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

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

(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

Noisy data
Optimization Problem Statement

We pose the minimization problem

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

where \(p(n)\) is a polynomial,

\[p(n) = a_0 + a_1 n + \cdots + a_d n^d, \] (3)

and \(\phi: \mathbb{R} \rightarrow \mathbb{R}\) is a sparsity-promoting penalty function, e.g.,

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

When \(\phi(u) = \lambda |u|\), then the regularization term in (2) is \(\lambda \|Dx\|_1\), where \(\|\cdot\|_1\) denotes the \(\ell_1\) norm.

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

---

Notation

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

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

\[
D = \begin{bmatrix}
-1 & 1 & & & \\
-1 & -1 & 1 & & \\
-1 & -1 & -1 & 1 & \\
-1 & -1 & -1 & -1 & 1 \\
\end{bmatrix}.
\] (5)

In MATLAB: $Dx$ is `diff(x)`.

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

\[
S := \begin{bmatrix}
0 & & & \\
1 & & & \\
1 & 1 & & \\
1 & 1 & 1 & \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
\end{bmatrix}
\] (6)

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

Note!

\[
DS = I
\] (7)
Notation

Tall Vandermonde matrix, \( V \),

\[
V = \begin{bmatrix}
1 & 1 & \cdots & 1 \\
1 & 2 & & 2^d \\
1 & 3 & & 3^d \\
& \vdots & \ddots & \vdots \\
& \vdots & & \vdots \\
1 & N & \cdots & N^d
\end{bmatrix}
\]

\( G \): Orthogonalize columns of \( V \)

\[
G^T G = I
\]

\( G \) is \( N \times (d + 1) \).

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

In MATLAB: \( G = \text{orth}(	ext{bsxfun(@power,(0:N-1)',0:d)))} \)
MM Optimization Algorithm$^2$

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

$$x^{(k+1)} = \arg\min_x G(x, x^{(k)}).$$

(8)

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

That is,

$$G(x, v) \geq F(x), \; \forall x,$$

and

$$G(v, v) = F(v)$$

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

---

The PATV problem (2) is equivalent to

\[(a^*, x^*) = \arg \min_{a,x} \frac{1}{2} \|y - Ga - x\|_2^2 + \sum_n \phi([Dx]_n)\]  

(9)

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

Note that \(a^*\) can be expressed explicitly:

\[a^* = GT(y - x).\]  

(10)

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

\[x^* = \arg \min_{x} \frac{1}{2} \|H(y - x)\|_2^2 + \sum_n \phi([Dx]_n)\]  

(11)

where \(H\) is given by

\[H := I - GG^T.\]  

(12)

---

3 In (9), polynomial \(p\) is represented using the orthonormal basis \(G\).
Adding a constant to $x^*$ does not change the value of the cost function (11).

1. $D$ annihilates constants.

2. $H$ annihilates constants (a constant signal is exactly represented as a polynomial of degree zero).

The minimizer $x^*$ is unique only up to an additive constant.

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

Then by (7),

$$Dx = DSu = lu = u$$

and (11) becomes

$$u^* = \arg \min_u \left\{ F(u) = \frac{1}{2} \left\| H(y - Su) \right\|_2^2 + \sum_n \phi(u(n)) \right\}.$$  

(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

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

(15)

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

$$\sum_n g(u(n), v(n)) \geq \sum_n \phi(u(n)).$$

(16)

Note:

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

(17)

where $W(v)$ is a diagonal matrix defined by

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

(18)

Therefore, a majorizor for $F(u)$ is given by

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

(19)
\( G(u, v) \) is quadratic in \( u \). Hence, minimizing \( G(u, v) \) wrt \( u \) gives

\[
u = (S^T H^T HS + W(v))^{-1} S^T H^T Hy \tag{20}\]

where \( W(v) \) depends on \( v \) per (18).

\( H^T H = H \), so

\[
u = (S^T HS + W(v))^{-1} S^T Hy. \tag{21}\]

Therefore, the MM update produces the sequence

\[
u^{(k+1)} = (S^T HS + W^{(k)})^{-1} S^T H y \tag{22}\]

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

\[
[W^{(k)}]_{n,n} = \frac{\phi'(u^{(k)}(n))}{u^{(k)}(n)} \tag{23}\]
Two problems with (22):

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

Solution\textsuperscript{4}: use the matrix inverse lemma (MIL).

Here, we need to use MIL twice.

First, write:

$$S^T HS + W^{(k)} = S^T (I - GG^T)S + W^{(k)}$$

$$= W^{(k)} + S^T S - S^T GG^T S$$

$$= A^{(k)} - BB^T$$

where

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

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

$$\left(S^T HS + W^{(k)}\right)^{-1} = [A^{(k)}]^{-1} + [A^{(k)}]^{-1} B \left[I - B^T [A^{(k)}]^{-1} B\right]^{-1} B^T [A^{(k)}]^{-1}$$

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

Use the MIL again:

\[
[A^{(k)}]^{-1} = [W^{(k)} + S^T S]^{-1}
\]

\[
= [W^{(k)}]^{-1} - [W^{(k)}]^{-1} [(S^T S)^{-1} + [W^{(k)}]^{-1}]^{-1} [W^{(k)}]^{-1}
\]

\[
= \Lambda^{(k)} - \Lambda^{(k)} [(S^T S)^{-1} + \Lambda^{(k)}]^{-1} \Lambda^{(k)}.
\]

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

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

\(\Lambda^{(k)}\) is diagonal.

Three observations:

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

\[
\phi(u) = \lambda |u| \quad \implies \quad u/\phi'(u) = |u|/\lambda
\]

\[
\phi(u) = (\lambda/\alpha) \log(1 + \alpha |u|) \quad \implies \quad u/\phi'(u) = |u|(1 + \alpha |u|)/\lambda
\]
2) The matrix $(S^T S)^{-1}$ is banded

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

Defining

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

then (28) can be written as

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

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

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

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

---

3) Matrix (27) also involves

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

$Q$ is small, $(d + 1) \times (d + 1)$, where $d$ is the order of the low-order polynomial $p(n)$. $B$ is tall and narrow, $(N - 1) \times (d + 1)$, and need be computed once and saved.

So at each iteration, the matrix $Q^{(k)}$ can be computed efficiently. \checkmark
Hence matrix (27) can be written as

\[
(S^T HS + W^{(k)})^{-1} = [A^{(k)}]^{-1} \left( I + B [Q^{(k)}]^{-1} B^T [A^{(k)}]^{-1} \right)
\]

(36)

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

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

\[
b = S^T Hv
\]

\[
u^{(k+1)} = [A^{(k)}]^{-1} \left[ b + B [Q^{(k)}]^{-1} B^T [A^{(k)}]^{-1} b \right]
\]

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: $y \in \mathbb{R}^N$, $d$, $\phi$

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

repeat

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

until convergence

9. $x = Su$
10. $a = G^T (y - x)$

output: $x$, $a$

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

When penalty $\phi$ is convex, then minimizer $u^*$ must satisfy optimality conditions.

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

Suppose $\phi$ is convex. If $u$ solves (14), then $u$ must satisfy:

$$[S^T H^T H(y - Su)]_n \in \partial \phi(u(n)), \quad \forall n$$

(37)

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

Using $H^T H = H$, $x = Su$, and $u = Dx$, we may write

$$[S^T H(y - x)]_n \in \partial \phi([Dx]_n), \quad \forall n.$$  

(38)

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

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

(39)

The optimality condition is conveniently illustrated by a scatter plot.
Example: $\ell_1$ 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

Moreover, the proposed MM algorithm requires no user specified parameter. The
ADMM algorithm requires two parameters ($\mu_i$) 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 $S^T H(y - x)$
versus $D x$. According to (38), the solution is optimal if the points in the scatter plot
lie on the dashed lines.

---

Example: $\ell_1$ norm penalty (convex)

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

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

$u = Dx$
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 $\ell_1$ norm.

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

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

![Estimated signal (PATV, log penalty) and Scatter plot](image)
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 $\ell_1$ norm and non-convex penalties
Simultaneous Low-pass Filtering and Total Variation Denoising (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

- LPA–ICI (RMSE = 0.115)
- Wavelet–HMT (RMSE = 0.097)
- Wavelet–TV (RMSE = 0.117)
- New algorithm (d) Estimated signal (RMSE = 0.087)
**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.

---

Low-Pass Filtering and Compound Sparse Denoising

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

\[
\arg\min_x \|H(y - x)\|_2^2 + \lambda_0\|x\|_1 + \lambda_1\|Dx\|_1
\]

Near infrared spectroscopic (NIRS) data.\textsuperscript{8}

---

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!