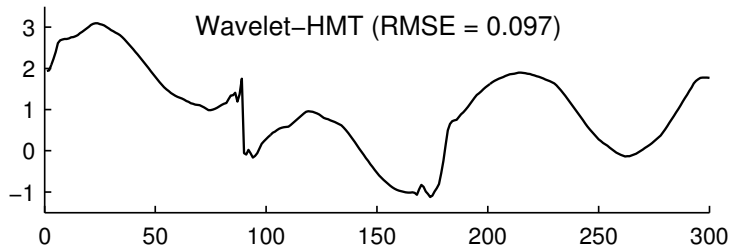
Observed noisy data: a low-frequency signal and with step-wise discontinuities (steps/jumps).

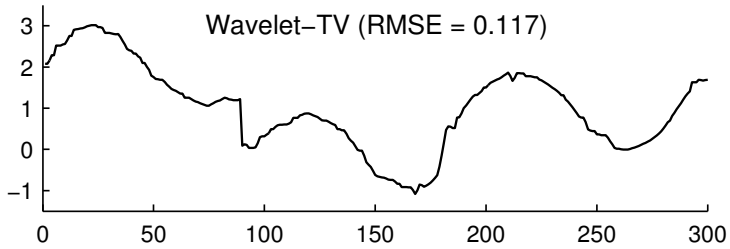
Goal: Simultaneous smoothing and change/jump detection.

Conventional low-pass filtering blurs discontinuities.

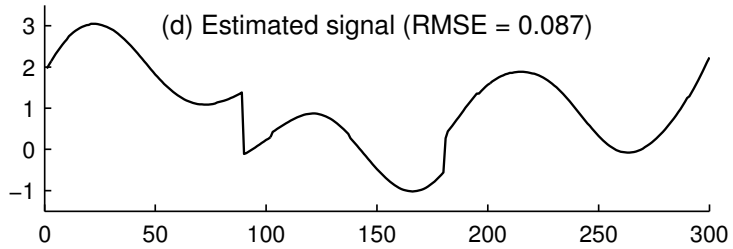
Comparison with other algorithms





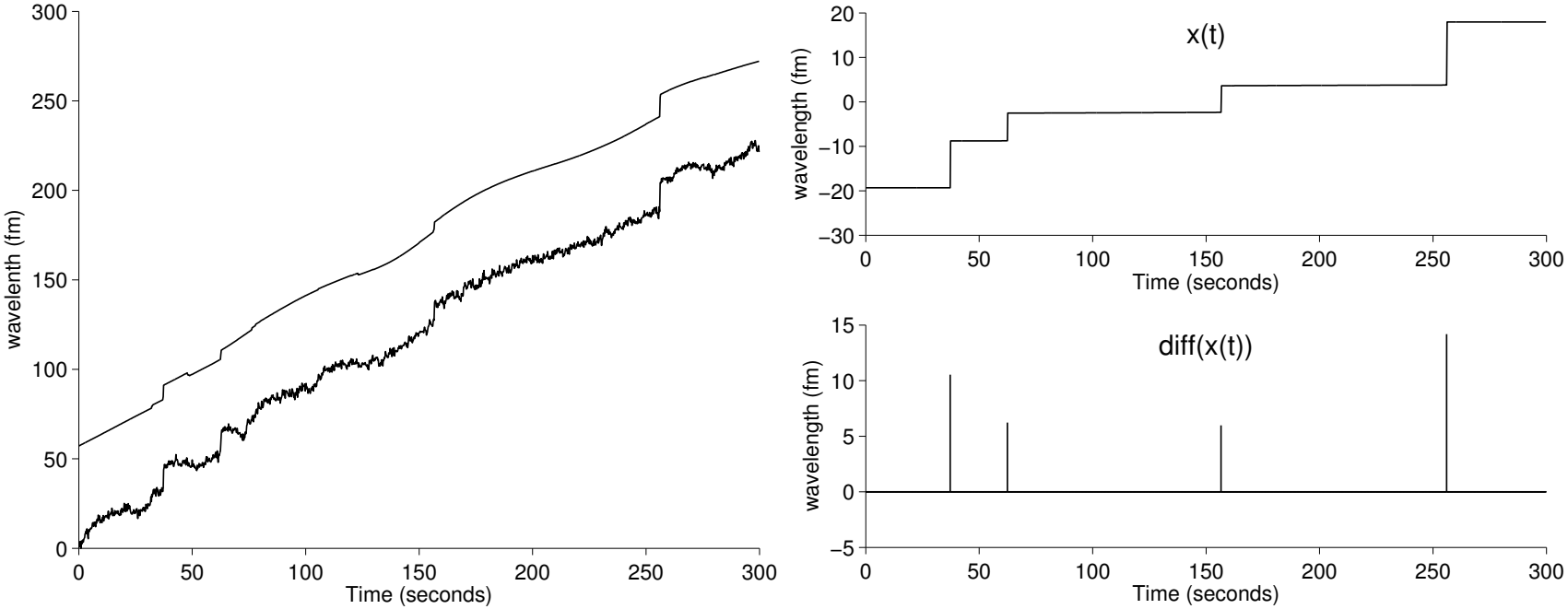


New algorithm



Simultaneous Low-pass Filtering and Total Variation Densoising (LPF+TVD)

Nano-particle Detection: The Whispering Gallery Mode (WGM) biosensor,[7] designed to detect individual nano-particles in real time, is based on detecting discrete changes in the resonance frequency of a microspherical cavity



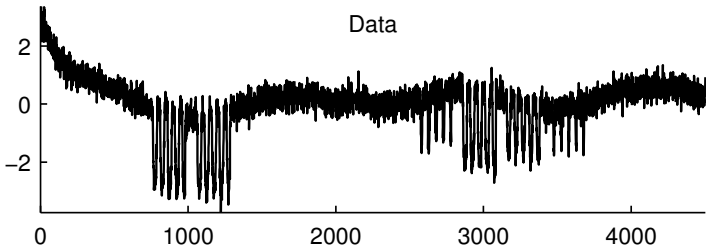
[7] S. Arnold et al. "Shift of Whispering-Gallery Modes in Microspheres by Protein Adsorption". In: *Opt. Lett* 28.4 (2003), pp. 272–274.

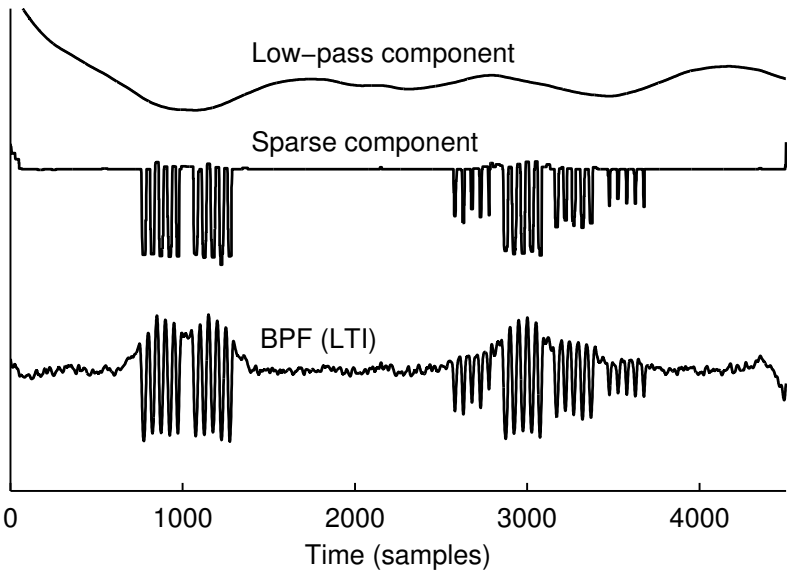
Low-Pass Filtering and Compound Sparse Denoising

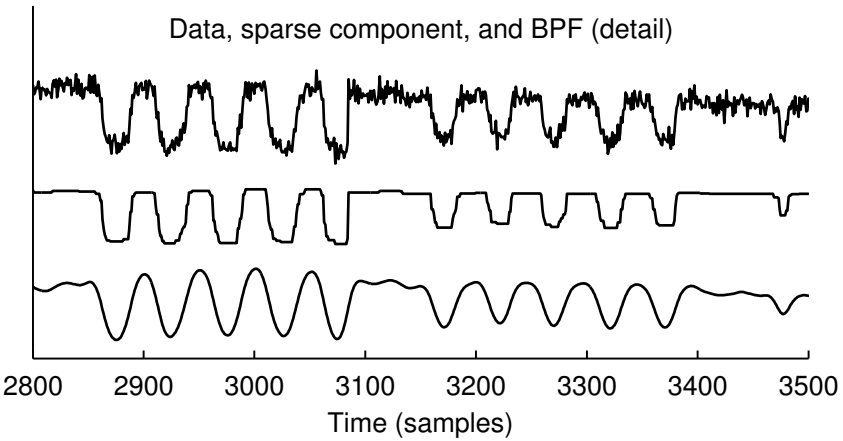
Signal model: (low-pass signal) + (sparse signal with sparse derivative) + (noise)

$$\underset{\mathbf{x}}{\operatorname{argmin}} \; \|\mathbf{H}(\mathbf{y}-\mathbf{x})\|_2^2 + \lambda_0 \|\mathbf{x}\|_1 + \lambda_1 \|\mathbf{D}\mathbf{x}\|_1$$

Near infrared spectroscopic (NIRS) data.[8]







LTI filtering can not separate the isolated pulses from low frequency trend.

[8] R. L. Barbour et al. "Optical tomographic imaging of dynamic features of dense-scattering media". In: *J. Opt. Soc. Am. A* 18.12 (Dec. 2001), pp. 3018–3036.

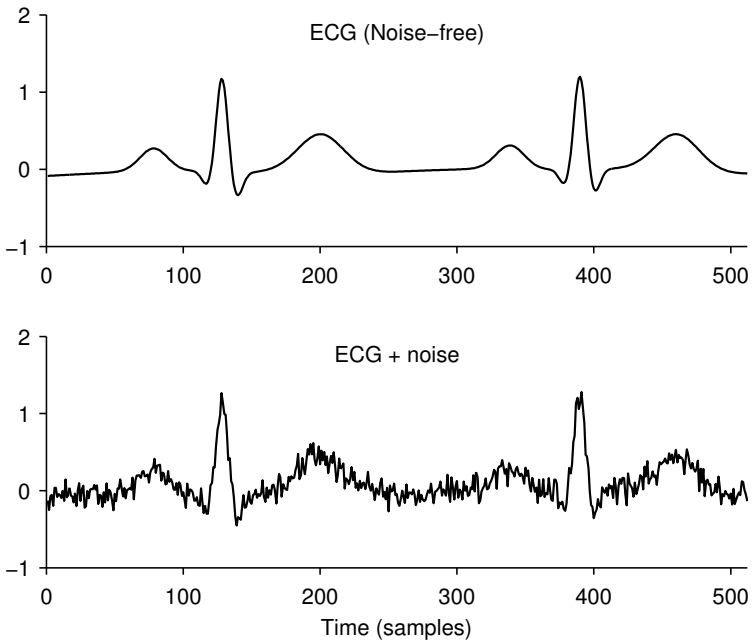
LPF + SDD (sparse derivative denoising)

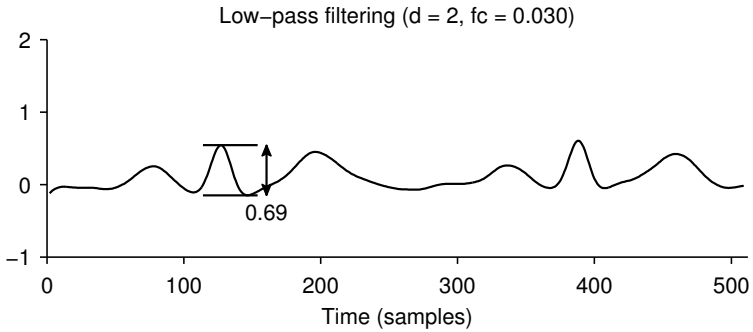
Signal model: (low-pass signal) + (sparse K-order derivative signal)

- Unifies low-pass filtering and sparse optimization.

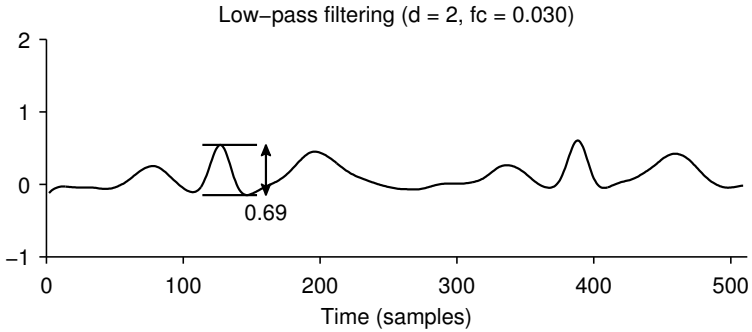
- More accurately captures signal features without sacrificing noise suppression.

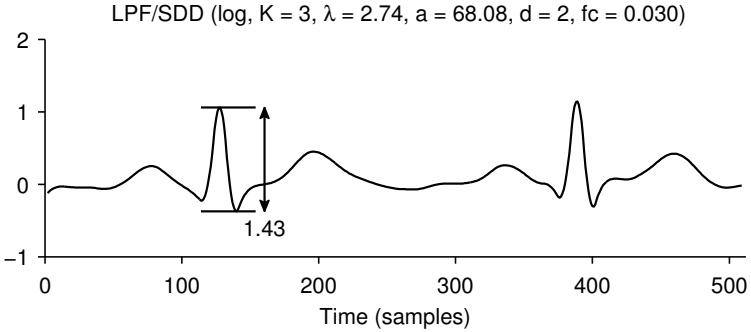
- Iterative algorithm using fast solvers for banded systems.

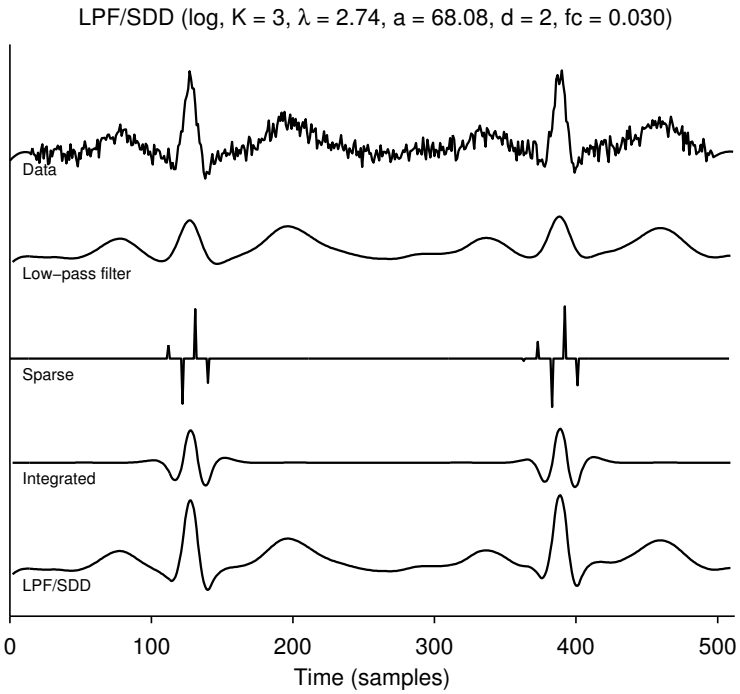


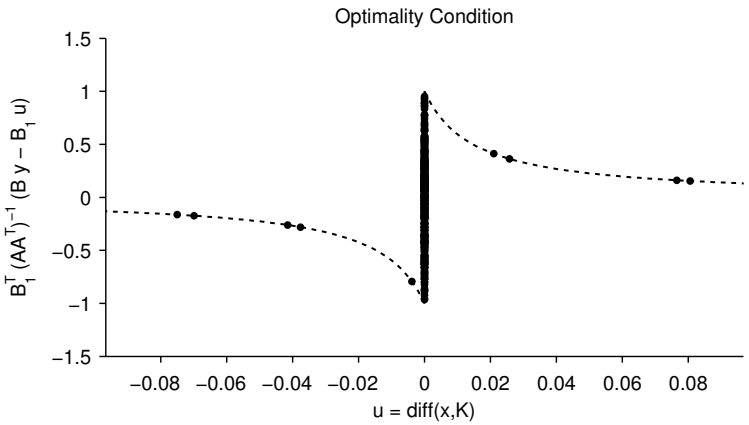


LPF/SDD with $K = 3$ and log penalty function (non-convex).









A low-pass filter attenuates the QRS peaks of the ECG signal. The new LPF+SDD algorithm better preserves them (singularities in the 3rd derivative).

Wavelet denoising without wavelets!