Linear Inverse Problems, Sparse Regularization, and Convex Optimization

Ivan Selesnick

March 23, 2017

Under-determined linear equations

Consider a system of under-determined system of equations

$$\mathbf{y} = \mathbf{A}\mathbf{x} \tag{1}$$

$$\begin{aligned} \mathbf{A} &: M \times N \quad (M < N) \\ \mathbf{y} &: M \times 1 \\ \mathbf{x} &: N \times 1 \end{aligned}$$

$$\mathbf{y} = \begin{bmatrix} y(0) \\ \vdots \\ y(M-1) \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} x(0) \\ \vdots \\ x(N-1) \end{bmatrix}$$

The system has more unknowns than equations.

We assume $\mathbf{A}\mathbf{A}^{\mathsf{H}}$ is invertible, therefore (1) has infinitely many solutions.

Norms

We will use the ℓ_2 and ℓ_1 norms.

$$\|\mathbf{x}\|_2^2 := \sum_n |x(n)|^2 \tag{2}$$

$$\|\mathbf{x}\|_1 := \sum_n |x(n)| \tag{3}$$

$\|\mathbf{x}\|_2^2$, i.e., the sum of squares, is referred to as the 'energy' of $\mathbf{x}$.

Least squares

To solve $\mathbf{y} = \mathbf{A}\mathbf{x}$, it is common to minimize the energy of $\mathbf{x}$.

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_2^2 \tag{4a}$$

$$\text{such that } \mathbf{y} = \mathbf{A}\mathbf{x}. \tag{4b}$$

The solution is

$$\hat{\mathbf{x}} = \mathbf{A}^{\mathsf{H}}\left(\mathbf{A}\mathbf{A}^{\mathsf{H}}\right)^{-1}\mathbf{y}. \tag{5}$$

When $\mathbf{y}$ is noisy, don't solve $\mathbf{y} = \mathbf{A}\mathbf{x}$ exactly. Instead, find approximate solution:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\{ \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_2^2 \right\} \tag{6}$$

The solution is

$$\hat{\mathbf{x}} = \left(\mathbf{A}^{\mathsf{H}}\mathbf{A} + \lambda\mathbf{I}\right)^{-1}\mathbf{A}^{\mathsf{H}}\,\mathbf{y}. \tag{7}$$

Large scale systems $\longrightarrow$ fast algorithms needed..

Sparse solutions

Another approach to solve $\mathbf{y} = \mathbf{A}\mathbf{x}$,

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \tag{8a}$$

$$\text{such that } \mathbf{y} = \mathbf{A}\mathbf{x} \tag{8b}$$

Problem (8) is the *basis pursuit* (BP) problem.

When $\mathbf{y}$ is noisy, don't solve $\mathbf{y} = \mathbf{A}\mathbf{x}$ exactly. Instead, find approximate solution.

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\{ \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_1 \right\} \tag{9}$$

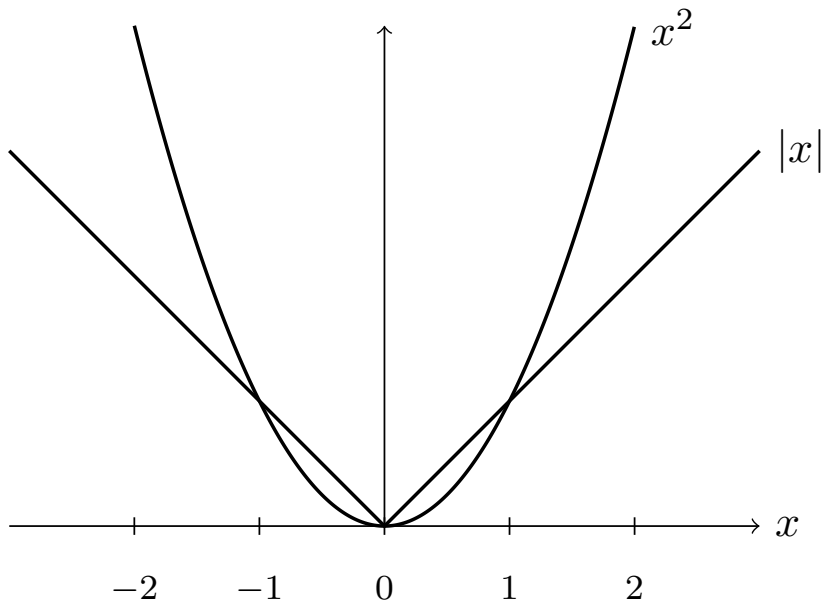
Problem (9) is the *basis pursuit denoising* (BPD) problem.

The BP/BPD problems can not be solved in explicit form, only by iterative numerical algorithms.

Least squares & BP/BPD

Least squares and BP/BPD solutions are quite different. Why?

To minimize $\|\mathbf{x}\|_2^2$ . . . , the largest values of $\mathbf{x}$ must be made small as they count much more than the smallest values.
$\Rightarrow$ least square solutions have many small values, as they are relatively unimportant $\Rightarrow$ least square solutions are not *sparse*.



Therefore, when it is known/expected that $\mathbf{x}$ is sparse, use the ℓ_1 norm; not the ℓ_2 norm.

Algorithms for sparse solutions

Objective function

- Non-differentiable
- Convex
- Large-scale

Algorithms

- MM: Majorization-Minimization
- ISTA: Iterative Shrinkage/Thresholding Algorithm
- FISTA: Fast ISTA
- SALSA (ADMM): Split Augmented Lagrangian Shrinkage Algorithm
- 'Matrix-free' algorithms

and more...

Parseval frames

The columns of $\mathbf{A}$ form a *Parseval frame* if $\boxed{\mathbf{A}\mathbf{A}^{\mathrm{H}} = p\mathbf{I}}$ with $p > 0$.

$$\begin{bmatrix} & \mathbf{A} & \end{bmatrix} \begin{bmatrix} & \mathbf{A}^{\mathrm{H}} & \\ & & \\ & & \end{bmatrix} = p\mathbf{I}$$

If $\boxed{\mathbf{A}\mathbf{A}^{\mathrm{H}} = p\mathbf{I}}$ then the solution to

$$\begin{aligned} \hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \ & \|\mathbf{x}\|_2^2 \\ \text{such that } & \mathbf{y} = \mathbf{A}\mathbf{x} \end{aligned}$$

is

$$\hat{\mathbf{x}} = \mathbf{A}^{\mathrm{H}}(\mathbf{A}\mathbf{A}^{\mathrm{H}})^{-1}\mathbf{y} \tag{10}$$

$$= \frac{1}{p}\mathbf{A}^{\mathrm{H}}\mathbf{y} \tag{11}$$

No matrix inversion needed.

Parseval frames

If

$$\boxed{\mathbf{A}\mathbf{A}^{\mathsf{H}} = p\mathbf{I}} \tag{12}$$

then the solution to

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\{ \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_2^2 \right\} \tag{13}$$

is

$$\hat{\mathbf{x}} = (\mathbf{A}^{\mathsf{H}}\mathbf{A} + \lambda \mathbf{I})^{-1} \mathbf{A}^{\mathsf{H}} \mathbf{y} \tag{14}$$

$$= \frac{1}{\lambda + p} \mathbf{A}^{\mathsf{H}} \mathbf{y} \tag{15}$$

using the matrix inverse lemma,

$$\left(\lambda \mathbf{I} + \mathbf{A}^{\mathsf{H}}\mathbf{A}\right)^{-1} = \frac{1}{\lambda}\mathbf{I} - \frac{1}{\lambda}\mathbf{A}^{\mathsf{H}} \left(\lambda \mathbf{I} + \mathbf{A}\mathbf{A}^{\mathsf{H}}\right)^{-1} \mathbf{A}. \tag{16}$$

• So, if $\boxed{\mathbf{A}\mathbf{A}^{\mathsf{H}} = p\mathbf{I}}$ then finding least square solutions is easy. No matrix inversion needed.

Some algorithms for BP/BPD also become computationally easier.

Example: Sparse Fourier coefficients using BP

The Fourier transform tells how to write the signal as a sum of sinusoids. But, it is not the only way.

Basis pursuit gives a sparse spectrum.

Suppose the M-point signal y is written as

$$y(m) = \sum_{n=0}^{N-1} c(n) \exp\left(\mathrm{j}\frac{2\pi}{N} mn\right), \qquad 0 \leqslant m \leqslant M-1 \tag{17}$$

where $c(n)$ is a length-N coefficient sequence, with $M \leqslant N$.

$$\mathbf{y} = \mathbf{A}\mathbf{c} \tag{18}$$

$$\mathbf{A}_{m,n} = \exp\left(\mathrm{j}\frac{2\pi}{N} mn\right), \quad 0 \leqslant m \leqslant M-1, \;\; 0 \leqslant n \leqslant N-1 \tag{19}$$

$\mathbf{c}$: length-N

The coefficients $c(n)$ are frequency-domain (Fourier) coefficients.

Example: Sparse Fourier coefficients using BP

1. If $N = M$, then $\mathbf{A}$ is the inverse N-point DFT matrix.
2. If $N > M$, then $\mathbf{A}$ is the first M rows of the inverse N-point DFT matrix.
   $\Rightarrow \mathbf{A}$ or $\mathbf{A}^{\mathsf{H}}$ can be implemented efficiently using the FFT.
   For example, in Matlab, $\mathbf{y} = \mathbf{A}\mathbf{c}$ is implemented as:

```
function y = A(c, M, N)
    v = N * ifft(c);
    y = v(1:M);
end
```

   Similarly, $\mathbf{A}^{\mathsf{H}}\,\mathbf{y}$ can be obtained by zero-padding and computing the DFT.
   In Matlab, $\mathbf{c} = \mathbf{A}^{\mathsf{H}}\,\mathbf{y}$ is implemented as:

```
function c = AT(y, M, N)
    c = fft([y; zeros(N-M, 1)]);
end
```

   $\Rightarrow$ Matrix-free algorithms.
3. Due to the orthogonality properties of complex sinusoids,

$$\boxed{\mathbf{A}\mathbf{A}^{\mathsf{H}} = N\,\mathbf{I}_M} \tag{20}$$

Example: Sparse Fourier coefficients using BP

When $N = M$, the coefficients $\mathbf{c}$ satisfying $\mathbf{y} = \mathbf{A}\mathbf{c}$ are uniquely determined.

When $N > M$, the coefficients $\mathbf{c}$ are not unique. Any vector $\mathbf{c}$ satisfying $\mathbf{y} = \mathbf{A}\mathbf{c}$ can be considered a valid set of coefficients. To find a particular solution we can minimize either $\|\mathbf{c}\|_2^2$ or $\|\mathbf{c}\|_1$.

Least squares:

$$\hat{\mathbf{c}} = \arg\min_{\mathbf{c}} \|\mathbf{c}\|_2^2 \tag{21a}$$

$$\text{such that } \mathbf{y} = \mathbf{A}\mathbf{c} \tag{21b}$$

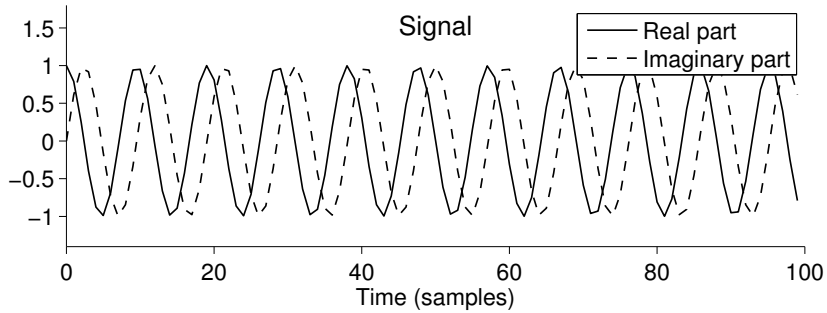
Basis pursuit:

$$\hat{\mathbf{c}} = \arg\min_{\mathbf{c}} \|\mathbf{c}\|_1 \tag{22a}$$

$$\text{such that } \mathbf{y} = \mathbf{A}\mathbf{c}. \tag{22b}$$

The two solutions can be quite different...

Example: Sparse Fourier coefficients using BP



Least square solution:

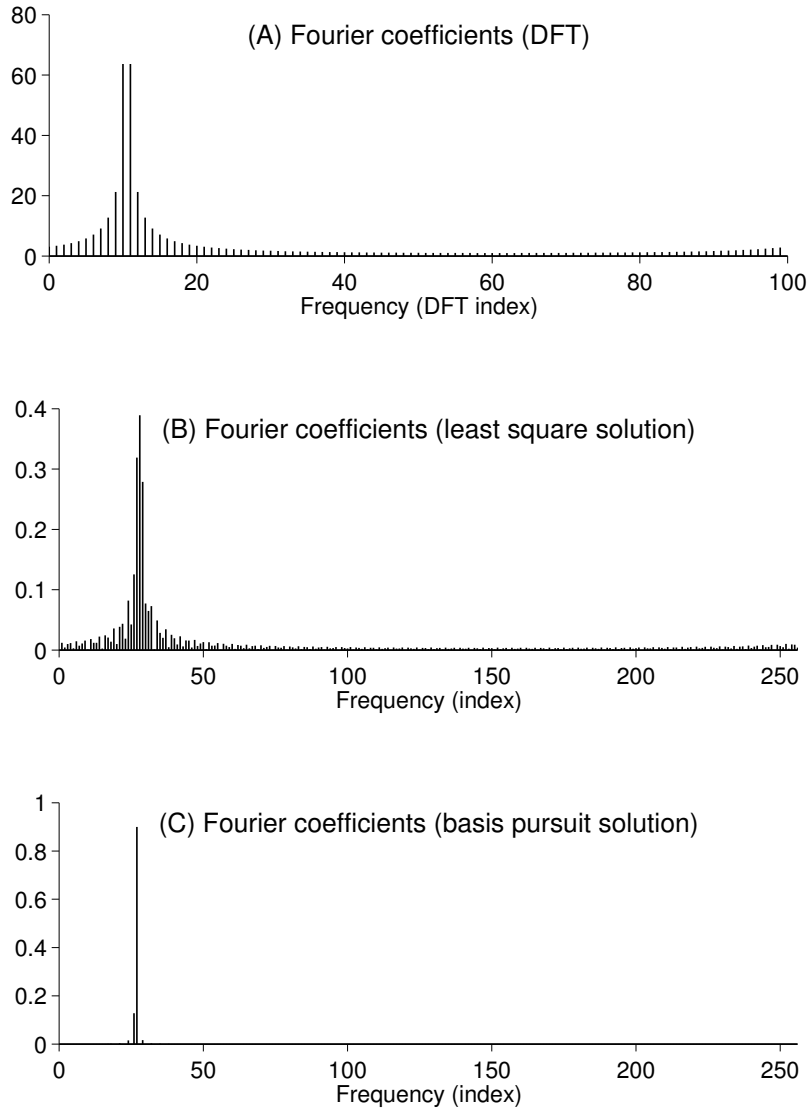
$$\begin{aligned}\hat{\mathbf{c}} &= \mathbf{A}^{\mathsf{H}}(\mathbf{A}\mathbf{A}^{\mathsf{H}})^{-1}\,\mathbf{y} \\ &= \frac{1}{N}\mathbf{A}^{\mathsf{H}}\mathbf{y} \qquad (\mathbf{A}\mathbf{A}^{\mathsf{H}} = N\,\mathbf{I})\end{aligned}$$

which is computed by

1. zero-pad the length-M signal $\mathbf{y}$ to length-N
2. compute its DFT

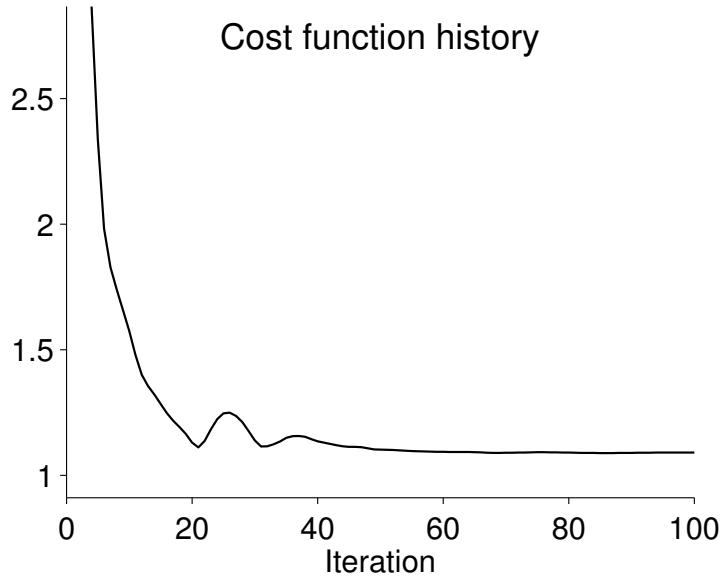
BP solution: Compute using algorithm SALSA

Example: Sparse Fourier coefficients using BP



The BP solution does not exhibit the leakage phenomenon.

Example: Sparse Fourier coefficients using BP



Cost function history of algorithm for basis pursuit solution

Example: Denoising using BPD

Digital LTI filters are often used for noise reduction (denoising).

But, if

- the noise and signal overlap in the frequency domain,
  or
- the respective frequency bands are unknown,

then it is difficult to use LTI filters.

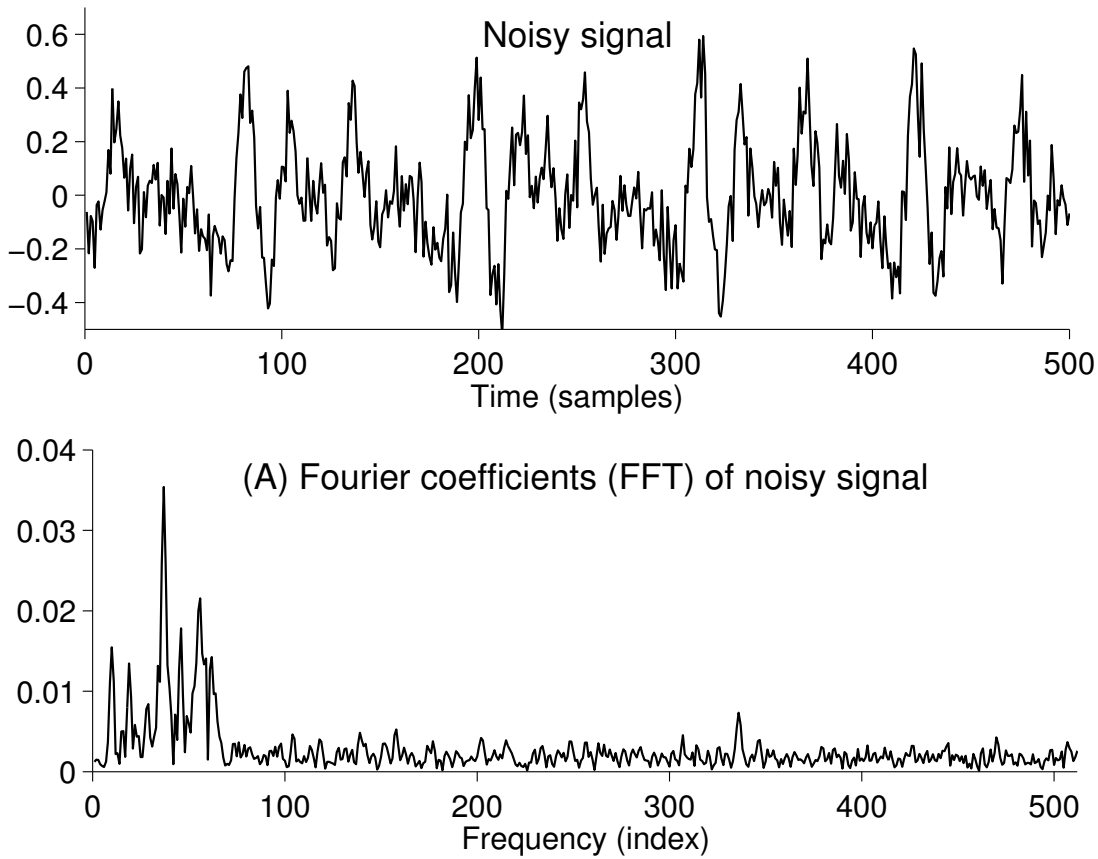
However, if the signal has sparse (or relatively sparse) Fourier coefficients, then BPD can be used for noise reduction.

Example: Denoising using BPD

Noisy speech signal y

$$y(m) = s(m) + w(m), \qquad 0 \leqslant m \leqslant M-1, \quad M = 500 \tag{23}$$

s : noise-free speech signal
w : noise sequence.



Example: Denoising using BPD

Assume the noise-free speech signal s has a sparse set of Fourier coefficients:

$$\mathbf{y} = \mathbf{A}\mathbf{c} + \mathbf{w}$$

$\mathbf{y}$: noisy speech signal, length M
$\mathbf{A}$: $M \times N$ DFT matrix (19)
$\mathbf{c}$: sparse Fourier coefficients, length N
$\mathbf{w}$: noise, length M

As $\mathbf{y}$ is noisy, find $\hat{\mathbf{c}}$ by solving the least square problem

$$\hat{\mathbf{c}} = \arg\min_{\mathbf{c}} \left\{ \|\mathbf{y} - \mathbf{A}\mathbf{c}\|_2^2 + \lambda \|\mathbf{c}\|_2^2 \right\} \tag{24}$$

or the basis pursuit denoising (BPD) problem

$$\hat{\mathbf{c}} = \arg\min_{\mathbf{c}} \left\{ \|\mathbf{y} - \mathbf{A}\mathbf{c}\|_2^2 + \lambda \|\mathbf{c}\|_1 \right\}. \tag{25}$$

Once $\hat{\mathbf{c}}$ is found, an estimate of the speech signal is given by $\hat{\mathbf{s}} = \mathbf{A}\hat{\mathbf{c}}$.

Example: Denoising using BPD

Least square solution:

$$\hat{\mathbf{c}} = (\mathbf{A}^{\mathrm{H}}\mathbf{A} + \lambda \mathbf{I})^{-1}\mathbf{A}^{\mathrm{H}}\,\mathbf{y} \tag{26}$$

$$= \frac{1}{\lambda + N}\,\mathbf{A}^{\mathrm{H}}\,\mathbf{y} \qquad (\mathbf{A}\mathbf{A}^{\mathrm{H}} = N\,\mathbf{I}) \tag{27}$$

using matrix inverse lemma.

$\Rightarrow$ least square estimate of the speech signal is

$$\hat{\mathbf{s}} = \mathbf{A}\hat{\mathbf{c}}$$

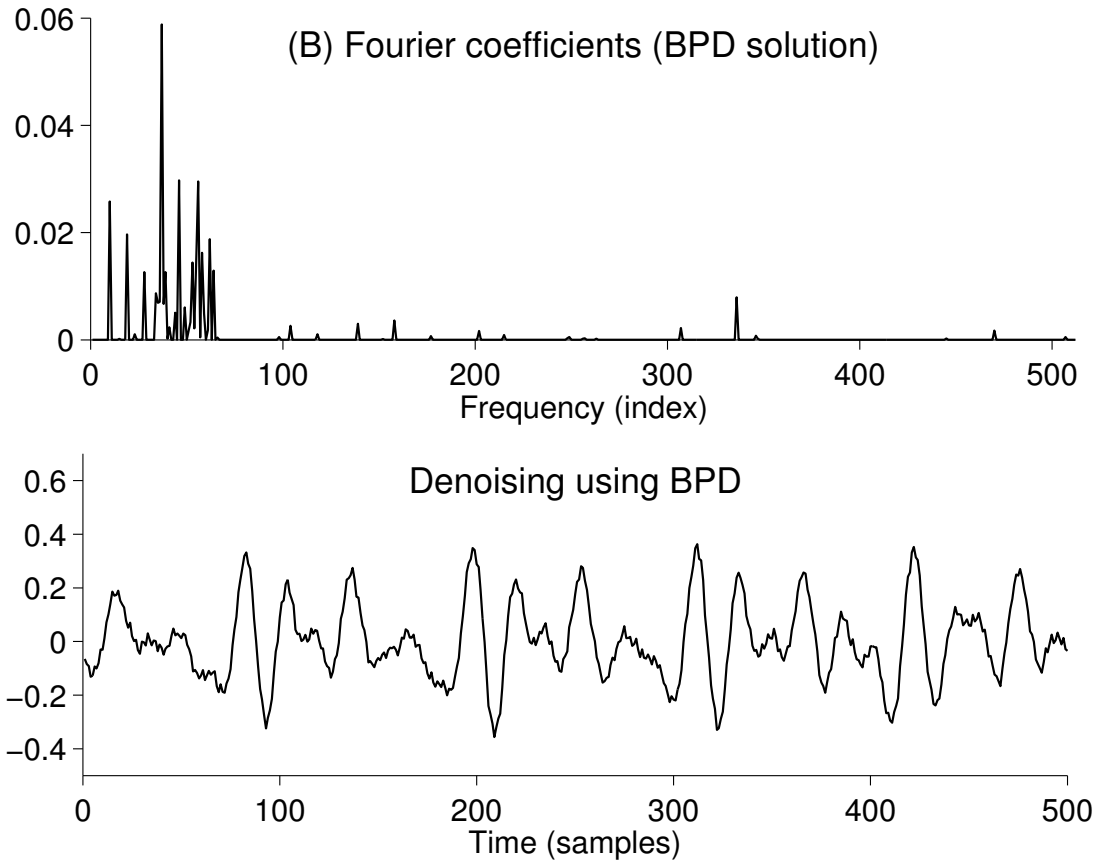
$$= \frac{N}{\lambda + N}\,\mathbf{y} \qquad \text{(least square solution)}.$$

But $\hat{\mathbf{s}}$ is only a scaled version of the noisy signal!

No filtering is achieved.

Example: Denoising using BPD

BPD solution



Obtained with algorithm SALSA. Effective noise reduction, unlike least squares!

Example: Deconvolution using BPD

If the signal of interest x is not only noisy but is also distorted by an LTI system with impulse response h, then the available data y is

$$y(m) = (h * x)(m) + w(m) \tag{28}$$

where '$*$' denotes convolution (linear convolution) and w is additive noise. Given the observed data y, we aim to estimate the signal x. We will assume that the sequence h is known.

Example: Deconvolution using BPD

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w} \tag{29}$$

x : length N
h : length L
y : length $M = N + L - 1$

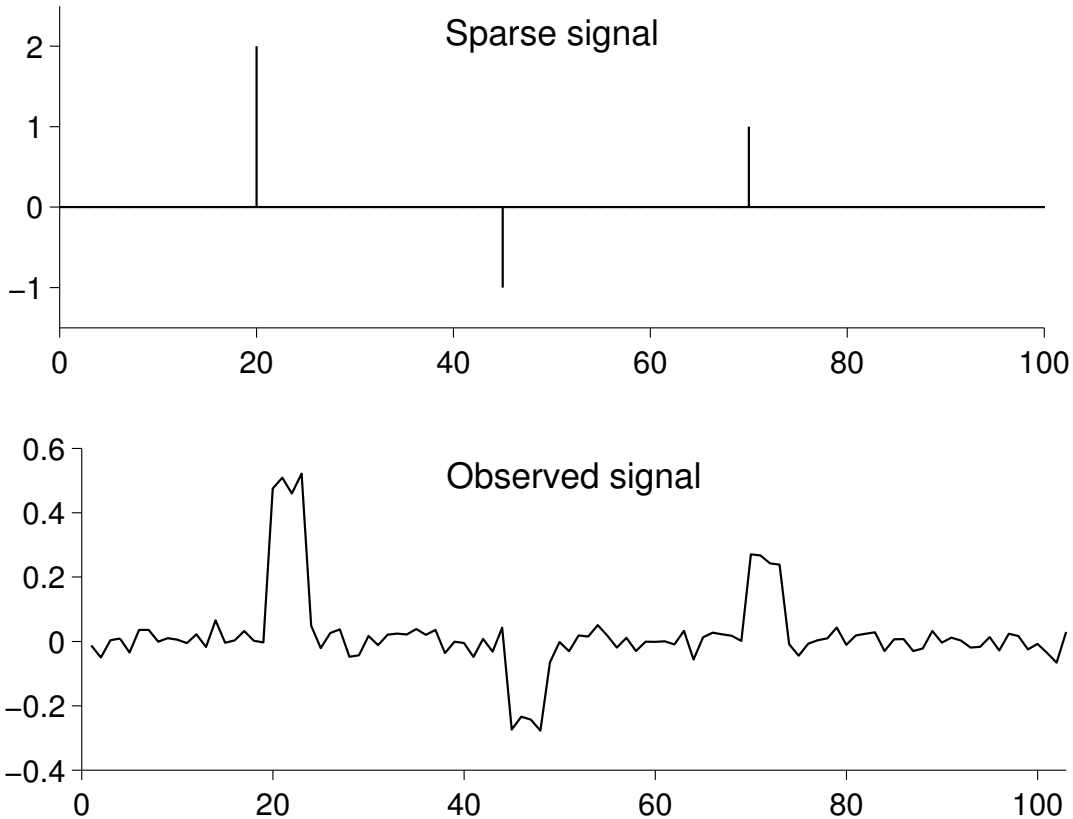
$$\mathbf{H} = \begin{bmatrix} h_0 & & & \\ h_1 & h_0 & & \\ h_2 & h_1 & h_0 & \\ & h_2 & h_1 & h_0 \\ & & h_2 & h_1 \\ & & & h_2 \end{bmatrix} \tag{30}$$

$\mathbf{H}$ is of size $M \times N$ with $M > N$ (because $M = N + L - 1$).

Example: Deconvolution using BPD

Sparse signal convolved by the 4-point moving average filter

$$h(n) = \begin{cases} \frac{1}{4} & n = 0, 1, 2, 3 \\ 0 & \text{otherwise} \end{cases}$$



Example: Deconvolution using BPD

Due to noise, solve the regularized least square problem

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\{ \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_2^2 \right\} \tag{31}$$

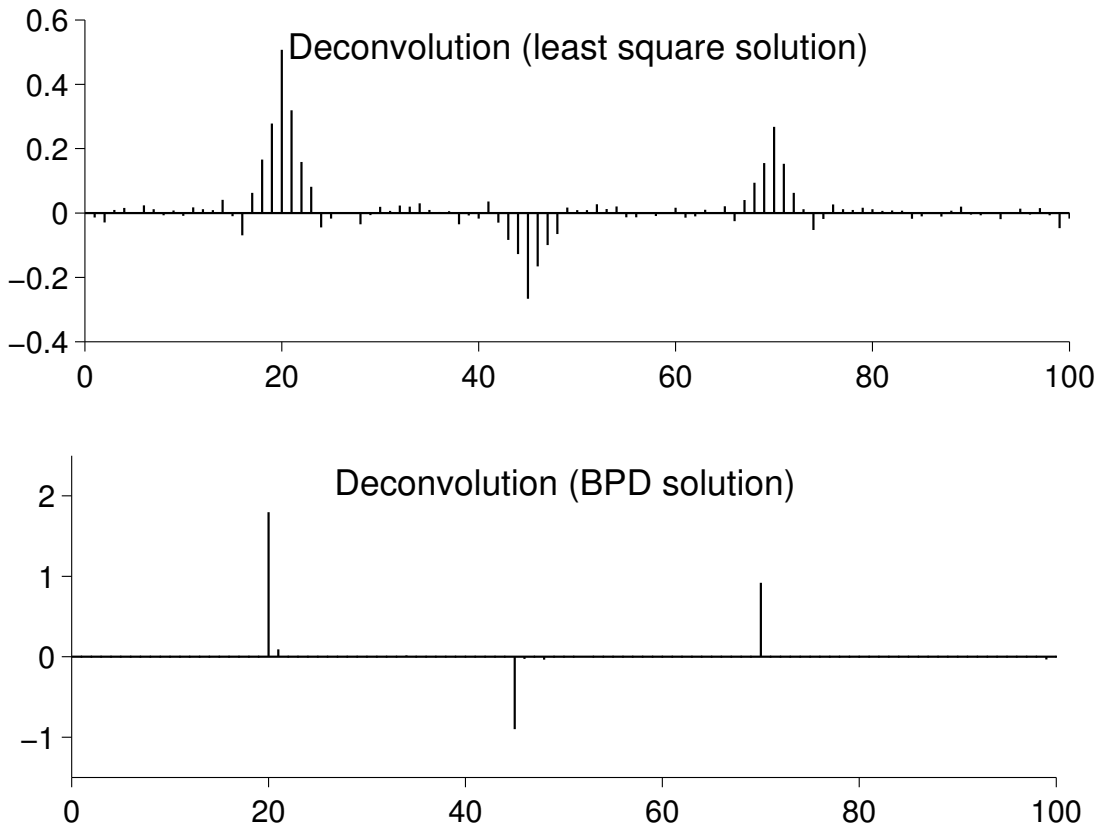
or the basis pursuit denoising (BPD) problem

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\{ \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_1 \right\}. \tag{32}$$

Least square solution:

$$\hat{\mathbf{x}} = (\mathbf{H}^{\mathsf{H}}\mathbf{H} + \lambda \mathbf{I})^{-1}\mathbf{H}^{\mathsf{H}}\,\mathbf{y}. \tag{33}$$

Example: Deconvolution using BPD

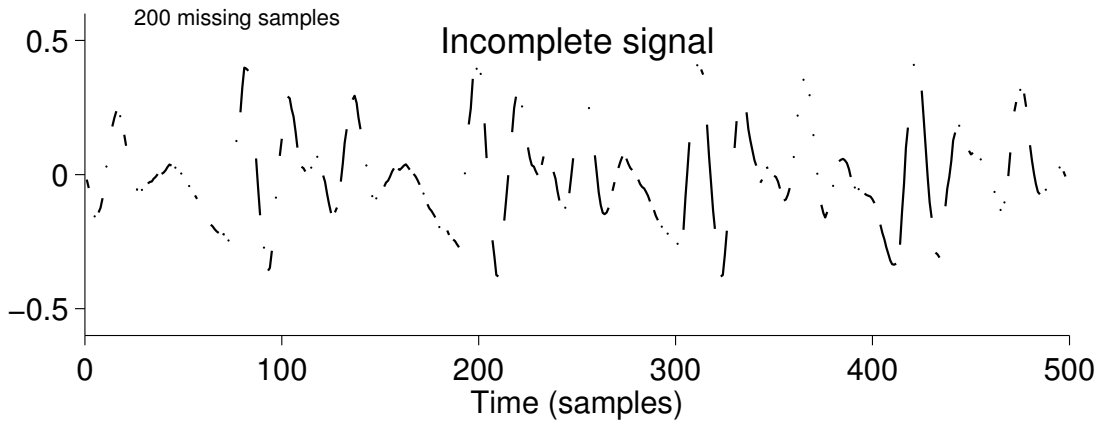


The BPD solution, obtained using SALSA, is more faithful to original signal.

Example: Filling in missing samples using BP

Due to data transmission/acquisition errors, some signal samples may be lost. Fill in missing values for *error concealment*.

Part of a signal or image may be intentionally deleted (image editing, etc). Convincingly fill in missing values according to the surrounding area to do *inpainting*.



Example: Filling in missing samples using BP

We write the incomplete data $\mathbf{y}$ as

$$\mathbf{y} = \mathbf{S}\mathbf{x} \tag{34}$$

$\mathbf{x}$: length M
$\mathbf{y}$: length $K < M$
$\mathbf{S}$: 'selection' (or 'sampling') matrix of size $K \times M$.

For example, if only the first, second and last elements of a 5-point signal $\mathbf{x}$ are observed, then the matrix $\mathbf{S}$ is given by:

$$\mathbf{S} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{35}$$

Problem: Given $\mathbf{y}$ and $\mathbf{S}$, find $\mathbf{x}$ such that $\mathbf{y} = \mathbf{S}\mathbf{x}$

$\Rightarrow$ Underdetermined system, infinitely many solutions.

Least square and BP solutions are very different...

Example: Filling in missing samples using BP

Properties of $\mathbf{S}$

1. $$\mathbf{S}\mathbf{S}^{\mathsf{T}} = \mathbf{I} \tag{36}$$

   where $\mathbf{I}$ is an $K \times K$ identity matrix. For example, with $\mathbf{S}$ in (35)

   $$\mathbf{S}\mathbf{S}^{\mathsf{T}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

2. $\mathbf{S}^{\mathsf{T}}\mathbf{y}$ sets the missing samples to zero.
   For example, with $\mathbf{S}$ in (35)

   $$\mathbf{S}^{\mathsf{T}}\mathbf{y} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} y(0) \\ y(1) \\ y(2) \end{bmatrix} = \begin{bmatrix} y(0) \\ y(1) \\ 0 \\ 0 \\ y(2) \end{bmatrix}. \tag{37}$$

Example: Filling in missing samples using BP

Suppose $\mathbf{x}$ has a sparse representation with respect to $\mathbf{A}$,

$$\mathbf{x} = \mathbf{A}\mathbf{c} \tag{38}$$

$\mathbf{c}$: sparse vector, length N, with $M \leqslant N$
$\mathbf{A}$: size $M \times N$.

The incomplete data $\mathbf{y}$ can then be written as

$$\mathbf{y} = \mathbf{S}\mathbf{x} \tag{39a}$$
$$= \mathbf{S}\mathbf{A}\mathbf{c}. \tag{39b}$$

Therefore, if $\hat{\mathbf{c}}$ satisfies

$$\mathbf{y} = \mathbf{S}\mathbf{A}\hat{\mathbf{c}} \tag{40}$$

then we may estimate $\mathbf{x}$ as

$$\hat{\mathbf{x}} = \mathbf{A}\hat{\mathbf{c}}. \tag{41}$$

Note that $\mathbf{y}$ is shorter than the coefficient vector $\mathbf{c}$, so (40) has infinitely many solutions.

Example: Filling in missing samples using BP

Any vector $\hat{\mathbf{c}}$ satisfying $\mathbf{y} = \mathbf{SA}\hat{\mathbf{c}}$ can be considered a valid set of coefficients.

To find a particular solution, solve the least squares (LS) problem

$$\hat{\mathbf{c}} = \arg\min_{\mathbf{c}} \|\mathbf{c}\|_2^2 \tag{42a}$$

$$\text{such that } \mathbf{y} = \mathbf{SAc} \tag{42b}$$

or the basis pursuit (BP) problem

$$\hat{\mathbf{c}} = \arg\min_{\mathbf{c}} \|\mathbf{c}\|_1 \tag{43a}$$

$$\text{such that } \mathbf{y} = \mathbf{SAc}. \tag{43b}$$

We will see ... the LS and BP solutions are very different.

Let us assume $\mathbf{A}$ satisfies

$$\mathbf{A}\mathbf{A}^{\mathsf{H}} = p\mathbf{I}, \tag{44}$$

for some positive real number p.

Example: Filling in missing samples using BP

The least square solution is

$$\hat{\mathbf{c}} = (\mathbf{SA})^{\mathsf{H}}((\mathbf{SA})(\mathbf{SA})^{\mathsf{H}})^{-1}\,\mathbf{y} \tag{45}$$
$$= \mathbf{A}^{\mathsf{H}}\mathbf{S}^{\mathsf{T}}(\mathbf{SAA}^{\mathsf{H}}\mathbf{S}^{\mathsf{T}})^{-1}\,\mathbf{y} \tag{46}$$
$$= \mathbf{A}^{\mathsf{H}}\mathbf{S}^{\mathsf{T}}(p\,\mathbf{SS}^{\mathsf{T}})^{-1}\,\mathbf{y} \qquad (\mathbf{AA}^{\mathsf{H}} = p\mathbf{I}) \tag{47}$$
$$= \mathbf{A}^{\mathsf{H}}\mathbf{S}^{\mathsf{T}}(p\,\mathbf{I})^{-1}\mathbf{y} \qquad (\mathbf{SS}^{\mathsf{T}} = \mathbf{I}) \tag{48}$$
$$= \frac{1}{p}\mathbf{A}^{\mathsf{H}}\mathbf{S}^{\mathsf{T}}\mathbf{y}. \tag{49}$$

Hence, the least square estimate $\hat{\mathbf{x}}$ is given by

$$\hat{\mathbf{x}} = \mathbf{A}\hat{\mathbf{c}} \tag{50}$$
$$= \frac{1}{p}\mathbf{AA}^{\mathsf{H}}\mathbf{S}^{\mathsf{T}}\mathbf{y} \quad \text{using (49)} \tag{51}$$
$$= \mathbf{S}^{\mathsf{T}}\mathbf{y}. \qquad (\mathbf{AA}^{\mathsf{H}} = p\mathbf{I}) \tag{52}$$

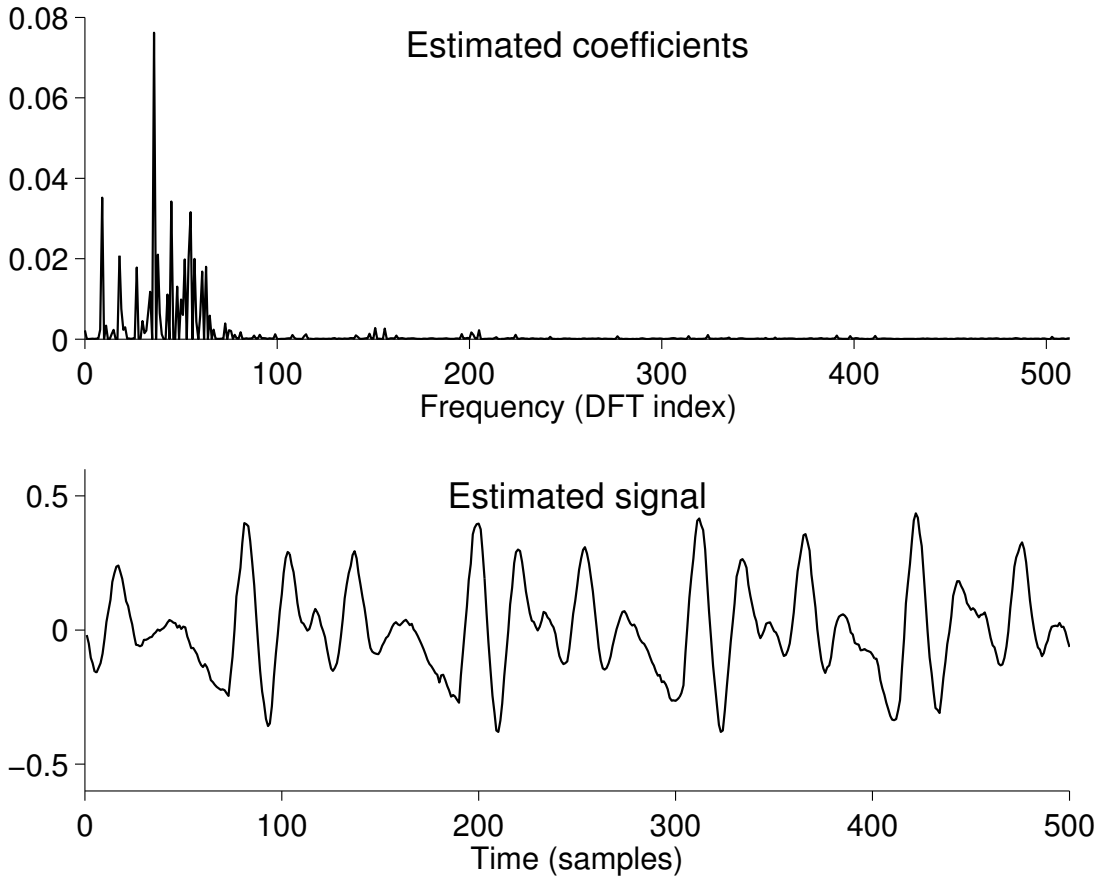
This estimate sets all the missing values to zero!

No estimation of the missing values. Least square solution of no use here.

Example: Filling in missing samples using BP

Short segments of speech can be sparsely represented using the DFT; therefore we set $\mathbf{A}$ equal to the $M \times N$ DFT (19) with $N = 1024$.

BP solution obtained using 100 iterations of SALSA:



The missing samples have been filled in quite accurately.

Total Variation Denoising (TVD)

The signal x is observed in additive white Gaussian noise (AWGN)

$$y(n) = x(n) + w(n), \quad n \in \{0,\ 1,\ 2, \ldots,\ N-1\}$$

Total variation denoising is defined by

$$\hat{x} = \arg\min_{x \in \mathbb{R}^N} \left\{ F(x) = \frac{1}{2} \sum_n |y(n) - x(n)|^2 + \lambda \sum_n |x(n) - x(n-1)| \right\},$$

written also as

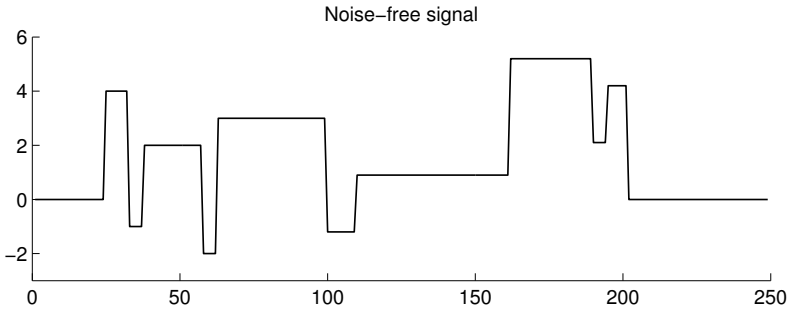
$$\hat{x} = \arg\min_{x \in \mathbb{R}^N} \left\{ F(x) = \frac{1}{2} \|y - x\|_2^2 + \lambda \|Dx\|_1 \right\}$$

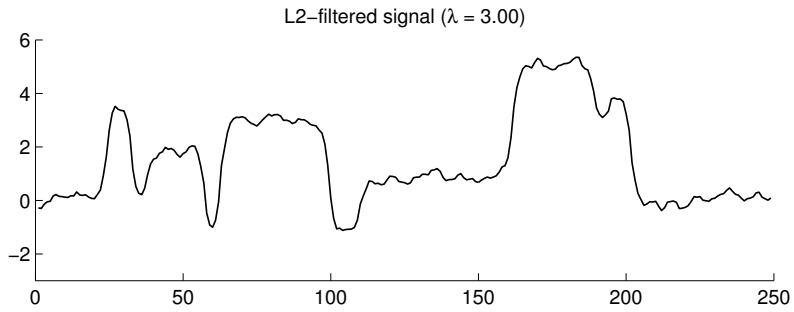
where

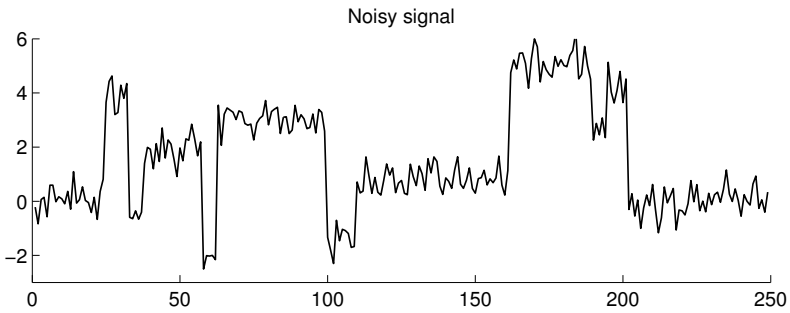
$$D = \begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{bmatrix}.$$

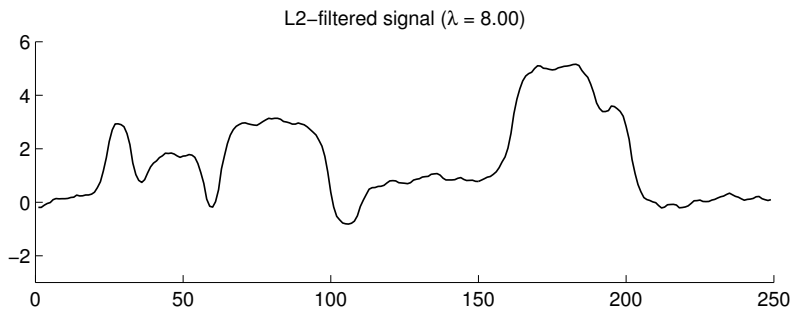
• L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, pp. 259–268, 1992.

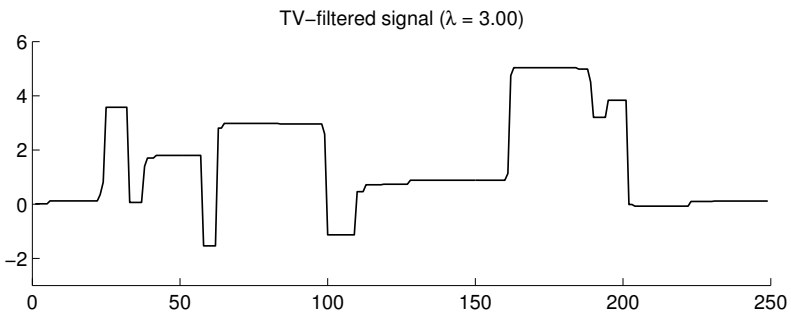
Total Variation Denoising





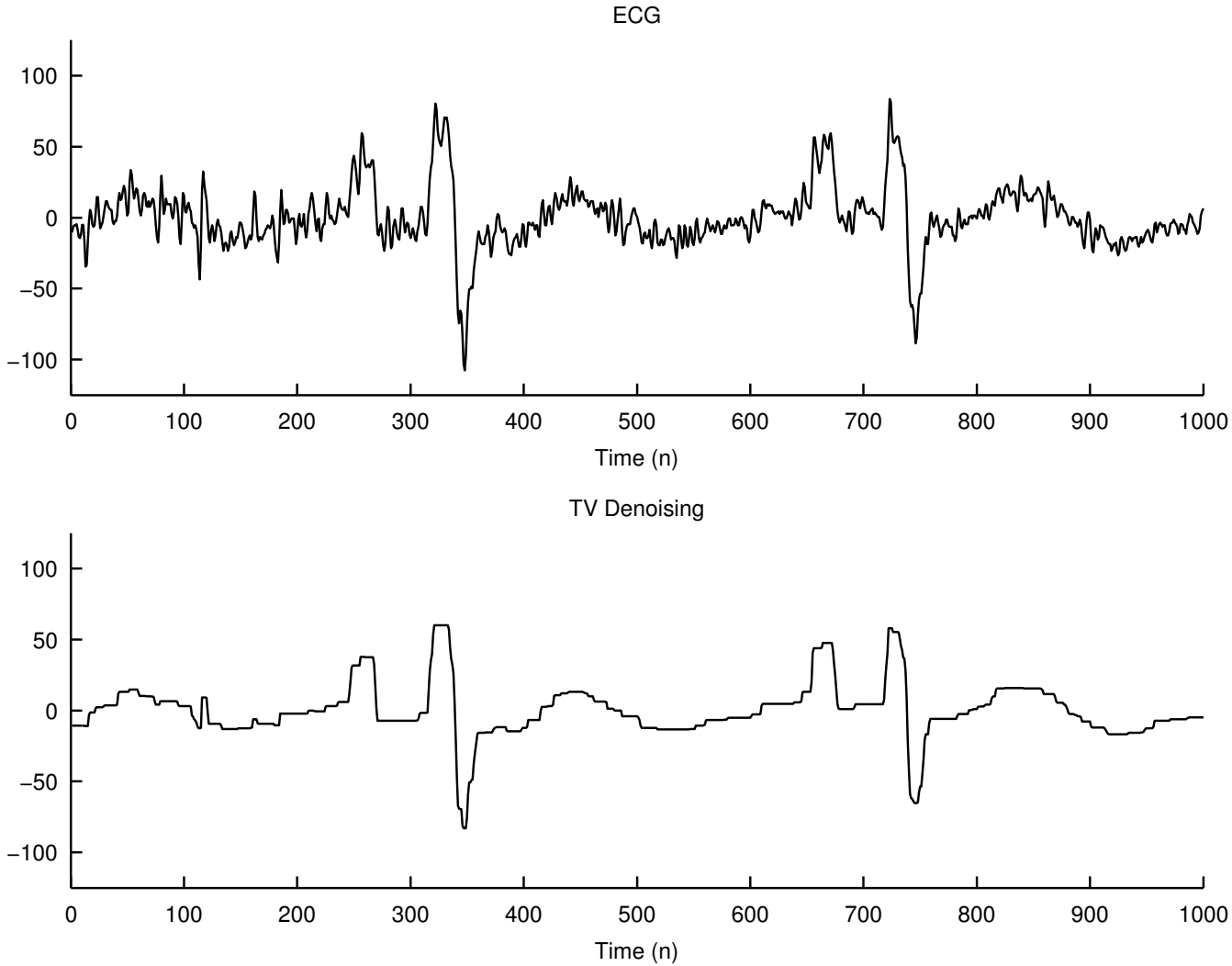






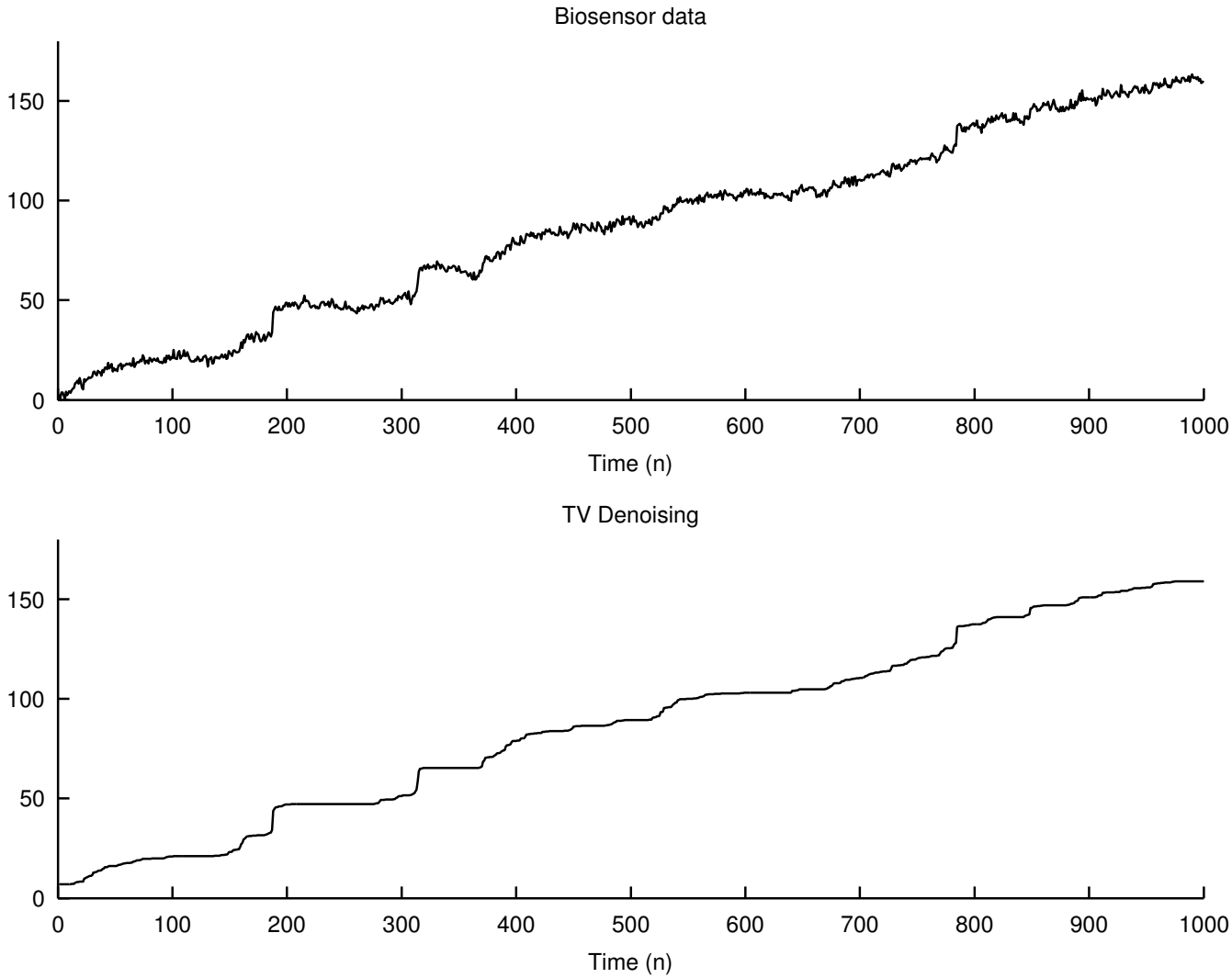
TV denoising preserves discontinuities more accurately than linear filtering.

Total Variation Denoising - staircase artifacts



TVD has staircase artifacts.

Total Variation Denoising - staircase artifacts



TVD has staircase artifacts.

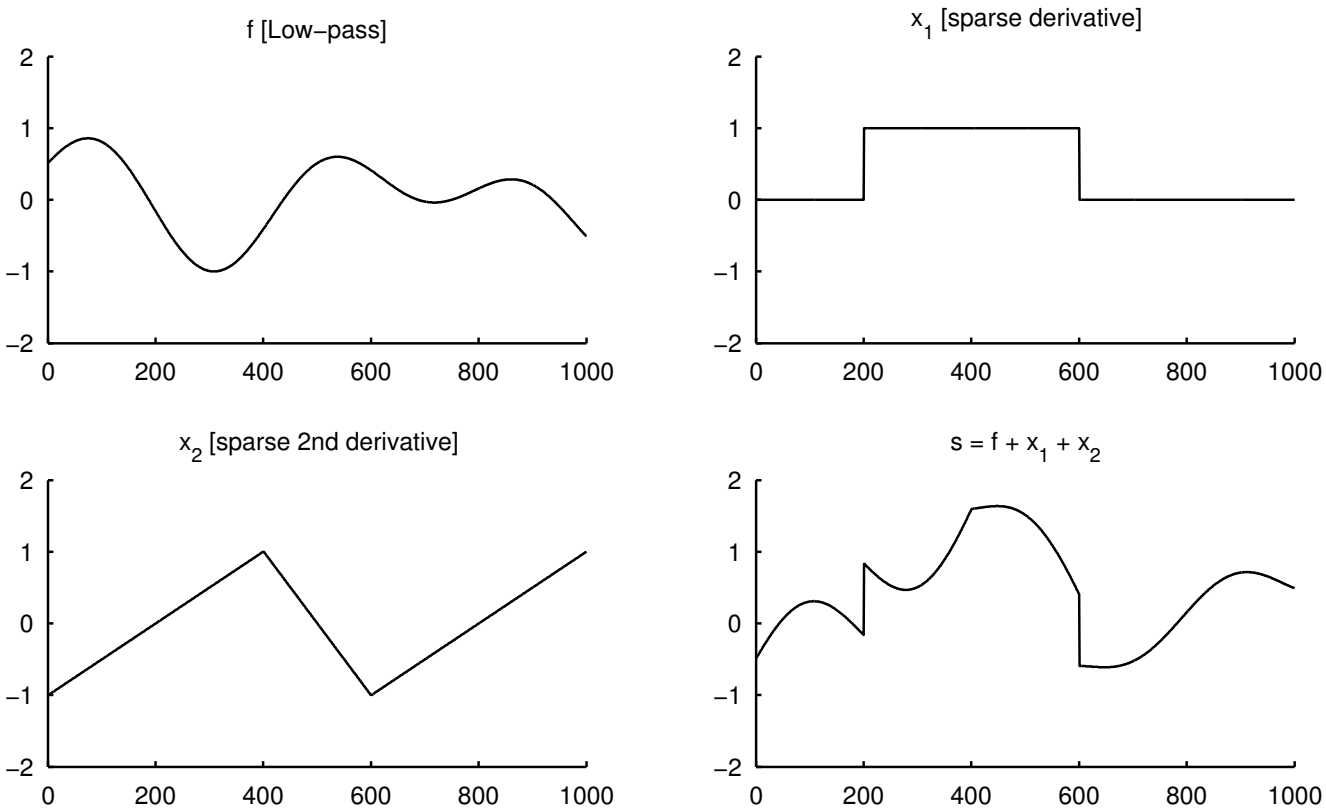
Sparse Singularity-Preserving Signal Smoothing (SIPS)

We assume the signal of interest is of the form

$$s = f + x_1 + x_2, \quad s, f, x_i \in \mathbb{R}^N$$

where

- f is a low-pass signal
- x_1 is approximately piecewise constant
- x_2 is approximately piecewise linear



Sparse Singularity-Preserving Signal Smoothing (SIPS)

Based on the signal model

$$s = f + x_1 + x_2, \quad s, f, x_i \in \mathbb{R}^N$$

we minimize the objective function

$$J(x_1, x_2) = \frac{1}{2}\|Hy - H(x_1 + x_2)\|_2^2 + \lambda_1 \sum_n \phi([Dx_1]_n) + \lambda_2 \sum_n \phi([D^2 x_2]_n)$$

where H is a high-pass filter.

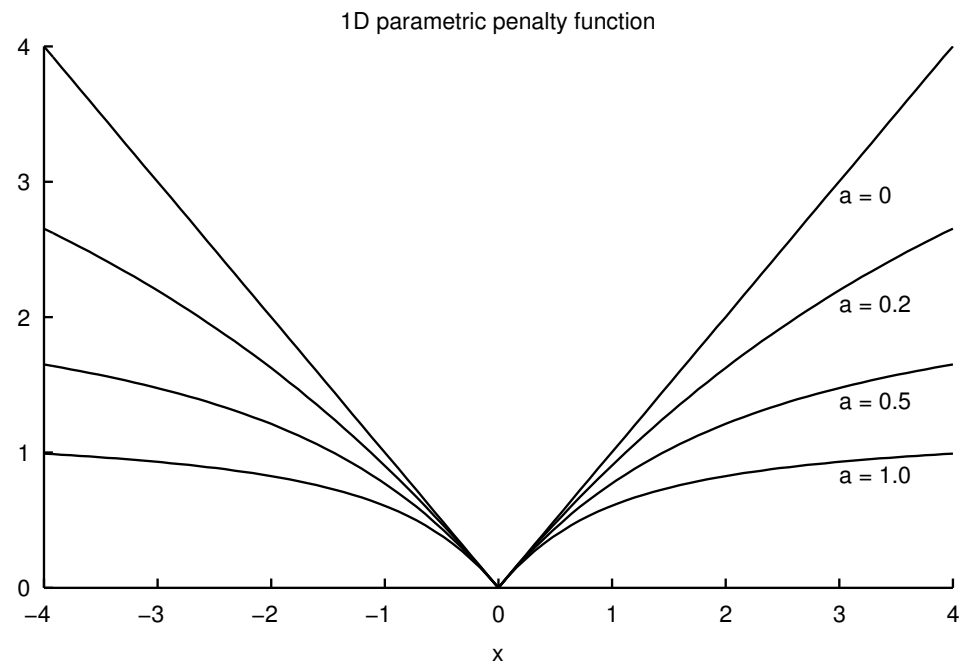
If ϕ is the absolute value function, the regularizer is $\left(\lambda_1 \|Dx_1\|_1 + \lambda_2 \|D^2 x_2\|_1\right)$.

Penalty Function

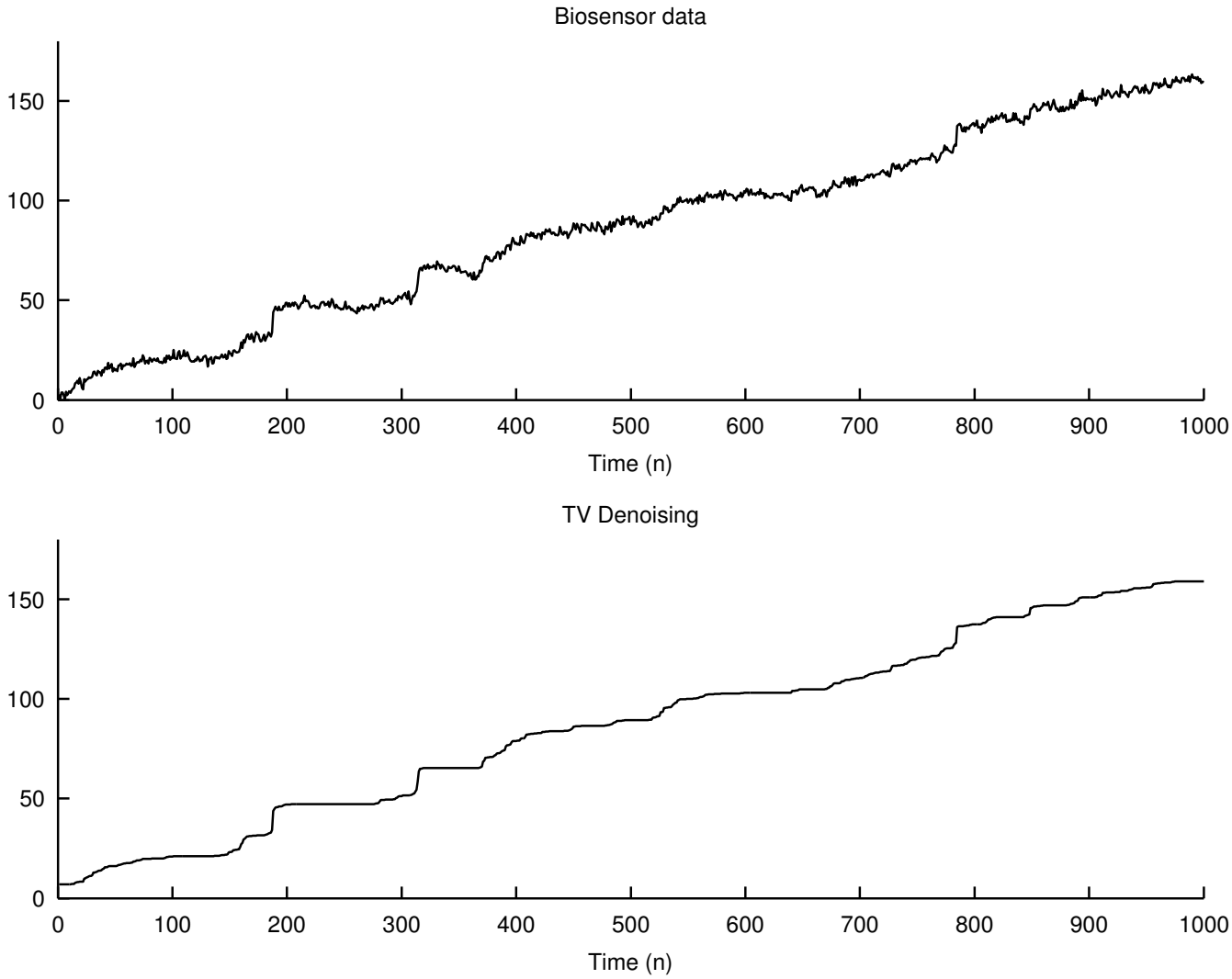
The penalty function ϕ can be taken to be

$$\phi(x) = \begin{cases} \frac{1}{a}\log(1 + a|x|), & a > 0 \\ |x|, & a = 0. \end{cases}$$

The parameter $a \geqslant 0$ controls the non-convexity of ϕ.

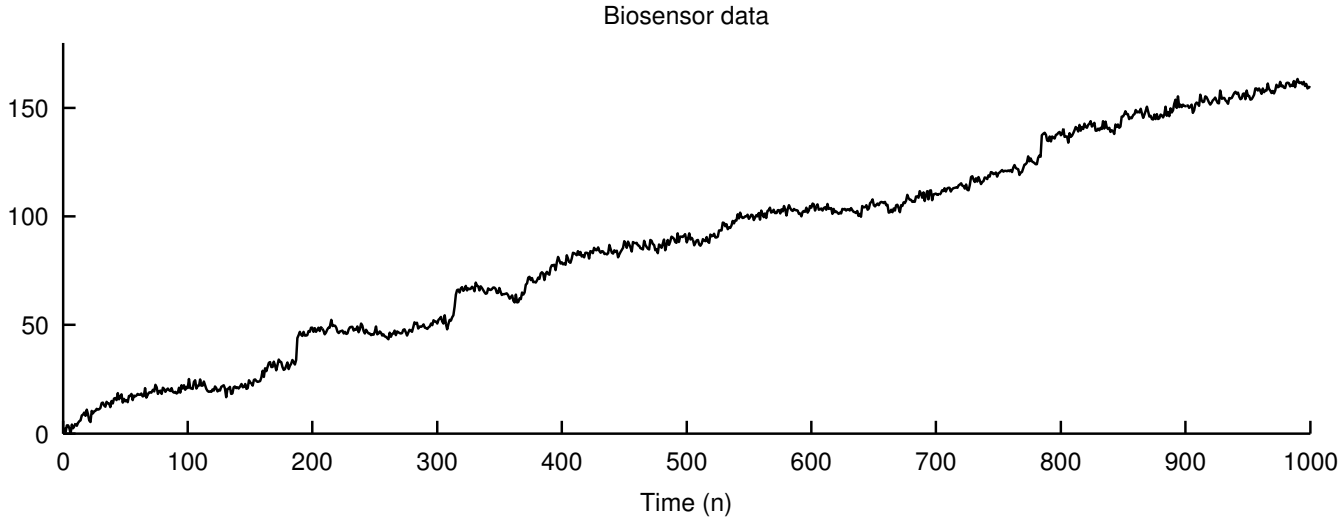


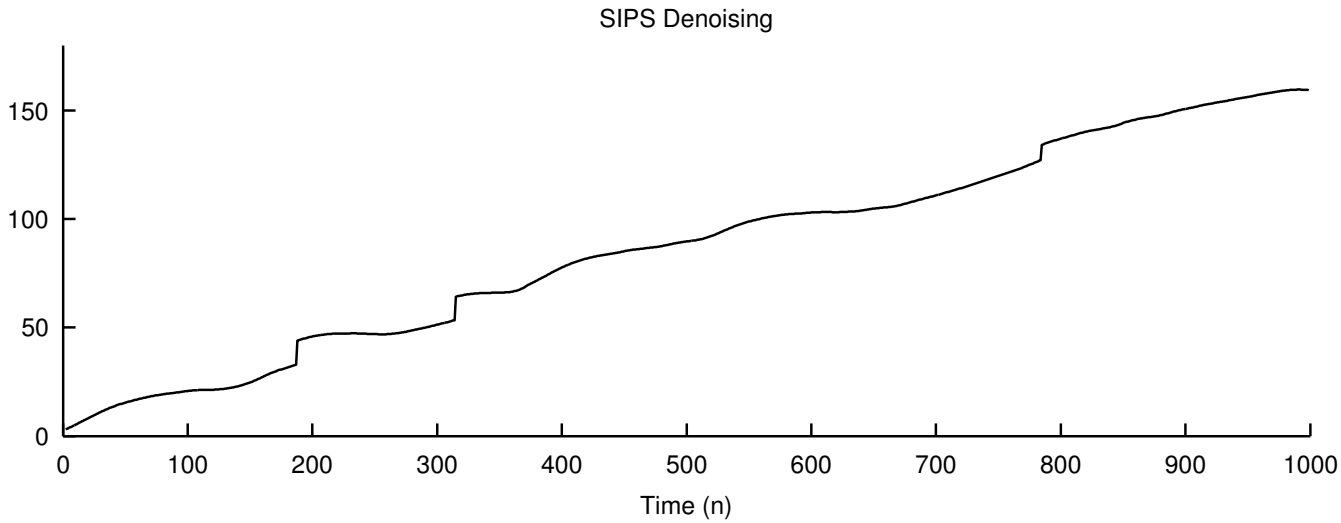
Total Variation Denoising (TVD)



TVD has staircase artifacts.

Sparse Singularity-Preserving Signal Smoothing (SIPS)





SIPS avoids staircase artifacts.

Sparse Singularity-Preserving Signal Smoothing (SIPS)

– Extension to higher-order singularities.

We assume the signal of interest is of the form

$$s = f + x_2 + x_3, \quad s, f, x_i \in \mathbb{R}^N$$

where

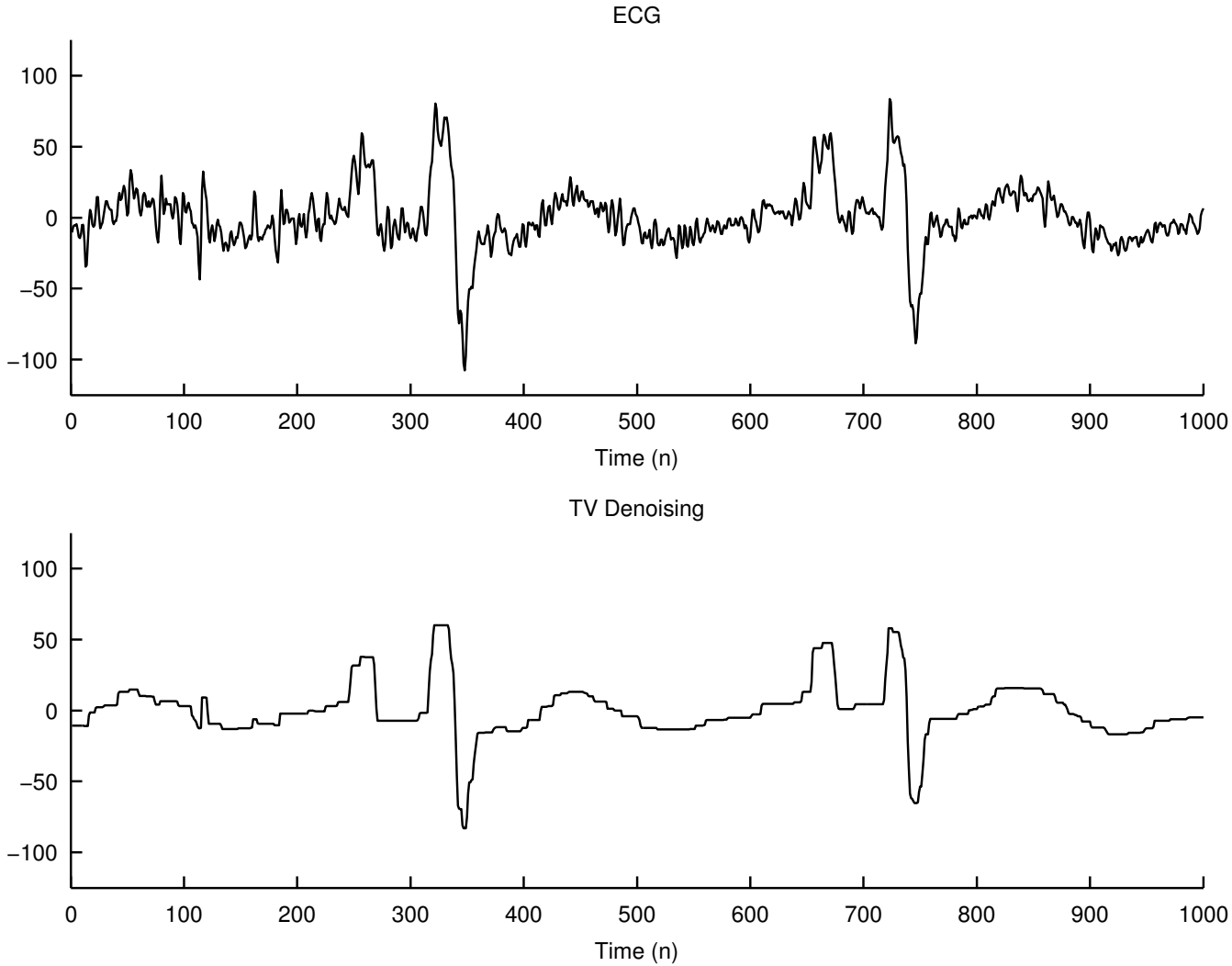
- f is a low-pass signal
- x_2 is approximately piecewise linear
- x_3 is approximately piecewise quadratic

We minimize the objective function

$$J(x_2, x_3) = \frac{1}{2}\|Hy - H(x_2 + x_3)\|_2^2 + \lambda_2 \sum_n \phi([D^2 x_2]_n) + \lambda_3 \sum_n \phi([D^3 x_3]_n)$$

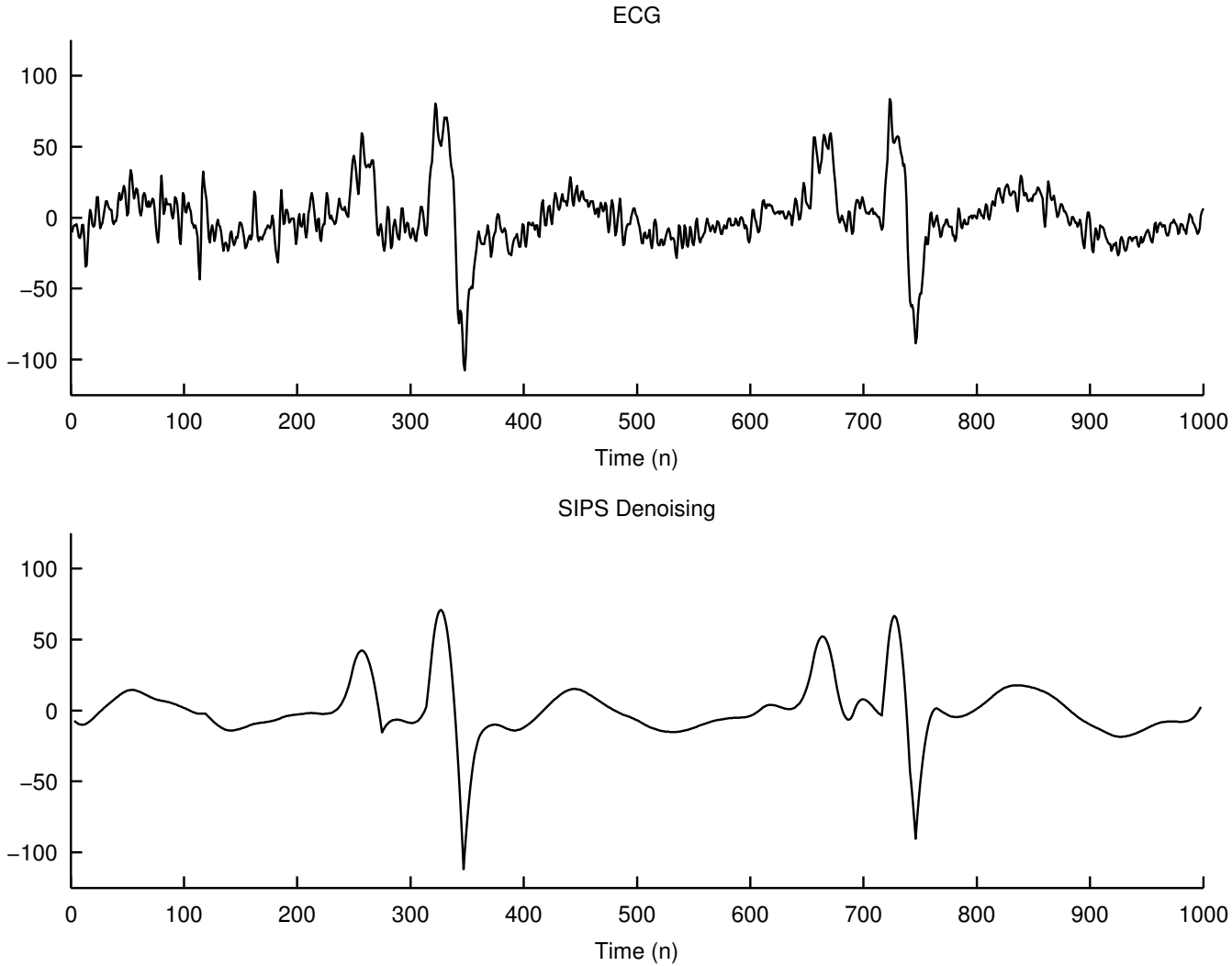
where H is a high-pass filter.

Total Variation Denoising (TVD)



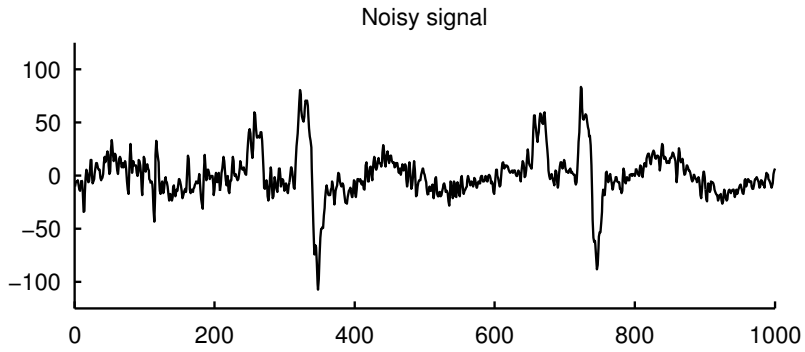
TVD has staircase artifacts.

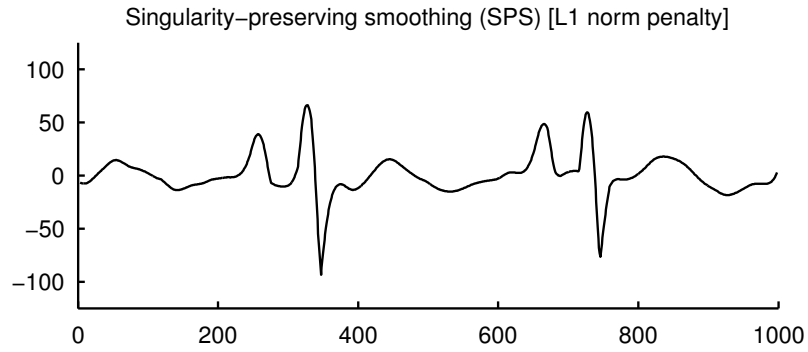
Sparse Singularity-Preserving Signal Smoothing (SIPS)

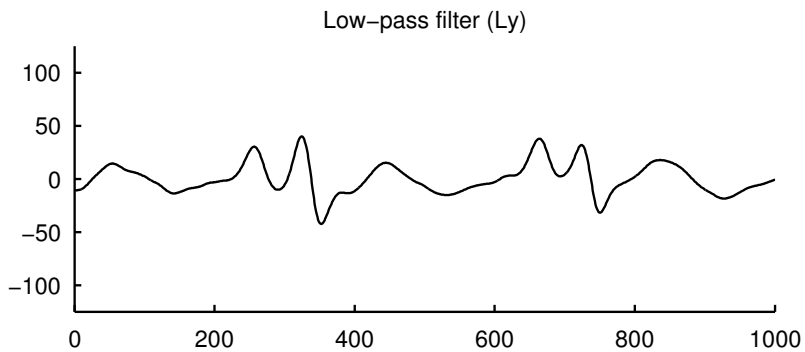


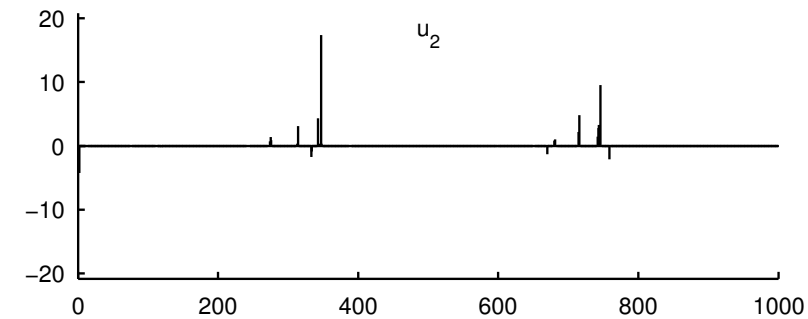
SIPS avoids staircase artifacts.

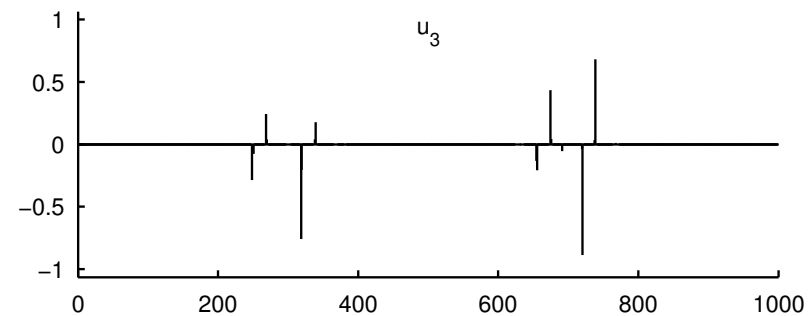
SIPS with L1 norm



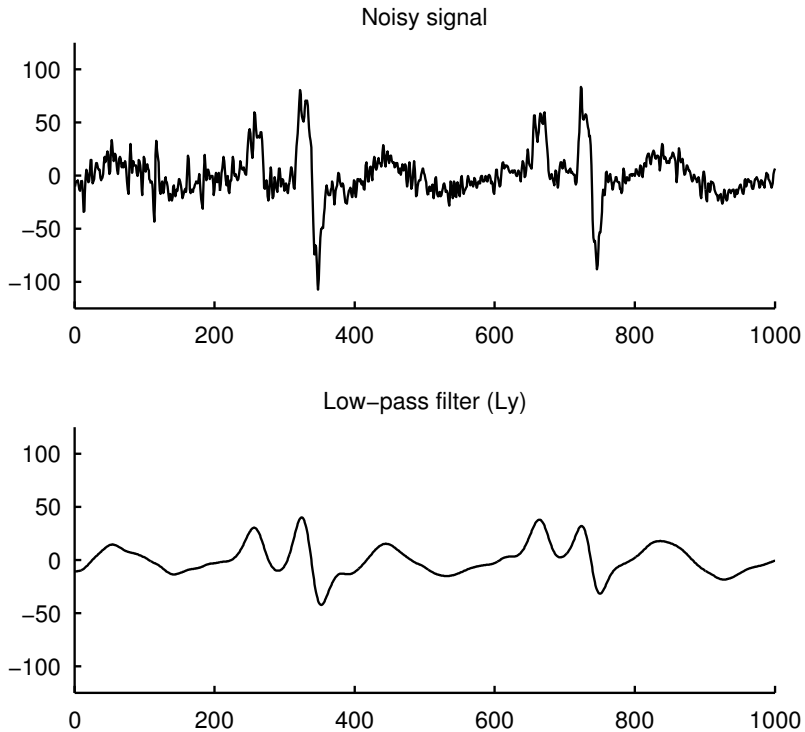


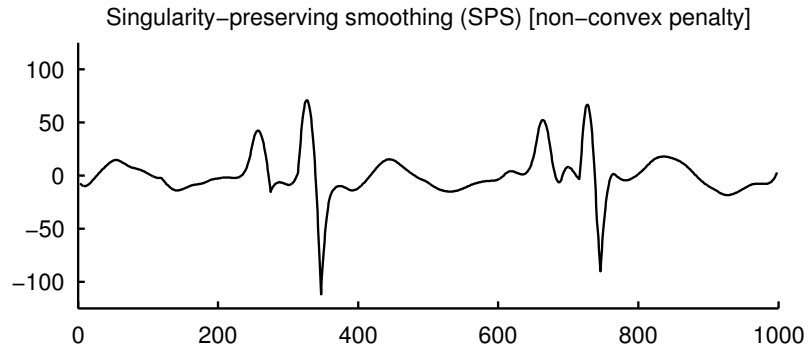


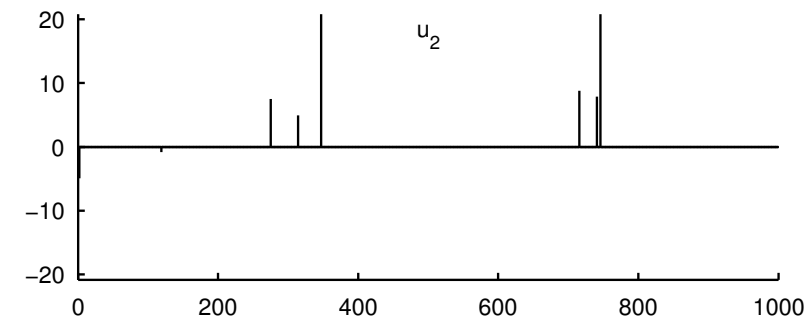


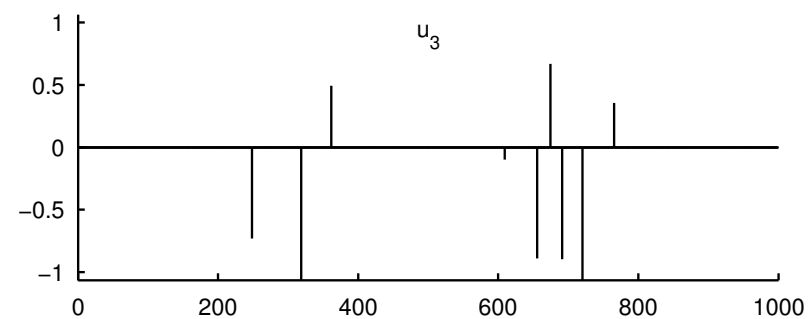


SIPS with non-convex penalty









Convex or non-convex: which is better for inverse problems?

Benefits of convex optimization

1. Absence of suboptimal local minima
2. Continuity of solution as a function of input data
3. Fewer complications when specifying regularization parameters
4. Availability of algorithms guaranteed to converge to a global optimum

But, non-convex regularization often performs better! Convex regularization under-estimates signal values (a 'bias toward zero').

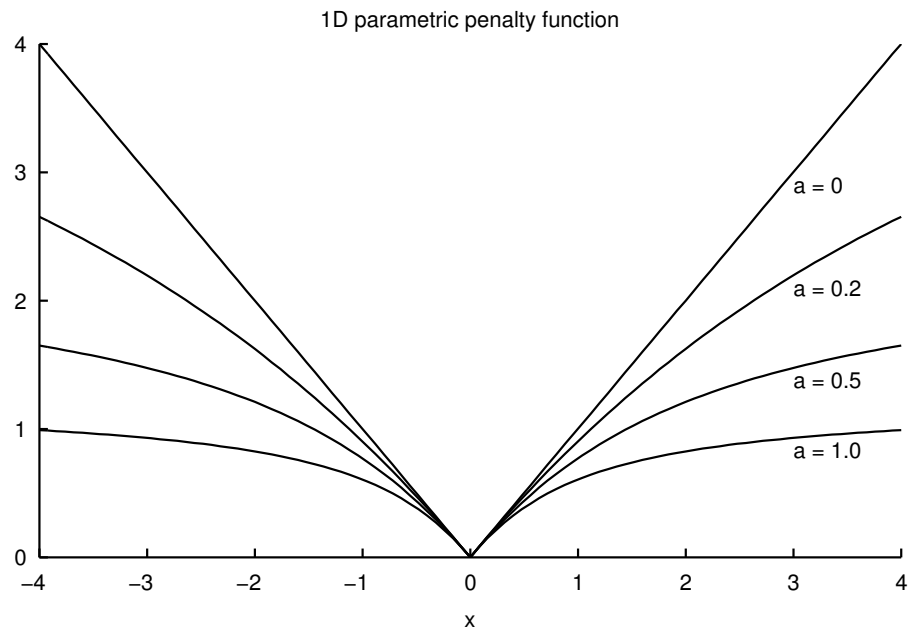
Non-convex regularization induces sparsity more effectively and is a popular alternative to convex functions.

Can we exploit the strong sparsity-inducing properties of non-convex penalties without forgoing the benefits of the convex approach?

Parameterized sparsity-inducing non-convex penalty

Let $\phi(\,\cdot\,; a)\colon \mathbb{R} \to \mathbb{R}$ be a penalty function with parameter $a \geqslant 0$ satisfying

1. ϕ is continuous on $\mathbb{R}$
2. ϕ is twice continuously differentiable, increasing, and concave on $\mathbb{R}_+$
3. $\phi(x; 0) = |x|$
4. $\phi(0; a) = 0$
5. $\phi(-x; a) = \phi(x; a)$
6. $\phi'(0^+; a) = 1$
7. $\phi''(x; a) \geqslant -a$ for all $x \neq 0$



Non-convex regularization, Convex optimization

Total variation denoising with convex regularization:

$$\hat{x} = \arg\min_{x \in \mathbb{R}^N} \left\{ F_0(x) = \frac{1}{2}\|y - x\|_2^2 + \lambda \|Dx\|_1 \right\}$$

With non-convex regularization:

$$\hat{x} = \arg\min_{x \in \mathbb{R}^N} \left\{ F_a(x) = \frac{1}{2}\|y - x\|_2^2 + \lambda \sum_n \phi([Dx]_n; a) \right\}$$

Can we constrain ϕ so that F_a is convex?

Proposition

F_a is strictly convex if

$$\inf_{x \neq 0} \phi''(x) > -\frac{1}{4\lambda}.$$

When ϕ satisfies properties above, F_a is strictly convex if

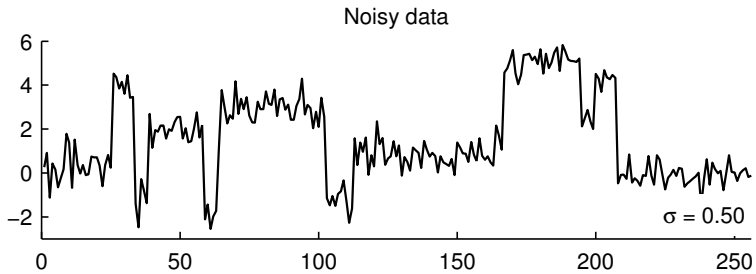
$$0 \leqslant a < \frac{1}{4\lambda}.$$

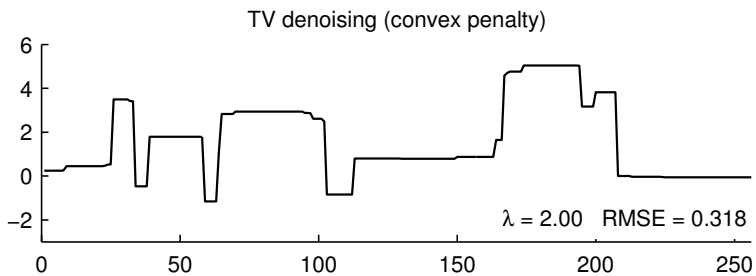
• I. W. Selesnick, A. Parekh, and I. Bayram, "Convex 1-D total variation denoising with non-convex regularization," *IEEE Signal Processing Letters*, vol. 22, pp. 141–144, Feb. 2015.

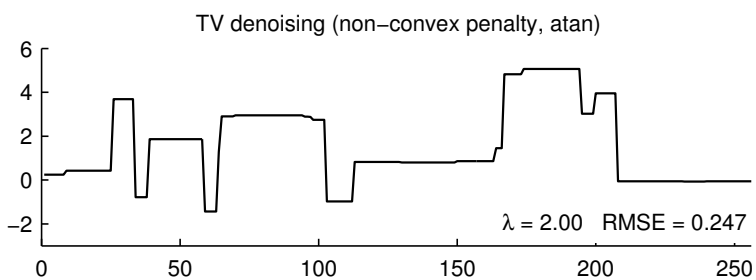
Convex TVD with non-convex regularization

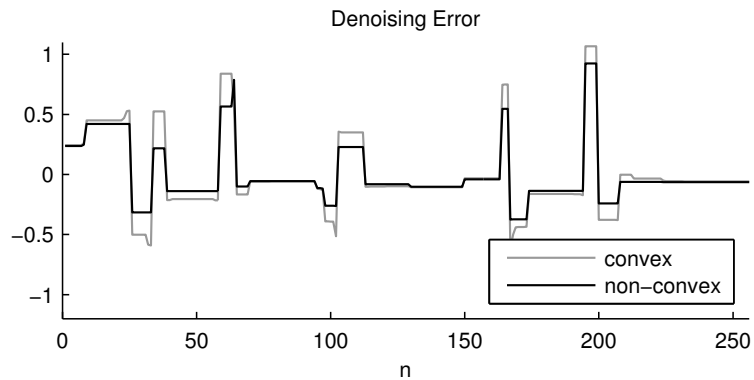
Set a to its maximal value to maximally induce sparsity of Dx while ensuring convexity of the objective function.

$$a = \frac{1}{4\lambda}$$









Convex TVD with non-convex regularization

TVD with non-convex regularization:

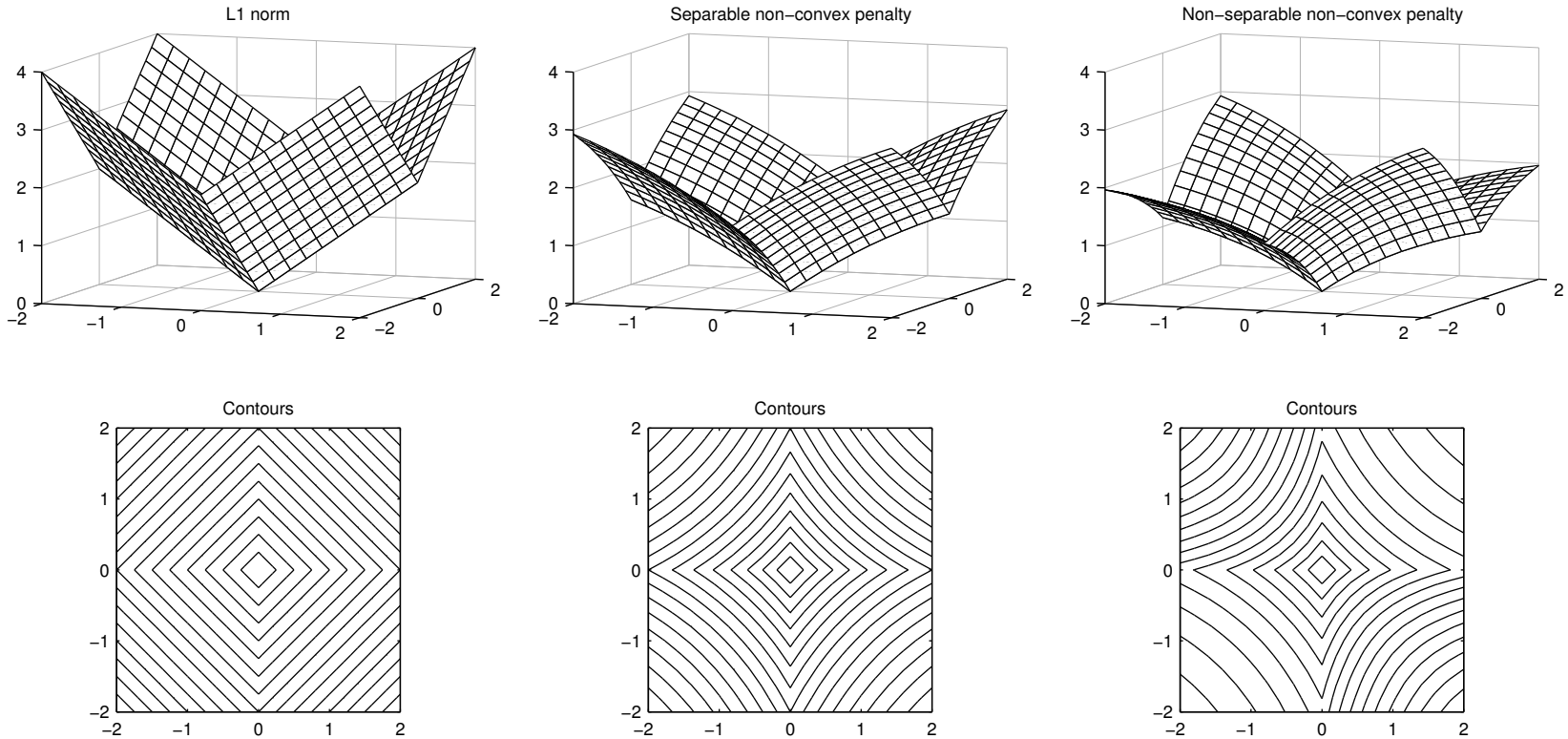
$$\hat{x} = \arg\min_{x \in \mathbb{R}^N} \left\{ F(x) = \frac{1}{2}\|y - x\|_2^2 + \lambda \sum_n \phi([Dx]_n; a) \right\}$$

TVD with non-separable non-convex regularization:

$$\hat{x} = \arg\min_{x \in \mathbb{R}^N} \left\{ F_{\text{nonsep}}(x) = \frac{1}{2}\|y - x\|_2^2 + \lambda \sum_n \psi(([Dx]_{n-1}, [Dx]_n); a) \right\}$$

where $\psi \colon \mathbb{R}^2 \to \mathbb{R}$.

Convex TVD with non-convex regularization



$a_1 = a_2 = 0$

(a) Separable, convex

$a_1 = a_2 > 0$

(b) Separable, non-convex

$a_1 > a_2 > 0$ (Proposed)

(c) Non-separable, non-convex

Convex TVD with non-convex regularization

Define $\psi(\,\cdot\,; a)\colon \mathbb{R}^2 \to \mathbb{R}$ as

$$\psi(x) = \begin{cases} (1-r)\,[\phi(x_1;\alpha) + \phi(x_2;\alpha)] + r\,\phi(x_1+x_2;\alpha), & x \in A_1 \\ (1+r)\,\phi(x_1;a_2) + \phi(rx_1+x_2;\alpha), & x \in A_2 \\ (1+r)\,\phi(x_2;a_2) + \phi(x_1+rx_2;\alpha), & x \in A_3 \end{cases} \tag{53}$$

where A_i are subsets of $\mathbb{R}^2$ defined as

$$A_1 = \{x \in \mathbb{R}^2 \mid x_1 x_2 \geqslant 0\}, \tag{54}$$

$$A_2 = \{x \in \mathbb{R}^2 \mid x_1\,(x_1+x_2) \leqslant 0\}, \tag{55}$$

$$A_3 = \{x \in \mathbb{R}^2 \mid x_2\,(x_1+x_2) \leqslant 0\} \tag{56}$$

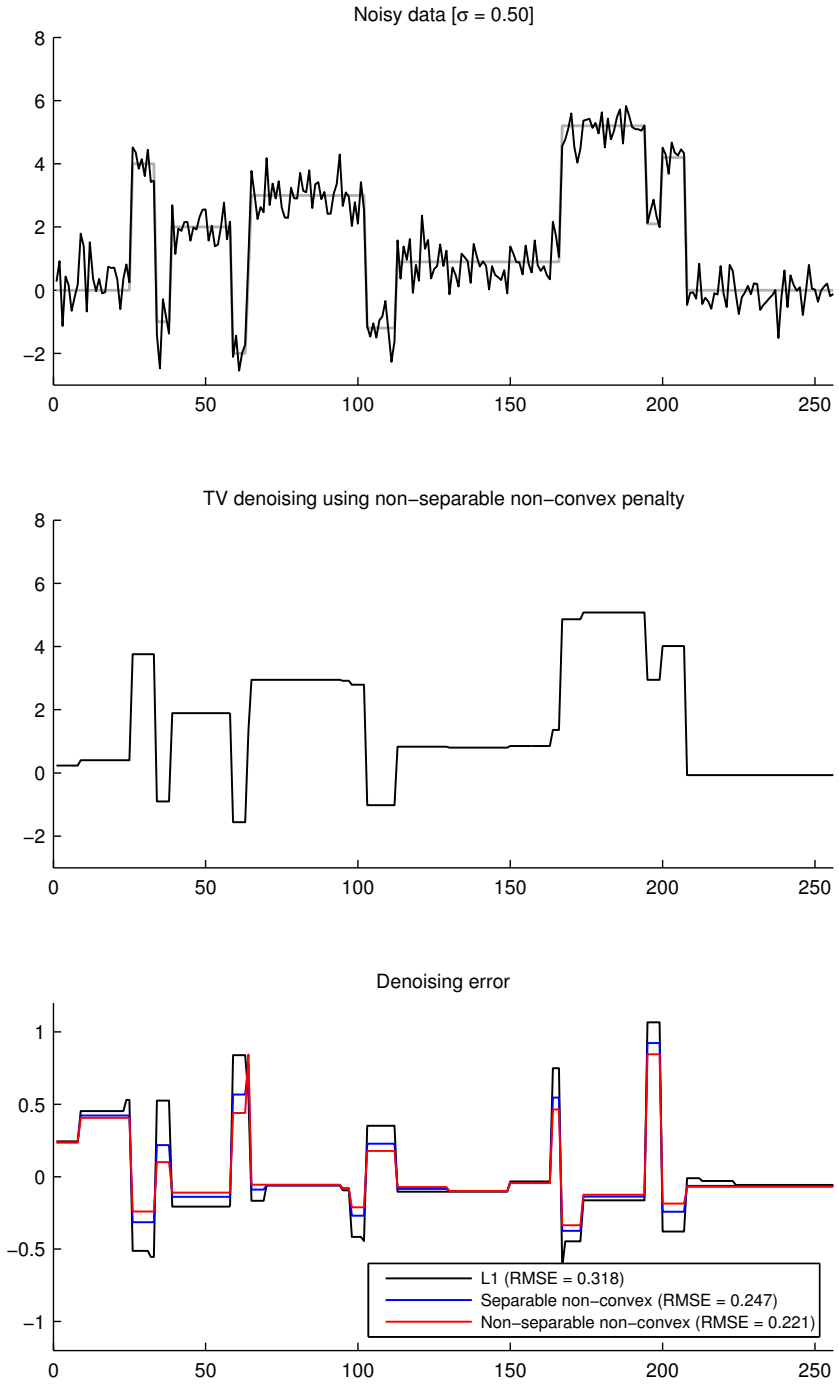
and α and r are given by

$$\alpha = \frac{a_1+a_2}{2}, \qquad r = \begin{cases} \frac{a_1-a_2}{a_1+a_2}, & a_1+a_2 > 0 \\ 0, & a_1 = a_2 = 0. \end{cases} \tag{57}$$

Proposition

F_{nonsep} is strictly convex if

$$a_1 \leqslant \frac{1}{2\lambda}, \quad a_2 \leqslant \frac{1}{4\lambda}.$$


Noisy data [σ = 0.50]
TV denoising using non−separable non−convex penalty
Denoising error
L1 (RMSE = 0.318)
Separable non−convex (RMSE = 0.247)
Non−separable non−convex (RMSE = 0.221)

Structured Sparsity with Overlapping Groups

A simple nonlinear thresholding approach to denoising is *basis pursuit denoising*

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_1 \right\} = \text{soft}(\mathbf{x}, \lambda)$$

Thresholding does not capture group structure (clustering/grouping behavior).

To account for clustering/grouping, we may use

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{x}\|_2^2 + \lambda \sum_n \sqrt{|x(n)|^2 + |x(n+1)|^2 + |x(n+2)|^2} \right\}$$

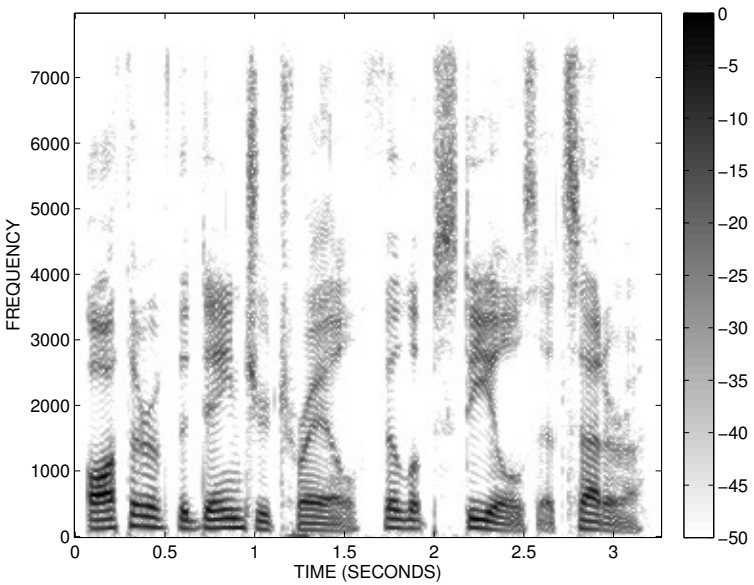
for group size 3.

The optimization is more challenging due to coupling among all signal values $x(n)$, but yields superior results for speech enhancement.

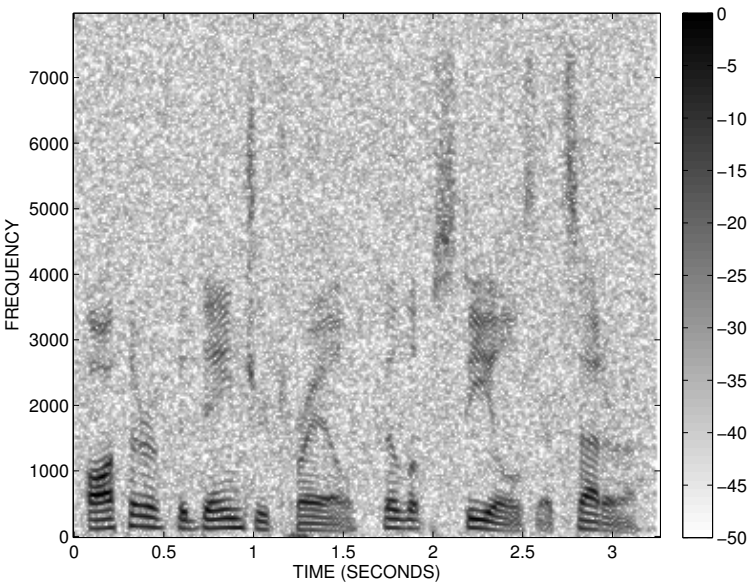
• P.-Y. Chen and I. W. Selesnick, "Translation-invariant shrinkage/thresholding of group sparse signals," *Signal Processing*, vol. 94, pp. 476–489, Jan. 2014.

Structured Sparsity with Overlapping Groups: Speech Enhancement

Speech signals exhibit structured sparsity in time-frequency domain.



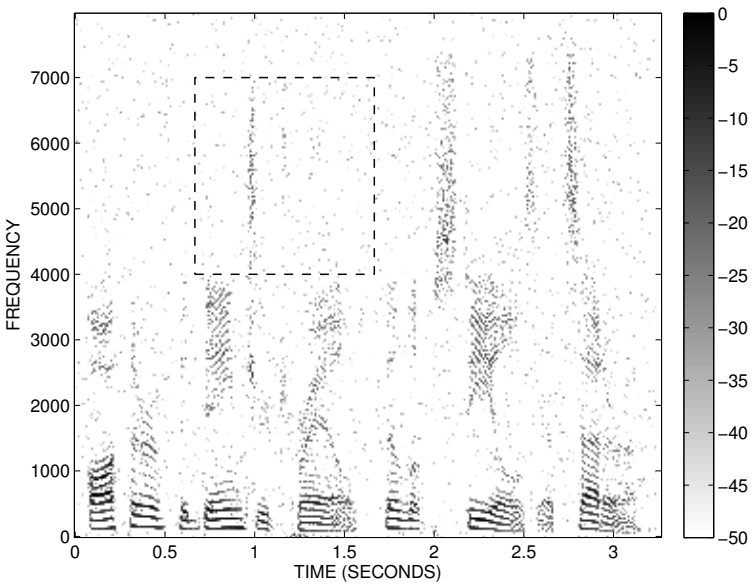
Noise-free signal



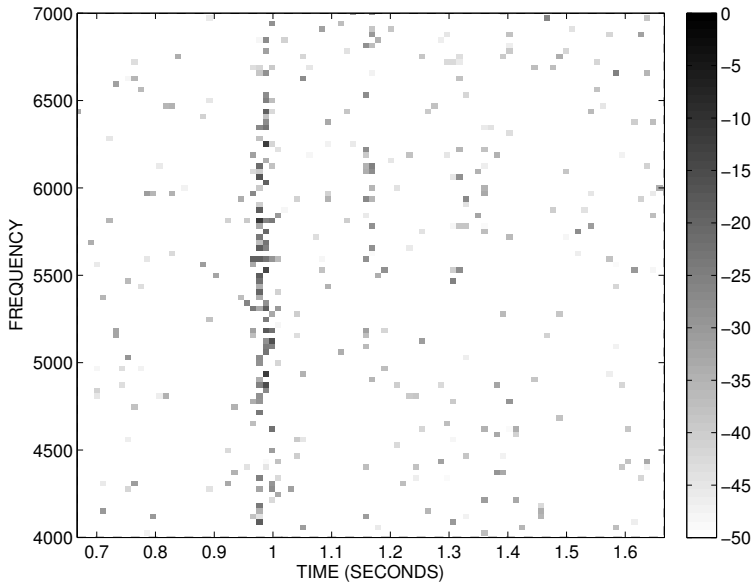
Noisy signal

Structured Sparsity with Overlapping Groups: Speech Enhancement

Scalar thresholding produces spurious noise spikes and *musical noise*.



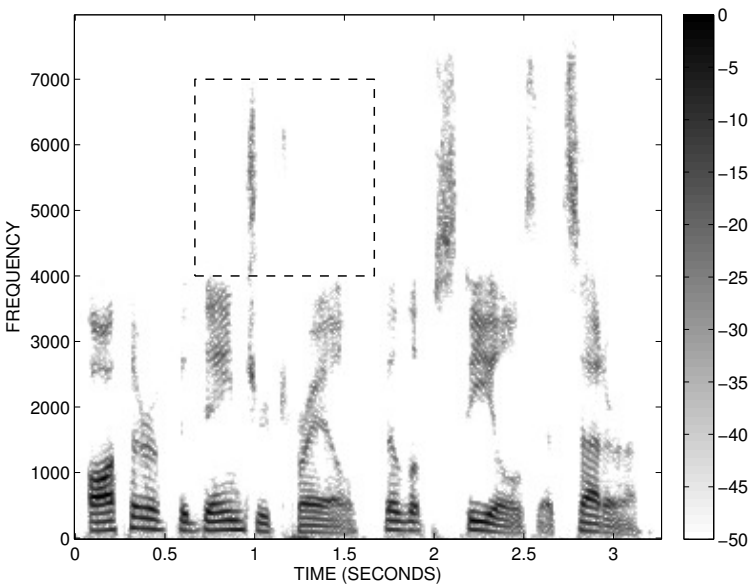
Scalar thresholding



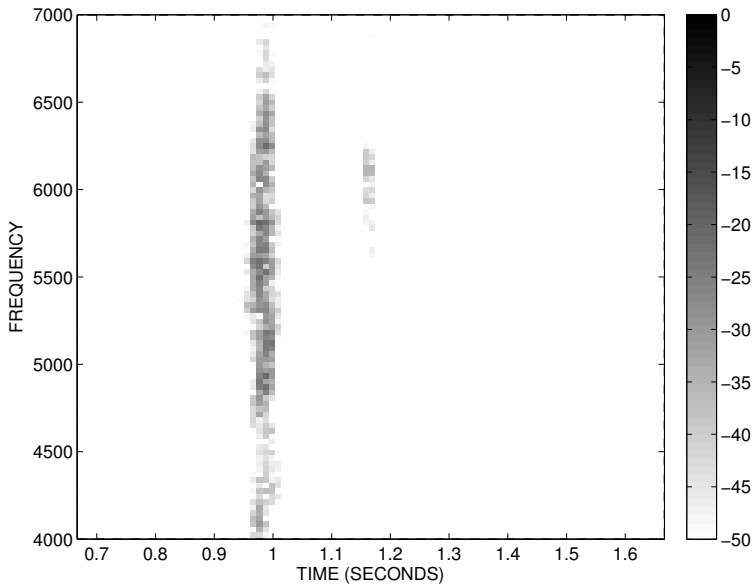
Magnified view

Structured Sparsity with Overlapping Groups: Speech Enhancement

New group shrinkage/thresholding (OGS) algorithm reduces musical noise.



OGS algorithm



Magnified view

Summary

1. Basis pursuit
2. Basis pursuit denoising
3. Sparse Fourier coefficients using BP
4. Denoising using BPD
5. Deconvolution using BPD
6. Filling in missing samples using BP
7. Total variation denoising (TVD)
8. Sparse singularity-preserving signal smoothing (SIPS)
9. Non-convex regularization, convex optimization
10. Group sparsity